Manuscript changes to referee comments

Deep learning for fast simulation of seismic waves in complex media

Introduction:
This document contains our previous responses and completed manuscript changes resulting from the two reviews of our “Deep learning for fast simulation of seismic waves in complex media” discussion paper.

We would like to thank the reviewers (Andrew Curtis and Andrew Valentine) for their in-depth and valuable comments on the paper and hope our responses below address their comments.

Where we have deemed appropriate, we have grouped similar comments from both reviewers together and provided a single response.

Ben Moseley, Tarje Nissen-Meyer & Andrew Markham

Reviewer comments:

[Reviewer 2] 1. The main worry I have about this manuscript is a combination of (a) the method is acoustic and uses relatively simple acoustic models, (b) the authors themselves state that it will be difficult to extend the proposed methods to more complex models and particularly to solid (elastic) models, and (c) that these two points seem to imply that this method is not, as the authors propose, an advancement on solid Earth science but rather is a paper about toy problems in acoustics that cannot be extended to elastic media. I am not saying that this is definitely what the authors believe, nor that it is true, but that is how their message came across to me as a reader. If true (and I would not actually be surprised if it is), this suggests that the paper might be more appropriate for an acoustics journal like JASA, rather than a solid Earth journal. Otherwise, the authors must better justify how this work advances solid Earth science.

It may be that the authors think that these methods could be extended to elastic media and to significantly more complex (heterogeneous) models, and they simply have not explained how. Or it may be that they think that this research closes off an avenue of research for solid Earth science which is useful to stop others from following – in effect they might decide to argue that they prove that this approach or network architecture will not be fruitful for the Earth sciences. Either is a positive step for science. One way or the other, the authors need to explain more clearly which (or which other message) we should take from their work, and why.

However, since this is a Discussion paper, in fact I think the best would be to rethink the paper slightly: I would begin by thinking through, and then presenting, a roadmap that might solve real modelling or inversion problems in the solid Earth sciences using models of more realistic complexity. Then show how this paper fits into that – either taking a first step in a possible direction to achieve it, or testing an avenue that while successful, turns out not to be practical for the real Earth. Either way, in the Discussion section they can explain how this work has advanced our state of knowledge about the overall strategy, and how we should move forward in future.

Author response:

We can see how our message comes across as though this is a toy problem that cannot be extended to elastic media and that the paper does not consider enough how it advances Solid Earth science.

The message we want to convey is that we are taking the first steps in understanding whether deep learning can aid seismic simulation tasks relevant to the Solid Earth sciences, and in that sense proposing a new research direction.

Whilst we only present methods for “toy” 2D acoustic simulation, and whilst we believe there are challenges to extending our methods to 3D, more heterogeneous media and elastic simulations tasks relevant to the Solid Earth, based on our first steps we believe further research is needed and the avenue could eventually yield practical and useful tools for Solid Earth science. We believe this ought to be done before any decision on whether this approach has the potential to scale up at production level for realistic Earth science applications.

We currently believe that extending our method to 3D is mostly a computational challenge, rather than requiring an entire conceptual recast to our network design. Similarly, our network structure is readily extendable to elastic simulation; the dimensionality of the network's inputs and outputs just needs to be increased. Whilst further research is needed to understand if these extensions are possible, we believe that our main contribution is to show that deep neural networks can simulate more complex 2D models for which no solutions other than numerically discretized ones exist and not just simple 1D models, which is a positive step towards using deep neural networks in real applications.

Proposed manuscript changes:

We propose to "reframe" the manuscript such that it includes a sort of "manifesto" or roadmap proposing and investigating deep learning for aiding Solid Earth simulation tasks:

1. Add more emphasis in the introduction on practical Solid Earth seismic simulation tasks, their challenges and how deep learning and our method fits into them
2. Add more detail in the discussion section on future research directions which could extend our methods so they can eventually be used in real Solid Earth simulation tasks. We would include new subsections discussing what we see are the main challenges towards this: 1) extending to more heterogeneous models, 2) extension to elastic simulation and 3) extension to 3D simulation.

Completed manuscript changes:

1. Abstract and introduction re-written to be more of a "manifesto" on the potential of deep learning for aiding simulation tasks in the Solid Earth sciences: added more emphasis on practical seismic simulation tasks, their challenges and how deep learning could aid them.
2. Added new subsections in the discussion, discussing in more detail 1) extension to more complex Earth models, 2) extension to elastic simulation and 3) extension to 3D simulation, as well as a summary on the potential practical applications of our method. In each case, we suggest how neural networks may be beneficial compared to the challenge of these extensions for conventional discrete methods.

Reviewer comments:

[Reviewer 1] • What do the authors foresee as the primary application(s) of their approach? The discussion seems to mainly envisage inversion-related use cases. Some comments on this:

1. *The 'fast seismic inversion' approach discussed in Section 3 is essentially a variant of the 'prior sampling' approach discussed in detail by Käufl et al (2016). The key strength and weakness of this as an inversion strategy is that all samples (i.e. the training set) are generated without reference to any observed data. This enables very rapid inference once data becomes avail- able, but it means that most training samples lie far from the observed data and are largely wasted from the perspective of any one inference. The end result is that inferences are considerably less well-constrained than would be possible with posterior sampling (see Fig. 9 of Käufl et al 2016). The bottom line is that prior sampling only seems a worthwhile strategy for (a) problems where time is of the essence, e.g.*

earthquake early warning, or (b) problems where the 'same' inference task needs to be solved many thousands of times with different data vectors.

2. *Using the learned model in Monte Carlo simulations seems superficially attractive, but comes with significant caveats. Fundamentally the inference remains entirely based on the information contained within the training dataset, and so all the limitations of prior sampling remain. The random walk would need to be constrained to only generate models compatible with the training data, if results are to be meaningful. Perhaps it would be possible to progressively retrain the learned simulation as the McMC proceeds, to ensure accuracy in relevant parts of the model space: this starts to move towards the Bayesian optimisation approaches discussed in (e.g.) Wang et al (2013). To play devil's advocate: if a problem is too complex to tackle using an McMC approach using physical simulations, can we really be confident that a learned model is sufficiently accurate to yield meaningful results? How big a training set is required to capture the full complexity of the physical problem?*

Some discussion and commentary on these issues, and other potential applications, would be appreciated.

<u>Author response:</u>

We agree that discussing in more detail the potential applications of our method would be valuable, as described in our previous response above. We also think that the specific issues highlighted on inversion in this comment are important to consider and we propose to expand our discussion on the pros and cons of our inversion technique to include these.

<u>Proposed manuscript changes:</u>

1. Add more detail in the discussion section on potential applications (as described in our response above)
2. Discuss in more detail the issues of prior sampling and limitations of the training dataset when discussing the pros and cons of our inversion technique.

<u>Completed manuscript changes:</u>

1. Added more emphasis on practical seismic simulation tasks (see previous response)
2. Added a subsection in the discussion on the limitations of our inversion technique (Section 4.4) which includes a discussion on the limitations of the training dataset.

<u>Reviewer comments:</u>

[Reviewer 1] • The main 'selling point' of the author's approach is that it enables seismograms to be generated significantly faster than would be possible using 'traditional' forward models. However, this comes with a number of caveats that I think need to be discussed more carefully.

1. *(As above) the author's approach is (I suspect) only effective for models that are sufficiently similar to those in the training set. The numerical forward code does not suffer from this restriction, and can handle complexities that aren't present in the authors' setup (e.g. anisotropy, variable density). How*

much of a speedup could be achieved by using a numerical code that had been designed with prior knowledge of the characteristics of the authors' training set? Put another way: the speedup could be made to seem even more impressive by using a code designed for a vastly more complex setting (e.g. SPECFEM3D) to build the training set. How fair is the comparison that is being presented?

2. *The headline speed comparisons ignore the costs of building a training set and then training the neural networks, which are significant. How many simulations does a user need to envisage performing before the author's approach becomes cost-effective overall? I think this is going to be a rather large number. Again, some discussion of the pros and cons of the author's approach would seem desirable.*

[Reviewer 1] • In general, we can make numerical simulations faster by introducing physical approximations. In such cases we typically have some intuition for how those approximations will impact upon results. Learned models offer a speedup without explicit physical approximations, but come with uncertainties that are difficult to quantify rigorously, and which may vary considerably depending on the particular set of inputs chosen. Would the authors like to comment on the pros and cons of the two different strategies for reducing computational costs?

[Reviewer 2] I also agree with most (but not all) of the comments from Reviewer 1 (Andrew Valentine), in particular his comment that a fair comparison for the layered-medium example would be between neural networks and other modelling methods that intrinsically assume a 1D Earth structure rather than full finite difference methods.

<u>Author response:</u>

For the first case where we consider simple layered 1D Earth models we agree that a fairer speed comparison would be against existing numerical methods which intrinsically assume 1D layers. For the 2D faulted Earth models we consider, to the best of our knowledge FD methods are the most efficient tools for these types of models, and therefore we believe this is a fair comparison.

We agree that there is more nuance to the comparison than just speed; for example, a discussion on where our approach break downs and the cost of training dataset generation and training would be useful.

<u>Proposed manuscript changes:</u>

1. For the case of 1D layered Earth models compare our approach (our WaveNet model) against the speed of an existing numerical method which intrinsically assumes 1D layers.
2. Include a discussion of where our approach breaks down and the relative cost of generating the training data and training our models.

<u>Completed manuscript changes:</u>

1. Added a comparison of the WaveNet model to an efficient quasi-analytic 2D ray tracing algorithm which assumes horizontally layered velocity models in Section 2.
2. Added a summary in the discussion which discusses the relative cost of training our models (Section 4.5).

Reviewer comments:

[Reviewer 2] 4. The paper appears to have committed the equivalent of an 'inverse crime'. If I understand correctly, the authors have trained networks in the forward and inverse directions using models with a certain parametrisation, and have tested the networks on models of exactly the same parametrization (Reviewer 1 touched on this too). While this would be reasonable if the models were themselves reasonably realistic, in a practical field like Geophysics I think it is necessary to test the networks on examples that lie outside of the range of the training set – not only using different models from those in the training set, but models that are not within the span of the algorithm used to generate the training set (as in the real Earth).

For example, (a) Earp and Curtis (2019, arXiv – https://arxiv.org/abs/1907.00541) and (b) Earp et al. (2019, arXiv – https://arxiv.org/abs/1908.09588) perform (a) probabilistic travel time tomographic imaging, and (b) probabilistic surface wave inversion for averaged shear wave structures with depth, using deep neural networks. The test examples using in both cases are created using a finer parametrization than was used in the network training set – thus the actually structure of the (synthetically) 'true' Earth is not attainable by the networks; nevertheless, they can be used as a useful check of whether in such cases (as in the real Earth) the networks behave sensibly – giving results that are spatial averages in some sense of the 'true' structure. To be clear I do not think that the above references are perfect in this regard and could certainly be improved (e.g., could use even more complex models for tests); nevertheless, the authors could usefully think about such tests for their work as it would strengthen the conclusions.

[Reviewer 1] • In any ML-based approach, the training data is central to the applicability of the method. The author's trained network appears effective for simulating waveforms in models that are generated using the same criteria as were used to make the training set. However, I suspect performance will be significantly worse for models that have significantly different character. This is something that deserves more discussion than it receives, perhaps with some examples. A particular concern in practical settings may be how an end-user can assess whether their input model is 'sufficiently close' to the training set.

Author response:

Whilst we were careful not to test the performance of the network using the same examples which were used to train the network, we only used examples drawn from the same distribution as the training distribution to test our networks. Because the training distribution only contains simple models we agree with the reviewers that this does not inform us on the performance of the network for models outside of the training set, and furthermore our current networks are likely to perform worse for such models.

For many seismological applications of forward and inverse modelling, we believe that the Earth models used are typically within a known range of parameters and therefore a training set could be constructed which appropriately spans the expected models, however we believe more research is needed in this area. For instance, dozens of published tomographic models could be used to define a base model range for future tomography modelling where thousands or millions of similar model simulations are needed.

Proposed manuscript changes:

1. We propose to test both our models on more realistic Earth models outside of the span of our training distribution and show the degradation observed. We also propose to suggest ways (such as nearest neighbour analysis) which could help a user determine how close an input model is to the training distribution, and potential future research ideas which could quantify the prediction uncertainty.
2. We will add more discussion on potential applications where we think this limitation is and is not permissible.

<u>Completed manuscript changes:</u>

1. Added generalisation tests for both the WaveNet and conditional autoencoder in Sections 2 and 3, testing them on velocity models significantly outside of the range of their training distribution. We also present a nearest neighbour approach to help assess how close an input velocity model is to the training distribution (Section 3).
2. Added subsections in the discussion which discuss the challenge of generalisation (Section 4.3) and potential applications where this is permissible (Section 4.5).

<u>Reviewer comments:</u>

[Reviewer 2] 2. The introduction is interesting and reviews some of the appropriate material, but is very sparsely justified, and as also stated by Reviewer 1, it does not include many key references. In my view, every sentence of a scientific work must either be a logical deduction from previous text, must have been deduced/proven in another paper, or may be an argument based on the material in another paper; in that latter two cases that paper needs to be discussed and cited. In this paper, the Introduction cites very few references and therefore contains unjustified (in the sense of, 'not justified') statements. Examples include:

One cannot write a paper on using neural networks to perform full waveform inversion (FWI) without citing Roeth and Tarantola (1994 – J. Geophys. Research). How does the FWI part of this paper improve on their work? That is not at all clear. There are many other papers using neural networks for imaging in Geophysics using waveforms or other types of data; you need to read and cite them, and describe how this work advances the field relative to those works. Currently the latter is not clear. The authors make the case for using neural networks for real-time applications – again first steps in this direction have already been taken (see Cao et al., 2019 – Geophysics, for example) and should be discussed.

[Reviewer 1] • Referencing, especially in the introduction, seems rather haphazard. If citations are to be given for broad, well-established topics such as the utility of seismic simulations in reservoir characterisation, I would expect these to be to major review papers or to 'classics': these are going to be most useful for a reader who is unfamiliar with the field. Without intending any criticism of the cited works, this does not really seem to be the case at present. Moreover, the authors' survey of the history of machine learning in geophysics is very short-sighted, ignoring anything more than a couple of years old. There are neural network papers in the geophysical literature from the late 1970s onwards, and it would be nice to see some acknowledgement of this body of work. Valentine & Trampert (2012) is probably the first instance of 'deep learning' in seismology, though the term had not been invented at that point (and we did not have the benefit of modern computational frameworks).

Author response:

We agree that the referencing in the introduction is sparse and that we only review examples of applications of deep learning to geophysics in the last couple of years, and that the paper would be stronger with more discussion of relevant work.

Proposed manuscript changes:

1. Ensure citations on broad, well-established topics in the introduction are major review papers or "classics".
2. Add and discuss more references where they are sparse, for example Roeth and Tarantola (1994 – J. Geophys. Research) when discussing FWI, and Cao et al., 2019 – Geophysics for real time simulation.
3. Include a fuller review of the applications of deep learning in geophysics, and include earlier examples, such as Valentine & Trampert (2012) and Devilee et al., (1999 – J. Geophys. Res).

Completed manuscript changes:

1. Citations changed and added in introduction to works which are major review papers, core textbooks or "classics".
2. Roeth and Tarantola (1994 – J. Geophys. Research) included when discussing FWI.
3. Added a review of the early applications of neural networks and deep learning in geophysics, including Valentine & Trampert (2012) and Devilee et al., (1999 – J. Geophys. Res) (Section 1.1).

Reviewer comments:

[Reviewer 2] 3. The authors promote the fact that they use 'deep learning', and their application certainly fits into that category. However, they must at least discuss why this is a positive feature of the method, and cite previous Geophysical applications of deep learning to support that discussion. Deep learning is usually defined to be the use of 4 or more layers within a neural network. While I agree with Reviewer 1 that his previous work (Valentine and Trampert) was an example of deep learning, the first that I know of in Geophysics was in fact Devilee et al., (1999 – J. Geophys. Res) – which came from the same university.

In my view there is therefore nothing new about the concept of deep learning: we were using it in Geophysics in the '90's. What has changed is the extent to which depth can be used to impose useful structure on networks (as the authors themselves have done in this manuscript – their Figure 9, and also in the paper cited by Reviewer 1); also the number of parameters that can now be used (the width of each layer) has increased hugely. In fact the number of parameters in the authors' application is relatively modest compared to some in machine learning literature, but is certainly comparable to other recent studies in Geophysics; the structure that the authors impose is both sensible and clearly useful in order to help to obtain stable results. These things should be discussed.

Author response:

We agree that deep learning is not a new technique in Geophysics and that two of the enabling recent advancements in this field are the ability to train deeper models with many more parameters. We agree the manuscript does not explicitly make this distinction and we propose to make this clearer.

Proposed manuscript changes:

1. Explicitly acknowledge that deep learning concepts are not new in geophysics and have been used in the past, and explain that, among other factors, such as the availability of more powerful hardware, advancements in training deep models with more parameters have enabled this work.

Completed manuscript changes:

1. Added explicit acknowledgement neural networks are not new in geophysics (Section 1.1) and explained the developments in deep learning which have driven the recent surge in deep learning research in geophysics (Section 1.1).

Deep learning for fast simulation of seismic waves in complex media

Ben Moseley[1], Tarje Nissen-Meyer[2], and Andrew Markham[1]

[1]Department of Computer Science, University of Oxford, UK

[2]Department of Earth Sciences, University of Oxford, UK

Correspondence: Ben Moseley (bmoseley@robots.ox.ac.uk)

Abstract. The simulation of seismic waves is a core task in many geophysical applications. Numerical methods such as Finite Difference (FD) modelling and Spectral Element Methods (SEM) are the most popular techniques for simulating seismic waves~~in complex media, but for many tasks~~, but disadvantages such as their computational cost ~~is prohibitively expensive~~prohibit their use for many tasks. In this work we ~~present two types of deep neural networks as fast alternatives for simulating seismic waves in horizontally layered and faulted 2D acoustic media. In contrast to the classical methods both networks~~ investigate the potential of deep learning for aiding seismic simulation in the Solid Earth sciences. We present two deep neural networks which are able to simulate the seismic response at multiple locations ~~within the media in a single inference step, without needing to iteratively model the seismic wavefield through time, resulting in~~ in horizontally layered and faulted 2D acoustic media an order of magnitude ~~reduction in simulation time. This speed improvement could pave the way to real-time seismic simulation and benefit seismic inversion algorithms based on forward modelling, such as full waveform inversion. Our~~ faster than traditional finite difference modelling. The first network is able to simulate ~~seismic waves~~ the seismic response in horizontally layered media ~~. We use~~ and uses a WaveNet network architecture ~~and show this is more accurate than a standard convolutional network design. Furthermore we show that seismic inversion can be carried out by retraining the network with its inputs and outputs reversed, offering a fast alternative to existing inversion techniques. Our~~ design. The second network is significantly more general than the first ~~; it~~ and is able to simulate ~~seismic waves~~ the seismic response in faulted media with arbitrary layers, fault properties and an arbitrary location of the seismic source on the surface of the media~~. It uses a convolutional autoencoder network designand is conditioned on the input source location. We investigate~~, using a conditional autoencoder design. We test the sensitivity of ~~different network designs and training hyperparameterson its simulation accuracy. We compare and contrast this network to the first network . To train both networks we introduce a time-dependent gain in the loss function which improves convergence. We discuss the relative merits of our approach with FD modelling and how our approach could be generalised to simulate more complexEarth models.~~ the accuracy of both networks to different network hyperparameters, and show that the WaveNet network can be retrained to carry out fast seismic inversion in the same media. We find that are there are challenges when extending our methods to more complex, elastic and 3D Earth models; for example the accuracy of both networks reduces when they are tested on models outside of their training distribution. We discuss further research directions which could address these challenges and potentially yield useful tools for practical simulation tasks.

1 Introduction

Seismic simulations are essential for ~~many areas of~~ addressing many outstanding questions in geophysics. In seismic hazards analysis, they are a key tool for quantifying the ground motion of potential earthquakes ~~(?)~~(Boore, 2003; Cui et al., 2010). In oil and gas prospecting, they allow the seismic response of hydrocarbon reservoirs to be modelled ~~(??)~~(Chopra and Marfurt, 2007; Lumley, 200 . In geophysical surveying ~~,~~ they show how the subsurface is illuminated by different survey designs ~~(?). Seismic simulations are used in global geophysics~~ (Xie et al., 2006). In global geophysics they are used to obtain snapshots of the Earth's interior dynamics ~~(?) and~~ by tomography (Hosseini et al., 2019; Bozdağ et al., 2016), to decipher source and path effects from individual seismograms ~~(?). They are also heavily used in seismic inversion , which estimates~~ (Krischer et al., 2017) and to model wave effects of complex structures (Thorne et al., 2020; Ni et al., 2002). In seismic inversion they are used to estimate the elastic properties of a medium given its seismic response ~~(?). In~~ (Tarantola, 1987; Schuster, 2017) and in Full Waveform Inversion ~~(FWI), a strategy widespread in the field of seismic imaging, simulations are used~~ (Fichtner, 2010; Virieux and Operto, 2009), a technique used to image the 3D structure of the subsurface, they are used up to tens of thousands of times to ~~iteratively estimate~~ improve on estimates of a medium's elastic properties~~(?).~~. In planetary science, seismic simulations play a central role in understanding novel recordings on Mars (Van Driel et al., 2019).

Numerous methods exist for simulating seismic waves~~(?). The most popular are~~, the most popular in fully heterogeneous media being Finite Difference (FD) ~~modelling~~ and Spectral Element Methods (SEM) ~~(??)~~(Igel, 2017; Moczo et al., 2007; Komatitsch and T . They are able to capture a large range of physics, including the effects of undulating solid-fluid interfaces ~~(?)~~(Leng et al., 2019), intrinsic attenuation ~~(?) and anisotropy (?)~~(van Driel and Nissen-Meyer, 2014a) and anisotropy (van Driel and Nissen-Meyer, 2014b). These methods solve for the propagation of the full seismic wavefield by discretising the elastodynamic equations of motion. For an acoustic heterogeneous medium these are given by the scalar linear equation of motion

$$\rho \nabla \cdot \left(\frac{1}{\rho} \nabla p \right) - \frac{1}{v^2} \frac{\partial^2 p}{\partial t^2} = -\rho \frac{\partial^2 f}{\partial t^2} \, , \tag{1}$$

where p is the acoustic pressure, f is a point source of volume injection (the seismic source), and $v = \sqrt{\kappa/\rho}$ is the velocity of the medium, with ρ the density of the medium and κ the adiabatic compression modulus ~~(?)~~(Long et al., 2013).

Whilst FD and spectral element methods are the primary means ~~to simulate seismic waves~~ of simulation in complex media, a major disadvantage of these methods is their computational cost ~~. FD modelling can involve millions of grid points~~ (Bohlen, 2002; Leng et al., 2016). Typical FD or SEM simulations can involve billions of degrees of freedom, and at each time step the wavefield must be iteratively updated at each 3D grid point. ~~This computational cost~~ For many practical geophysical applications this is often prohibitively expensive~~; supercomputers are typically required for large simulations (?). A faster method for seismic simulation would enable many applications~~. ~~For example , it would benefit real-time seismic simulation and seismic inversion methods which are heavily limited by the computational cost of forward simulation (?).~~ For example, in global seismology one may be interested in modelling waves up to 1 Hz in frequency to resolve small-scale heterogeneities in the mantle and a single simulation of this type with conventional techniques can cost around 40 million CPU hours

(Leng et al., 2019). At crustal scales, industrial seismic imaging requires wave modelling up to tens of Hertz in frequency carried out hundreds of thousands of times for each explosion in a seismic survey, and such requirements can easily fill the largest supercomputers on Earth. Any improvement in efficiency is welcome, not least due to the high financial and environmental costs of high-performance computing.

In some applications, large parts of the Earth model may be relatively smooth or simple. This simplicity can be taken advantage of, for example in the complexity-adapted SEM introduced by Leng et al. (2016), and can deliver a large speed-up compared to standard numerical modelling. Pseudo-analytical methods such as ray tracing and amplitude-versus-offset modelling (Aki and Richards, 1980; Vinje et al., 1993) are another approach which can provide significant speed-ups, albeit being approximate. We note that many applications are constrained and driven by a sparse set of observations on the surface of an Earth model. For these applications we are typically only interested in modelling the seismic response at these points to deciper seismic origin or the 3D structure beneath the surface, yet fully numerical methods still need to iterate the entire wavefield through all points in the model at all points in time. Any shortcut to avoid computing these massive 4D wavefields might lead to drastic efficiency improvements. In short, the points above suggest that alternative and advantageous methods to capture accurate wave physics may be possible for these challenging problems.

The field of ~~deep learning has~~ machine learning has seen an explosion in growth over the last decade. This has been primarily driven by advancements in deep learning, which has provided more powerful algorithms allowing much more difficult problems to be learned (Goodfellow et al., 2016). This progress has led to a surge in the use of deep learning techniques across many areas of science. In particular, deep neural networks have recently shown promise in ~~its~~ their ability to make ~~approximate~~ fast yet sufficiently accurate predictions of physical phenomena (Guo et al., 2016; Lerer et al., 2016; Paganini et al., 2018). These approaches are able to learn about highly non-linear physics and often offer much faster inference times than traditional simulation~~(??).~~ .

In this work we ~~study the ability~~ ask whether the latest deep learning techniques can aid seismic simulation tasks relevant to the Solid Earth sciences. We investigate the use of deep neural networks ~~for simulating seismic waves in 2D acoustic media~~and discuss the challenges and opportunities when using them for practical seismic simulation tasks. Our contribution is as follows;

– We present two deep neural networks which are able to simulate seismic waves in 2D acoustic media ~~. Both networks are~~ an order of magnitude faster than FD simulation. ~~Our~~ The first network uses a WaveNet network architecture ~~(?) and simulates~~ (van den Oord et al., 2016) and is able to accurately simulate the pressure response from a fixed point source at multiple locations in a horizontally layered velocity model. ~~We show that this network design is more accurate than a standard convolutional neural network. We also show that seismic inversion can be carried out by retraining the network with its inputs and outputs reversed, offering a fast alternative to existing inversion techniques, at least for layered media.~~

– ~~Our second network~~ The second is significantly more general~~than the first~~; it uses a conditional autoencoder network design and ~~it~~ is able to simulate ~~seismic waves~~ the seismic response at multiple locations in faulted media with arbitrary layers, fault properties and an arbitrary location of the source on the surface of the media. ~~The network is conditioned~~

~~on the input source location. We compare this network with our WaveNet network. We also investigate~~ In contrast to the classical methods both networks simulate the seismic response in a single inference step, without needing to iteratively model the seismic wavefield through time, resulting in a significant speed-up compared to FD simulation.

– We test the sensitivity of ~~its simulation accuracy with~~ the accuracy of both networks to different network designs~~and training hyperparameters.~~

– ~~For both networks we~~ , present a loss function with a time-varying gain which improves training convergence ~~. Finally, we discuss the relative merits of our approach with FD modelling how our approach could be generalised to simulate more complex~~ and show that fast seismic inversion in horizontal layered media can also be carried out by retraining the WaveNet network.

– We find challenges when extending our methods to more complex, elastic and 3D Earth models and discuss further research directions which could address these challenges and yield useful tools for practical simulation tasks.

In Section 2 we consider the simple case of simulating seismic waves in horizontally layered 2D acoustic Earth models using a WaveNet deep neural network. In Section 3 we move on to the task of simulating more complex faulted Earth models. In Section 4 we discuss the challenges of extending our approach and future research directions.

1.1 Related Work

1.2 ~~Related Work~~

The use of machine learning and neural networks in geophysics is not new (Van Der Baan and Jutten, 2000). For example, Murat and Rudman (1992) used neural networks to carry out automated first break picking, Dowla et al. (1990) used a neural network to discriminate between earthquakes and nuclear explosions and Poulton et al. (1992) used them for electromagnetic inversion of a conductive target. In seismic inversion, Röth and Tarantola (1994) used a neural network to estimate the velocity of 1D, layered, constant thickness velocity profiles from seismic amplitudes and Nath et al. (1999) used neural networks for cross-well travel-time tomography. However, these early approaches only used shallow network designs with small numbers of free parameters which limits the expressivity of neural networks and the complexity of problems they can learn about (Goodfellow et al., 2016).

~~Applying deep learning to physics problems is a burgeoning field of research and there is much active work in this area. ?~~ The field of machine learning has grown rapidly over the last decade, primarily because of advances in deep learning. The availability of larger datasets, discovery of methods which allow deeper networks to be trained and availability of more powerful computing architectures (mostly GPUs) has allowed much more complex problems to be learnt (Goodfellow et al., 2016), leading to a surge in the use of deep learning in many different research areas. In physics, Lerer et al. (2016) presented a deep convolutional network which could accurately predict whether randomly stacked wooden towers would fall or remain stable, given 2D images of the tower. ~~?~~ Guo et al. (2016) demonstrated that convolutional neural networks could estimate flow fields in complex Computational Fluid Dynamics (CFD) calculations two orders of magnitude faster than a traditional GPU-accelerated

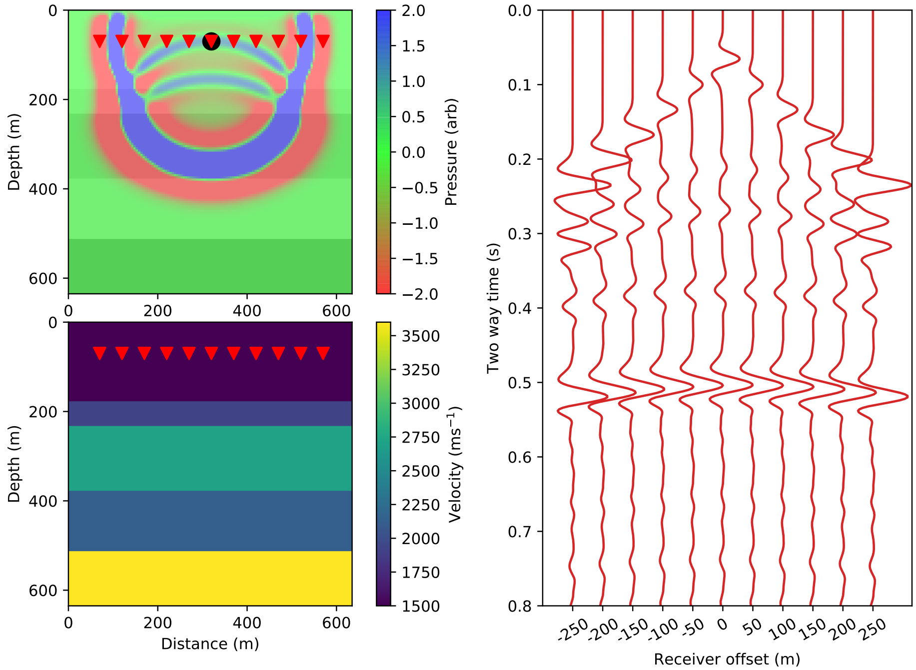

Figure 1. Ground truth FD simulation example. Left, top: A 20 Hz Ricker seismic source is emitted close to the surface and propagates through a 2D horizontally layered acoustic Earth model. The black circle shows the source location. 11 receivers are placed at the same depth as the source with a horizontal spacing of 50 m (red triangles). The full wavefield is overlain for a single snapshot in time. Note seismic reflections occur at each velocity interface. Left, bottom: The Earth velocity model. The Earth model has a constant density of 2200 kgm^{-2}. Right: The resulting ground truth pressure response recorded by each of the receivers, using FD modelling. A $t^{2.5}$ gain is applied to the receiver responses for display.

CFD solver and Paganini et al. (2018) used a conditional generative adversarial network to simulate particle showers in particle colliders.

~~Geophysicists are also starting to use deep learning for seismic-related problems. ?~~ A resurgence is occurring in geophysics too (Bergen et al., 2019; Kong et al., 2019). Early examples of deep learning include Devilee et al. (1999), who used deep probabilistic neural networks to estimate crustal thicknesses from surface wave velocities and Valentine and Trampert (2012) who used a deep autoencoder to compress seismic waveforms. More recently, Perol et al. (2018) presented an earthquake identification method using convolutional networks which is orders of magnitude faster than traditional techniques. In seismic inversion, ~~?~~ Araya-Polo et al. (2018) proposed an efficient deep learning concept for carrying out seismic tomography using the semblance of common mid-point receiver gathers as input. ~~?~~ Wu and Lin (2018) proposed a convolutional autoencoder network to carry out seismic inversion, whilst ~~?~~ Yang and Ma (2019) adapted a U-net network design for the same purpose. ~~?~~ Richardson (2018) demonstrated that a recurrent neural network framework can be used to carry out FWI. ~~?~~

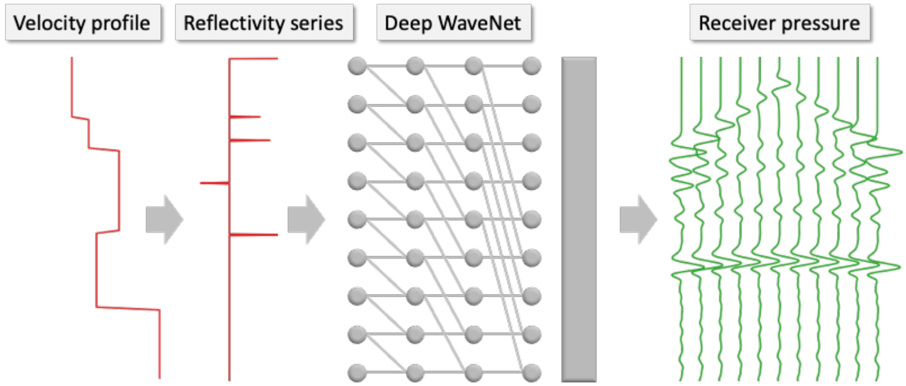

Figure 2. Our WaveNet simulation workflow. Given a 1D Earth velocity profile as input (left), our WaveNet deep neural network (middle) outputs a simulation of the pressure responses at the 11 receiver locations in Fig 1. The raw input 1D velocity profile sampled in depth is converted into its normal incidence reflectivity series sampled in time before being input into the network. The network is composed of 9 time-dilated causally-connected convolutional layers with a filter width of 2 and dilation rates which increase exponentially with layer depth. Each hidden layer of the network has same length as the input reflectivity series, 256 channels and a ReLU activation function. A final causally-connected convolutional layer with a filter width of 101 samples, 11 output channels and an identity activation is used to generate the output simulation.

Sun and Demanet (2018) showed a method for using deep learning to extrapolate low frequency seismic energy to improve the convergence of FWI algorithms.

~~For seismic simulation ?~~ In seismic simulation Zhu et al. (2017) presented a multi-scale convolutional network for predicting the evolution of the full seismic wavefield in heterogeneous media. Their method was able to approximate wavefield kinematics over multiple time steps, although it suffered from the accumulation of error over time and did not offer a reduction in computational time. ~~?~~ Moseley et al. (2018) showed that a convolutional network with a recursive loss function can simulate the full wavefield in horizontally layered acoustic media. ~~?~~ Krischer and Fichtner (2017) used a generative adversarial network to simulate seismograms from radially symmetric and smooth Earth models.

In this work we present ~~a fast method~~ fast methods for simulating seismic waves in ~~faulted and arbitrarily layered~~ horizontally layered and faulted 2D acoustic media, which offer a significant reduction in computation time compared to Zhu et al. (2017) . We also present a fast method for seismic inversion of horizontally layered acoustic media, which is more general than the original approach proposed by Röth and Tarantola (1994) because it is able to invert velocity models with varying numbers of layers and varying layer thicknesses. We restrict ourselves to 2D acoustic media and discuss implications for 3D elastic media below.

2 Fast seismic simulation in 2D horizontally layered acoustic media using WaveNet

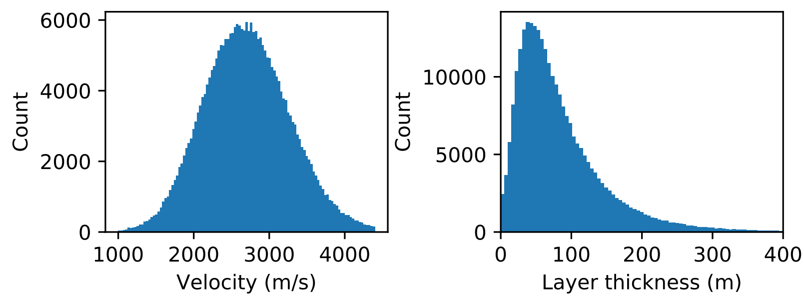

Figure 3. Distribution of layer velocity and layer thickness over all examples in the training set.

~~In this section we consider~~

First we consider the simple case of simulating seismic waves in horizontally layered 2D acoustic Earth models. We train a deep neural network with a WaveNet architecture to simulate the seismic response recorded at multiple receiver locations in the Earth model, horizontally offset from a point source emitted at the surface of the model. As mentioned above, many seismic applications are concerned with sparse observations similar to this setup. Whilst we concentrate on simple velocity models here, more complex faulted Earth models are considered in Section 3.

An example simulation we wish to learn is shown in Fig. 1 ~~.~~

~~Our~~ and our simulation workflow is shown in Fig. 2. The input to the network is a horizontally layered velocity profile and the output of the network is a simulation of the pressure response recorded at each ~~of~~ receiver location. We will now discuss deep neural networks, our WaveNet architecture, our simulation workflow and our training methodology in more detail below.

2.1 Deep neural networks and the WaveNet network

~~Distribution of layer velocity and layer thickness over all examples in the training set.~~

A neural network is a network of simple computational elements, known as neurons, which perform mathematical operations on multidimensional arrays, or tensors ~~.~~ (Goodfellow et al., 2016). The composition of these neurons together defines a mathematical function of the network's input. Each neuron has a set of free parameters, or weights, which are tuned using optimisation~~techniques such that the networkcan learn a function of its inputs~~, allowing the network's function to be learned, given a set of training data. In deep learning, ~~these~~ the neurons are typically arranged in multiple layers, which allows the network to learn highly non-linear functions.

A standard building block in deep learning is the convolutional layer, where all neurons in the layer share the same weight tensor and each neuron has a limited field of view of its input tensor. The output of the layer is ~~mathematically~~ achieved by cross correlating the weight tensor with the input tensor. Multiple weight tensors, or filters, can be used to increase the depth of the output ~~layer. Such networks~~ tensor. Such designs have achieved state of the art performance across a wide range of machine learning tasks (Gu et al., 2018).

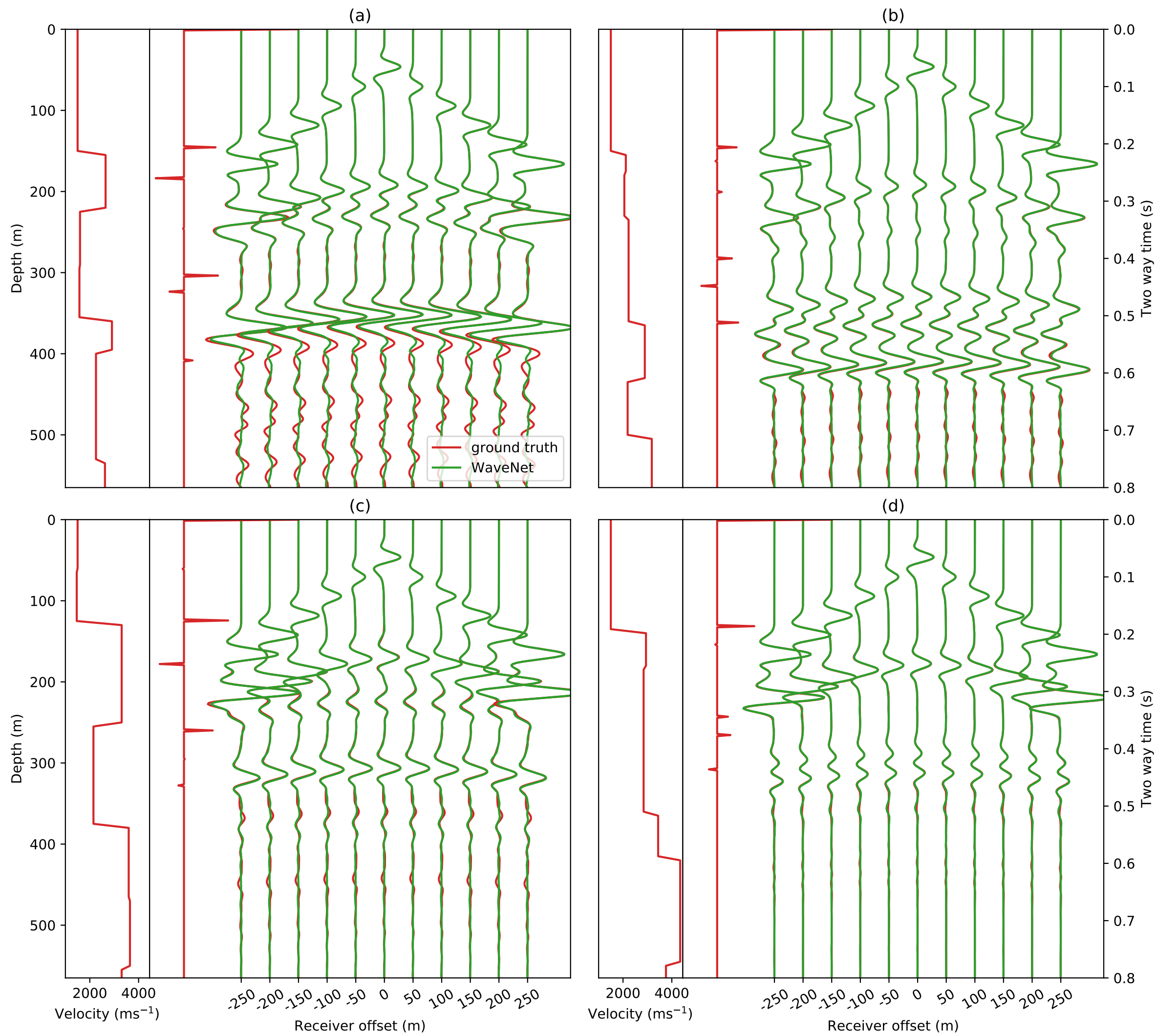

Figure 4. WaveNet simulations for 4 randomly selected examples in the test set. Red shows the input velocity model, its corresponding reflectivity series and the ground truth pressure response from FD simulation at the 11 receiver locations. Green shows the WaveNet simulation given the input reflectivity series for each example. A $t^{2.5}$ gain is applied to the receiver responses for display.

The WaveNet network ~~(?) proposed~~ proposed by van den Oord et al. (2016) makes multiple alterations to the standard convolutional layer for its use with time series. Each convolutional layer is made causal; that is, the receptive field of each neuron only contains samples from the input layer whose sample times are before or the same as the current neuron's sample time. Furthermore the WaveNet exponentially dilates the width of its causal connections with layer depth. This allows the field of view of its neurons to increase exponentially with layer depth, without ~~increasing the number of weights in the network~~needing a large number of layers. These modifications are made to honour time series prediction tasks which are causal and to better

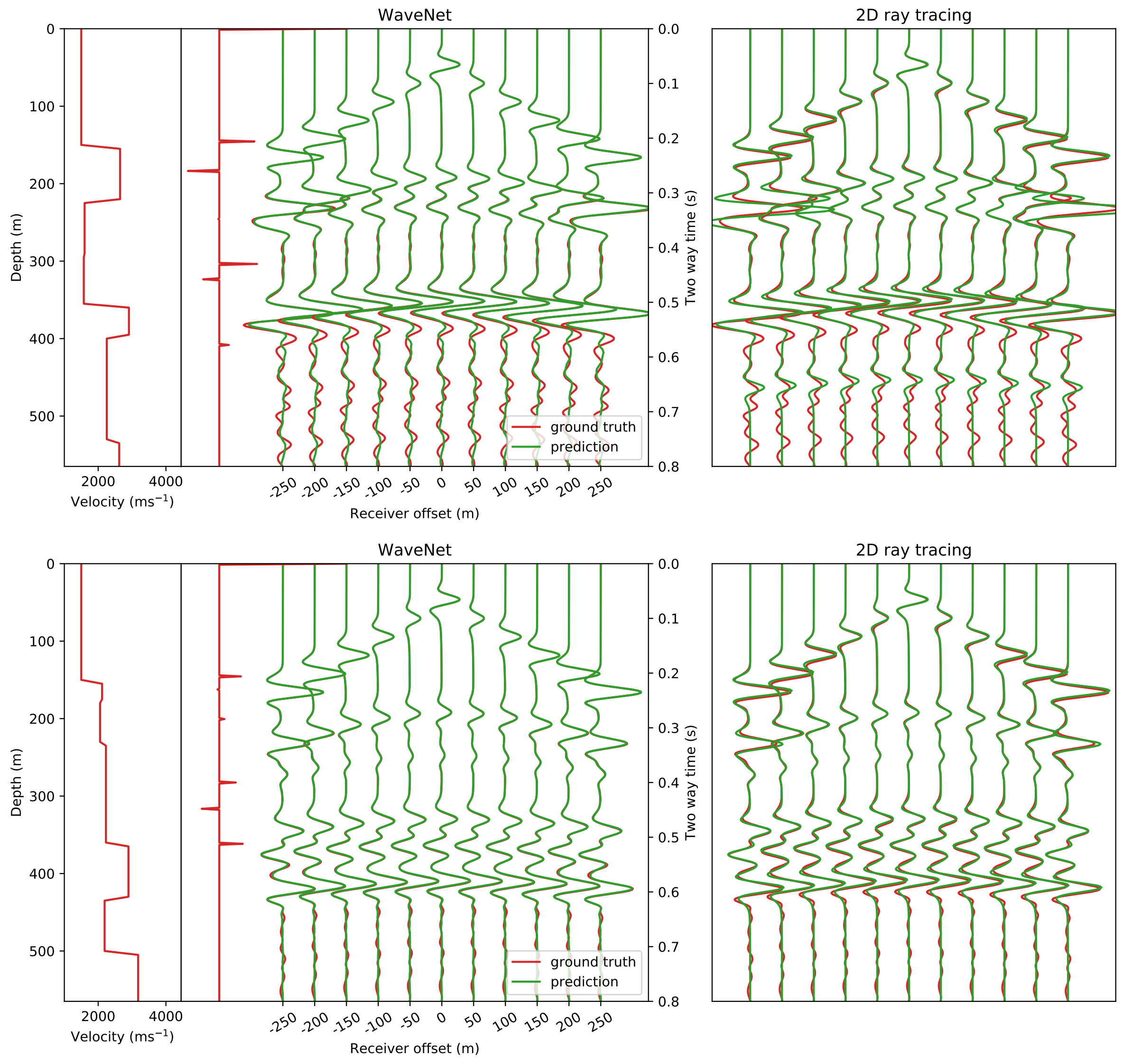

Figure 5. Comparison of WaveNet simulation to 2D ray tracing. We compare the WaveNet simulation to 2D ray tracing for 2 of the examples in Fig 4. Red shows the input velocity model, its corresponding reflectivity series and the ground truth pressure responses from FD simulation. Green shows the WaveNet simulation (left) and 2D ray tracing simulation (right). A $t^{2.5}$ gain is applied to the receiver responses for display.

model input data which varies over multiple time scales. The WaveNet network recently achieved state of the art performance in text to speech synthesis.

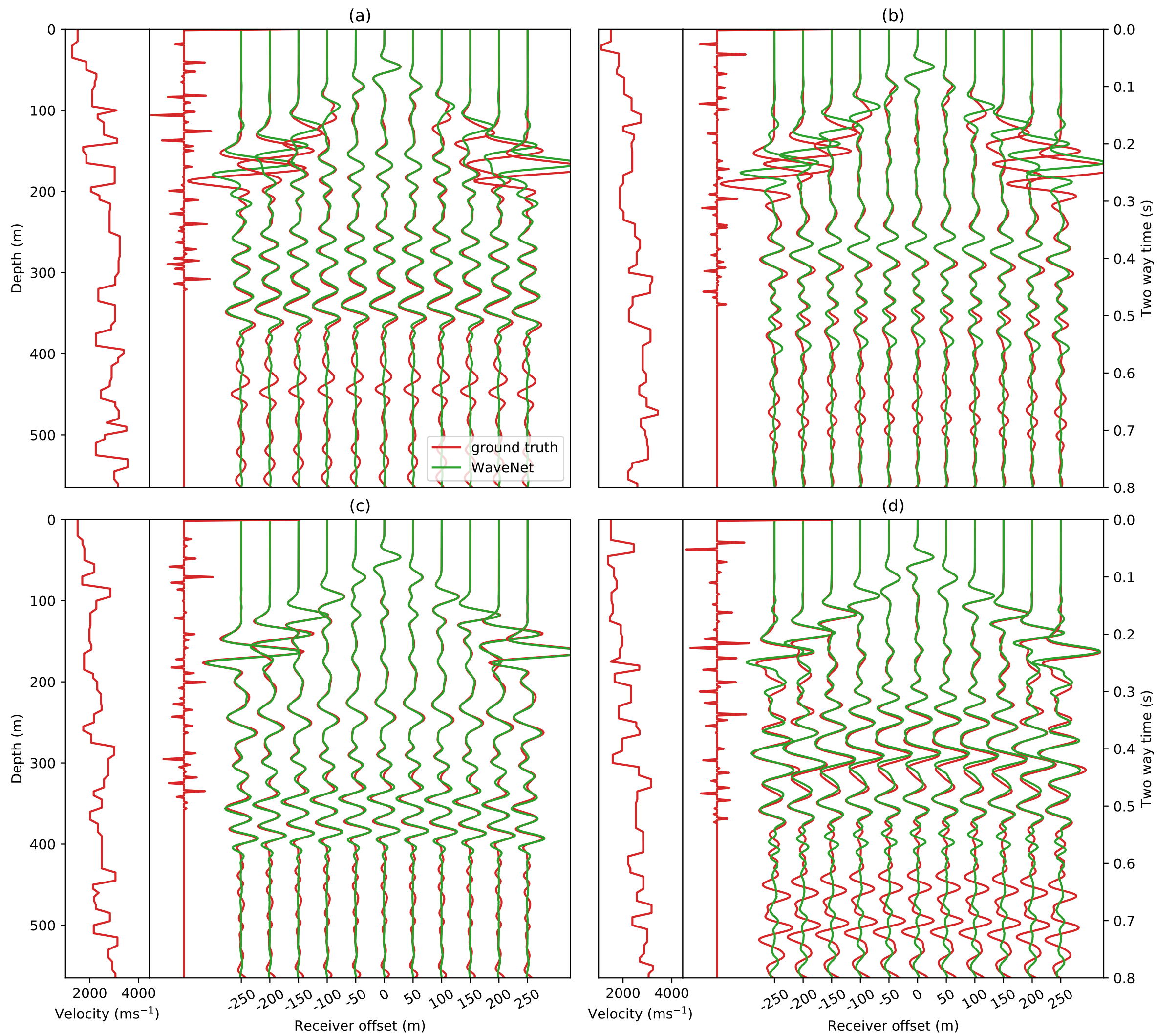

Figure 6. Generalisation ability of the WaveNet. The WaveNet simulations (green) for 4 velocity models with a much smaller average layer thicknesses than the training distribution are compared to ground truth FD simulation. Red shows the input velocity model, its corresponding reflectivity series and the ground truth pressure responses from FD simulation.

2.2 Simulation workflow

Our workflow consists of a preprocessing step, where we convert each input velocity model into its corresponding normal incidence reflectivity series sampled in time (Fig. 2, left), followed by a simulation step, where ~~we use the~~ it is passed to a WaveNet network to simulate the pressure response recorded by each receiver (Fig. 2, middle).

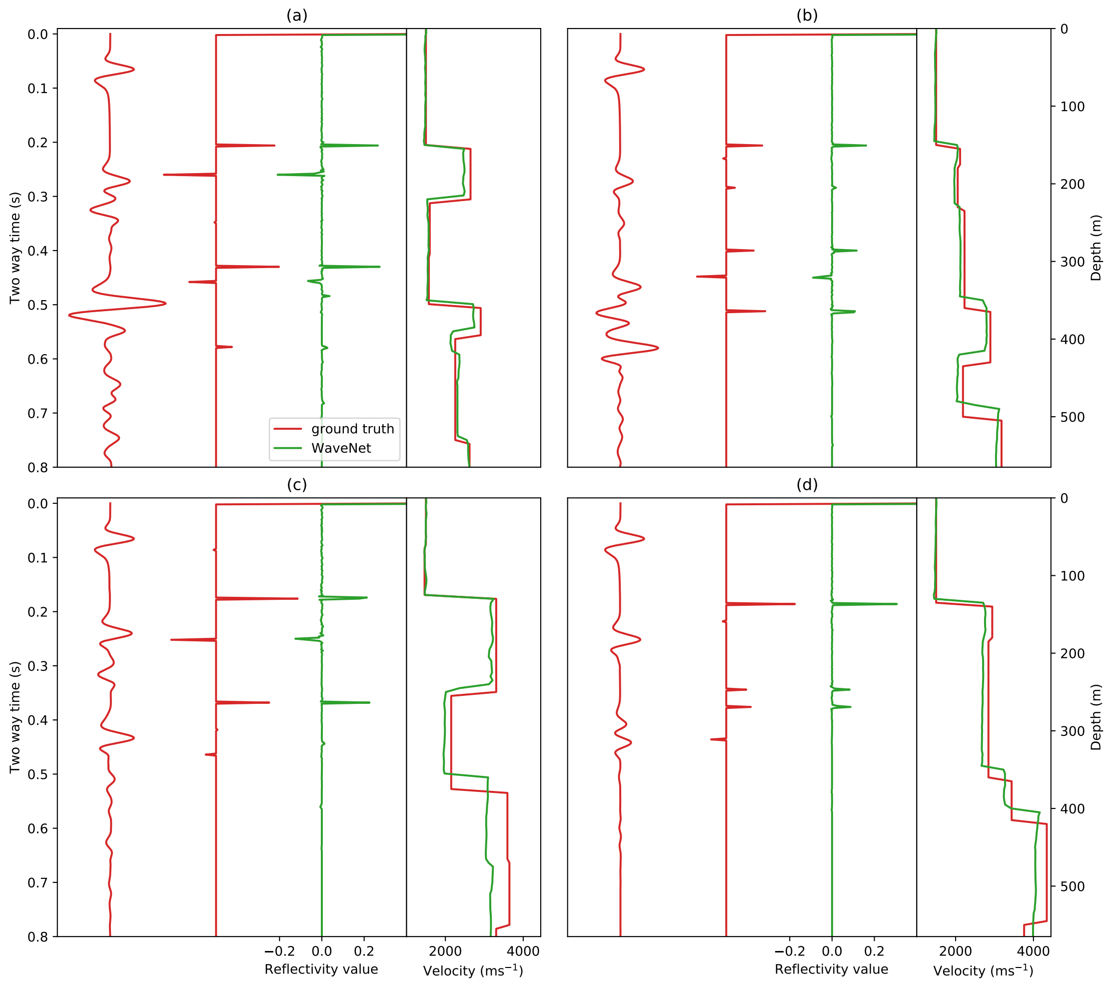

Figure 7. Top: Inverse WaveNet predictions for 4 examples in the test set. Red shows the input pressure response at the zero-offset receiver location, the ground truth reflectivity series and its corresponding velocity model. Green shows the inverse WaveNet reflectivity series prediction and the resulting velocity prediction.

~~We chose to convert the velocity model to its corresponding normal incidence reflectivity series because, for the case of horizontally layered Earth models, the normal incidence reflectivity series and the pressure responses are causally correlated.~~ The reflectivity series is typically used in exploration seismology ~~(?)~~ (Russell, 1988) and contains values of the ratio of the amplitude of the reflected wave to the incident wave for each interface in a velocity model. For acoustic waves at normal incidence, these values are given by

$$R = \frac{\rho_2 v_2 - \rho_1 v_1}{\rho_2 v_2 + \rho_1 v_1}\,, \tag{2}$$

where ρ_1,v_1 and ρ_2,v_2 are the densities and P-wave velocities across the interface. The series is usually expressed in time and each reflectivity value occurs at the time at which the primary reflection of the source from the corresponding velocity interface arrives at a given receiver. ~~These~~ The arrival times can be computed by carrying out a depth-to-time conversion of the reflectivity values using the input velocity model.

We chose to convert the velocity model to its reflectivity series and use the causal WaveNet architecture to constrain our workflow. For horizontally layered velocity models and receivers horizontally offset from the source, the receiver pressure recordings are causally correlated to the normal incidence reflectively series of the zero-offset receiver. Intuitively, a seismic reflection recorded after a short time has only travelled through a shallow part of the velocity model and the pressure responses are at most dependent on the past samples in this reflectivity series.

~~We constrain our workflow such that it honours this causal correlation, by~~ By preprocessing the input velocity model into its corresponding reflectivity series and ~~by~~ using the causal WaveNet network to simulate the receiver response ~~.~~ we constrain our workflow so that it honours this causal correlation.

We input the 1D profile of a 2D horizontally layered velocity model, with a depth of 640 m and a step size of 5 m. We use Eq. 2 and a standard 1D depth to time conversion to convert the velocity model into its normal incidence reflectivity series. The output reflectivity series has a length of 1 s and a sample rate of 2 ms. An example output reflectivity series is shown in Fig. 2 (left).

~~This~~ The reflectivity series is ~~then passed to our~~ passed to the WaveNet network, which contains 9 causally-connected convolutional layers (Fig. 2, middle). Each convolutional layer has the same length as the input reflectivity series, 256 hidden channels, a receptive field width of 2 samples and a Rectified Linear Unit (ReLU) activation function ~~(?)~~(Nair and Hinton, 2010). Similar to the original WaveNet design, we use exponentially increasing dilations at each layer to ensure that the first sample in the input reflectivity series is in the receptive field of the last sample of the output simulation. We add a final causally-connected convolutional layer with 11 output channels, a filter width of 101 samples and an identity activation to generate the output simulation, where each output channel corresponds to a receiver prediction. This results in the network having 1,333,515 free parameters in total.

2.3 Training data generation

To train the network, we generate 50,000 synthetic ground truth example simulations using the SEISMIC_CPML code, which performs 2$^{\mathrm{nd}}$-order acoustic FD modelling ~~(?)~~(Komatitsch and Martin, 2007). Each example simulation uses a randomly sampled 2D horizontally layered velocity model with a width and depth of 640 m and a sample rate of 5 m in both directions. (Fig. 1, bottom left). For all simulations we use a constant density model of 2200 kgm^{-2}.

In each simulation the layer velocities and layer thickness are randomly sampled from log-normal distributions. We also add a small velocity gradient randomly sampled from a normal distribution to each model such that the velocity values tend to increase with depth, to be more Earth-realistic. The distributions over layer velocities and layer thicknesses for the entire training set are shown in Fig. 3.

We use a 20 Hz Ricker source emitted close to the surface and record the pressure response at 11 receiver locations placed symmetrically around the source, horizontally offset every 50 m (Fig. 1, top left). We use a convolutional perfectly matched layer boundary condition such that waves which reach the edge of the model are absorbed with negligible reflection. We run each simulation for 1 s and use a 0.5 ms sample rate to maintain accurate FD fidelity. We downsample the resulting receiver pressure responses to 2 ms before using them for training.

We run 50,000 simulations and extract a training example from each simulation, where each training example consists of a 1D layered velocity profile and the recorded pressure response at each of the 11 receivers. We withhold 10,000 of these examples as a validation set to measure the generalisation performance of ~~our~~ the network during training.

2.4 Training process

The network is trained using the Adam stochastic gradient descent algorithm ~~(?)~~(Kingma and Ba, 2014). This algorithm computes the gradient of a loss function with respect to the free parameters of the network over a randomly selected subset, or batch, of the training examples. This gradient is used to iteratively update the parameter values, with a step size controlled by a learning rate parameter. We ~~use~~ propose a L2 loss function with time-varying gain function for this task, given by

$$L = \frac{1}{N}\|G(\hat{Y}-Y)\|_2^2 \,, \tag{3}$$

where $\hat{Y}$ is the simulated receiver pressure response from the network, Y is the ground truth receiver pressure response from FD modelling and N is the number of training examples in each batch. The gain function G has the form $G = t^g$ where t is the sample time and g is a hyperparameter which determines the strength of the gain. We add this to empirically account for the ~~spherical spreading~~ attenuation of the wavefield ~~by~~ caused by spherical spreading, by increasing the weight of samples at later times. In this Section we use a fixed value of $g = 2.5$. We use a learning rate of $1\text{x}10^{-5}$, a batch size of 20 training examples and run training over 500,000 gradient descent steps.

2.5 ~~Results~~Comparison to 2D ray tracing

~~WaveNet simulations for 4 randomly selected examples in the test set. Red shows the input velocity model, its corresponding reflectivity series and the ground truth pressure response from FD simulation at the 11 receiver locations. Green shows the WaveNet simulation given the input reflectivity series for each example. A~~ $\cancel{t^{2.5}}$ ~~gain is applied to the receiver responses for display.~~ We compare the WaveNet simulation to an efficient, quasi-analytical 2D ray-tracing algorithm which assumes horizontally layered media. We modify the 2D horizontally layered ray-tracing bisection algorithm from the CREWES seismic modelling library (Margrave and Lamoureux, 2018) to include Zoeppritz modelling of the reflection and transmission coefficients at each velocity interface (Aki and Richards, 1980) and 2D spherical spreading attenuation (Gutenberg, 1936; Newman, 1973) during ray tracing. The output of the algorithm is a primary reflectivity series for each receiver, which we convolve with the source signature used in FD modelling to obtain an estimate of the receiver responses.

~~Comparison of WaveNet simulation to 1D convolutional model. We compare our WaveNet simulation for 3 of the examples in Fig 4 to a simple 1D convolutional model. Red shows the input velocity model, its corresponding reflectivity series and the ground truth pressure response at the zero-offset receiver. Green shows the WaveNet simulation at the zero-offset receiver and blue shows the 1D normal incidence convolutional model. Bottom right: histogram of the average absolute amplitude difference between the ground truth FD simulation and the zero-offset simulation from the WaveNet and 1D convolutional model, over the test set of 1000 examples. A $t^{2.5}$ gain is applied to the receiver responses for display.~~

2.6 Inversion workflow

~~During training the loss~~ As an additional test, we are also able to retrain the WaveNet network to carry out fast seismic inversion in the same media, which offers a fast alternative to existing inversion algorithms. We retrain the WaveNet network with its inputs and output reversed. Its input is now a set of 11 recorded receiver responses and its output is a prediction of the corresponding normal incidence reflectivity series. To recover a prediction of the velocity model we carry out a standard 1D time-to-depth conversion of the output reflectivity values followed by integration. We use the same WaveNet architecture described in Sect. 2.2, except that we invert its structure to maintain the causal correlation between the receiver responses and reflectivity series. We also use 128 instead of 256 hidden channels for each hidden layer. We use exactly the same training data and training strategy described in Sect. 2.3 and 2.4, except that we now use the loss function given by

$$L = \frac{1}{N} \| \hat{R} - R \|_2^2 , \tag{4}$$

where R is the true reflectivity series and $\hat{R}$ is the predicted reflectivity series.

2.7 Results

Whilst training the WaveNet the losses over the training and validation datasets converge to similar values, suggesting the network is generalising well to examples in the validation dataset. To assess the performance of the trained network, we generate a random test set of 1000 unseen examples. The ~~WaveNet~~ simulations for 4 randomly selected examples from this test set are compared to the ground truth FD modelling simulation in Fig. 4. We also compare ~~our~~ the WaveNet simulation to ~~a simple 1D convolutional approximation of the zero-offset receiver response at normal incidence (?), given by $\tilde{Y} = R * S$, where R is the reflectivity series in time and S is the source signature, shown in~~ 2D ray tracing in Fig. 5.

For nearly all time samples ~~our~~ the network is able to simulate the receiver pressure responses. ~~Unlike the 1D convolutional model, the~~ The WaveNet is able to predict the Normal Moveout (NMO) of the primary layer reflections with receiver offset, the direct arrivals at the start of each receiver recording and the spherical spreading loss of the wavefield over time, though ~~we notice~~ the network struggles to accurately simulate the multiple reverberations at the end of the receiver recordings. We plot the histogram of the average absolute amplitude difference between the ground truth FD simulation and the ~~zero-offset~~ simulation from the WaveNet and ~~1D convolutional model~~ 2D ray tracing over the test set in Fig. ~~5. We~~ A1 (bottom right) and observe that the WaveNet simulation has a lower average ~~loss than the 1D convolutional model~~difference than 2D ray tracing.

We ~~also investigate the accuracy of the WaveNet simulation with different network designs.~~ compare the sensitivity of the network's accuracy to two different convolutional network designs in Fig. A1~~compares the WaveNet simulation to the simulations from using two different convolutional network designs. Both convolutional~~ . Their main differences to the WaveNet design is that both networks use standard rather than causal convolutional layers and the second network uses exponential dilations whilst the first does not. Both networks have 9 convolutional layers, each with 256 hidden channels, filter sizes of 3, ReLU activations for all hidden layers and an identity activation function for the output layer~~. The second network uses exponential dilations whilst the first does not. Both networks have~~ , with 1,387,531 free parameters in total. We observe that the convolutional network without dilations does not converge during training~~. We plot the histogram of the~~ , whilst the dilated convolutional network has a higher average absolute amplitude difference ~~between the~~ over the test set from the ground truth FD simulation ~~and the simulations from the WaveNet and the dilated convolutional network over the test set. The dilated convolutional network has a higher average loss~~ than the WaveNet network ~~.~~ (Fig. A1 (bottom right)).

~~We measure the average time taken to generate 100 simulations using the SEISMIC_CPML library on a single core of a 2.2 GHz Intel Core i7 processor to be 73 ± 1 s. Using the same core the WaveNet network is able to generate 100 simulations in an average time of 3.79 ± 0.03 s (19 times quicker). Using the TensorFlow library (TensorFlow, 2015) and a Nvidia Tesla K80 GPU produces simulations with an average time of 0.133 ± 0.001 s (549 times quicker). This speedup is likely to be higher than if the GPU was used for accelerating existing numerical methods (?). The WaveNet network takes approximately 12 hours to train on one Nvidia Tesla K80 GPU, although this training step is only required once and subsequent simulation steps are fast.~~

3 ~~Fast seismic inversion in 2D horizontally layered acoustic media using WaveNet~~

~~Comparison of network architecture on simulation accuracy. Top left shows the WaveNet simulated pressure response for a randomly selected example in the test set (green) compared to ground truth (red). Top right and bottom left show the simulated response when using convolutional network designs with and without exponential dilations. Bottom right: histogram of the average absolute amplitude difference between the ground truth FD simulation and the simulations from the WaveNet and the dilated convolutional network, over the test set of 1000 examples. A $t^{2.5}$ gain is applied to the receiver responses for display.~~

~~Top: Inverse WaveNet predictions for 4 examples in the test set. Red shows the input pressure response at the zero-offset receiver location, the ground truth reflectivity series and its corresponding velocity model. Green shows the inverse WaveNet reflectivity series prediction and the resulting velocity prediction.~~

~~In this section we retrain our WaveNet network to carry out fast seismic inversion in horizontally layered 2D acoustic Earth models. This offers a fast alternative to existing inversion algorithms.~~

2.1 ~~Inversion workflow~~

~~We are able to perform seismic inversion in the same media by retraining the WaveNet network with its inputs and output reversed. Its input is now a set of 11 recorded receiver responses and its output is a~~ The generalisation ability of the WaveNet outside of its training distribution is tested in Fig. 6. We generate four velocity models with a much smaller average layer thickness than the training set and compare the WaveNet simulation to the ground truth FD simulation. We find that the WaveNet is able to make an accurate prediction of the ~~corresponding normal incidence reflectivity series. To recover a prediction of the velocity model, we carry out a standard 1D time-to-depth conversion of the output reflectivity values followed by integration. We use the same WaveNet architecture described in Sect. 2.1, except that we invert its structure to maintain the causal correlation between the receiver responses and reflectivity series. We also use 128 instead of 256 hidden channels for each hidden layer. We use exactly the same training data and training strategy described in Sect. 2.3 and 2.4, except that we now use the loss function given by~~

$$\cancel{L = \frac{1}{N} \|\hat{R} - R\|_2^2 ,}$$

~~where R is the true reflectivity series and $\hat{R}$ is the predicted reflectivity series~~seismic response, but it struggles to simulate the multiple reflections and sometimes the interference between the direct arrival and primary reflections.

2.1 ~~Results~~

~~During training the loss~~ When training the inverse WaveNet the losses over the training and validation datasets converge to similar values and we test the performance of the trained network using a test set of 1000 unseen examples. ~~Our predictions~~ Predictions of the reflectivity series and velocity models for 4 randomly selected examples from this test set are shown in Fig. 7. The inverse WaveNet network is able to predict the underlying velocity model for each example. We observe that in some cases small velocity errors propagate with depth, which is likely a result of the integration of the reflectivity series.

We ~~measure~~ compare the average time taken to generate 100 ~~velocity predictions~~ simulations (or 100 velocity model predictions for the inverse WaveNet) to FD simulation and 2D ray tracing in Table 1. We find that on a single ~~core of a 2.2 GHz Intel Core i7 processor to be 1.27 ± 0.02 s. On a Nvidia Tesla K80 GPU this reduces to 0.051 ± 0.001 s. This is~~ CPU core the WaveNet is 19 times faster than FD simulation, and using a GPU and the TensorFlow library (TensorFlow, 2015) it is 549 times faster. This speedup is likely to be higher than if the GPU was used for accelerating existing numerical methods (Rietmann et al., 2012). In this case, the specialised 2D ray tracing algorithm offers a similar speed up to the WaveNet network. The inverse WaveNet is able to produce velocity predictions in the same order of magnitude time as the forward network, which is likely to be a fraction of the time needed for existing seismic inversion algorithms which rely on forward simulation. ~~We note the~~ Its prediction time is faster than the forward WaveNet ~~network because the inverse network~~ because it has less hidden channels in its architecture and therefore requires less computation. Both networks take approximately 12 hours to train on one Nvidia Tesla K80 GPU, although this training step is only required once and subsequent simulation steps are fast.

3 Fast seismic simulation in 2D faulted acoustic media using a conditional autoencoder

~~Our WaveNet network~~

Method	Average CPU time (s)	Average GPU time (s)	Training time (days)
2D FD simulation	73 ± 1 (1x)	-	-
2D ray tracing	2.2 ± 0.1 (33x)	-	-
WaveNet (forward)	3.79 ± 0.03 (19x)	0.133 ± 0.001 (549x)	0.5
Conditional autoencoder	3.3 ± 0.1 (22x)	0.180 ± 0.003 (406x)	4
WaveNet (inverse)	1.27 ± 0.02	0.051 ± 0.001	0.5

Table 1. Speed comparison of simulation and inversion methods. The time shown is the average time taken to generate 100 simulations (or 100 velocity predictions for the inverse WaveNet) on either a single core of a 2.2 GHz Intel Core i7 processor or a Nvidia Tesla K80 GPU. For simulation methods the speed up factor compared to FD simulation is shown in brackets.

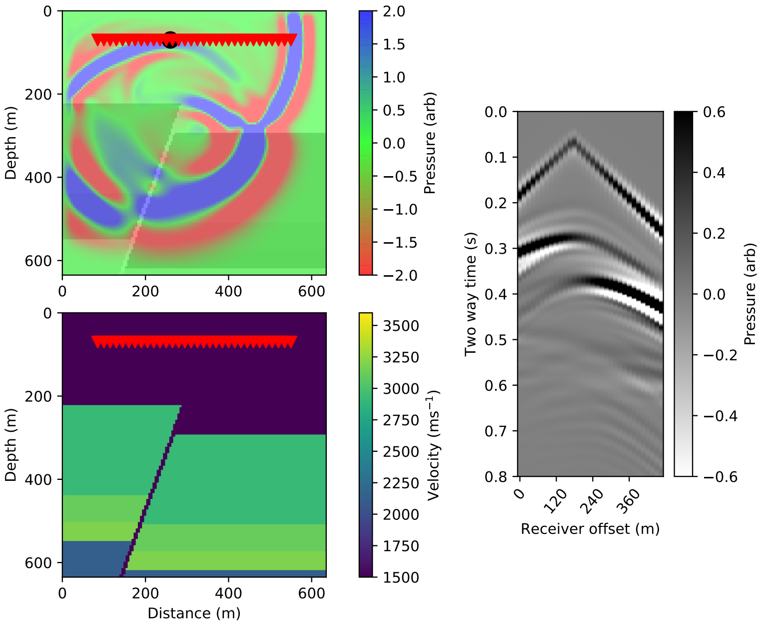

Figure 8. Ground truth FD simulation example, with a 2D faulted media. Left, top: The black circle shows the source location. 32 receivers are placed at the same depth as the source with a horizontal spacing of 15 m (red triangles). The full wavefield pressure is overlain for a single snapshot in time. Left, bottom: The Earth velocity model. Right: The resulting ground truth pressure response recorded by each receiver, using FD modelling. A $t^{2.5}$ gain is applied to the receiver responses for display.

The WaveNet architecture we implemented is limited in that it is only able to simulate horizontally layered Earth models. In this section we present a second network which is significantly more general; it simulates seismic waves in 2D faulted acoustic media with arbitrary layers, fault properties and an arbitrary location of the seismic source on the surface of the media.

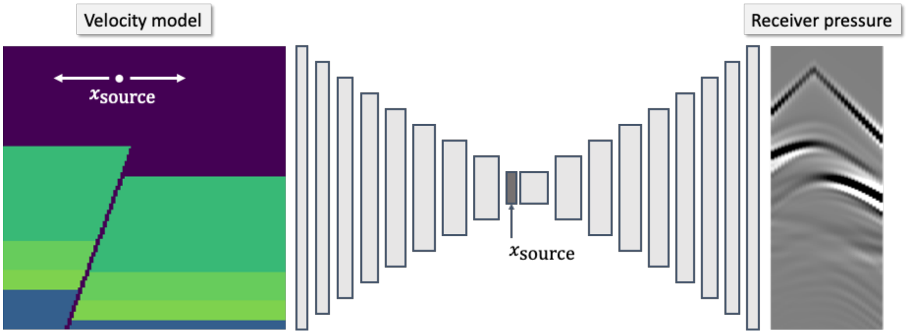

Figure 9. Our conditional autoencoder simulation workflow. Given a 2D velocity model and source location as input, a conditional autoencoder network outputs a simulation of the pressure responses at the receiver locations in Fig. 8. The network is composed of 24 convolutional layers and concatenates the input source location with its latent vector.

This is a much more challenging task to learn for multiple reasons. Firstly, the media varies along both dimensions and the resulting seismic wavefield has more complex kinematics than the wavefields in horizontally layered media. Secondly, we allow the output of the network to be conditioned on the input source location which requires the network to learn the effect of the source location. Thirdly, we input the velocity model directly into the network without conversion to a reflectivity series beforehand; the network must learn to carry out its own depth to time conversion to simulate the receiver responses. We chose this approach over our WaveNet workflow because we note that for non-horizontally layered media the pressure responses are not causally correlated to the normal incidence reflectivity series in general and ~~therefore the causality assumption in our WaveNet workflow~~ our previous causality assumption does not hold.

Similar to Section 2, we simulate the seismic response recorded by a set of receivers horizontally offset from a point source emitted within the Earth model. An example simulation we wish to learn is shown in Fig. 8. We will now discuss ~~our~~ the network architecture and training process in more detail below.

~~Ground truth FD simulation example, with a 2D faulted media. Left, top: The black circle shows the source location. 32 receivers are placed at the same depth as the source with a horizontal spacing of 15 m (red triangles). The full wavefield pressure is overlain for a single snapshot in time. Left, bottom: The Earth velocity model. Right: The resulting ground truth pressure response recorded by each receiver, using FD modelling. A $t^{2.5}$ gain is applied to the receiver responses for display.~~

~~Our conditional autoencoder simulation workflow. Given a 2D velocity model and source location as input, a conditional autoencoder network outputs a simulation of the pressure responses at the receiver locations in Fig. 8. The network is composed of 24 convolutional layers and concatenates the input source location with its latent vector.~~

~~Conditional autoencoder simulations for 8 randomly selected examples in the test set. White circles show the input source location. The left simulation plots show the network predictions, the middle simulation plots show the ground truth FD simulations and the right simulation plots show the difference. A $t^{2.5}$ gain is applied for display.~~

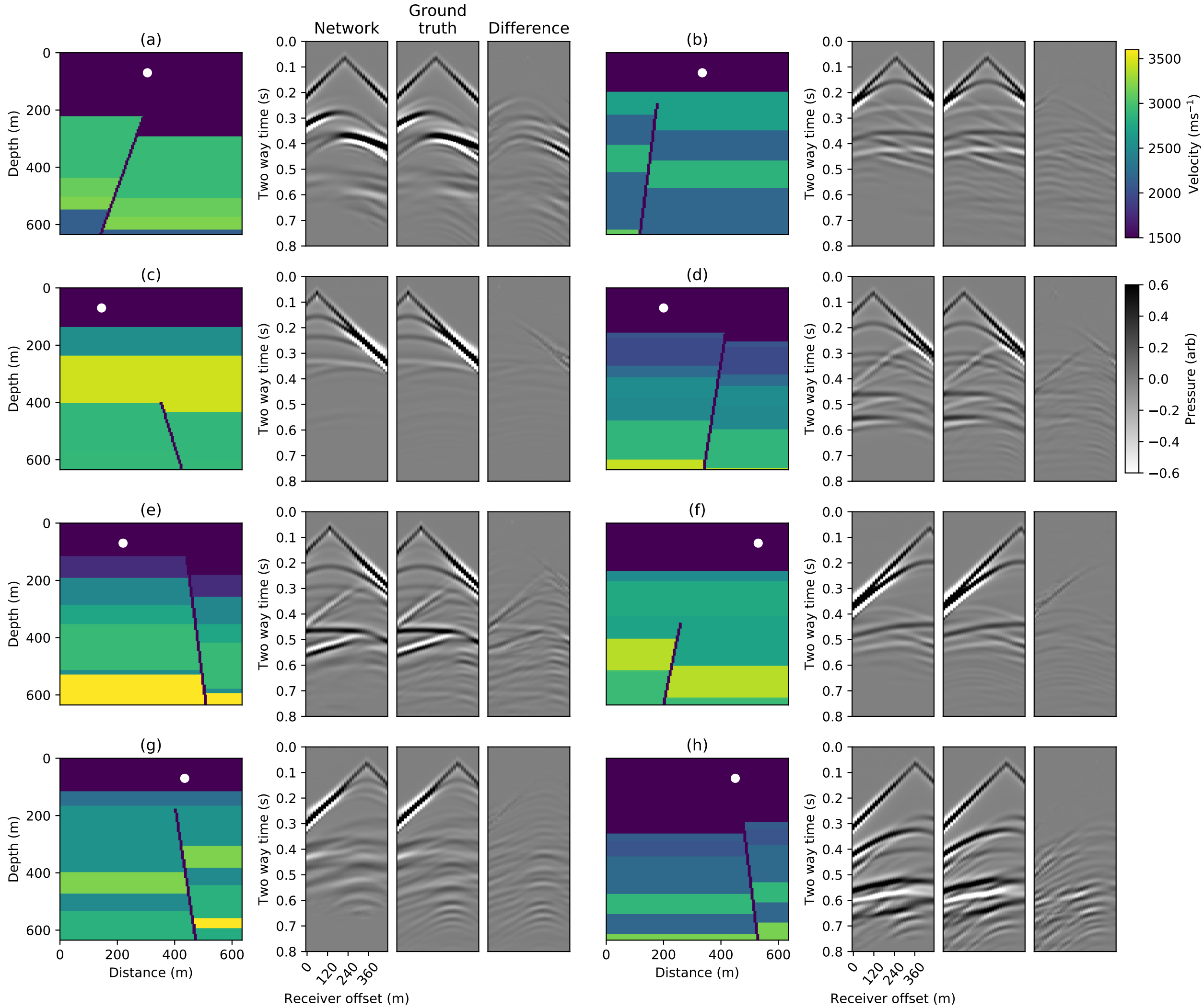

Figure 10. Conditional autoencoder simulations for 8 randomly selected examples in the test set. White circles show the input source location. The left simulation plots show the network predictions, the middle simulation plots show the ground truth FD simulations and the right simulation plots show the difference. A $t^{2.5}$ gain is applied for display.

~~Conditional autoencoder simulation accuracy when varying the source location. The network simulation is shown for 6 different source locations whilst keeping the velocity model fixed. The source positions are regularly spaced across the surface of the velocity model (white circles). Example simulations for 2 different velocity models in the test set are shown, where each row corresponds to a different velocity model. The pairs of simulation plots in each row from left to right correspond to the network prediction (left in the pair) and the ground truth FD simulation (right in the pair), when varying the source location from left to right in the velocity model. A $t^{2.5}$ gain is applied for display.~~

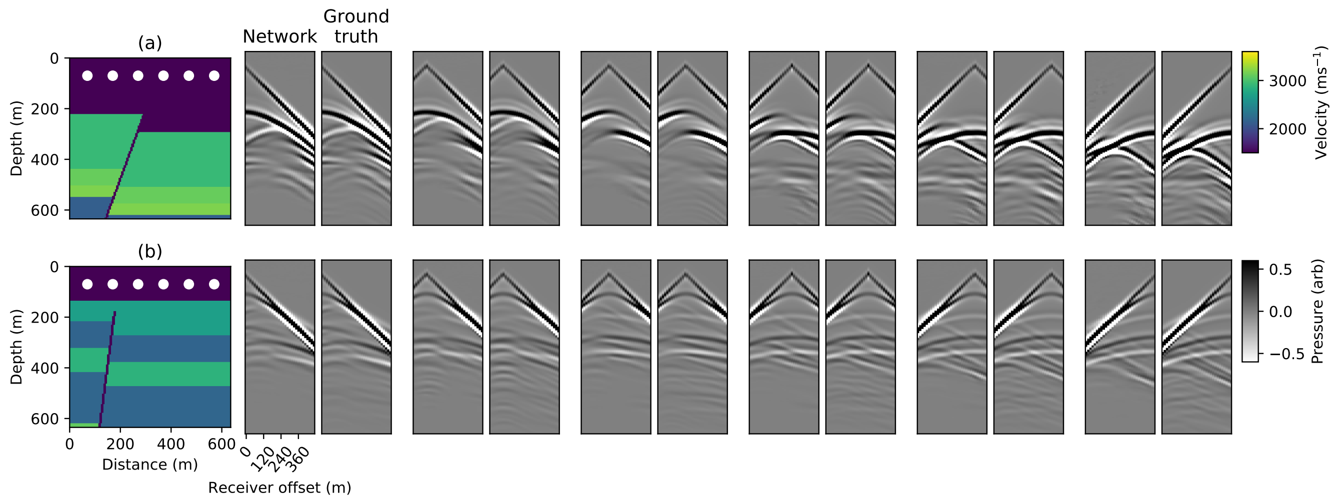

Figure 11. Conditional autoencoder simulation accuracy when varying the source location. The network simulation is shown for 6 different source locations whilst keeping the velocity model fixed. The source positions are regularly spaced across the surface of the velocity model (white circles). Example simulations for 2 different velocity models in the test set are shown, where each row corresponds to a different velocity model. The pairs of simulation plots in each row from left to right correspond to the network prediction (left in the pair) and the ground truth FD simulation (right in the pair), when varying the source location from left to right in the velocity model. A $t^{2.5}$ gain is applied for display.

~~Comparison of different network designs and training hyperparameters on simulation accuracy. Top right shows a randomly selected velocity model and source location from the test set and its corresponding ground truth FD simulation. Bottom compares simulations and their difference to the ground truth when using our proposed conditional autoencoder (baseline), when halving the number of hidden channels for all layers (thin), when using an L2 loss function during training (L2 loss), when using gain exponents of $g = 0$ and $g = 5$ in the loss function and when removing 2 layers from the encoder and 8 layers from the decoder (shallow). Top right: histogram of the average absolute amplitude difference between the ground truth FD simulation and the simulation from the different cases over the test set. A $t^{2.5}$ gain is applied for display.~~

~~Comparison of the WaveNet and conditional autoencoder simulation accuracy. The left plot shows a velocity model, reflectivity series and ground truth receiver pressure responses for a randomly selected example in the horizontally layered velocity model test set in red. Green shows the WaveNet simulation. The middle plot shows the conditional autoencoder simulation for the same velocity model. The right plot shows the histogram of the average absolute amplitude difference between the ground truth FD simulation and the WaveNet and conditional autoencoder simulations over this test set. A $t^{2.5}$ gain is applied for display.~~

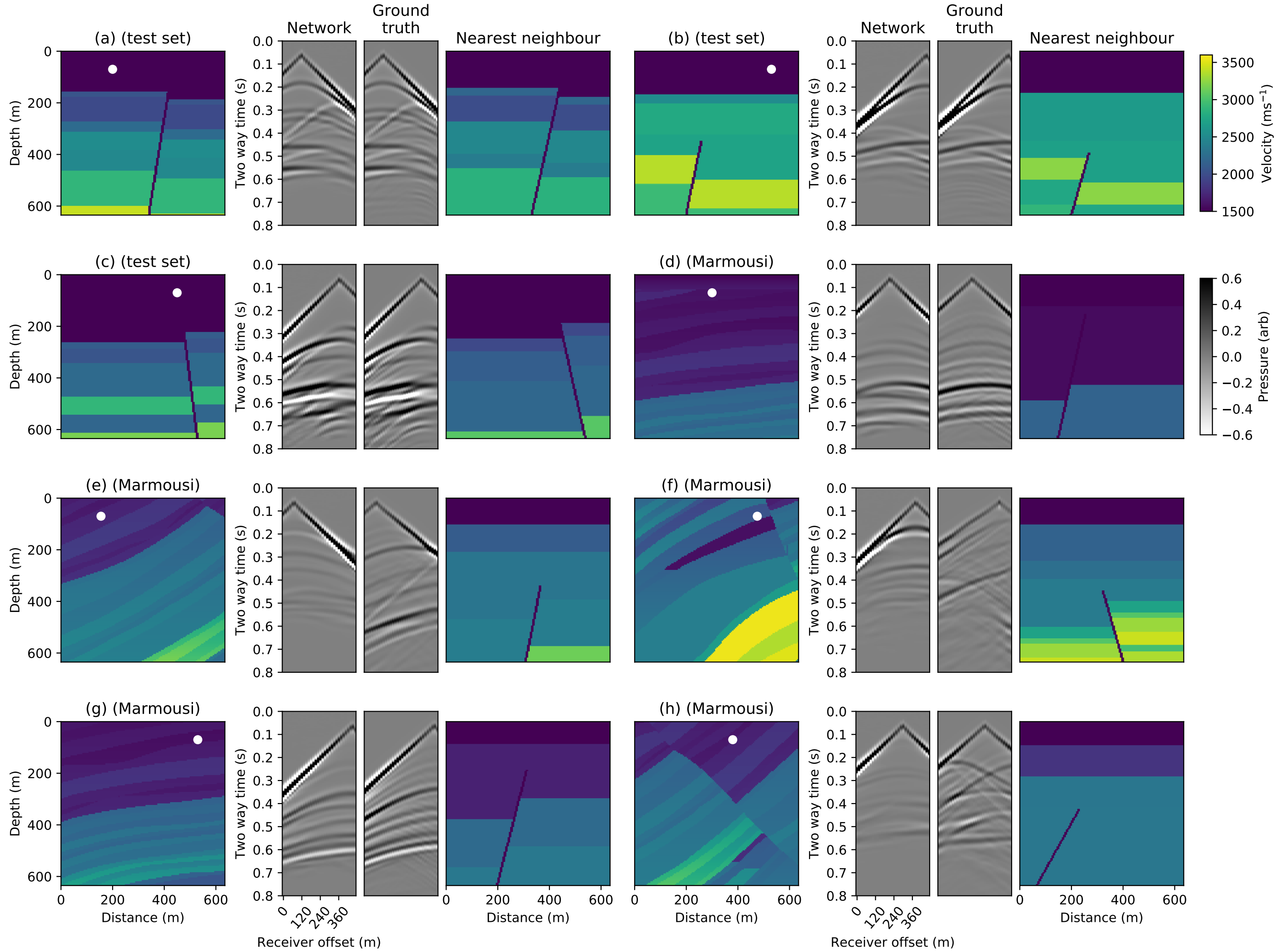

Figure 12. Generalisation ability of the conditional autoencoder. The conditional autoencoder simulations for 5 velocity models taken from different regions of the Marmousi P-wave velocity model are shown (examples (d)-(h)). For each example, left shows the input velocity model and source location, the middle simulations plots show the network prediction (left) and the ground truth FD simulation (right) and right shows the nearest neighbour in the training set to the input velocity model. Simulations from 3 of the test velocity models in Fig. 10 are also shown with their nearest neighbours (examples (a)-(c)). A $t^{2.5}$ gain is applied for display.

3.1 Conditional autoencoder architecture

Our simulation workflow is shown in Fig. 9. Instead of preprocessing the input velocity model to its associated reflectivity model, we input the velocity model directly into ~~our network. Our network is now also~~ the network. The network is conditioned on the source position, which ~~can~~ is allowed to vary along the surface of the Earth model. The output of the network is a simulation of the pressure responses recorded at ~~the~~ 32 fixed receiver locations in the model shown in Fig. 8.

We use a conditional autoencoder network design, shown in Fig 9. The network is composed of 10 convolutional layers which reduce the spatial dimensions of the input velocity model until it has a 1x1 shape with 1024 hidden channels. We term this tensor the latent vector. The input source surface position is concatenated onto the latent vector and 14 ~~transposed~~ convolutional layers are used to expand the size of the latent vector until its output shape is the same as the target receiver gather. We choose this encoder-decoder architecture to force the network to compress the velocity model into a set of salient features before expanding them to infer the receiver responses. All hidden layers use ReLU activation functions and the final output layer uses an identity activation function. The resulting network has 18,382,296 free parameters. The full parameterisation of the network is shown in Table A1.

~~Layer Type in, out channels kernel size stride padding 1 Conv2d (1,8) (3,3) (1,1) (1,1) 14 Conv2d (512,512) (3,3) (1,1) (1,1) 2 Conv2d (8,16) (2,2) (2,2) 0 15 Conv2d (512,512) (3,3) (1,1) (1,1) 3 Conv2d (16,16) (3,3) (1,1) (1,1) 16 ConvT2d (512,256) (2,4) (2,4) 0 4 Conv2d (16,32) (2,2) (2,2) 0 17 Conv2d (256,256) (3,3) (1,1) (1,1) 5 Conv2d (32,32) (3,3) (1,1) (1,1) 18 Conv2d (256,256) (3,3) (1,1) (1,1) 6 Conv2d (32,64) (2,2) (2,2) 0 19 ConvT2d (256,64) (2,4) (2,4) 0 7 Conv2d (64,128) (2,2) (2,2) 0 20 Conv2d (64,64) (3,3) (1,1) (1,1) 8 Conv2d (128,256) (2,2) (2,2) 0 21 Conv2d (64,64) (3,3) (1,1) (1,1) 9 Conv2d (256,512) (2,2) (2,2) 0 22 ConvT2d (64,8) (2,4) (2,4) 0 10 Conv2d (512,1024) (2,2) (2,2) 0 23 Conv2d (8,8) (3,3) (1,1) (1,1) 11 Concat (1024,1025) 24 Conv2d (8,8) (3,3) (1,1) (1,1) 12 ConvT2d (1025,1025) (2,2) (2,2) 0 25 Conv2d (8,1) (1,1) (1,1) 0 13 ConvT2d (1025,512) (2,4) (2,4) 0~~

~~Conditional autoencoder layer parameters. Each entry shows the parameterisation of each convolutional layer. The padding column shows the padding on each side of the input tensor for each spatial dimension.~~

3.2 Training process

We use the same training data generation process described by Section 2.3. When generating velocity models, we add a fault to the model. We randomly sample the length, normal or reverse direction, slip distance and orientation of the fault. Example velocity models drawn from this process are shown in Fig. 10. We generate 100,000 example velocity models and for each model chose three random source locations along the top of the model. This generates a total of 300,000 synthetic ground truth example simulations to use for training the network. We withhold 60,000 of these examples to use as a validation set during training.

We train using the same training process and loss function described in Section 2.4, except that we employ a L1 norm instead of a L2 norm in the loss function (Eq. 3). We use a learning rate of $1\mathrm{x}10^{-4}$, a batch size of 100 examples and run training over 3,000,000 gradient descent steps. We use batch normalisation ~~(?)~~ (Ioffe and Szegedy, 2015) after each convolutional layer to help regularise the network during training.

3.3 Results

During training the ~~loss~~ losses over the training and validation datasets converge to similar values and we test the performance of the trained network using a test set of 1000 unseen examples. The output simulations for 8 randomly selected velocity models and source positions from this set are shown in Fig. 10. We observe that the network is able to simulate the ~~recorded pressure~~

~~response. The network is able to simulate the~~ kinematics of the primary reflections and in most cases is able to ~~approximate~~ capture their relative amplitudes. ~~The network is also able to generalise over different source locations. We demonstrate this capability further in Fig. 11, where we~~ We also plot the network simulation when varying the source location over 2 velocity models from the test set ~~.~~ in Fig. 11 and find that the network is able to generalise well over different source locations.

We test the accuracy of the ~~network~~ simulation when using different network designs and training hyperparameters~~.~~, shown in Fig. A2~~compares an example simulation~~. We compare example simulations from the test set when using our baseline conditional autoencoder network, when halving the number of hidden channels for all layers, when using an L2 loss function during training, when using gain exponents of $g = 0$ and $g = 5$ in the loss function and when removing 2 layers from the encoder and 8 layers from the decoder. We plot the histogram of the average absolute amplitude difference between the ground truth FD simulation and the ~~simulation from all of these cases~~ network simulation over the test set ~~. We~~ for all of the cases above, and observe that in all ~~the~~ cases the simulations are less accurate than our baseline approach. Without the gain in the loss function, the network only learns to simulate the direct arrival and the first few reflections in the receiver responses. With a gain exponent of $g = 5$, the network simulation is unstable and it fails to simulate the first 0.2 seconds of the receiver responses. When using the network with less layers the simulations have edge artefacts~~. The~~, whilst the network with half the number of hidden channels is closest to the baseline accuracy. ~~We also~~ In testing we find that training a ~~flat convolutional neural~~ network with the same number of layers but without using a bottleneck design to reduce the velocity model to a 1x1x1024 latent vector does not converge.

We compare the accuracy of the conditional autoencoder to the WaveNet network ~~.~~ in Fig. A3. We plot the simulation from both networks for an example model in the horizontally layered velocity model test set ~~in Fig. A3. We also plot~~ and the histogram of the average absolute amplitude difference between the ground truth FD simulation and the WaveNet and conditional autoencoder simulations over this ~~test~~ set. Both networks are able to accurately simulate the receiver responses and the WaveNet simulation is slightly more accurate than the conditional autoencoder, though of course the latter is more general.

We ~~measure the average time taken for~~ test the generalisation ability of the conditional autoencoder ~~to~~ outside of its training distribution by inputting randomly selected 640×640 m boxes from the publicly available 2D Marmousi P-wave velocity model (Martin et al., 2006) into the network. These velocity models contain much more complex faulting at multiple scales, higher dips and more layer variability than our training dataset. The resulting network simulations are shown in Fig. 12. We calculate the nearest neighbour to the input velocity model in the set of training velocity models, defined as the training model with the lowest L1 difference summed over all velocity values from the input velocity model, and show this alongside each example.

We find that the network is not able to accurately simulate the full seismic response from velocity models which have large dips and/or complex faulting (examples (e), (f) and (h)) that are absent in the training set. This observation is similar to most studies which analyse the generalisability of deep neural networks outside their training set (e.g. Zhang and Lin (2018) and Earp and Curtis (2019)). However, encouragingly, the network is able to mimic the response from velocity models with small dips ((d) and (g)), even though the nearest training-set neighbour contains a fault whereas the Marmousi layers are continuous.

We compare the average time taken to generate 100 simulations using ~~a single core of a 2.2 GHz Intel Core i7 processor to be 3.3 ± 0.1 s (~~the conditional autoencoder network to FD simulation in Table 1. We find that on a single CPU core the network is 22 ~~times faster than FD modelling on the same core). Similar to the WaveNet network , this is an order of magnitude~~ times faster than FD ~~modelling. Using PyTorch (Pytorch, 2016) and a Nvidia Tesla K80 GPU produces simulations with an average time of 0.180 ± 0.003 s (~~simulation and when using a GPU and the PyTorch library (Pytorch, 2016) it is 406 times faster~~than FD modelling). The conditional autoencoder~~ . This is comparable to the speed up obtained with the WaveNet. It is likely that 2D ray tracing will not offer the same speed up as observed in Section 2.7, because computing ray paths through these models is likely to be more demanding. The network takes approximately 4 days to train on one Nvidia Titan V GPU. This is 8 times longer than training the WaveNet network, ~~though we made little~~ although we made no effort to optimise ~~the~~ its training time. We ~~also~~ find that when using only 50,000 training examples the validation loss increases and the network overfits to the training dataset.

4 ~~Discussion and Further Work~~

~~We present two neural networks which offer fast methods for simulating seismic waves~~

4 Discussion

Both our deep neural networks accurately model the seismic response in horizontally layered and faulted 2D acoustic media ~~. Our WaveNet network~~ one to two orders of magnitude faster than FD modelling. The WaveNet is able to ~~simulate the pressure response at multiple receiver locations in~~ carry out fast simulation of horizontally layered velocity models, ~~whilst our~~ and the conditional autoencoder is ~~more general in that it is able to simulate the response from 2D faulted velocity models, conditioned on the source location.~~

~~We find that using causality in our WaveNet network generates more accurate simulations than when using a standard convolutional network without causality. This suggests that adding this physics constraint helps the network simulate the pressure responses. We also show that using exponential dilations appears to be a crictical design choice; using a convolutional network without dilations does not converge. This is likely because without dilations the network's field of view does not cover the entire input reflectivity series.~~

~~Our WaveNet network can also carry out~~ able to generalise to faulted media with arbitrary layers, fault properties and an arbitrary location of the seismic source on the surface of the media. This is a significantly harder task than simulating horizontally layered media with the WaveNet network. The WaveNet can also be adapted to carry out fast seismic inversion, which offers a fast alternative to existing inversion algorithms. ~~We note that seismic inversion is typically an ill-defined problem and it is likely that the predictions the inverse WaveNet architecture makes are biased towards the velocity models it is trained on. This uncertainty could be quantified, for example by using Bayesian deep learning methods (?). We have also yet to compare our inverse WaveNet network to existing seismic inversion techniques, or test it with real seismic data. An alternative method~~

~~for inversion is to use our forward networks in existing seismic inversion algorithms based on optimisation, such as FWI (?). Both our WaveNet and conditional autoencoder networks are fully differentiable and could therefore be used to generate fast approximate gradient estimates in these methods~~

Whilst these results are encouraging and suggest that deep learning is valuable for both simulation and inversion, there are further challenges when extending our simulation methods to more complex, elastic and 3D Earth models in practical simulation tasks. We believe that further research will help to understand whether deep learning can aid in these more general settings and discuss this in more detail below.

~~Our conditional autoencoder shows excellent generalisation; it is~~

4.1 Extension to elastic simulation

An important ability for practical geophysical applications is to be able to simulate seismic waves in ~~faulted mediawith arbitrary layers, fault properties and an arbitrary location of the seismic source on the surface of the media. This is a significantly harder task than simulating the horizontally layered mediawith the WaveNet network. The network shows good accuracy in simulating the kinematics of reflections, but for velocity models with many layers or strong contrasts it sometimes struggles to accurately simulate their relative amplitudes. Our loss function or network designcould be investigated further to reduce these amplitude differences.~~ (visco-)elastic media, rather than acoustic media. The architectures of our networks are readily extendable in this regard; S-wave velocity and density models could be added as additional input channels to our networks and the number of output channels in the networks could be increased so that multi-component particle velocity vectors are output. The same training scheme could be used, with training data generated using elastic FD simulation instead of acoustic simulation and a loss function which compares vector fields instead of scalar fields. Thus, with some simple changes to our design, this challenge is at least conceptually simple to address, though further research is required to understand if it is feasible. The cost of traditional elastic simulation exceeds the cost of acoustic simulation by orders of magnitude and has prevented the seismic industry from fully embracing this crucial step. We postulate that the difference in simulation times between future elastic and acoustic simulation networks might be smaller compared to fully discretised methods such as FD, as a consequence of the networks not needing to compute the entire discretised wavefield. While this is speculative at this point, it is intriguing to investigate.

~~Our ablation tests show that simulation accuracy is sensitive to the network design and training hyperparameters. Using an appropriate gain function in the loss function appears critical; with too little or too much gain convergence is unstable. Empirically our gain function with $g = 2.5$ performs well. When using less layers in the network we find the simulation has edge artefacts. This suggest that there is a minimum limit on the number of layers when using this encoder-decoder network design~~

4.2 Extension to 3D simulation

Another important extension is to move from 2D to 3D simulation. ~~We also find that using a flat network without a bottleneck design does not converge. Our hypothesis is that the bottleneck encourages a depth-to-time conversion by slowly reducing the~~

~~spatial dimensions of~~ A major challenge here is likely to be the increased computational cost of generating training data with conventional methods, which for instance is significantly higher in 3D when using FD modelling. In terms of network design, our autoencoder could be extended to 3D simulation by increasing the dimensionality of its input, hidden and output tensors. In this case we would expect a similar order of magnitude acceleration of simulation time to 2D, because the network would still directly estimate the seismic response without needing to iteratively model the seismic wavefield through time. However, this approach is likely to be practically challenging because increasing the dimensionality would increase the number of weights and likely the training time. Finding an alternative representation, such as meshes or oct-trees (Ahmed et al., 2018) to reduce the dimensionality of the problem, or a way to exploit symmetry in the wave equation to reduce complexity, may be critical. Furthermore, whilst we only used the wavefield at each receiver location to train our networks, finding a way to use the entire wavefield from FD simulation to train the network may help reduce the number of training simulations required. We note that generating training data is an amortized cost because the ~~velocity model before expanding them into time.~~ network only needs to be trained once, which in the case of seismic inversion with millions of production runs could become negligible. Another intriguing aspect is to investigate whether deep neural network simulation costs scale more favourably with increasing frequency ω, compared to fully discrete methods which scale with ω^4; in this study we only consider simulation at a fixed frequency.

~~We find that the WaveNet network has marginally better performance than the conditional autoencoder when simulating layered velocity models. This may be because the WaveNet network is more physically-constrained for this task; it uses causality and has 18 times less free parameters. It is an open question how best to represent causality in networksfor simulating more arbitrary Earth models. We find that both networks have more difficulty simulating multiple reverberations, perhaps because they have more complex wave physics than the primary reflections. Other components such as Long Short-Term Memory (LSTM) or Recursive Neural Network (RNN) cells could be tested inside the networks for improving its prediction of these signals.~~

4.3 Generalisation to more complex Earth models

~~A key future challenge is to generalise our networks to~~ Perhaps the largest challenge in designing appropriate networks is to improve their generality so they can simulate more complex Earth models. ~~Whilst our generalisation from~~ We have shown that deep neural networks can move beyond simulating simple horizontally layered velocity models to ~~faulted models is promising, we have yet to consider arbitrary Earth models. Furthermore we focus on acoustic simulation and do not consider elastic or viscoelastic simulation. To generalise further, our conditional autoencoder network requires~~ more complex faulted models where, to the best of our knowledge, no analytical solutions exist, which we believe is a positive step. However, both our networks performed worse on velocity models outside of their training distributions. Furthermore, to be able to generalise to more complex velocity models the conditional autoencoder required more free parameters, more time to train and more training examples than the WaveNet network. ~~Elastic and aniostropic parameters would need to be added as additional inputs to the networks and ground truth simulations with more complex Earth models would need to be used to train them. Novel network designs which incorporate physics constraints directly in their architecture or in their loss function may help; it may also be~~

~~useful to use~~ Generalisation outside of the training distribution is a well known and common challenge of deep neural networks in general (Goodfellow et al., 2016).

A naive approach would be to increase the range of the training data to improve the generality of the network, however this would quickly become computationally intractable when trying to simulate all possible Earth models. We note that for many practical applications it may be acceptable to use a training distribution with a limited range; for example, in many of the seismic applications such tomography, FWI, and seismic hazard assessment, a huge number of forward simulations of comparatively few Earth models are carried out.

A promising research direction may be to better regularise the networks by adding more physics-based constraints into the workflow. We found that using causality in the WaveNet generated more accurate simulations than when using a standard convolutional network; this suggested that adding this constraint helped the network simulate the seismic response, although it is an open question how best to represent causality when simulating more arbitrary Earth models. We also found that a bottleneck design helped the conditional autoencoder to converge; our hypothesis is that this encouraged a depth-to-time conversion by slowly reducing the spatial dimensions of the velocity model before expanding them into time. More advanced network designs, for example using attention-like mechanisms (Vaswani et al., 2017) to help the network focus on ~~the relevant part~~ relevant parts of the velocity model~~to carry out simulation~~, rather than using convolutional layers with full fields of view~~(?)~~.

~~Another key challenge is to move from 2D simulation to 3D simulation. This is challenging because the computation cost of generating training data in 3D is significantly higher than in 2D. However, though large, this could be an amortized cost; the network would only~~ , or using Long Short-Term Memory (LSTM) cells to help the network model multiple reverberations could be tested. Another interesting direction would be to use the wave equation (Eq. 1) to directly regularise the loss function, similar to the physics-based machine learning approach proposed by Raissi et al. (2019).

We find the nearest neighbour test is a useful way to understand if an input velocity model is close to the training distribution and therefore if the network's output simulation is likely to be accurate. Probabilistic approaches, such as Bayesian deep learning (Gal, 2016), could be investigated for their ability to provide more quantitative uncertainty estimates on the network's output simulation.

4.4 Inversion with WaveNet

Finally, we discuss our inversion approach using WaveNet. We note that seismic inversion is typically an ill-defined problem and it is likely that the predictions the network made are biased towards the velocity models it was trained on. The accuracy of the network reduced when it was tested on inputs outside of its training distribution and this is likely to degrade further when tested with real, noisy seismic data. Further research could try to quantify this uncertainty, for example by using Bayesian deep learning. We have also yet to compare our inverse WaveNet network to existing seismic inversion techniques, such as posterior sampling or FWI. An alternative method for inversion is to use our forward networks in existing seismic inversion algorithms based on optimisation, such as FWI. Both the WaveNet and conditional autoencoder networks are fully differentiable and could therefore be used to generate fast approximate gradient estimates in these methods, although similar limitations on their generality are likely to exist and one would need to be ~~trained once and after this inference steps are still likely to be fast. Whilst we extract only the wavefield at each receiver location to train our network, using the entire wavefield from FD simulation during training may help reduce the number of~~ careful to keep the inversion routine within the training distribution of the network.

4.5 Summary

Given the potentially large training costs and the challenge of generality, it may be that current deep learning techniques are most advantageous to practical simulation tasks where many similar simulations are required, such as inversion or statistical seismic hazard analysis, and least useful for problems with a very small number of simulations per model family. In seismology, however, we suspect that most current and future challenges fall into the former category, which renders these initial results promising. Further research is required to understand how best to design the training ~~simulations~~set for a particular simulation application, as well as how to help deep neural networks generalise to unseen velocity models outside of the training distribution.

5 Conclusions

We have investigated the potential of deep learning for aiding seismic simulation tasks in geophysics. We presented two deep neural networks ~~for carrying~~ which are able to carry out fast and largely accurate simulation of seismic waves. Both networks are 20 - 500 times faster than FD modelling and simulate seismic waves in horizontally layered and faulted 2D acoustic media. The first network uses a WaveNet architecture and simulates seismic waves in horizontally layered media. We showed that this network can also be used to carry out fast seismic inversion of the same media. ~~Our~~ The second network is significantly more general than the first; it simulates seismic waves in faulted media with arbitrary layers, fault properties and an arbitrary location of the seismic source on the surface of the media. Our ~~approaches could pave the way to real-time seismic simulation and benefit seismic inversion algorithms based on forward simulation. Our work suggests that deep learning is a valuable tool for both seismic simulation and inversion~~main contribution is to show that deep neural networks can move beyond simulating simple horizontally layered velocity models to more complex faulted models where, to the best of our knowledge, no analytical solutions exist, which we believe is a positive step towards understanding their potential. We discussed the challenges of extending our approaches to practical geophysical applications and future research directions which could address them, noting where it may be favourable for using these network architectures.

Code and data availability. All our training data used was generated synthetically, using the SEISMIC_CPML FD modelling library. Our WaveNet code is already publicly available on Github here: https://github.com/benmoseley/seismic-simulation-wavenet. We are happy to release the code to reproduce all our results on Github on publication of this paper.

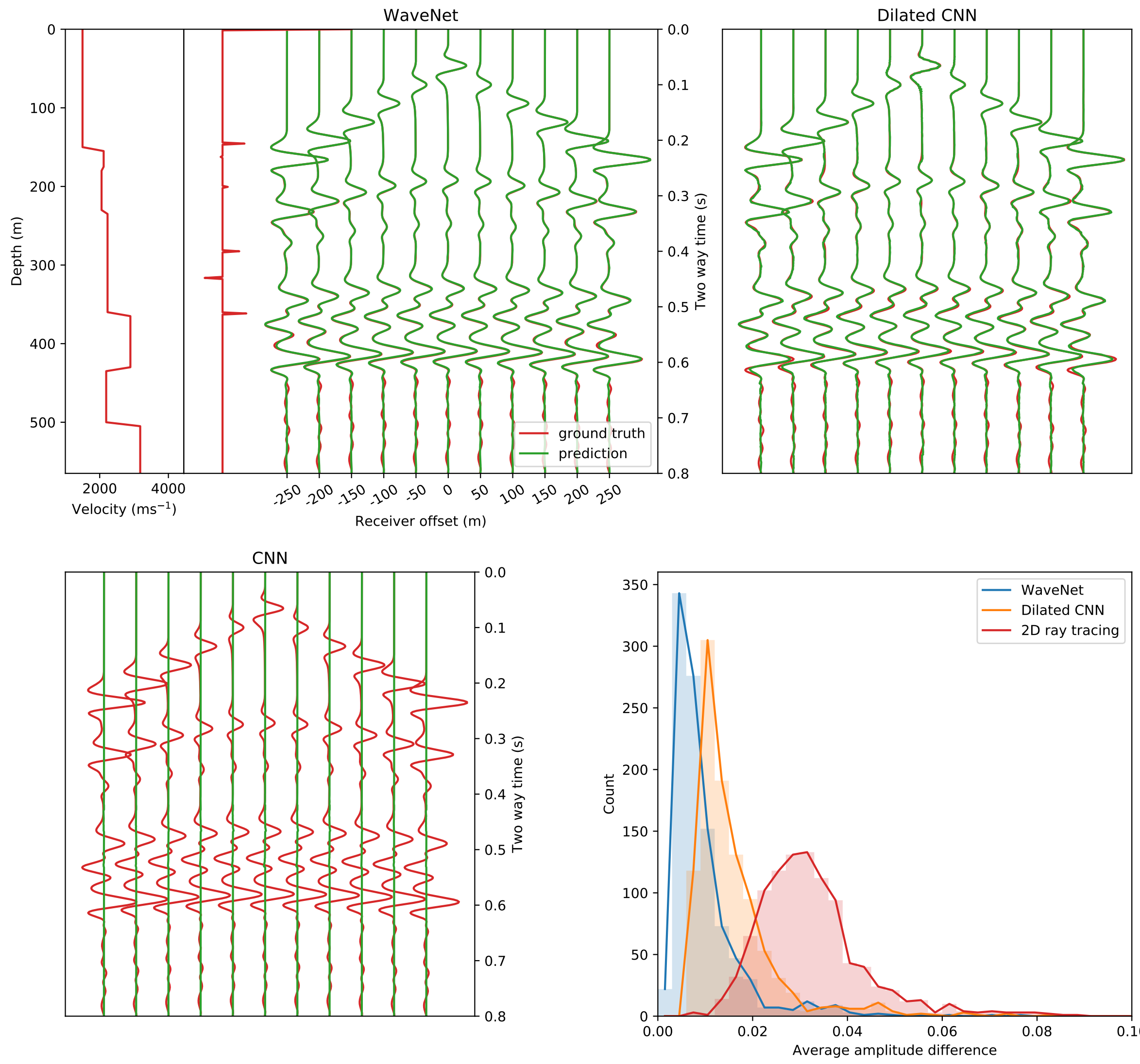

Figure A1. Comparison of different network architectures on simulation accuracy. Top left shows the WaveNet simulated pressure response for a randomly selected example in the test set (green) compared to ground truth FD simulation (red). Top right and bottom left show the simulated response when using two convolutional network designs with and without exponential dilations. Bottom right shows the histogram of the average absolute amplitude difference between the ground truth FD simulation and the simulations from the WaveNet, the dilated convolutional network and 2D ray tracing over the test set of 1000 examples. A $t^{2.5}$ gain is applied to the receiver responses for display.

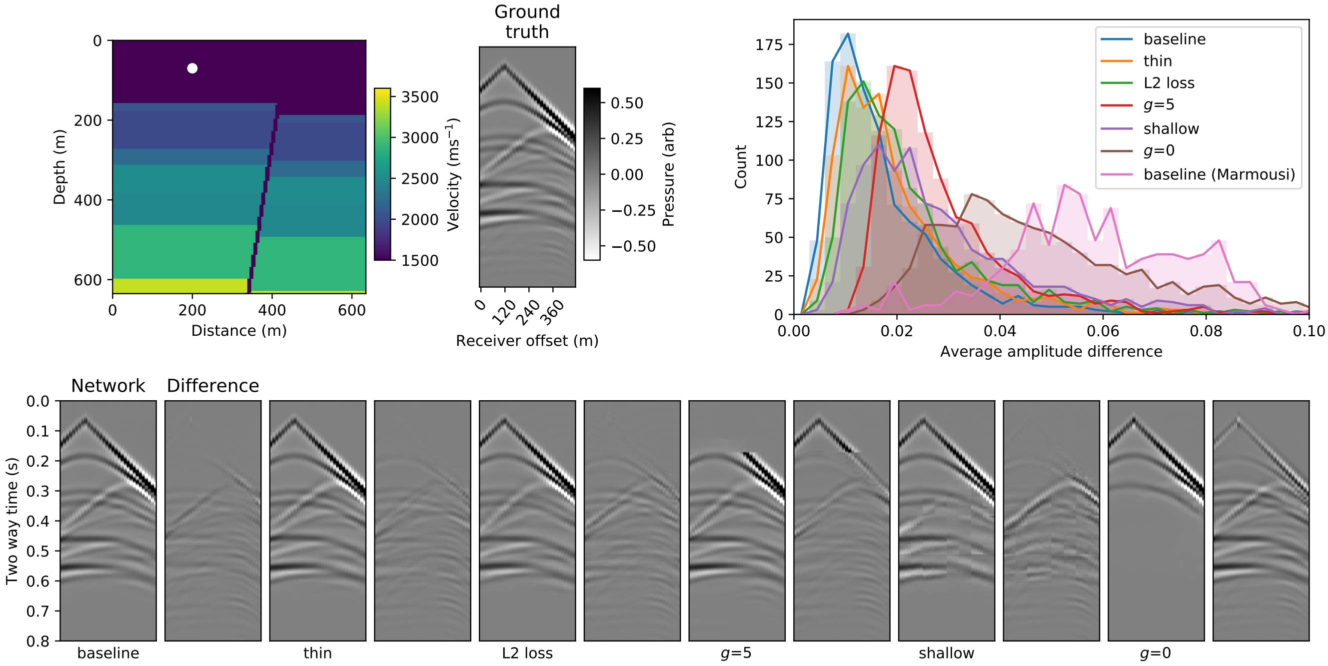

Figure A2. Comparison of different conditional autoencoder network designs and training hyperparameters on simulation accuracy. Top left shows a randomly selected velocity model and source location from the test set and its corresponding ground truth FD simulation. Bottom compares simulations and their difference to the ground truth when using our proposed conditional autoencoder (baseline), when halving the number of hidden channels for all layers (thin), when using an L2 loss function during training (L2 loss), when using gain exponents of $g = 0$ and $g = 5$ in the loss function and when removing 2 layers from the encoder and 8 layers from the decoder (shallow). Top right shows the histogram of the average absolute amplitude difference between the ground truth FD simulation and the simulation from the different cases over the test set. The histogram of the baseline network over the Marmousi test dataset is also shown. A $t^{2.5}$ gain is applied for display.

Appendix A: Supplementary figures and tables

Author contributions. TNM and AM were involved in the conceptualisation, supervision and review of the work. BM was involved in the conceptualisation, data creation, methodology, investigation, software, data analysis, validation and writing.

Competing interests. TNM is a Topical Editor for the Solid Earth Editorial Board.

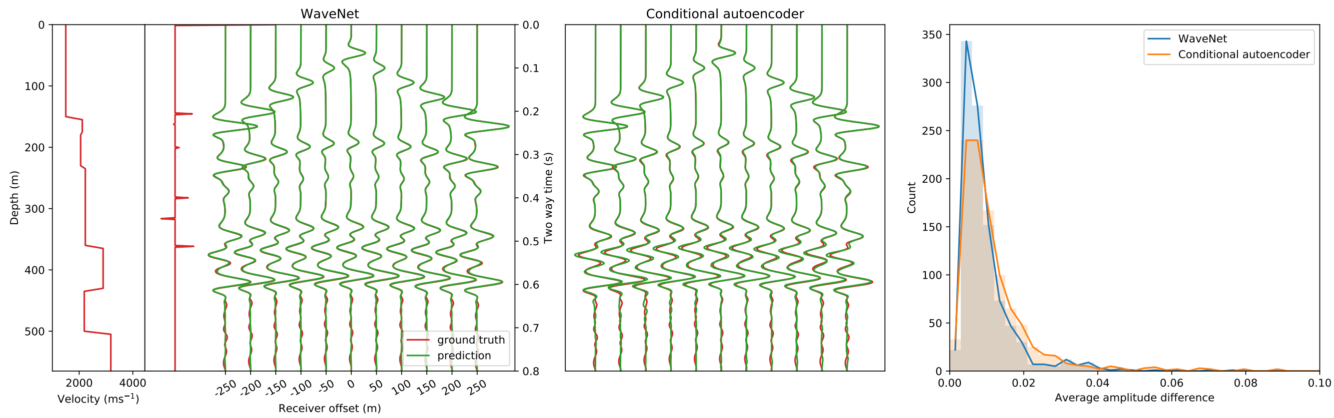

Figure A3. Comparison of the WaveNet and conditional autoencoder simulation accuracy. The left plot shows a velocity model, reflectivity series and ground truth FD simulation for a randomly selected example in the horizontally layered velocity model test set in red. Green shows the WaveNet simulation. The middle plot shows the conditional autoencoder simulation for the same velocity model. The right plot shows the histogram of the average absolute amplitude difference between the ground truth FD simulation and the WaveNet and conditional autoencoder simulations over this test set. A $t^{2.5}$ gain is applied for display.

Layer	Type	in, out channels	kernel size	stride	padding						
1	Conv2d	(1,8)	(3,3)	(1,1)	(1,1)	14	Conv2d	(512,512)	(3,3)	(1,1)	(1,1)
2	Conv2d	(8,16)	(2,2)	(2,2)	0	15	Conv2d	(512,512)	(3,3)	(1,1)	(1,1)
3	Conv2d	(16,16)	(3,3)	(1,1)	(1,1)	16	ConvT2d	(512,256)	(2,4)	(2,4)	0
4	Conv2d	(16,32)	(2,2)	(2,2)	0	17	Conv2d	(256,256)	(3,3)	(1,1)	(1,1)
5	Conv2d	(32,32)	(3,3)	(1,1)	(1,1)	18	Conv2d	(256,256)	(3,3)	(1,1)	(1,1)
6	Conv2d	(32,64)	(2,2)	(2,2)	0	19	ConvT2d	(256,64)	(2,4)	(2,4)	0
7	Conv2d	(64,128)	(2,2)	(2,2)	0	20	Conv2d	(64,64)	(3,3)	(1,1)	(1,1)
8	Conv2d	(128,256)	(2,2)	(2,2)	0	21	Conv2d	(64,64)	(3,3)	(1,1)	(1,1)
9	Conv2d	(256,512)	(2,2)	(2,2)	0	22	ConvT2d	(64,8)	(2,4)	(2,4)	0
10	Conv2d	(512,1024)	(2,2)	(2,2)	0	23	Conv2d	(8,8)	(3,3)	(1,1)	(1,1)
11	Concat	(1024,1025)				24	Conv2d	(8,8)	(3,3)	(1,1)	(1,1)
12	ConvT2d	(1025,1025)	(2,2)	(2,2)	0	25	Conv2d	(8,1)	(1,1)	(1,1)	0
13	ConvT2d	(1025,512)	(2,4)	(2,4)	0						

Table A1. Conditional autoencoder layer parameters. Each entry shows the parameterisation of each convolutional layer. The padding column shows the padding on each side of the input tensor for each spatial dimension.

Acknowledgements. The authors would like to thank the Computational Infrastructure for Geodynamics (www.geodynamics.org) for releasing the open-source SEISMIC_CPML FD modelling libraries.

We would also like to thank Tom Le Paine for his fast WaveNet implementation on GitHub from which our code was based on (github.com/tomlepaine/fast-wavenet).

References

Ahmed, E., Saint, A., Shabayek, A. E. R., Cherenkova, K., Das, R., Gusev, G., Aouada, D., and Ottersten, B.: A survey on Deep Learning Advances on Different 3D Data Representations, ArXiv e-prints, 2018.

Aki, K. and Richards, P. G.: Quantitative seismology, W. H. Freeman and Co., 1980.

Araya-Polo, M., Jennings, J., Adler, A., and Dahlke, T.: Deep-learning tomography, The Leading Edge, 37, 58–66, 2018.

Bergen, K. J., Johnson, P. A., De Hoop, M. V., and Beroza, G. C.: Machine learning for data-driven discovery in solid Earth geoscience, Science, 363, 2019.

Bohlen, T.: Parallel 3-D viscoelastic finite difference seismic modelling, Computers & Geosciences, 28, 887–899, 2002.

Boore, D. M.: Simulation of ground motion using the stochastic method, Pure and Applied Geophysics, 160, 635–676, 2003.

Bozdağ, E., Peter, D., Lefebvre, M., Komatitsch, D., Tromp, J., Hill, J., Podhorszki, N., and Pugmire, D.: Global adjoint tomography: first-generation model, Geophysical Journal International, 207, 1739–1766, 2016.

Chopra, S. and Marfurt, K. J.: Seismic Attributes for Prospect Identification and Reservoir Characterization, Society of Exploration Geophysicists and European Association of Geoscientists and Engineers, 2007.

Cui, Y., Olsen, K. B., Jordan, T. H., Lee, K., Zhou, J., Small, P., Roten, D., Ely, G., Panda, D. K., Chourasia, A., Levesque, J., Day, S. M., and Maechling, P.: Scalable Earthquake Simulation on Petascale Supercomputers, in: 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, November, pp. 1–20, IEEE, 2010.

Devilee, R. J. R., Curtis, A., and Roy-Chowdhury, K.: An efficient, probabilistic neural network approach to solving inverse problems: Inverting surface wave velocities for Eurasian crustal thickness, Journal of Geophysical Research: Solid Earth, 104, 28 841–28 857, 1999.

Dowla, F. U., Taylor, S. R., and Anderson, R. W.: Seismic discrimination with artificial neural networks: Preliminary results with regional spectral data, Bulletin of the Seismological Society of America, 80, 1346–1373, 1990.

Earp, S. and Curtis, A.: Probabilistic Neural-Network Based 2D Travel Time Tomography, ArXiv e-prints, 2019.

Fichtner, A.: Full Seismic Waveform Modelling and Inversion, Springer, 2010.

Gal, Y.: Uncertainty in Deep Learning, Ph.D. thesis, University of Cambridge, 2016.

Goodfellow, I., Bengio, Y., and Courville, A.: Deep Learning, MIT Press, 2016.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., and Chen, T.: Recent advances in convolutional neural networks, Pattern Recognition, 77, 354–377, 2018.

Guo, X., Li, W., and Iorio, F.: Convolutional Neural Networks for Steady Flow Approximation, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, KDD '16, pp. 481–490, 2016.

Gutenberg, B.: The amplitudes of waves to be expected in seismic prospecting, Geophysics, 1, 252–256, 1936.

Hosseini, K., Sigloch, K., Tsekhmistrenko, M., Zaheri, A., Nissen-Meyer, T., and Igel, H.: Global mantle structure from multifrequency tomography using P, PP and P-diffracted waves, Geophysical Journal International, 220, 96–141, 2019.

Igel, H.: Computational seismology: a practical introduction, Oxford University Press, 2017.

Ioffe, S. and Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: 32nd International Conference on Machine Learning, ICML 2015, vol. 1, pp. 448–456, International Machine Learning Society (IMLS), 2015.

Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, ArXiv e-prints, 2014.

Komatitsch, D. and Martin, R.: An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation, Geophysics, 72, SM155–SM167, 2007.

Komatitsch, D. and Tromp, J.: Introduction to the spectral element method for three-dimensional seismic wave propagation, Geophysical Journal International, 139, 806–822, 1999.

Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., and Gerstoft, P.: Machine learning in seismology: Turning data into insights, Seismological Research Letters, 90, 3–14, 2019.

Krischer, L. and Fichtner, A.: Generating Seismograms with Deep Neural Networks, AGU Fall Meeting Abstracts, p. kr, 2017.

Krischer, L., Hutko, A. R., van Driel, M., Stähler, S., Bahavar, M., Trabant, C., and Nissen-Meyer, T.: On-Demand Custom Broadband Synthetic Seismograms, Seismological Research Letters, 88, 1127–1140, 2017.

Leng, K., Nissen-Meyer, T., and van Driel, M.: Efficient global wave propagation adapted to 3-D structural complexity: a pseudospectral/spectral-element approach, Geophysical Journal International, 207, 1700–1721, 2016.

Leng, K., Nissen-Meyer, T., van Driel, M., Hosseini, K., and Al-Attar, D.: AxiSEM3D: broad-band seismic wavefields in 3-D global earth models with undulating discontinuities, Geophysical Journal International, 217, 2125–2146, 2019.

Lerer, A., Gross, S., and Fergus, R.: Learning Physical Intuition of Block Towers by Example, Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, pp. 430–438, 2016.

Long, G., Zhao, Y., and Zou, J.: A temporal fourth-order scheme for the first-order acoustic wave equations, Geophysical Journal International, 194, 1473–1485, 2013.

Lumley, D. E.: Time-lapse seismic reservoir monitoring, Geophysics, 66, 50–53, 2001.

Margrave, G. F. and Lamoureux, M. P.: Numerical Methods of Exploration Seismology, Cambridge University Press, 2018.

Martin, G. S., Wiley, R., and Marfurt, K. J.: Marmousi2: An elastic upgrade for Marmousi, Leading Edge (Tulsa, OK), 25, 156–166, 2006.

Moczo, P., Robertsson, J. O., and Eisner, L.: The Finite-Difference Time-Domain Method for Modeling of Seismic Wave Propagation, Advances in Geophysics, 48, 421–516, 2007.

Moseley, B., Markham, A., and Nissen-Meyer, T.: Fast approximate simulation of seismic waves with deep learning, ArXiv e-prints, 2018.

Murat, M. E. and Rudman, A. J.: Automated first arrival picking: a neural network approach, Geophysical Prospecting, 40, 587–604, 1992.

Nair, V. and Hinton, G.: Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair, in: Proceedings of ICML, vol. 27, pp. 807–814, 2010.

Nath, S. K., Chakraborty, S., Singh, S. K., and Ganguly, N.: Velocity inversion in cross-hole seismic tomography by counter-propagation neural network, genetic algorithm and evolutionary programming techniques, Geophysical Journal International, 138, 108–124, 1999.

Newman, P.: Divergence effects in a layered earth, Geophysics, 38, 481–488, 1973.

Ni, S., Tan, E., Gurnis, M., and Helmberger, D.: Sharp sides to the African superplume, Science, 296, 1850–1852, 2002.

Paganini, M., De Oliveira, L., and Nachman, B.: Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters, Physical Review Letters, 120, 1–6, 2018.

Perol, T., Gharbi, M., and Denolle, M.: Convolutional neural network for earthquake detection and location, Science Advances, 4, e1700 578, 2018.

Poulton, M. M., Sternberg, B. K., and Glass, C. E.: Location of subsurface targets in geophysical data using neural networks, Geophysics, 57, 1534–1544, 1992.

Pytorch: https://www.pytorch.org, 2016.

Raissi, M., Perdikaris, P., and Karniadakis, G. E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, 378, 686–707, 2019.

Richardson, A.: Seismic Full-Waveform Inversion Using Deep Learning Tools and Techniques, ArXiv e-prints, 2018.

Rietmann, M., Messmer, P., Nissen-Meyer, T., Peter, D., Basini, P., Komatitsch, D., Schenk, O., Tromp, J., Boschi, L., and Giardini, D.: Forward and adjoint simulations of seismic wave propagation on emerging large-scale GPU architectures, International Conference for High Performance Computing, Networking, Storage and Analysis, SC, pp. 1–11, 2012.

Röth, G. and Tarantola, A.: Neural networks and inversion of seismic data, Journal of Geophysical Research, 99, 6753, 1994.

Russell, B. H.: Introduction to Seismic Inversion Methods, Society of Exploration Geophysicists, 1988.

Schuster, G. T.: Seismic Inversion, Society of Exploration Geophysicists, 2017.

Sun, H. and Demanet, L.: Low frequency extrapolation with deep learning, 2018 SEG International Exposition and Annual Meeting, SEG 2018, pp. 2011–2015, 2018.

Tarantola, A.: Inverse problem theory: methods for data fitting and model parameter estimation, Elsevier, 1987.

TensorFlow: https://www.tensorflow.org, 2015.

Thorne, M. S., Pachhai, S., Leng, K., Wicks, J. K., and Nissen-Meyer, T.: New Candidate Ultralow-Velocity Zone Locations from Highly Anomalous SPdKS Waveforms, Minerals, 10, 211, 2020.

Valentine, A. P. and Trampert, J.: Data space reduction, quality assessment and searching of seismograms: autoencoder networks for waveform data, Geophysical Journal International, 189, 1183–1202, 2012.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K.: WaveNet: A Generative Model for Raw Audio, ArXiv e-prints, 2016.

Van Der Baan, M. and Jutten, C.: Neural networks in geophysical applications, Geophysics, 65, 1032–1047, 2000.

van Driel, M. and Nissen-Meyer, T.: Optimized viscoelastic wave propagation for weakly dissipative media, Geophysical Journal International, 199, 1078–1093, 2014a.

van Driel, M. and Nissen-Meyer, T.: Seismic wave propagation in fully anisotropic axisymmetric media, Geophysical Journal International, 199, 880–893, 2014b.

Van Driel, M., Ceylan, S., Clinton, J. F., Giardini, D., Alemany, H., Allam, A., Ambrois, D., Balestra, J., Banerdt, B., Becker, D., Böse, M., Boxberg, M. S., Brinkman, N., Casademont, T., Chèze, J., Daubar, I., Deschamps, A., Dethof, F., Ditz, M., Drilleau, M., Essing, D., Euchner, F., Fernando, B., Garcia, R., Garth, T., Godwin, H., Golombek, M. P., Grunert, K., Hadziioannou, C., Haindl, C., Hammer, C., Hochfeld, I., Hosseini, K., Hu, H., Kedar, S., Kenda, B., Khan, A., Kilchling, T., Knapmeyer-Endrun, B., Lamert, A., Li, J., Lognonné, P., Mader, S., Marten, L., Mehrkens, F., Mercerat, D., Mimoun, D., Möller, T., Murdoch, N., Neumann, P., Neurath, R., Paffrath, M., Panning, M. P., Peix, F., Perrin, L., Rolland, L., Schimmel, M., Schröer, C., Spiga, A., Stähler, S. C., Steinmann, R., Stutzmann, E., Szenicer, A., Trumpik, N., Tsekhmistrenko, M., Twardzik, C., Weber, R., Werdenbach-Jarklowski, P., Zhang, S., and Zheng, Y.: Preparing for InSight: Evaluation of the blind test for martian seismicity, Seismological Research Letters, 90, 1518–1534, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I.: Attention Is All You Need, ArXiv e-prints, 2017.

Vinje, V., Iversen, E., and Gjoystdal, H.: Traveltime and amplitude estimation using wavefront construction, Geophysics, 58, 1157–1166, 1993.

Virieux, J. and Operto, S.: An overview of full-waveform inversion in exploration geophysics, Geophysics, 74, 2009.

Wu, Y. and Lin, Y.: InversionNet: A Real-Time and Accurate Full Waveform Inversion with CNNs and continuous CRFs, ArXiv e-prints, pp. 1–14, 2018.

Xie, X.-B., Jin, S., and Wu, R.-S.: Wave-equation-based seismic illumination analysis, Geophysics, 71, S169–S177, 2006.

Yang, F. and Ma, J.: Deep-learning inversion: A next-generation seismic velocity model building method, Geophysics, 84, R583–R599, 2019.

Zhang, Z. and Lin, Y.: Data-driven Seismic Waveform Inversion: A Study on the Robustness and Generalization, ArXiv e-prints, pp. 1–13, 2018.

Zhu, W., Sheng, Y., and Sun, Y.: Wave-dynamics simulation using deep neural networks, Stanford Report, 2017.