Theory-based machine learning elastic full waveform inversion with various parameterizations

Tianze Zhang, Kris Innanen, Jian Sun, Daniel Trad

2019 10 13th

University of Calgary ES 136

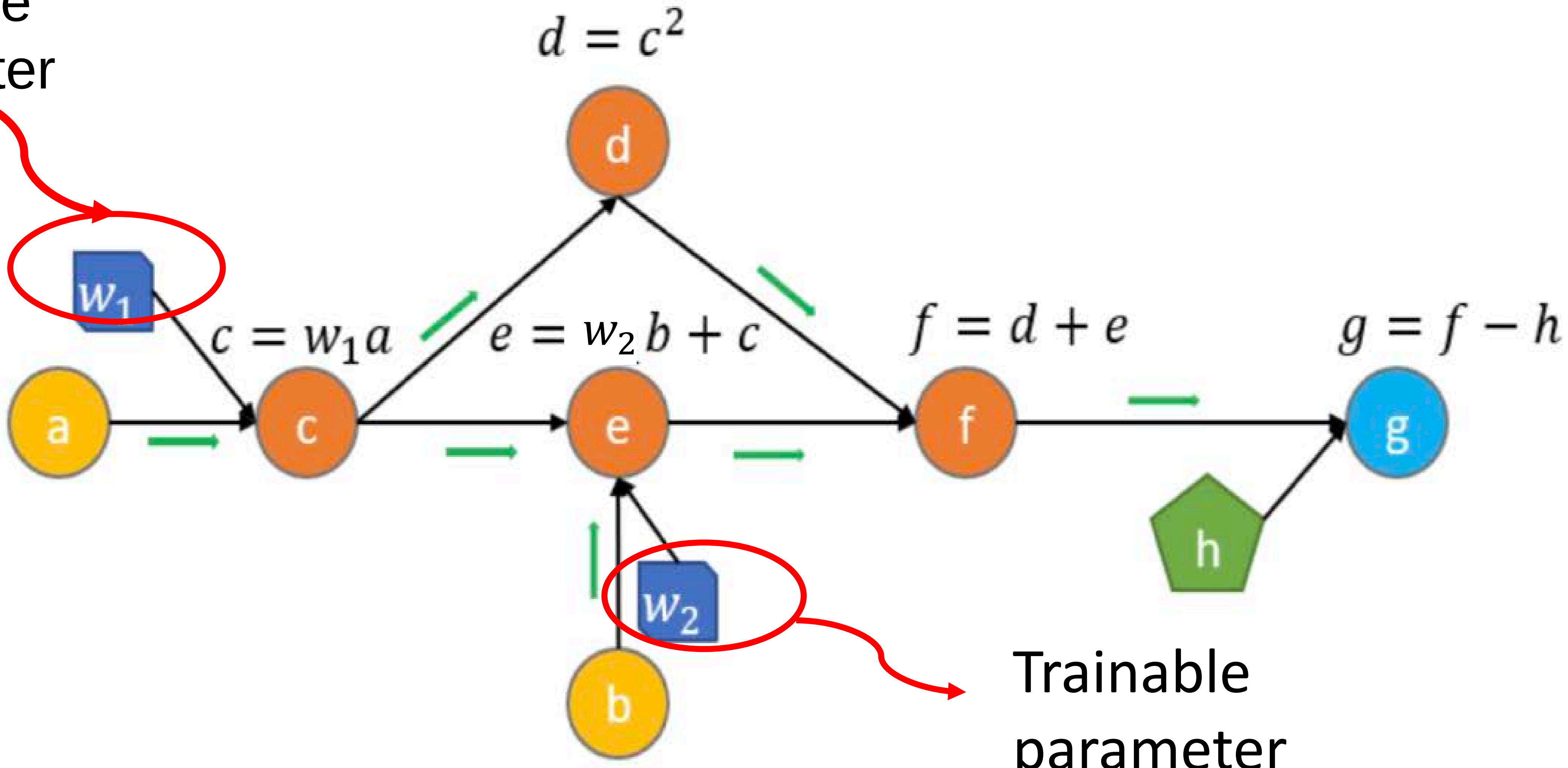
CREWES Teck Talk

Outline

- Introduction neural networks
- Recurrent neural networks (RNN)
- An RNN formulation of elastic full waveform inversion
- Noise stress test
- Conclusions

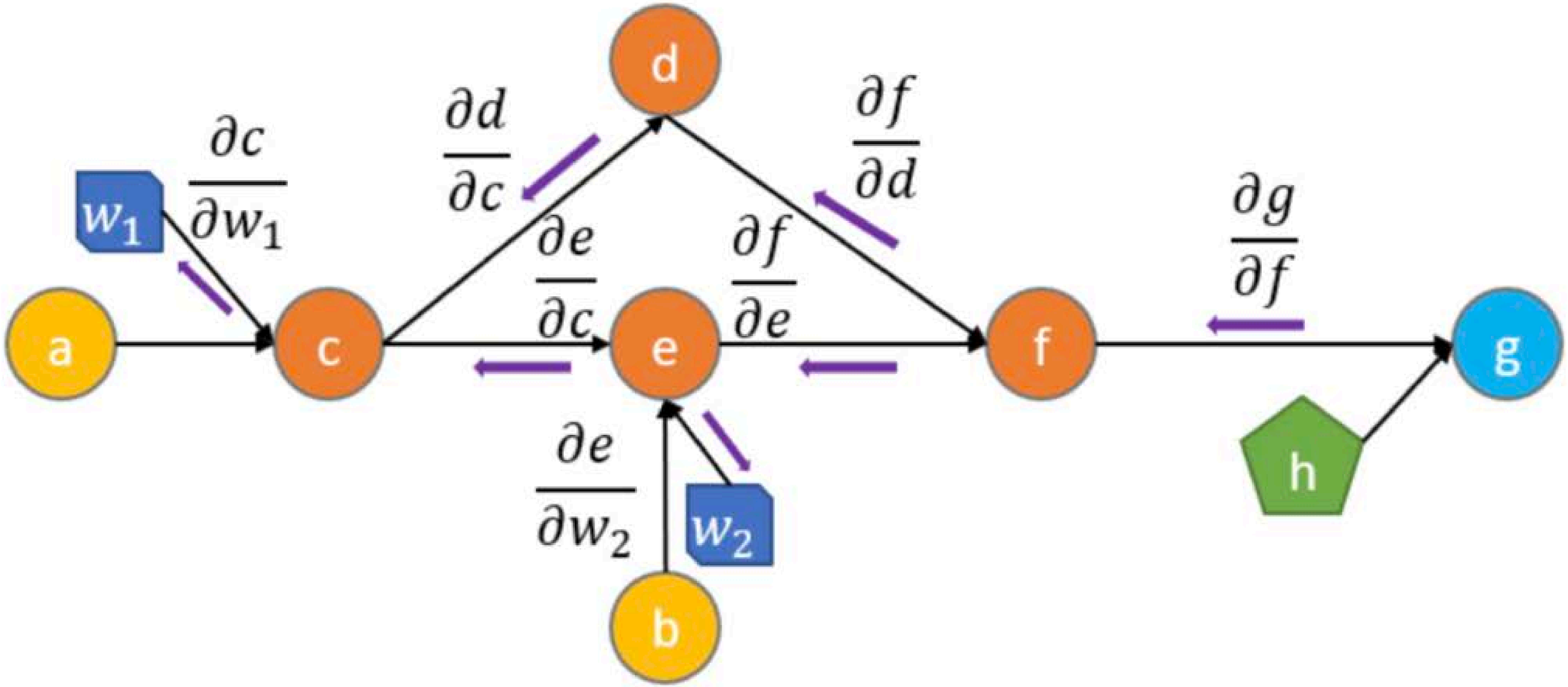
1. Feed forward network

Trainable parameter

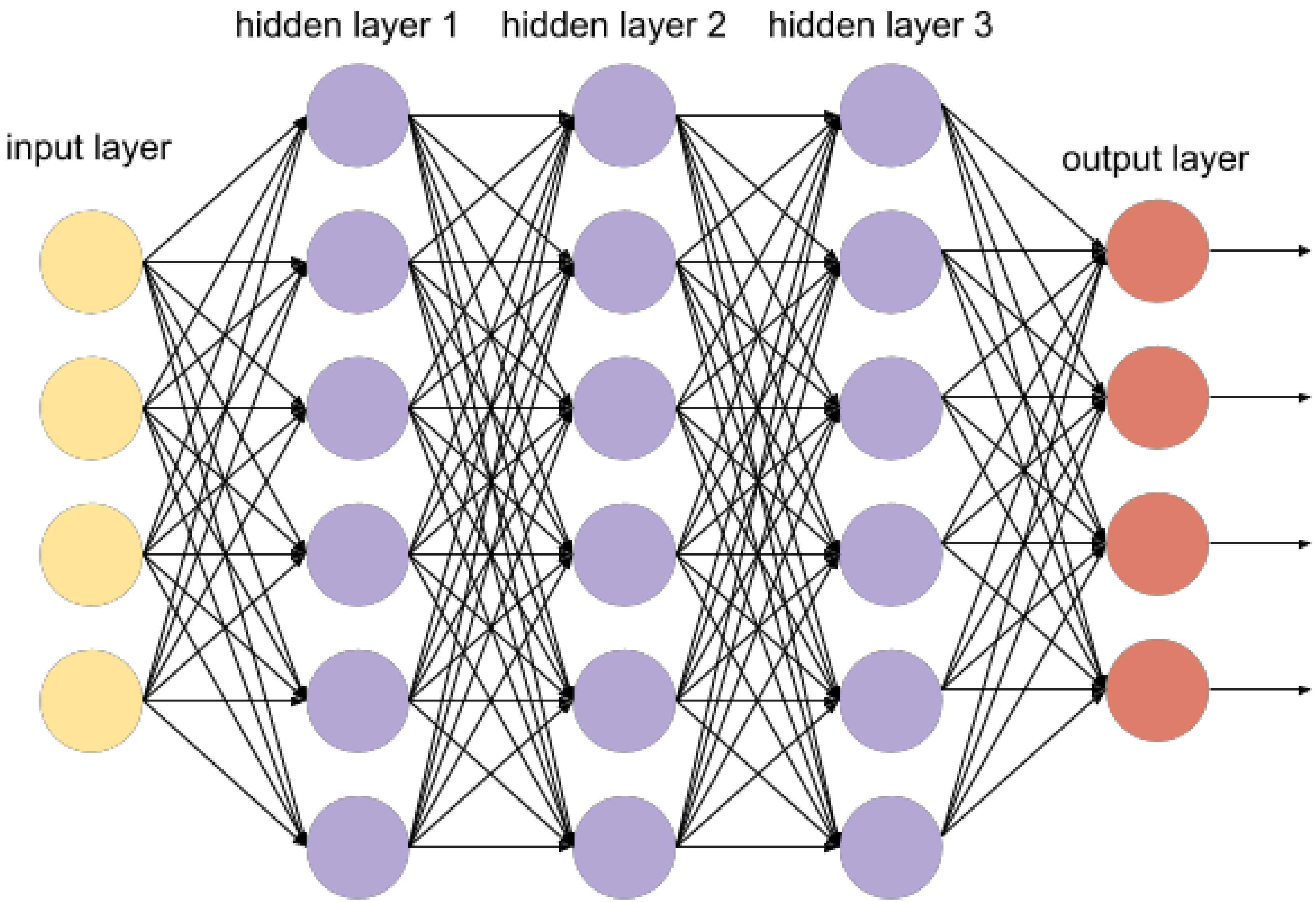


Trainable parameter

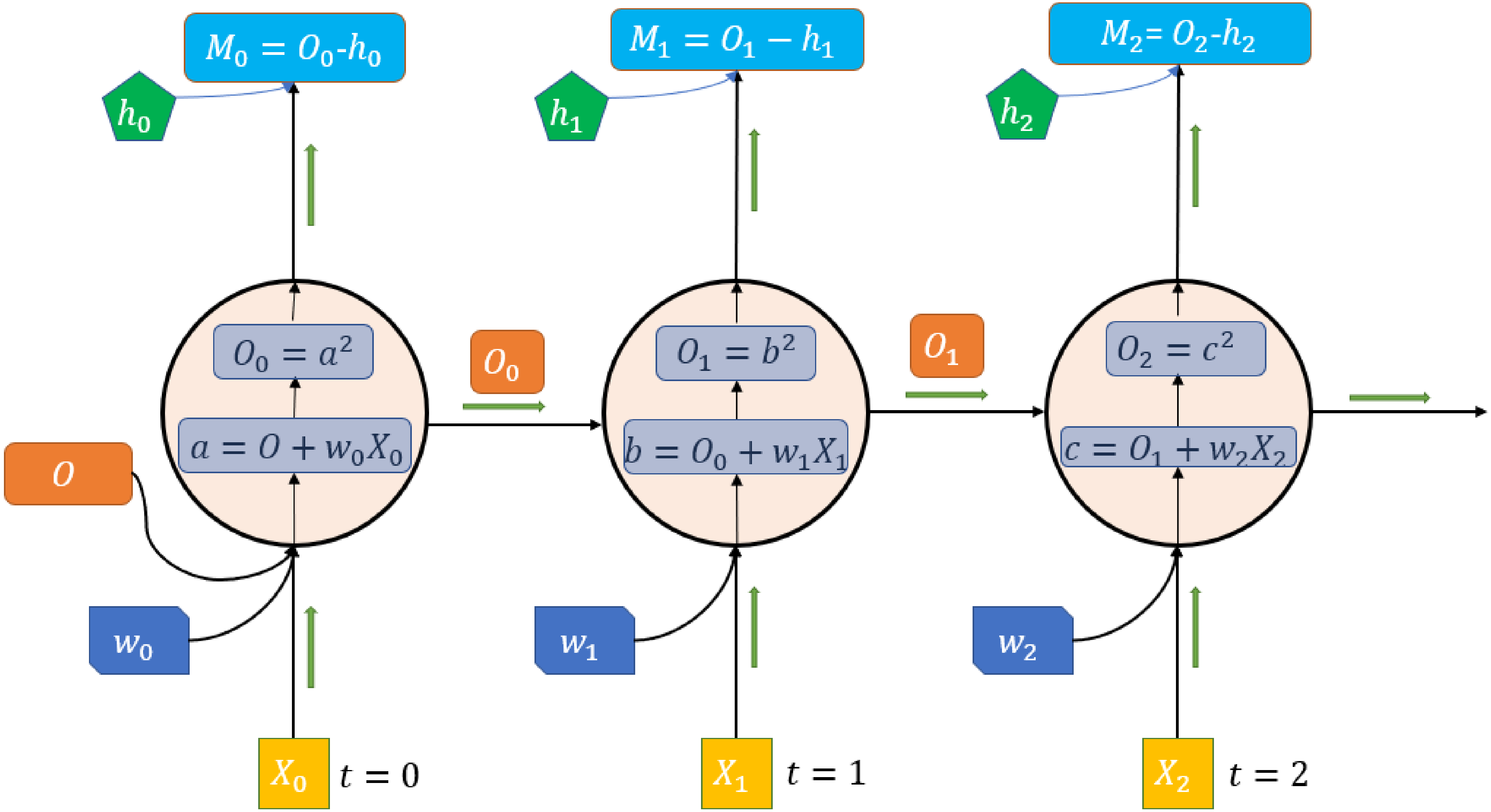
1. Feed forward network



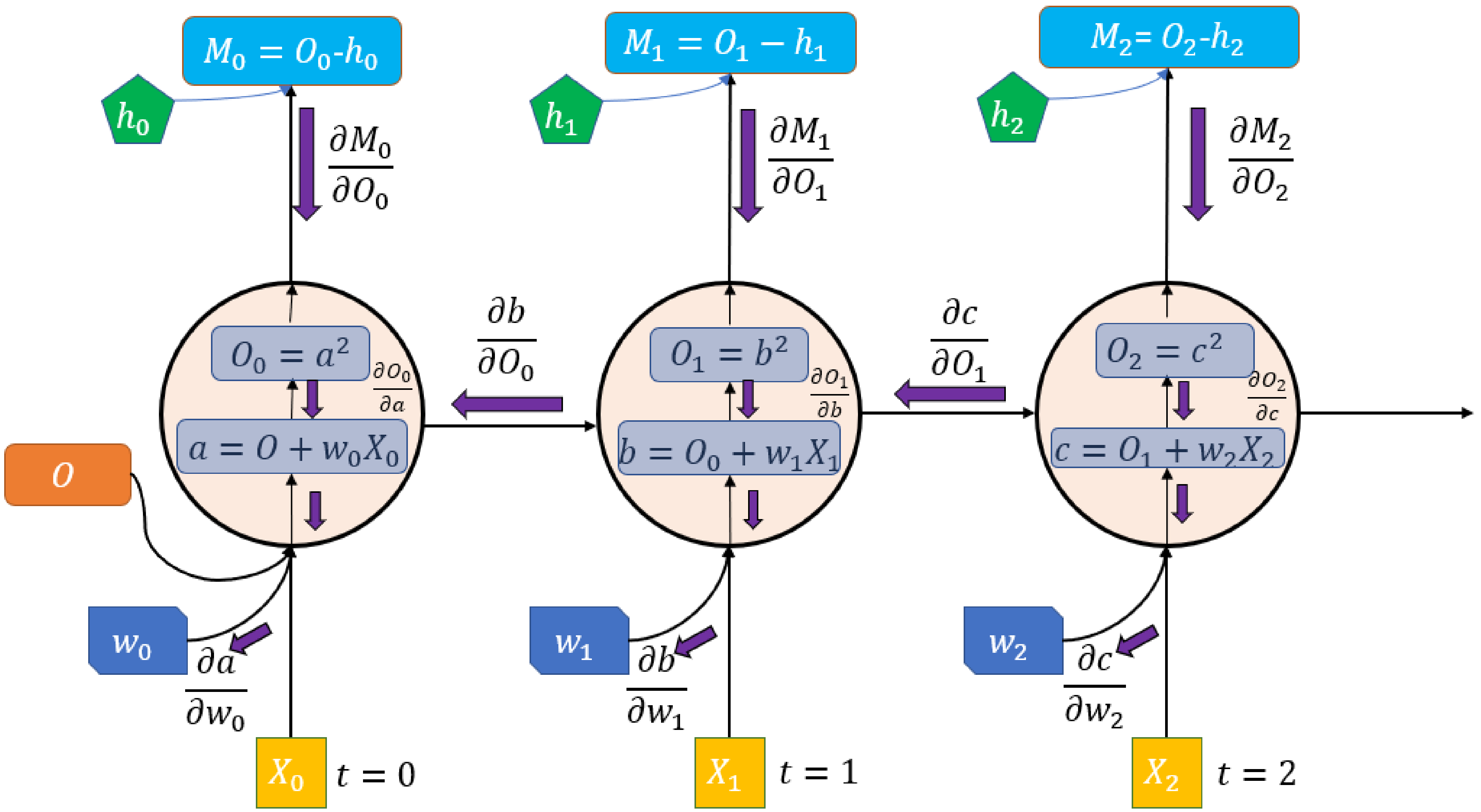
1. Fully connected network



2. Recurrent neural network (RNN) forward



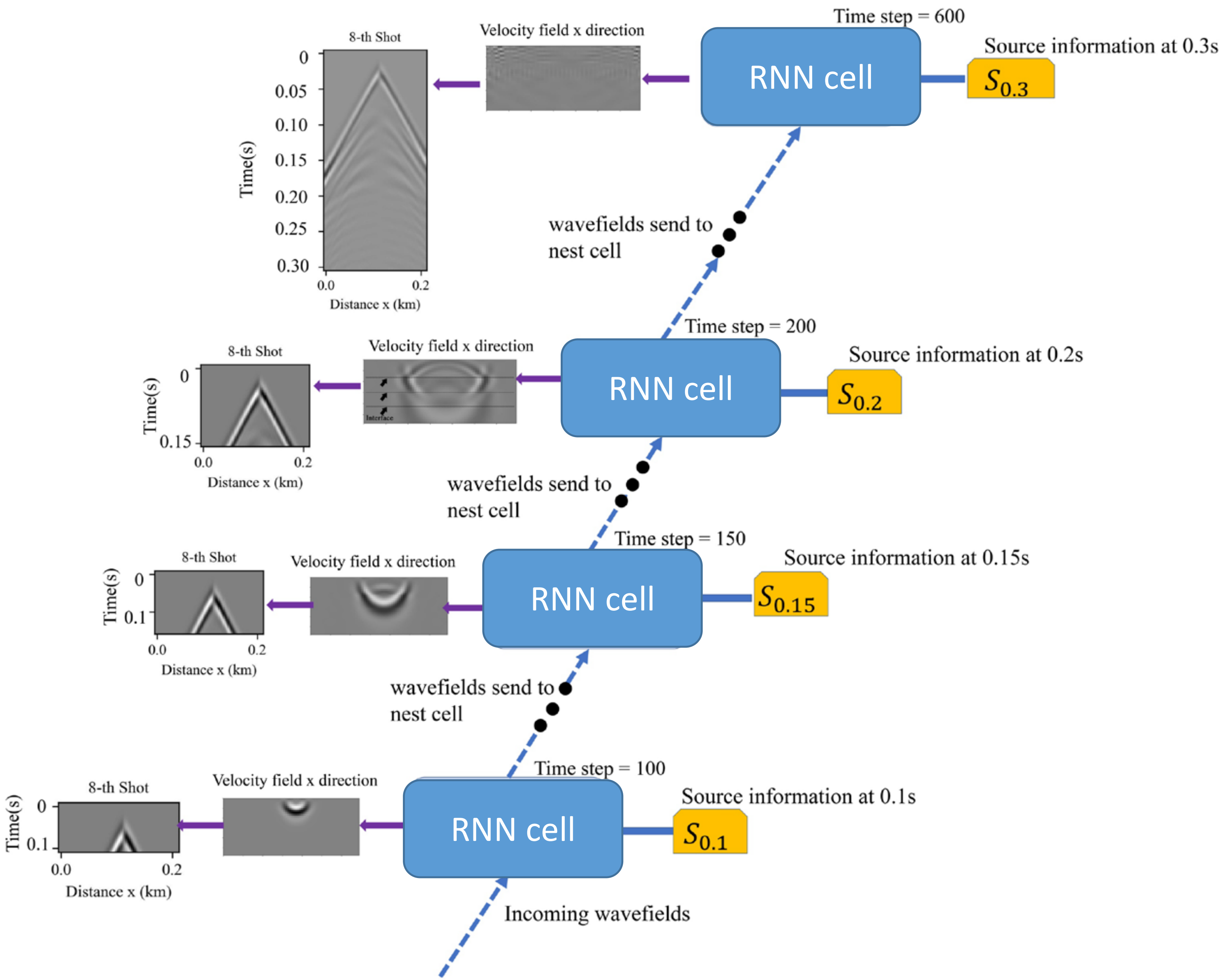
2. Recurrent neural network (RNN) backward



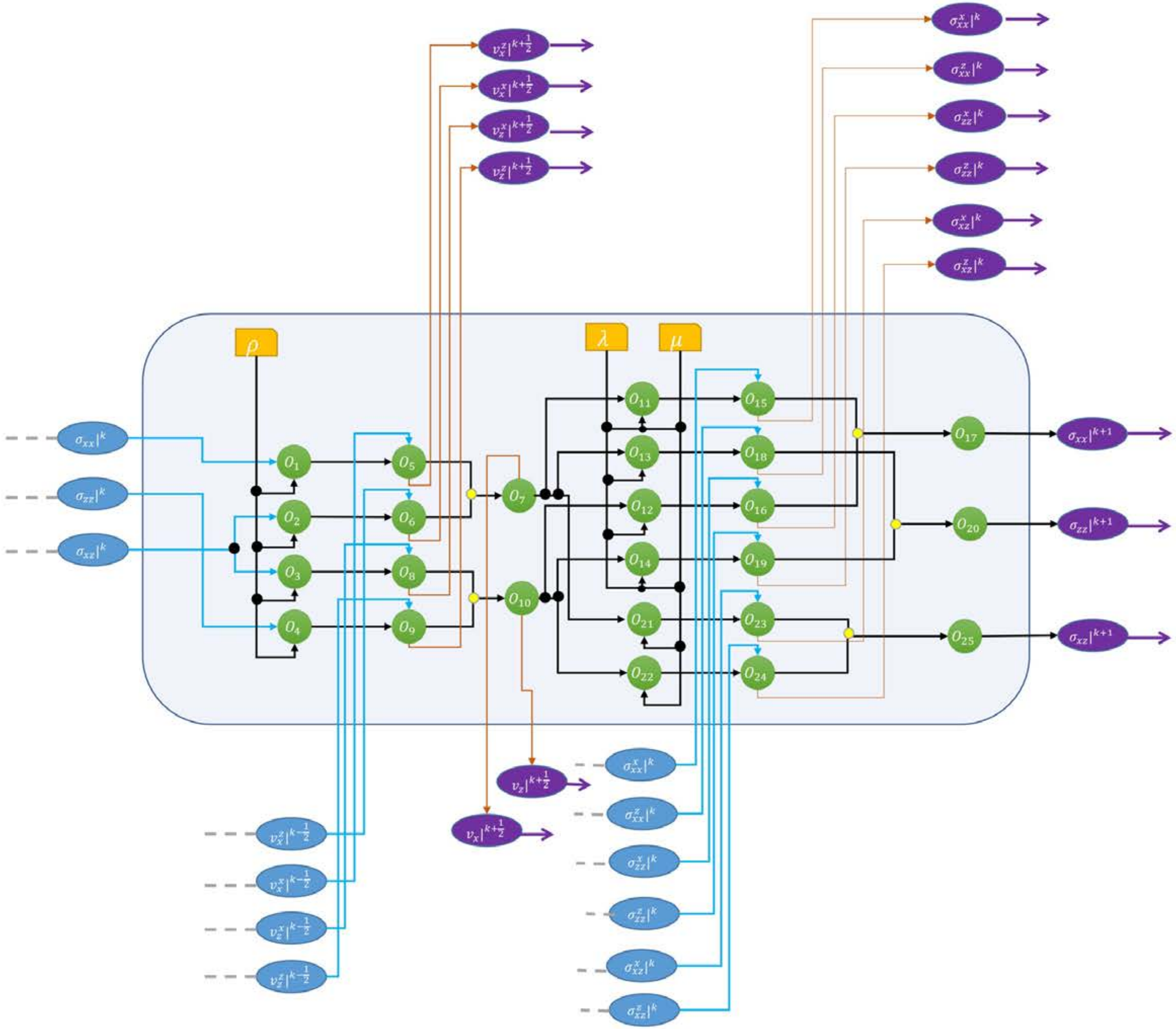
3.1 Isotropic elastic wave equation

$$
\begin{cases}
\dfrac{\partial \mathbf{v}_x}{\partial t} = \dfrac{1}{\rho}\left(\dfrac{\partial \boldsymbol{\sigma}_{xx}}{\partial x} + \dfrac{\partial \boldsymbol{\sigma}_{xz}}{\partial z}\right) \\
\dfrac{\partial \mathbf{v}_z}{\partial t} = \dfrac{1}{\rho}\left(\dfrac{\partial \boldsymbol{\sigma}_{xz}}{\partial x} + \dfrac{\partial \boldsymbol{\sigma}_{zz}}{\partial z}\right) \\
\dfrac{\partial \boldsymbol{\sigma}_{xx}}{\partial t} = (\lambda + 2\mu)\dfrac{\partial \mathbf{v}_x}{\partial x} + \lambda\dfrac{\partial \mathbf{v}_x}{\partial x} \\
\dfrac{\partial \boldsymbol{\sigma}_{zz}}{\partial t} = (\lambda + 2\mu)\dfrac{\partial \mathbf{v}_z}{\partial z} + \lambda\dfrac{\partial \mathbf{v}_x}{\partial x} \\
\dfrac{\partial \boldsymbol{\sigma}_{xz}}{\partial t} = \mu\left(\dfrac{\partial \mathbf{v}_x}{\partial z} + \dfrac{\partial \mathbf{v}_z}{\partial x}\right)
\end{cases}
$$

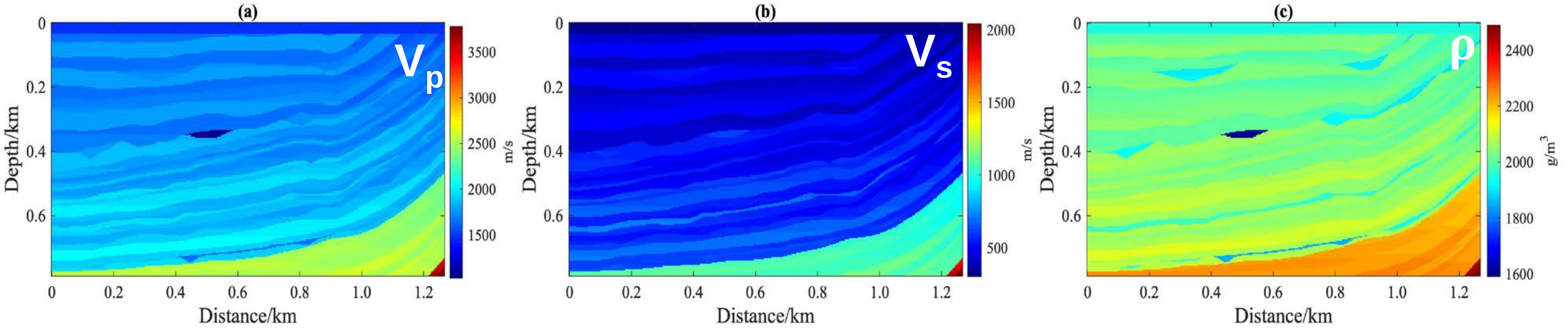
3.2 RNN cell designed according to elastic wave equation



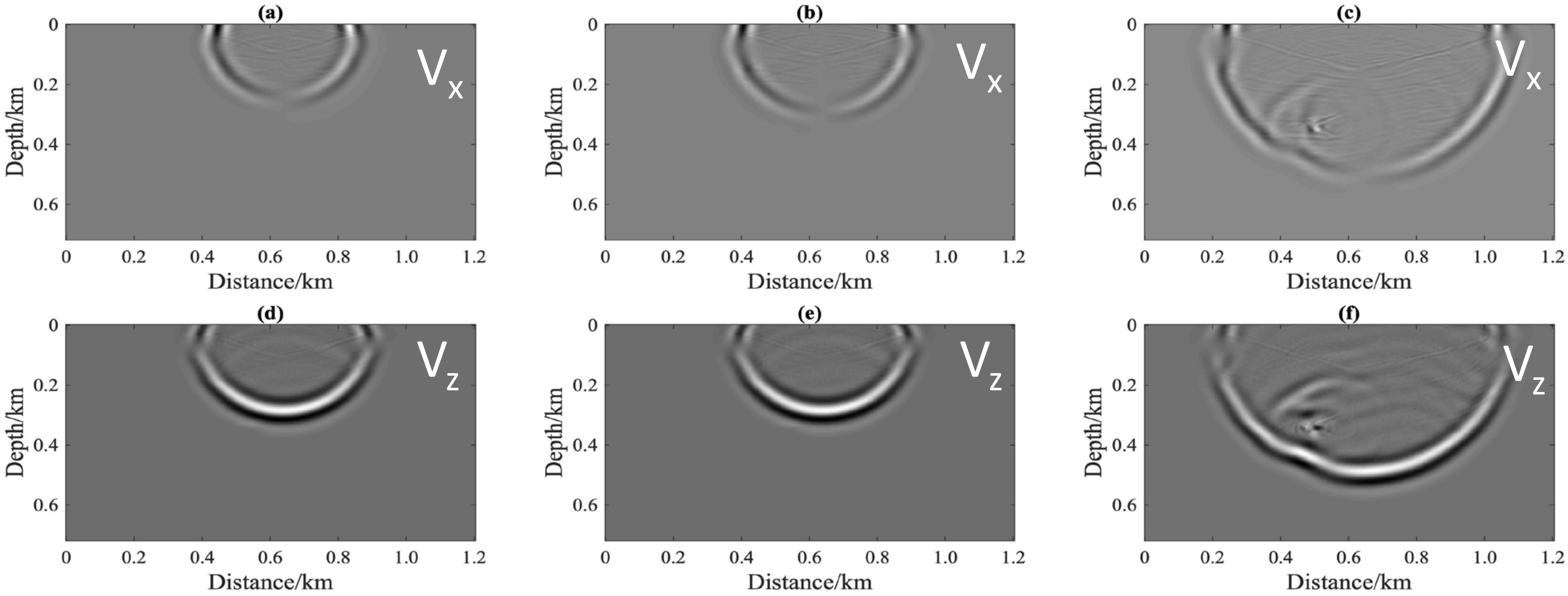
3.2 RNN cell designed according to elastic wave equation



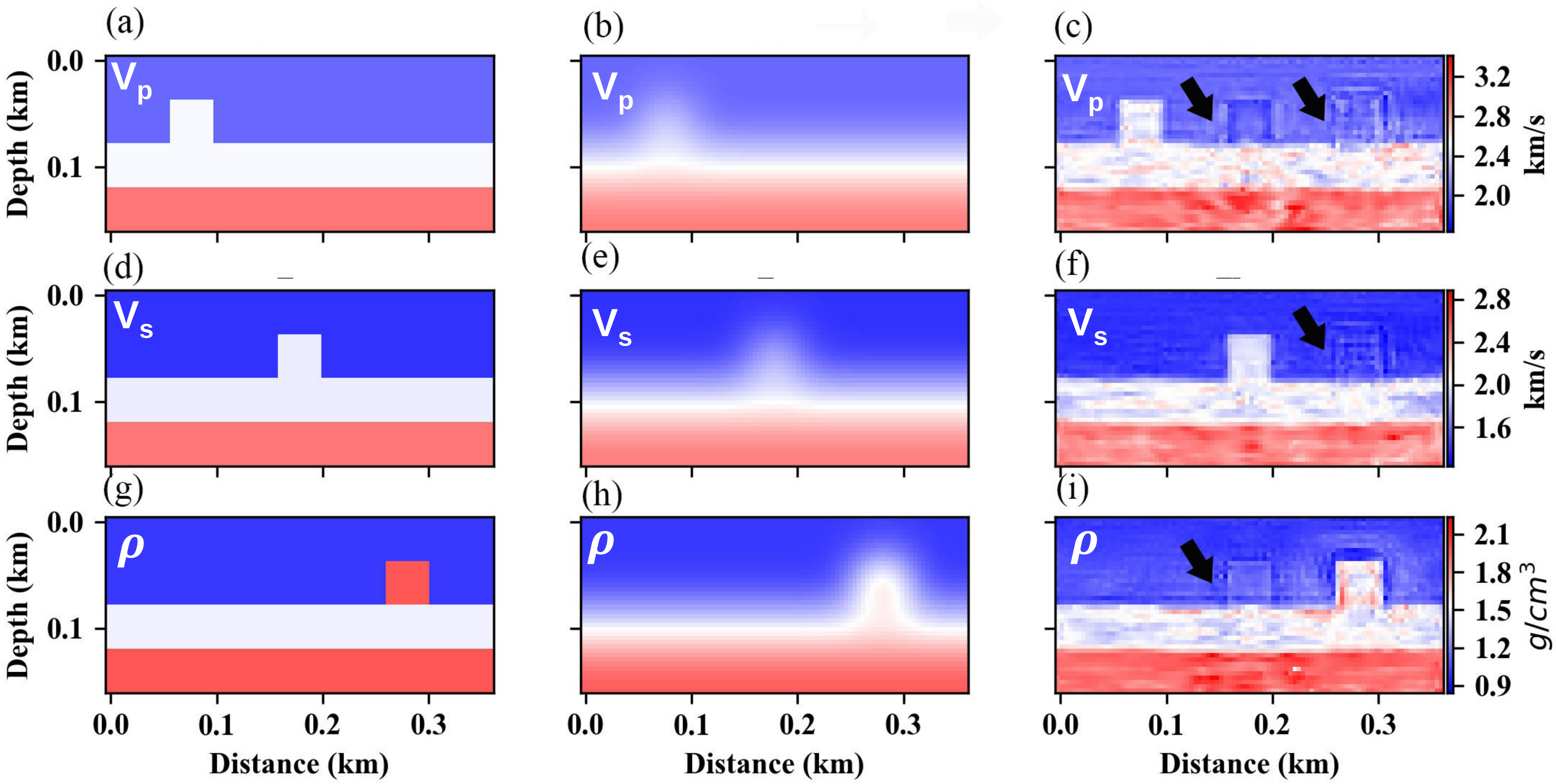
3.3 Elastic media to test forward modeling



3.4 Velocity fields generated by elastic RNN



3.5 Velocity parameterization-Toy model



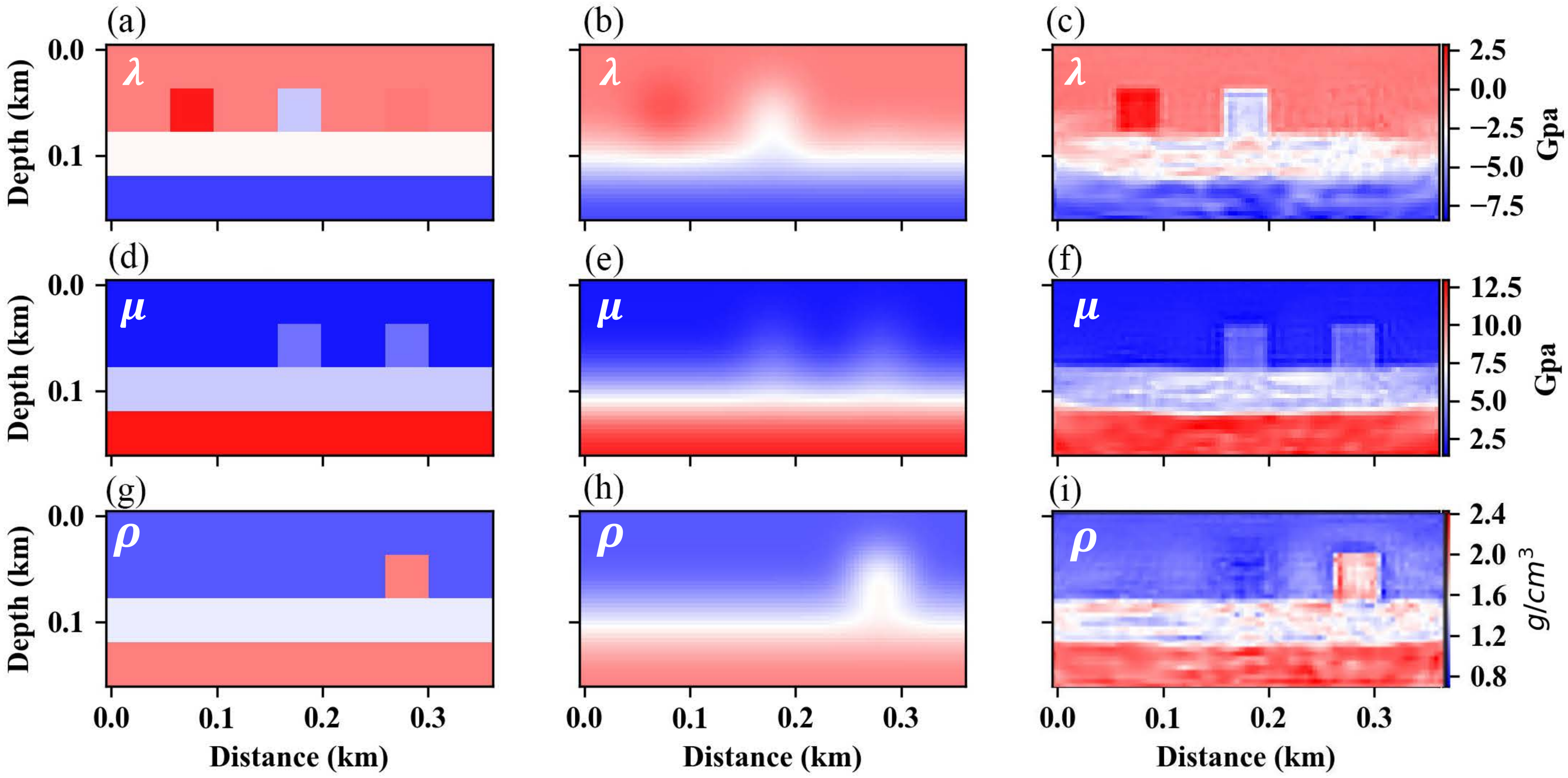
20 shots (top)

90 receivers (top)

f_{dom} 35Hz

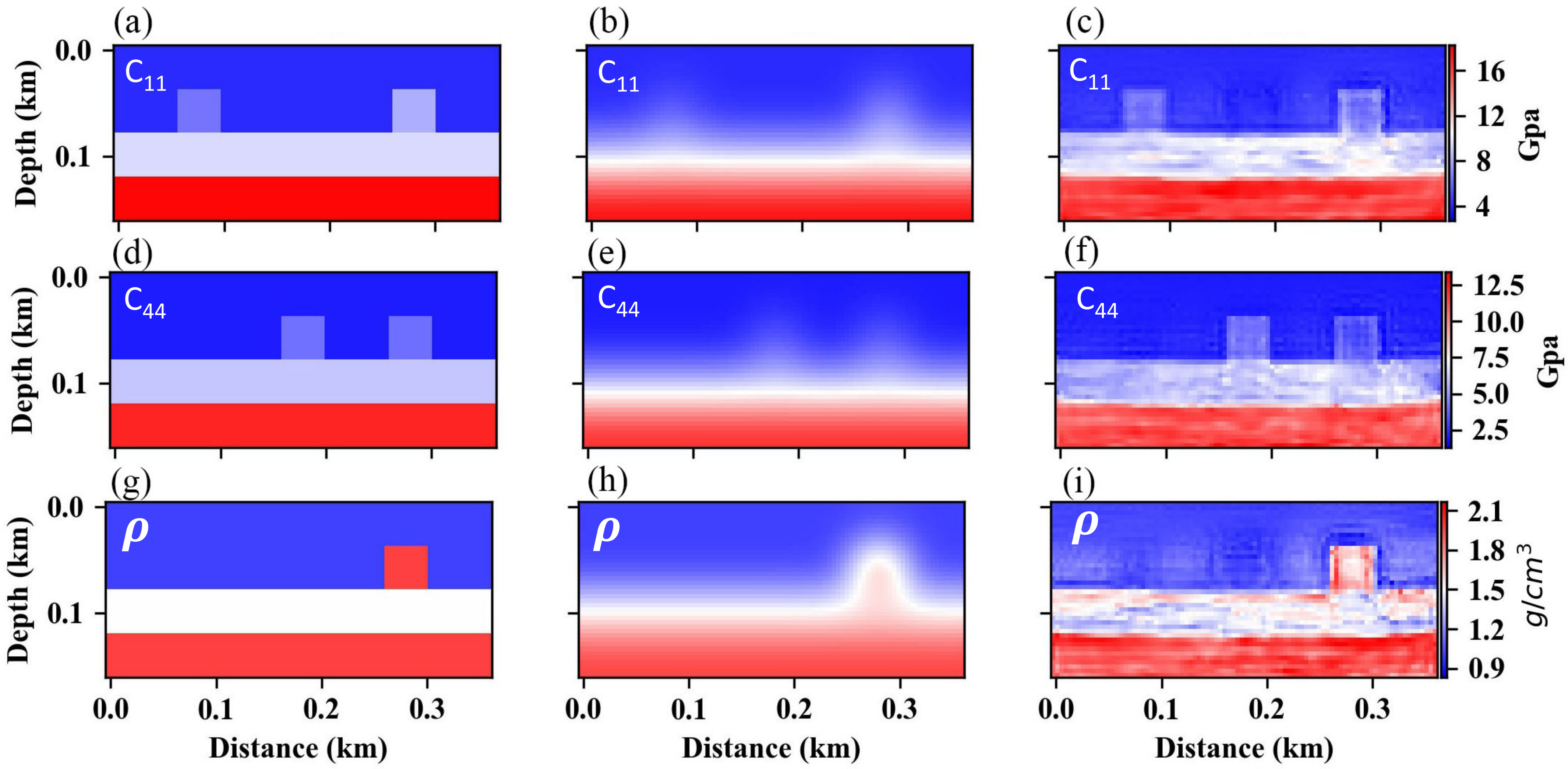
T_{max} 0.25s

3.6 Modulus parameterization-Toy model



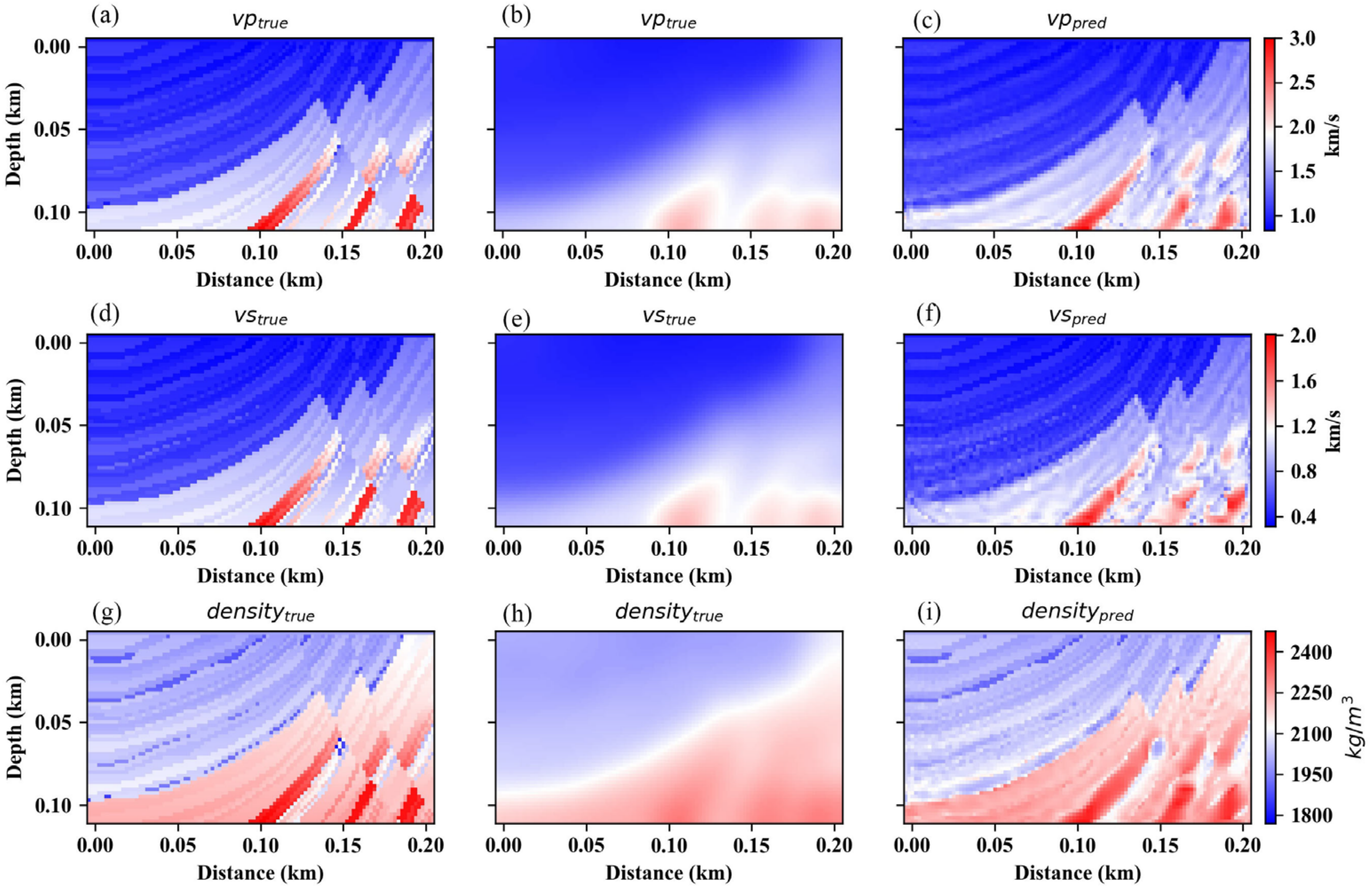
20 shots (top)
90 receivers (top)
f_{dom} 35Hz

3.7 Stiffness matrix parameterization-Toy model



20 shots (top)
90 receivers (top)
f_{dom} 35Hz
T_{max} 0.25s

3.8 Velocity parameterization:Part of Marmousi model



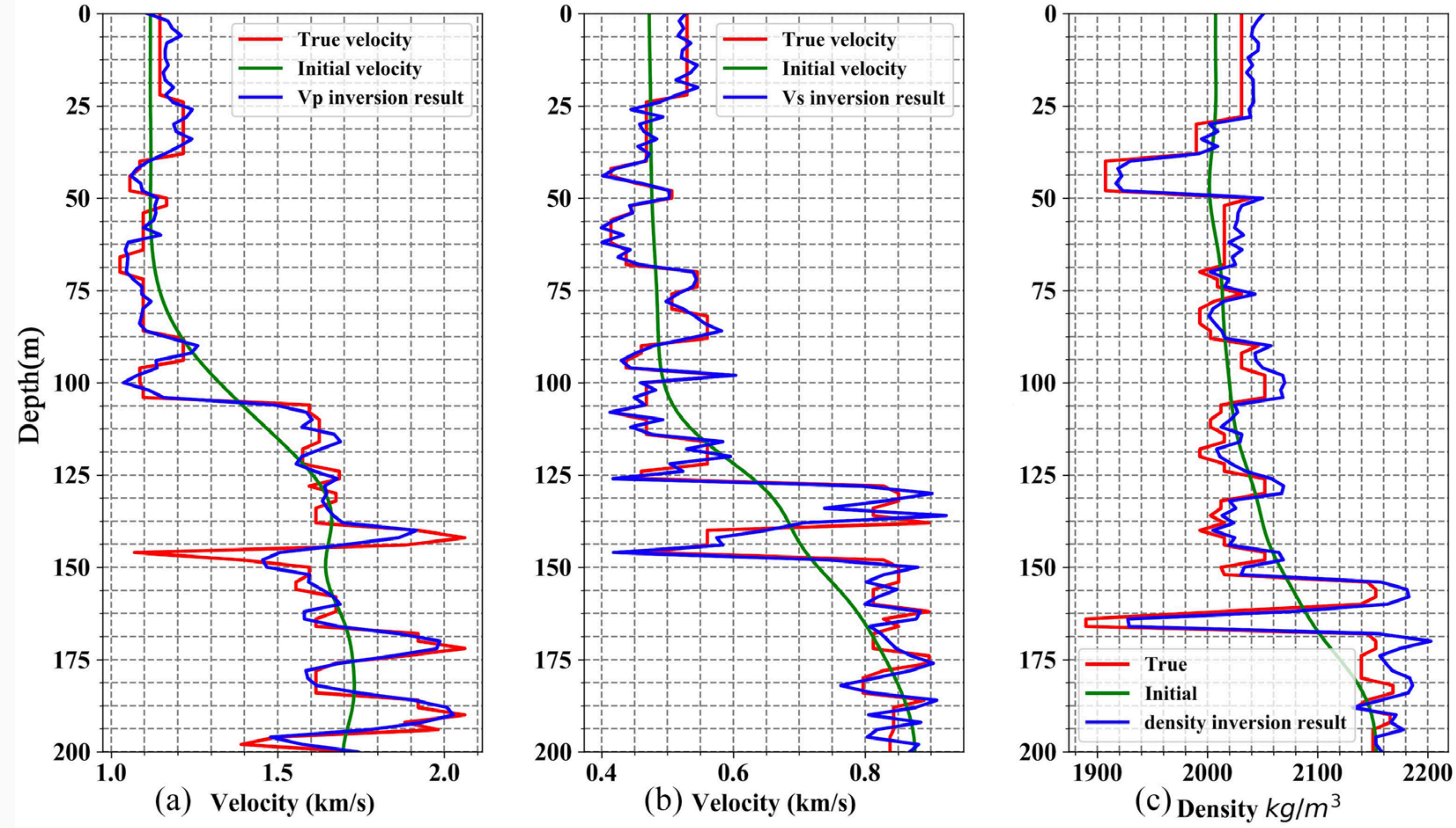
10 shots (top)

100 receivers (top)

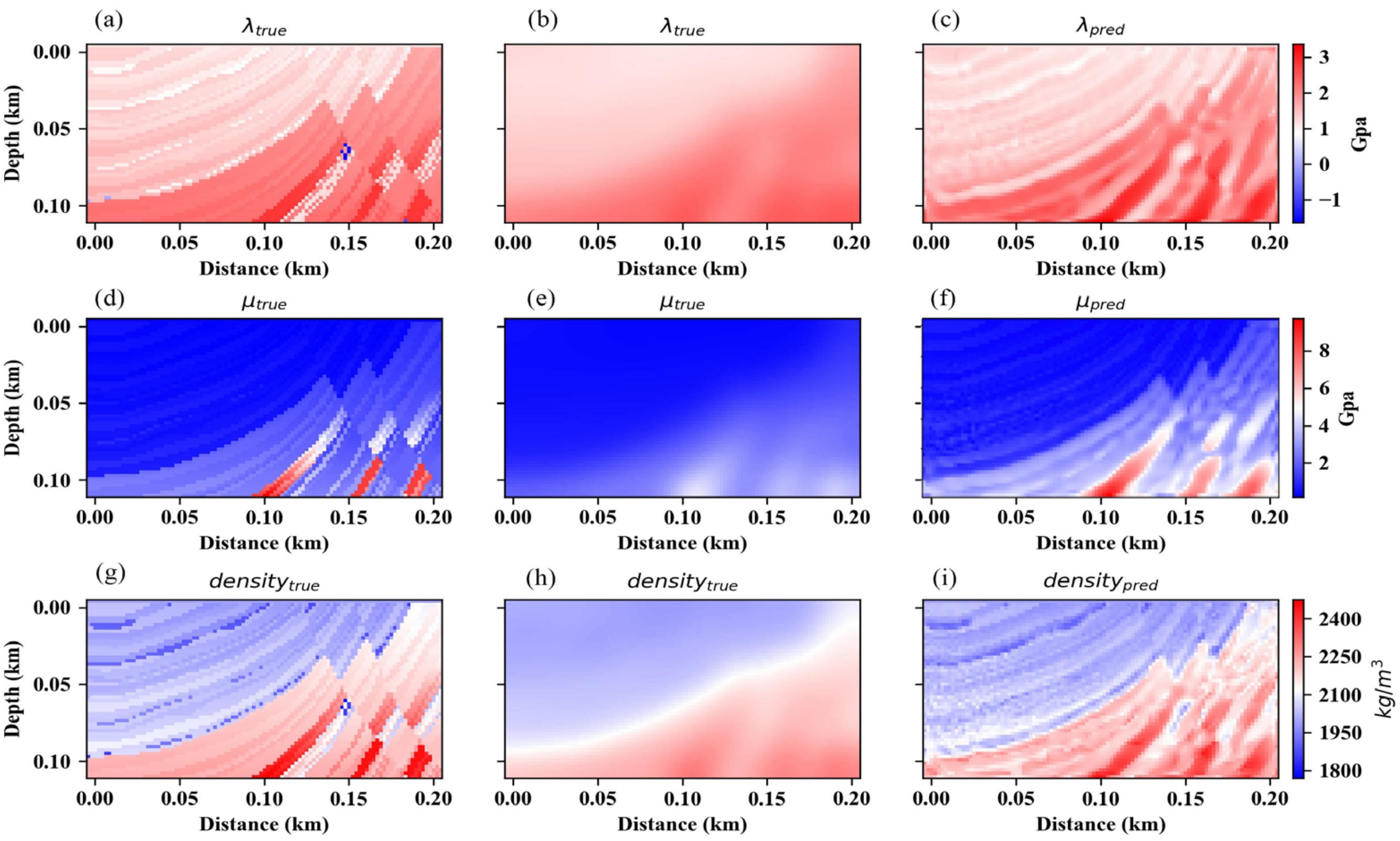
f_{dom} 35Hz

T_{max} 0.25s

3.8 Velocity parameterization:Part of Marmousi model



3.9 Modulus parameterization Part of Marmousi model



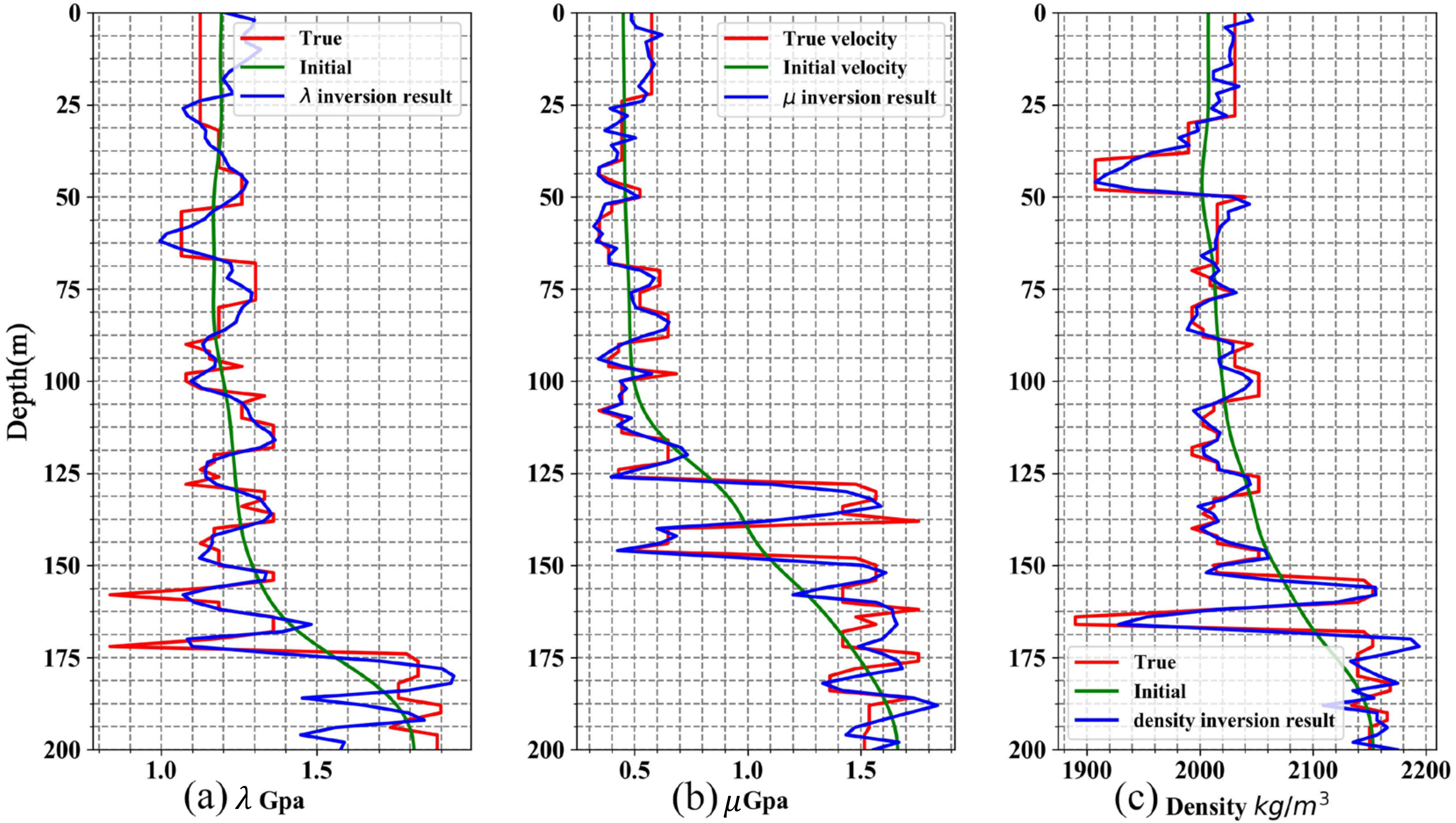
10 shots (top)

100 receivers (top)

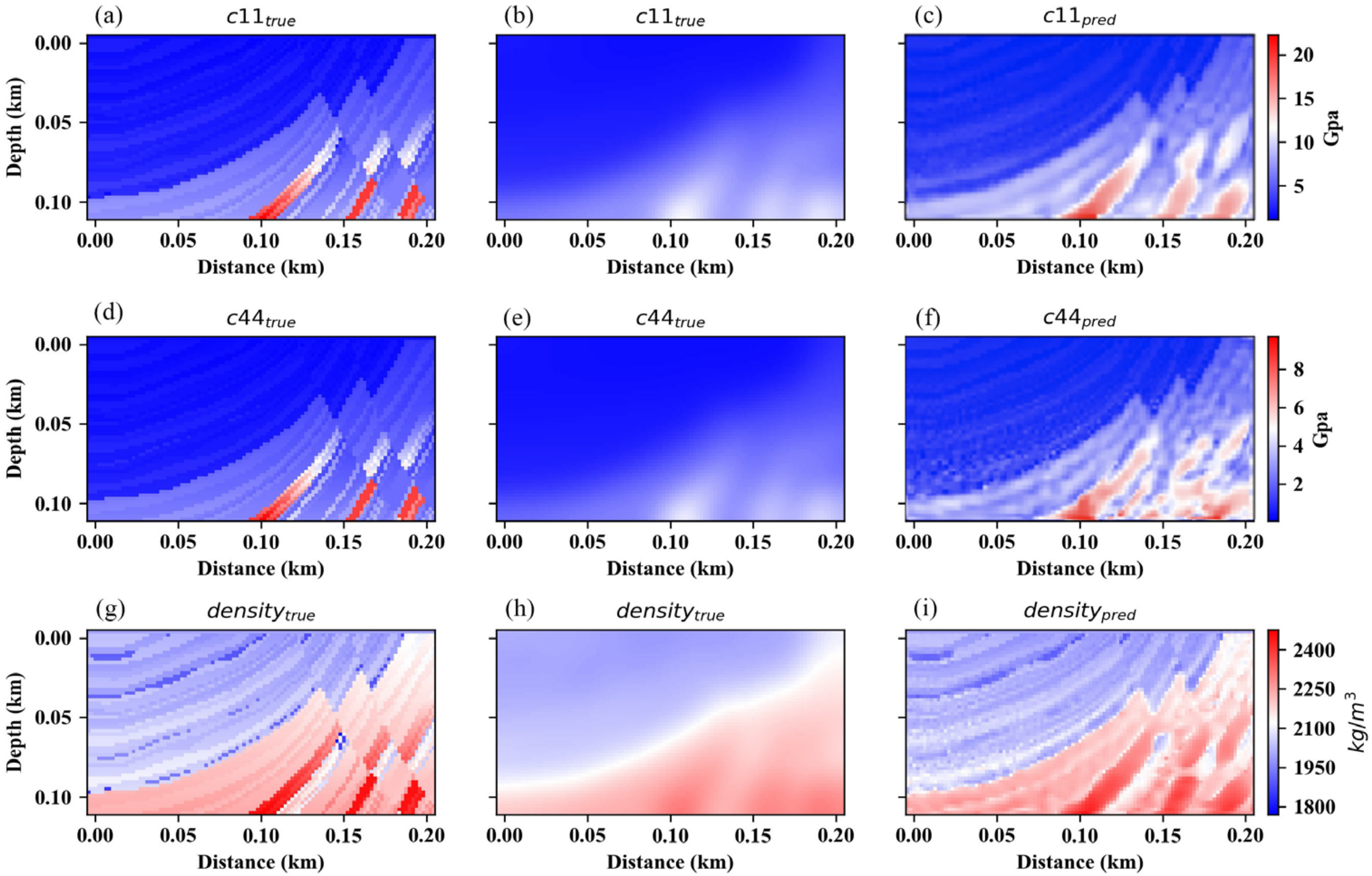
f_{dom} 35Hz

T_{max} 0.25s

3.9 Modulus parameterization Part of Marmousi model



3.10 Stiffness matrix parameterization Part of Marmousi model



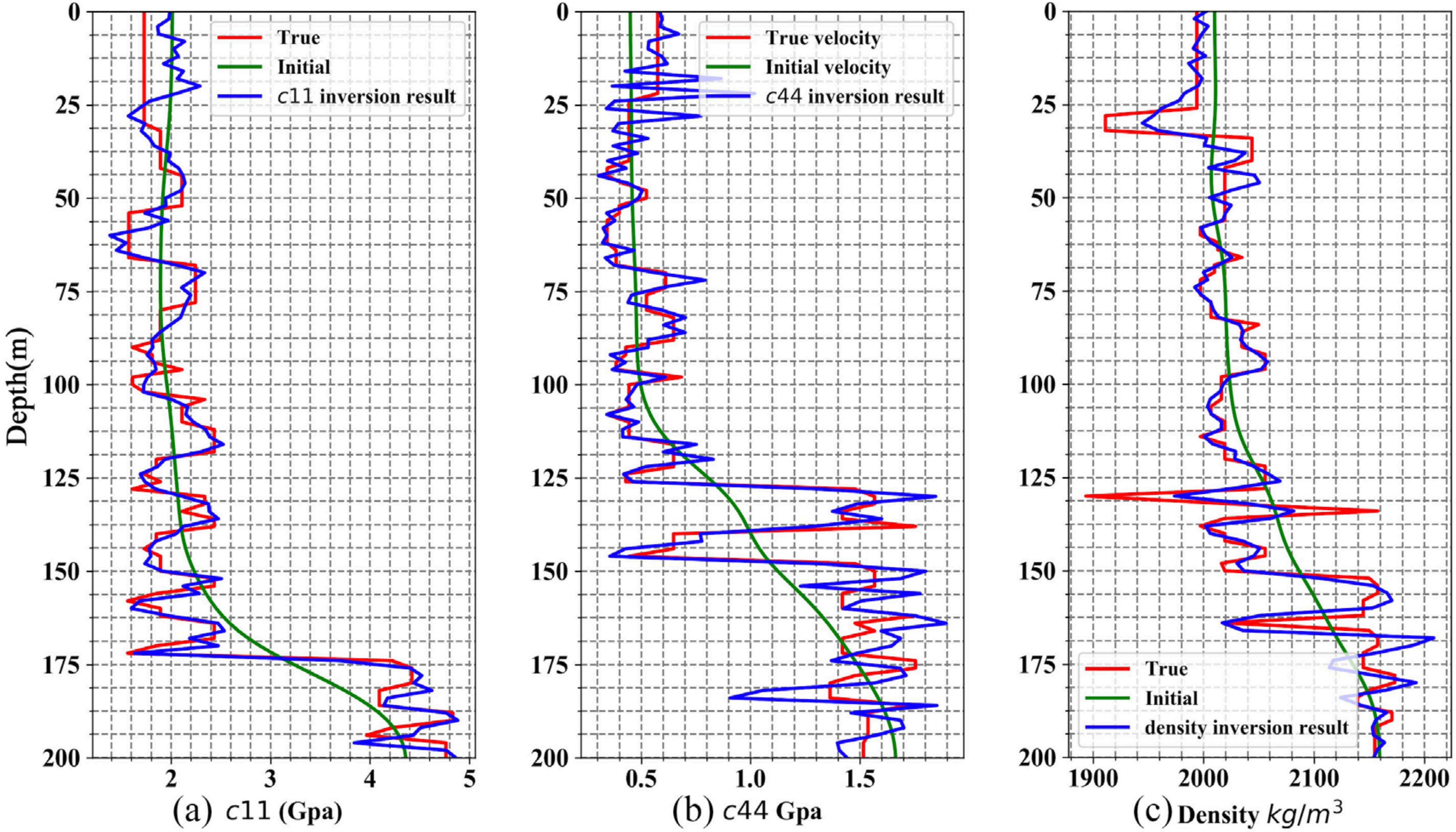
10 shots (top)

100 receivers (top)

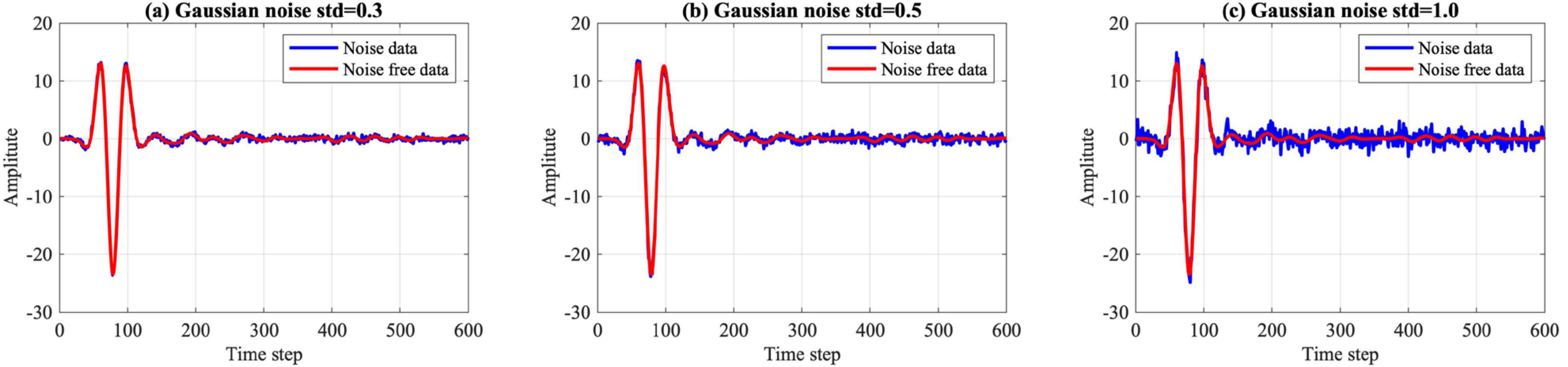
f_{dom} 35Hz

T_{max} 0.25s

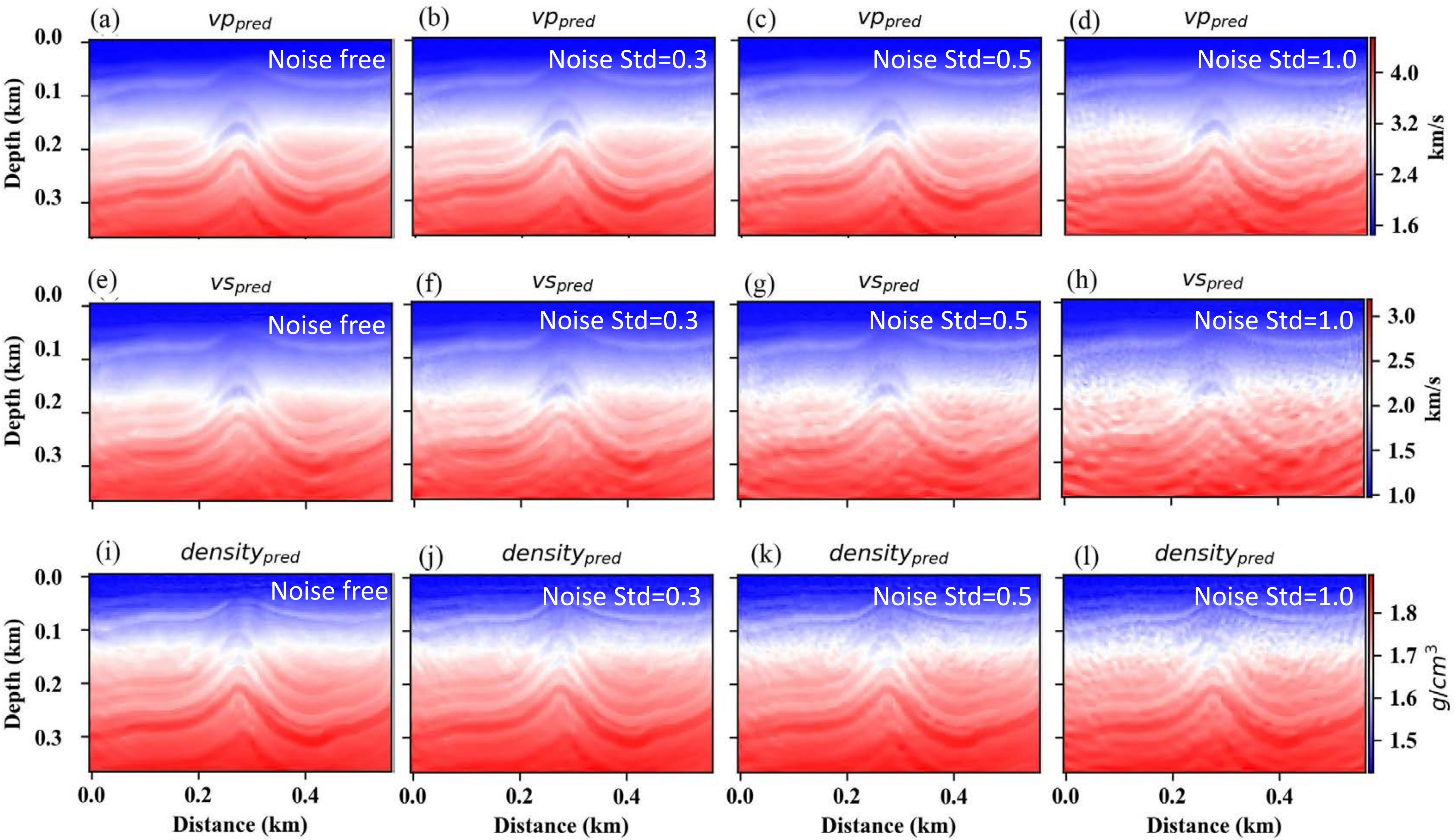
3.10 Stiffness matrix parameterization Part of Marmousi model



4. Noise stress test



4. Noise stress test



5 Conclusions

- RNN formulation of elastic FWI is robust and admits a range of optimization choices (e.g., Adam)
- These are gradient based; cross-talk is managed prior to inversion
- Modelling error:
  - Likely to cause issues for RNN/FWI
  - Can potentially be addressed through training
  - Can a deep learning FWI algorithm “teach itself” which physics rules to use?

Thanks

Thanks CREWES sponsors and students
Thanks Kris Innanen, Daniel Trad, Jian Sun and Zhan Niu for their discussion.