Seismic Velocity Inversion and Uncertainty Quantification Using Conditional Normalizing Flows

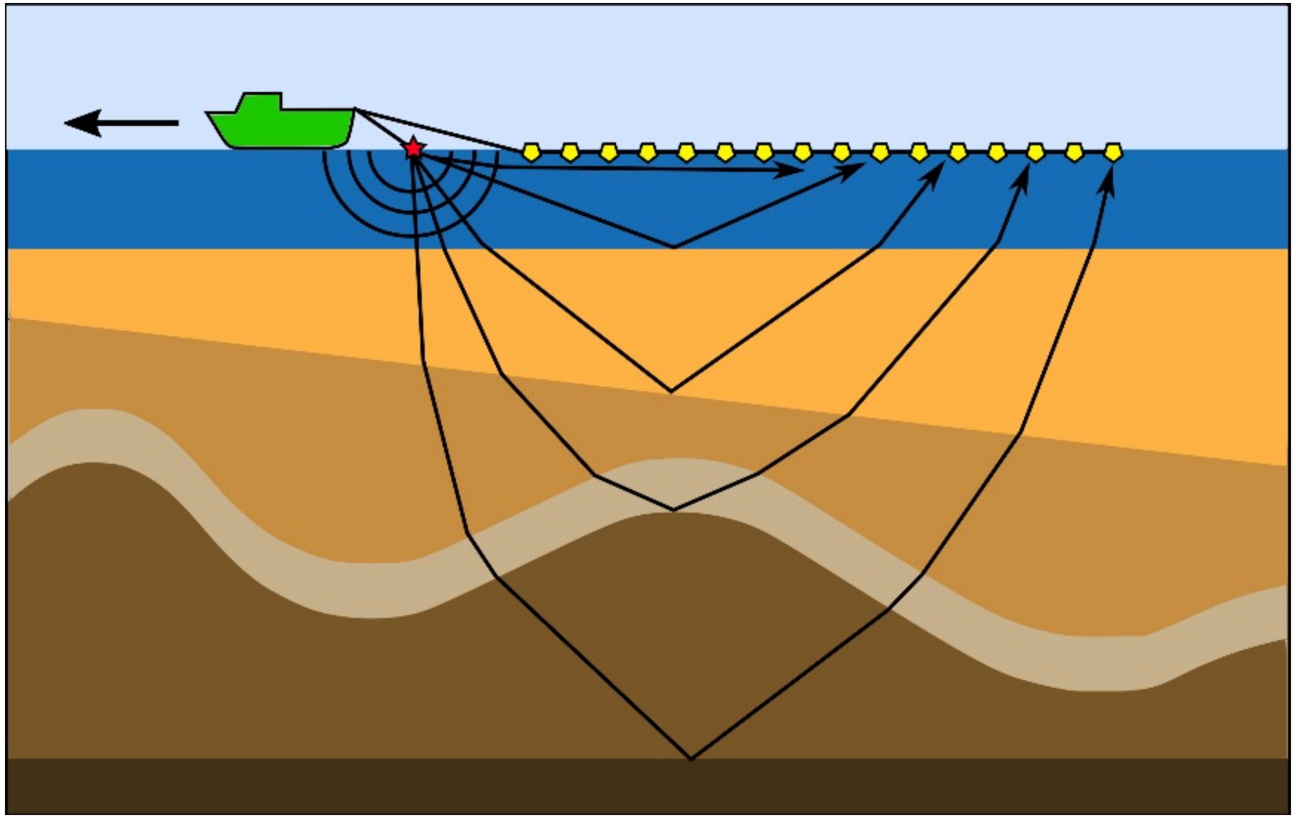
Yuxiao Ren[1,2], Philipp A. Witte[3], Ali Siahkoohi[2], Mathias Louboutin[2], Ziyi Yin[2], Felix J. Herrmann[2]

[1] Shandong University, China
[2] Georgia Institute of Technology, USA
[3] Georgia Institute of Technology, now Microsoft

SLIM
Georgia Institute of Technology

Motivation

- Seismic velocity inversion plays a vital role in seismic exploration.
- It is an ill-posed problem where different solutions may fit the data equally well.



Bayesian Inversion

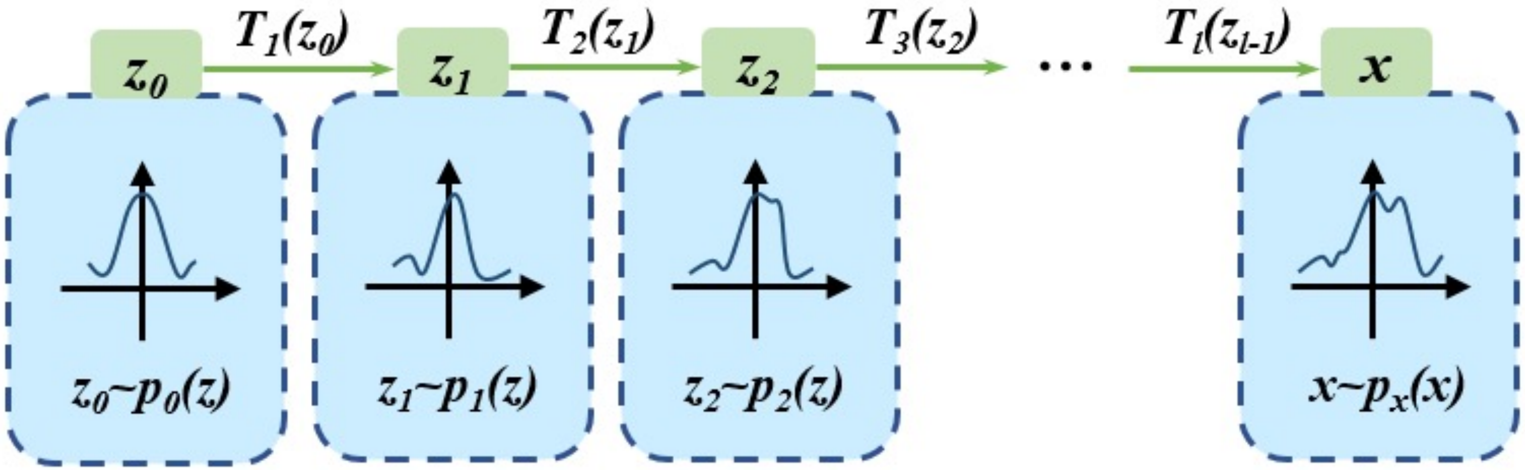
- From the perspective of Bayesian inversion, the multiple solutions can be considered as posterior samples of the solution distribution
- Given the observed data $\mathbf{d}$, seismic velocity model $\mathbf{m}$ can be expressed as the posterior distribution

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})\,p(\mathbf{m})}{p(\mathbf{d})} \propto p(\mathbf{d}|\mathbf{m})\,p(\mathbf{m})$$

$\mathbf{m}$	True velocity model
$p(\mathbf{d})$	Probability of seismic data $\mathbf{d}$ being observed
$p(\mathbf{m})$	Probability of seismic velocity distribution $\mathbf{m}$
$p(\mathbf{d}\|\mathbf{m})$	Conditional probability of observing $\mathbf{d}$ given seismic model $\mathbf{m}$
$p(\mathbf{m}\|\mathbf{d})$	Conditional probability of model $\mathbf{m}$ when $\mathbf{d}$ is observed

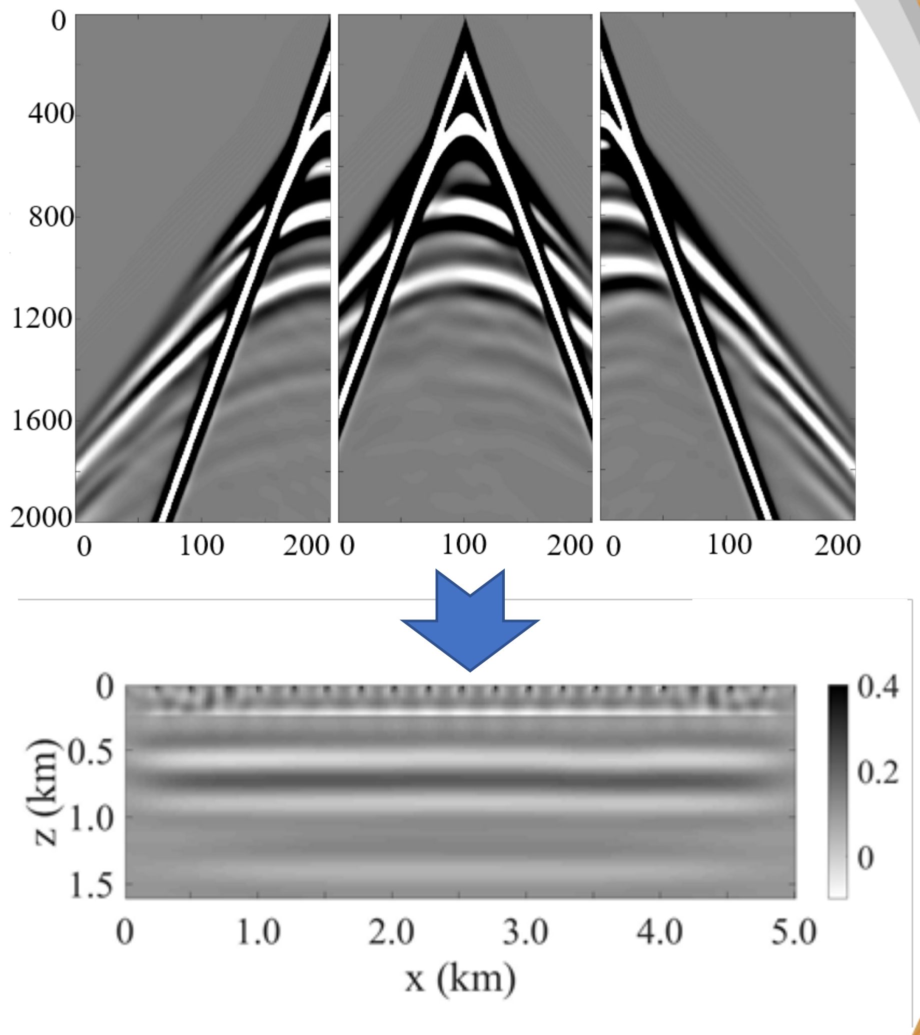
Normalizing flows

- A special kind of invertible neural network
- By connecting several invertible network units in series, it could easily sample the **posterior distribution** via memory-efficient training
- A good choice to address the seismic inversion problem and evaluate the reliability of the inverted velocity models.



The proposed method

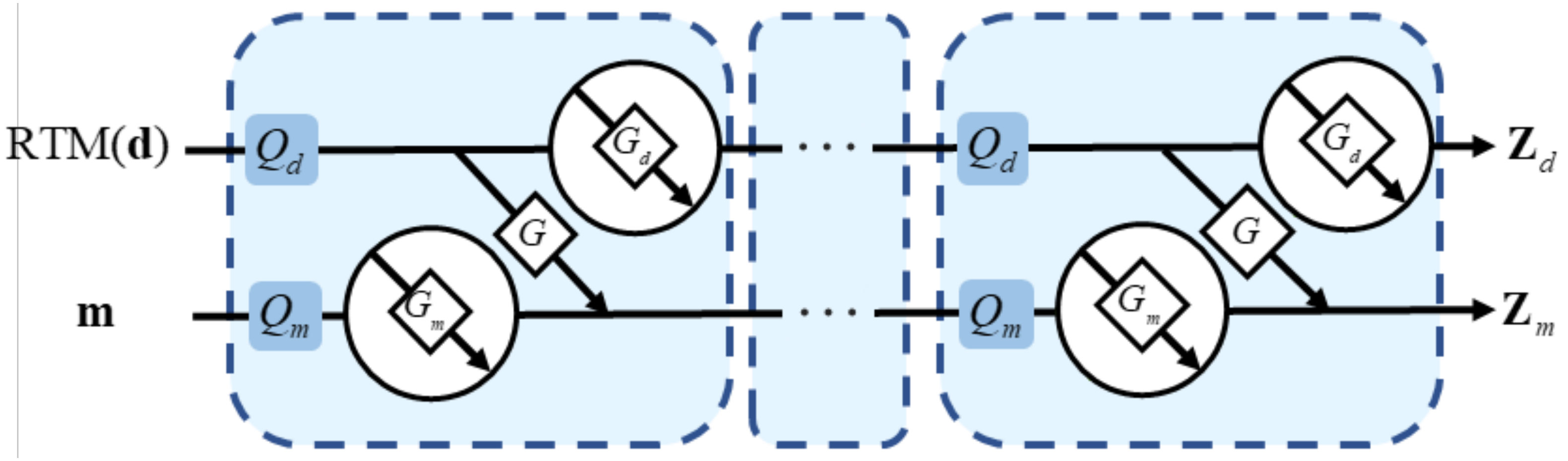
- HINT structure[1] is adopted for building a conditional normalizing flow (CNF) named **i-VelInvNet**
- Instead of seismic data, **RTM images** are used as network input
- **Two advantages:**
  - ✓ Same dimension as the velocity model, reduce network input dimension.
  - ✓ effectively reflect the subsurface geological features, easier for CNF to establish a mapping to the true velocity model



[1] Kruse J, Detommaso G, Scheichl R, et al. HINT: Hierarchical Invertible Neural Transport for Density Estimation and Bayesian Inference[J]. arXiv preprint arXiv:1905.10687, 2020.

The proposed method

- **i-VelInvNet architecture**



$\mathbf{m}$ & RTM($\mathbf{d}$): network input

$\mathbf{z}_\mathrm{d}$ & $\mathbf{z}_\mathrm{m}$: network output

Q_d & Q_m: orthogonal matrices

G、G_m & G_d: spatial down-sampling network[2]

Forward pass

$$\begin{cases} \left[\mathbf{m}^1,\mathbf{m}^2\right]^\mathrm{T} = Q_\mathrm{m}\mathbf{m},\ \left[\mathbf{d}^1,\mathbf{d}^2\right]^\mathrm{T} = Q_\mathrm{d}\mathrm{RTM}(\mathbf{d}) \\ \mathbf{z}_\mathrm{m} = \mathrm{G}_\mathrm{m}\left(\left[\mathbf{m}^1,\mathbf{m}^2\right]^\mathrm{T}\right)\odot\mathbf{s}+\mathbf{t},\ [\mathbf{s},\mathbf{t}] = G\left(\left[\mathbf{d}^1,\mathbf{d}^2\right]^\mathrm{T}\right) \\ \mathbf{z}_\mathrm{d} = \mathrm{G}_\mathrm{d}\left(\left[\mathbf{d}^1,\mathbf{d}^2\right]^\mathrm{T}\right) \end{cases}$$

Backward pass

$$\begin{cases} \left[\mathbf{d}^1,\mathbf{d}^2\right]^\mathrm{T} = \mathrm{G}_\mathrm{d}^{-1}(\mathbf{z}_d) \\ \left[\mathbf{m}^1,\mathbf{m}^2\right]^\mathrm{T} = \mathrm{G}_\mathrm{m}^{-1}\left((\mathbf{z}_\mathrm{m}-\mathbf{t})./\mathbf{s}\right),\ [\mathbf{s},\mathbf{t}] = G\left(\left[\mathbf{d}^1,\mathbf{d}^2\right]^\mathrm{T}\right) \\ \mathbf{m} = Q_\mathrm{m}^{-1}\left[\mathbf{m}^1,\mathbf{m}^2\right]^\mathrm{T},\ \mathrm{RTM}(\mathbf{d}) = Q_\mathrm{d}^{-1}\left[\mathbf{d}^1,\mathbf{d}^2\right]^\mathrm{T} \end{cases}$$

[2] PUTZKY P, WELLING M. Invert to learn to invert[J]. arXiv preprint arXiv:1911.10914, 2019.

The proposed method

- **i-VelInvNet loss function**

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2}\left\|T\left(\mathrm{RTM},\mathbf{m}\right)\right\|_2^2 - \log\left|\det \nabla T\left(\mathrm{RTM},\mathbf{m}\right)\right|\right]$$

θ : Network parameters, T: Forward pass

- **Inference process**

  ✓ for a given unseen observation data, compute $\mathbf{z}_{\mathrm{d}}$

  ✓ Randomly sampling $\mathbf{z}_{\mathrm{m}} \sim N\left(0,\mathbf{I}_{\mathrm{m}}\right)$ and compute the posterior sample

$$p_T\left(\mathbf{m}\,|\,\mathbf{d}\right) = \tilde{T}^{-1}\begin{bmatrix}\mathbf{z}_{\mathrm{d}} \\ \mathbf{z}_{\mathrm{m}}\end{bmatrix} = \tilde{T}^{-1}\begin{bmatrix}\tilde{T}_{\mathrm{d}}\left(\mathrm{RTM}\left(\mathbf{d}\right)\right) \\ N\left(0,\mathbf{I}_{\mathrm{m}}\right)\end{bmatrix}$$

- **Uncertainty analysis of the inverted results**

Workflow of the proposed method

Step 1: Initialization

Step 2: Compute RTM

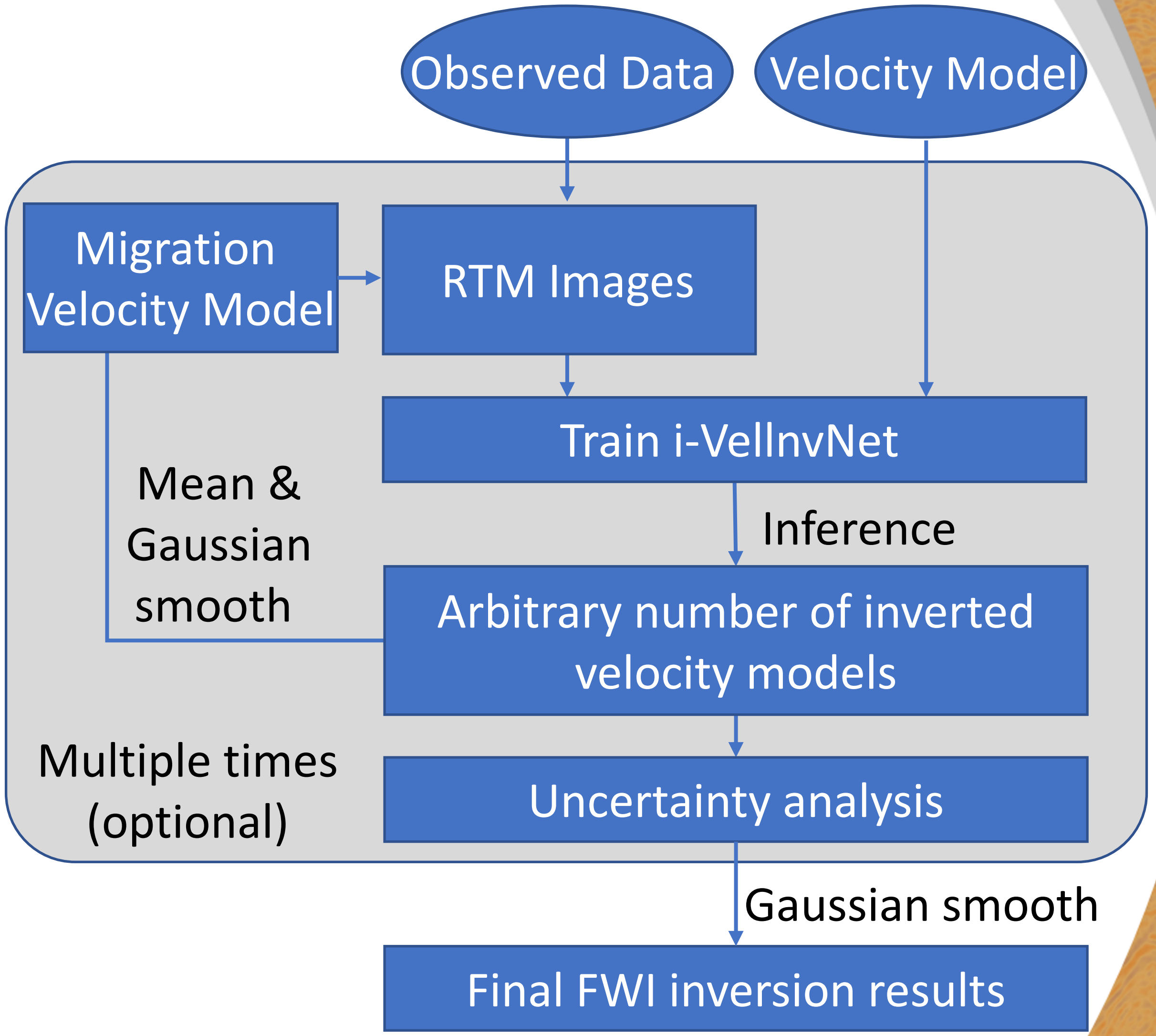
Step 3: Train i-VelInvNet

Step 4: Posterior sampling based on the trained i-VelInvNet

Step 5: Uncertainty analysis of the inverted results

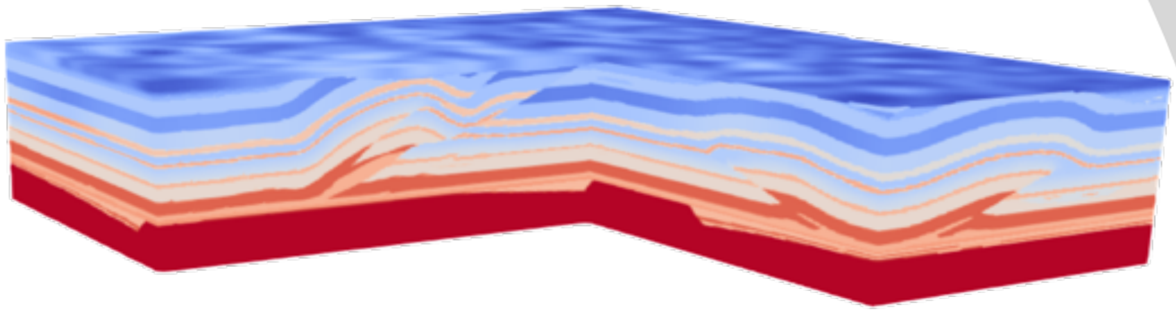
Step 6: Repeat step 2 ~ 5 for better posterior samples (optional)

Step 7: Gaussian smooth for a good initial model and perform FWI for the final inversion results



EXPT.1: Check invertibility of i-VelInvNet

- **Training dataset setup**
  - ✓ **2000 models:** 2D sections of the 3D overthrust model
  - ✓ **Model size:** 64 × 200
  - ✓ **Grid spacing:** 25 m
  - ✓ **Observation setup:** 21 sources, 201 receivers, 6Hz Ricker wavelet



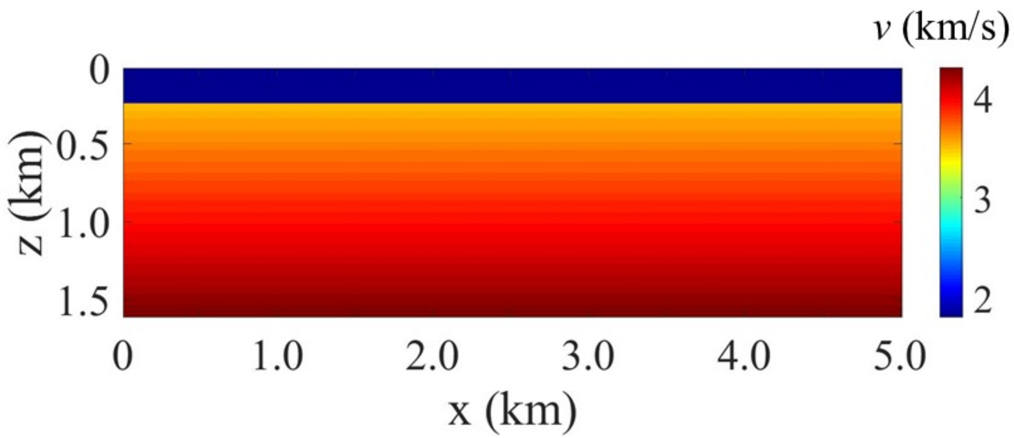
the SEG/EAGE overthrust model

- **How to check Invertibility ?**

Forward $[\mathbf{z}_{\mathrm{d}}, \mathbf{z}_{\mathrm{m}}]^{\mathrm{T}} = T(\mathrm{RTM}, \mathbf{m}),$

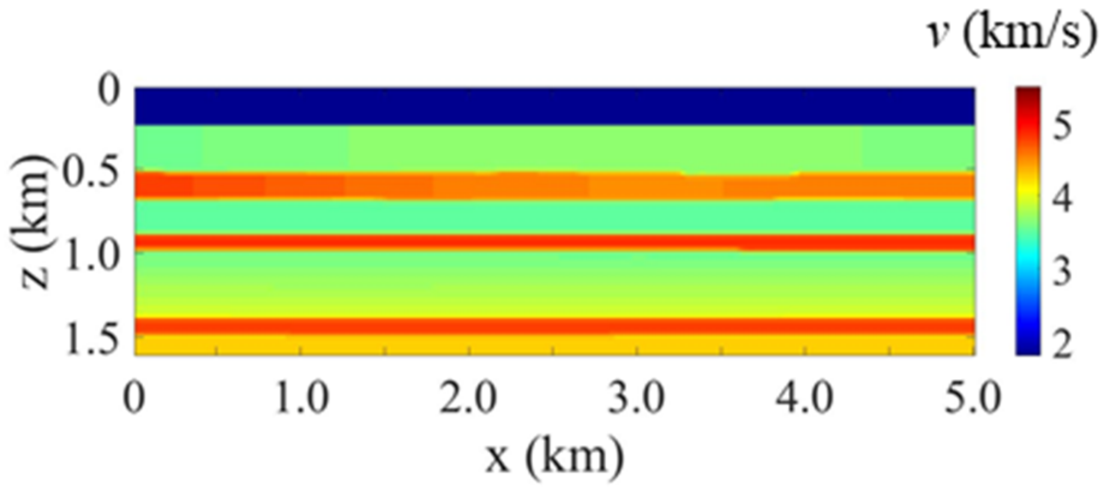
Backward $[\mathrm{RTM}', \mathbf{m}']^{\mathrm{T}} = T^{-1}(\mathbf{z}_{\mathrm{d}}, \mathbf{z}_{\mathrm{m}})$

Check $\mathbf{m} = \mathbf{m}'$ $\quad \mathrm{RTM} = \mathrm{RTM}'$

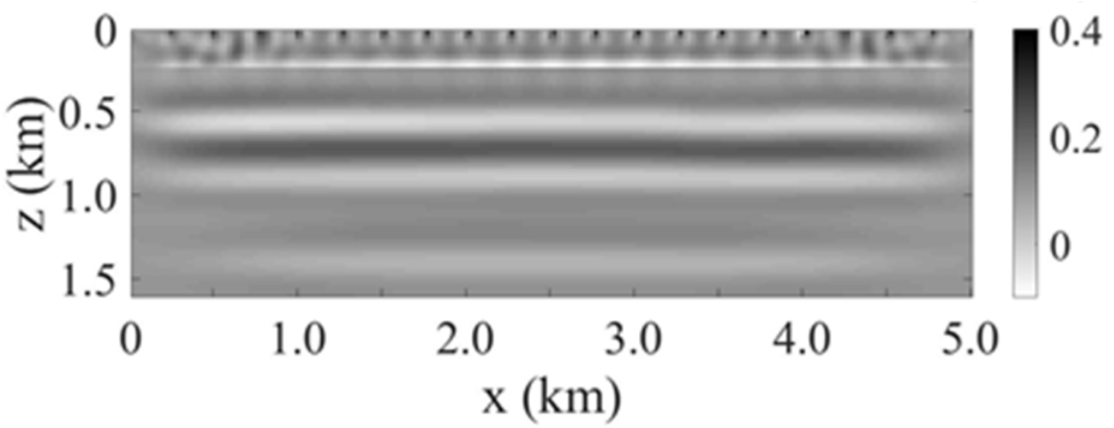


Migration velocity model

EXPT.1: Check invertibility of i-VelInvNet

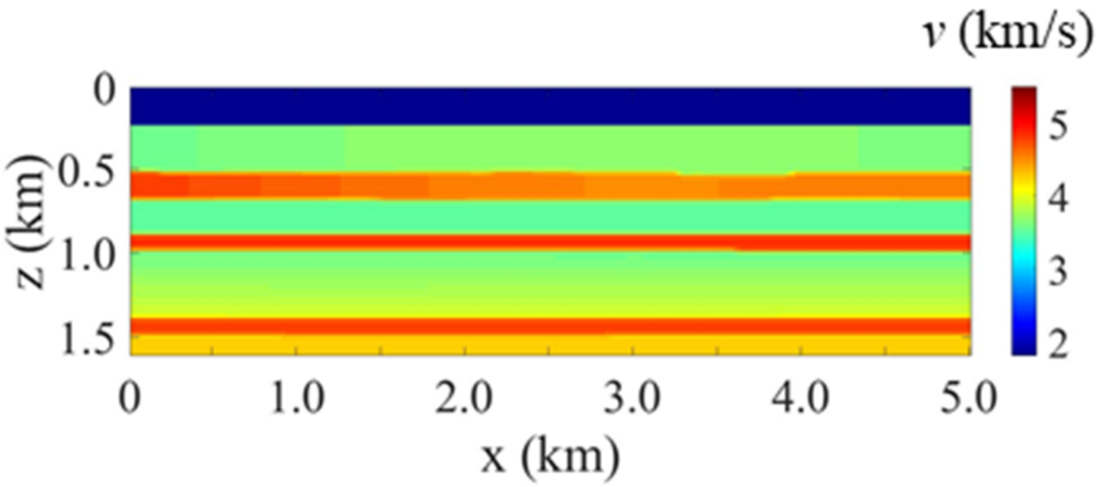


True velocity model

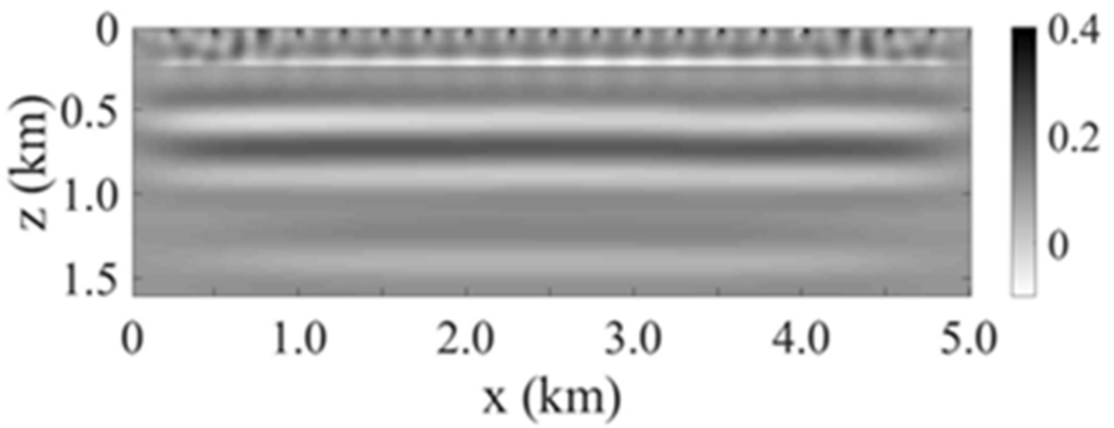


RTM image

EXPT.1: Check invertibility of i-VelInvNet

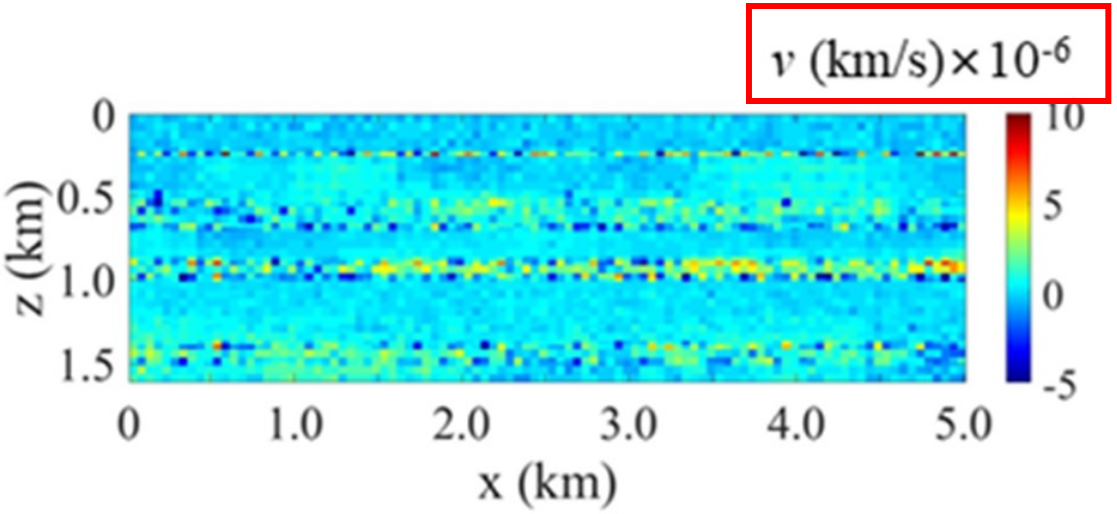


Predicted velocity model

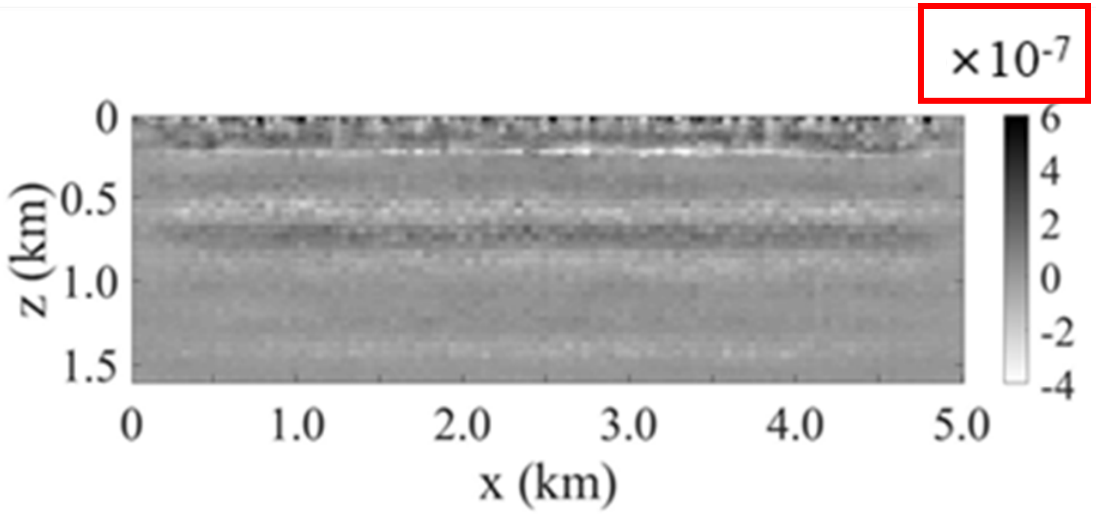


Predicted RTM image

EXPT.1: Check invertibility of i-VelInvNet



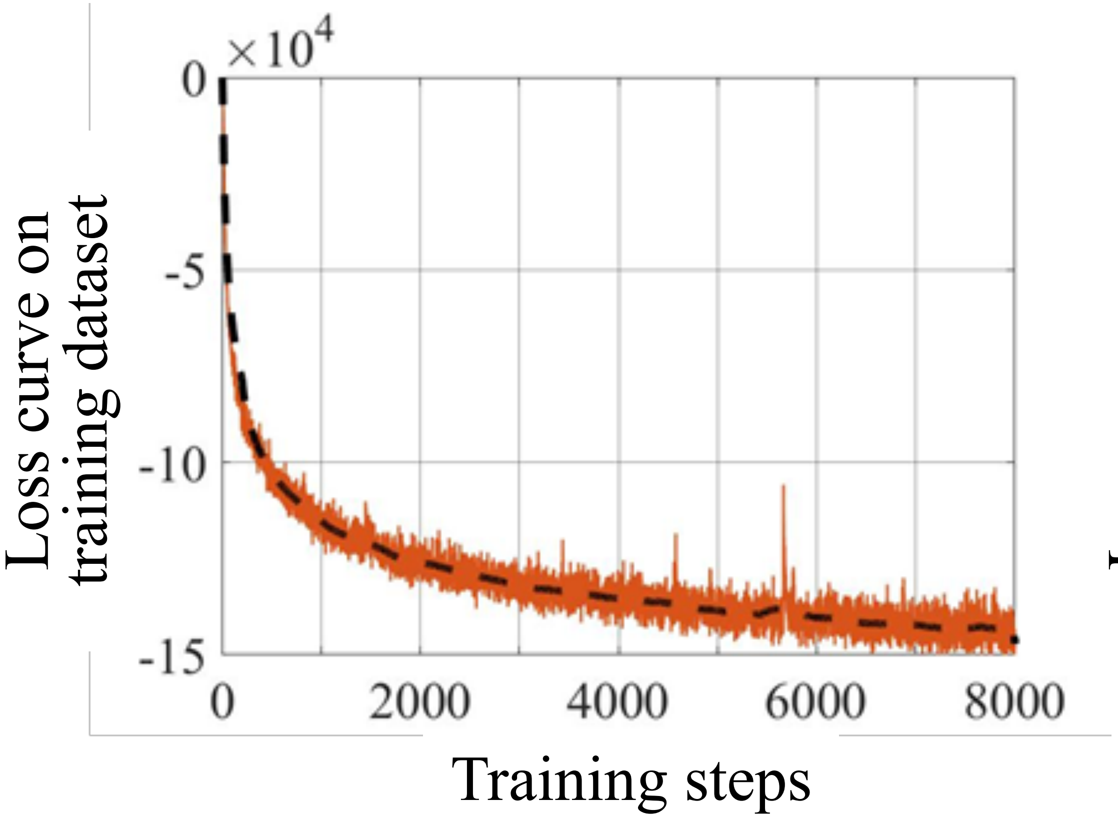
Difference of the velocity model

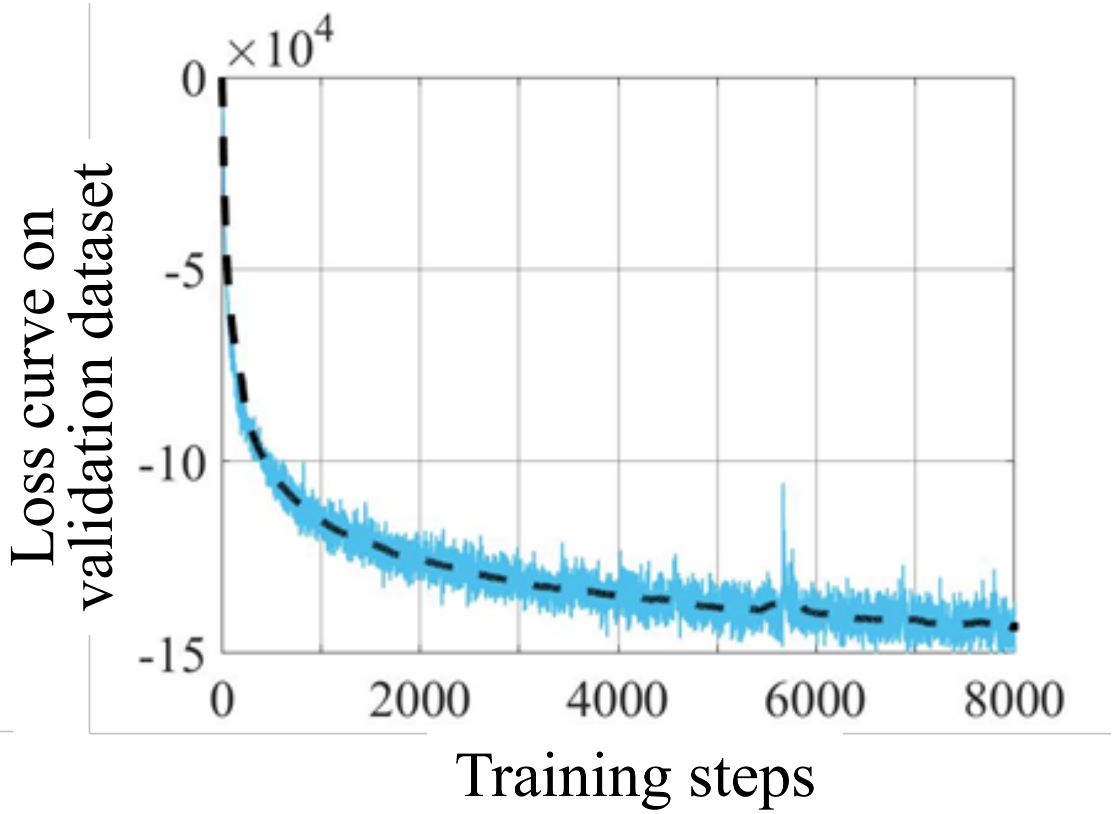


Difference of the RTM images

EXPT.2: Inversion results of i-VelInvNet

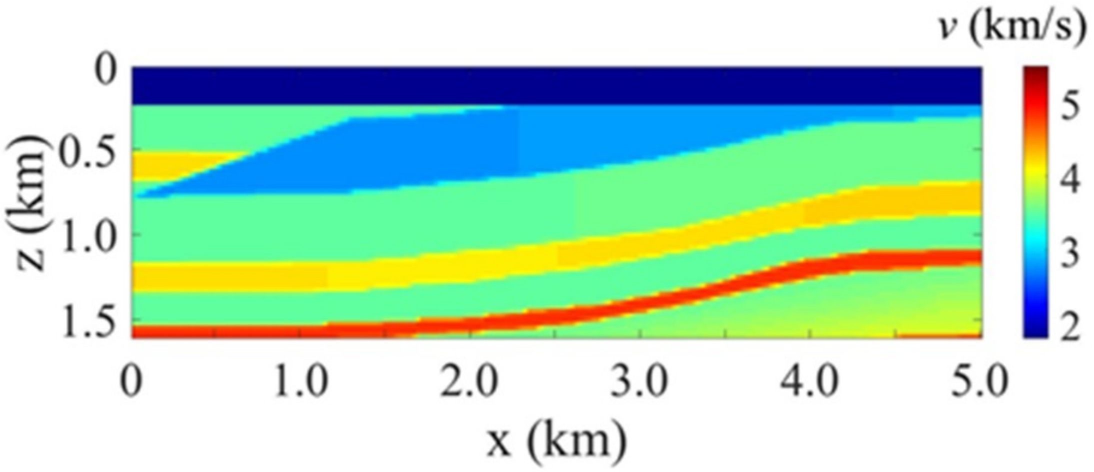
- Train : valid : test = 1200 : 400 : 400
- Training steps = 8000



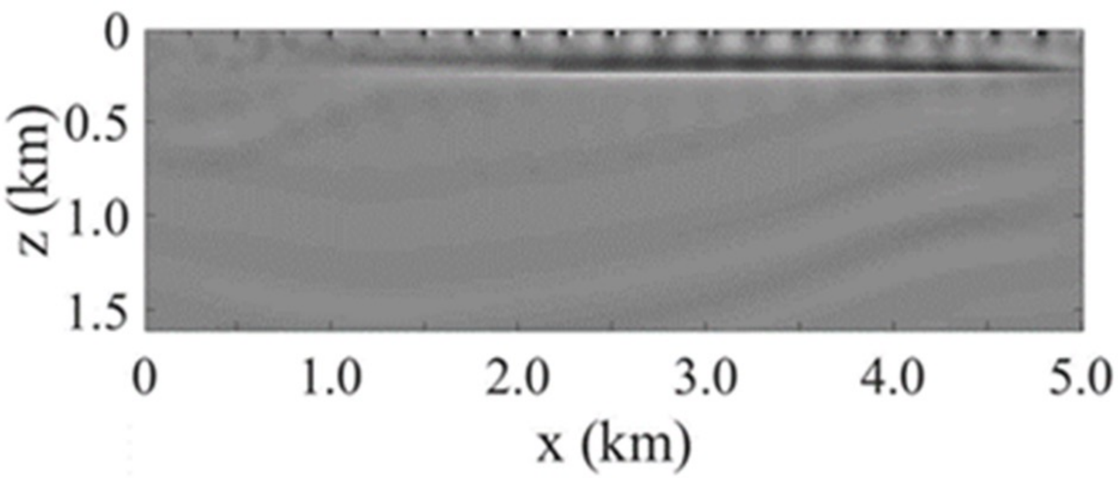


EXPT.2: Inversion results of i-VelInvNet

- Randomly select a velocity from the test dataset
- Compute the observation data and RTM image as i-VelInvNet input
- Fix z_d and feed 100 random z_m for 100 posterior samples of the inverted velocity model via the backward pass of the trained i-VelInvNet.



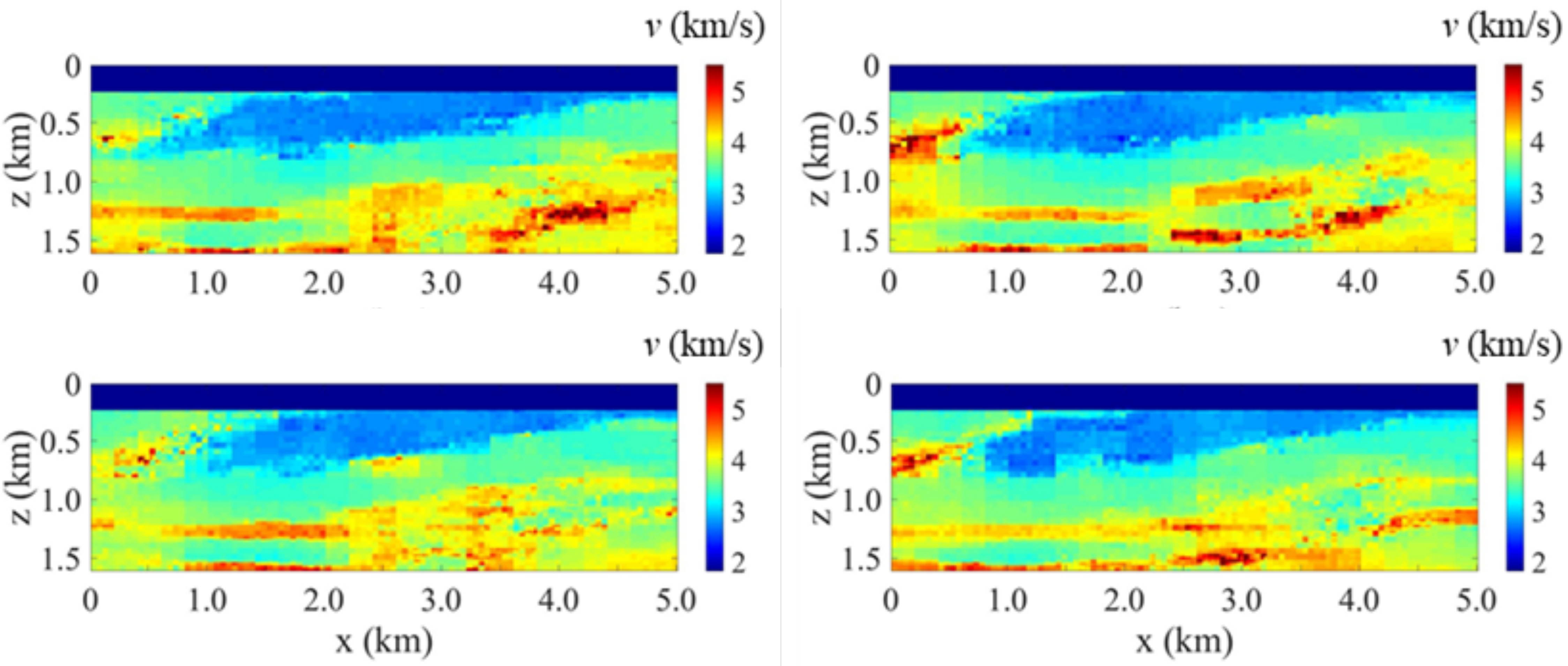
True velocity model



RTM image

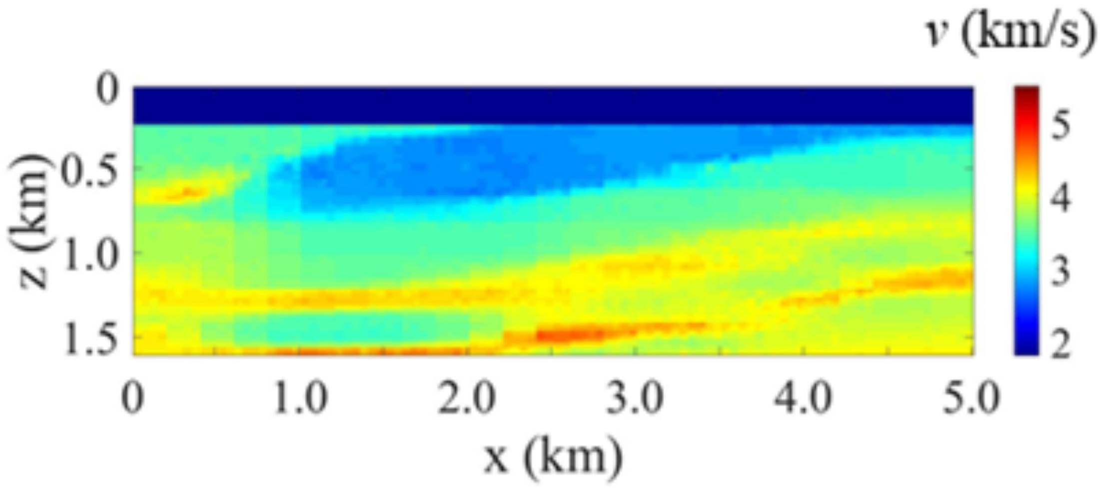
EXPT.2: Inversion results of i-VelInvNet

- 4 of the 100 **posterior samples** of the i-VelInvNet inversion results

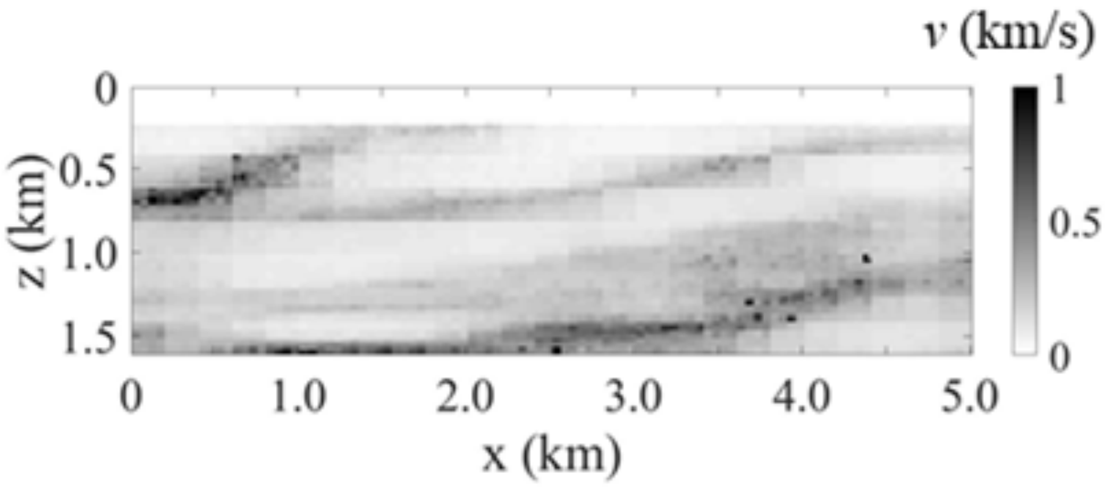


EXPT.2: Inversion results of i-VelInvNet

- **Uncertainty analysis of the posterior samples**



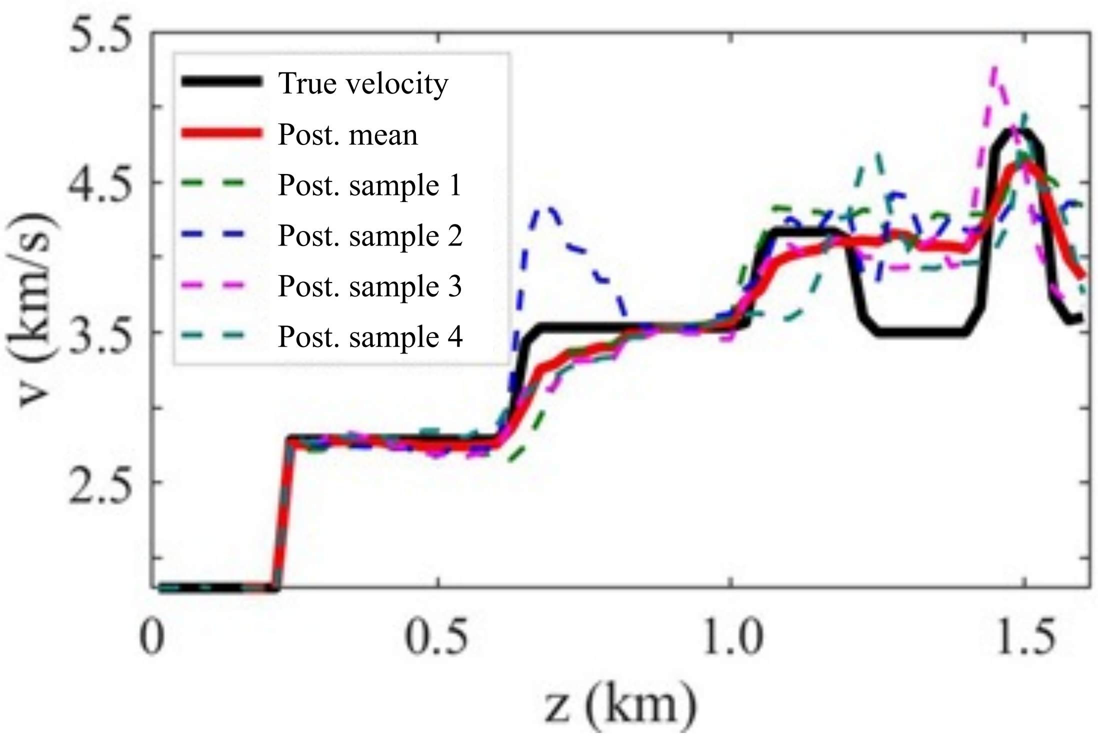
Posterior mean



Standard deviation

EXPT.2: Inversion results of i-VelInvNet

- **Quantitative comparison**



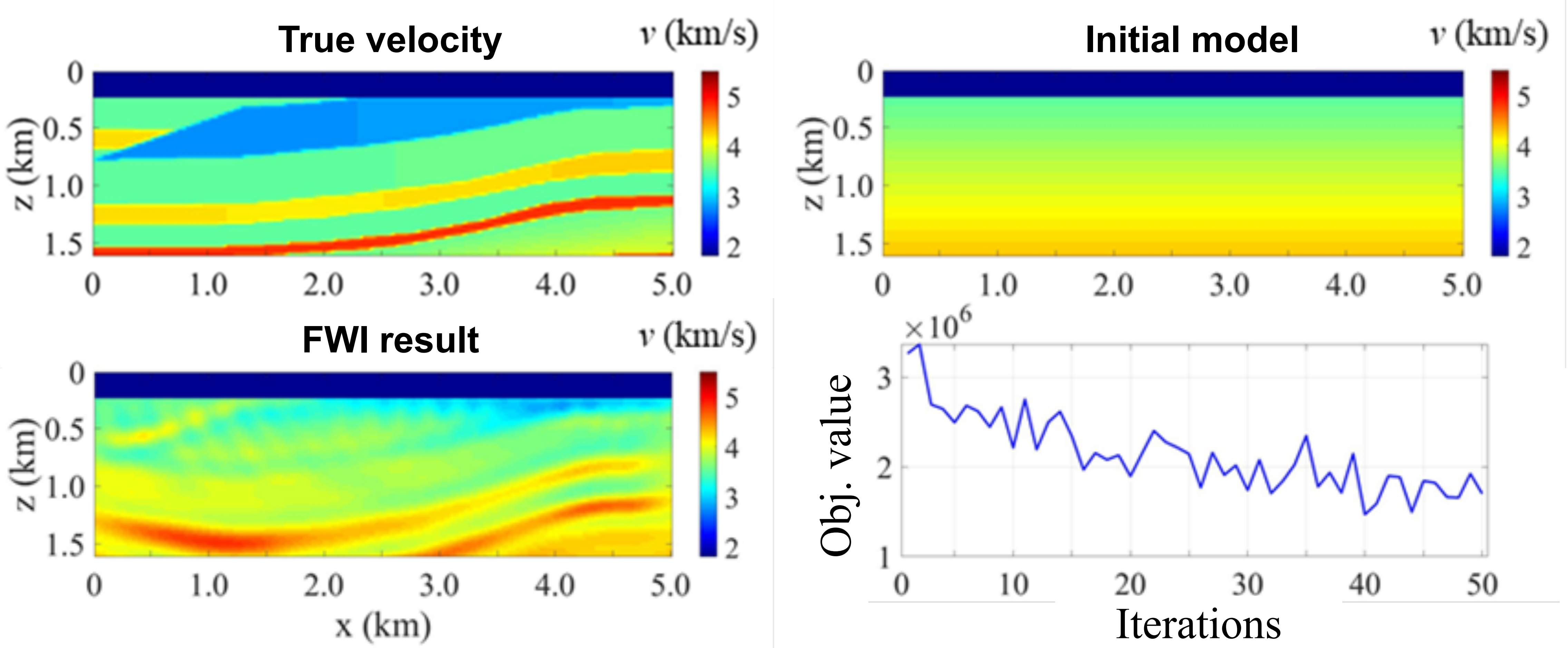
Velocity value at x=2.5km

Metrics	Posterior samples	Posterior mean
MAE↓	0.0238	**0.0195**
MSE↓	0.0012	**0.0008**
PSNR↑	29.8356	**31.8120**
SSIM↑	0.8215	**0.8779**
MSSIM↑	0.8803	**0.9188**

SLIM

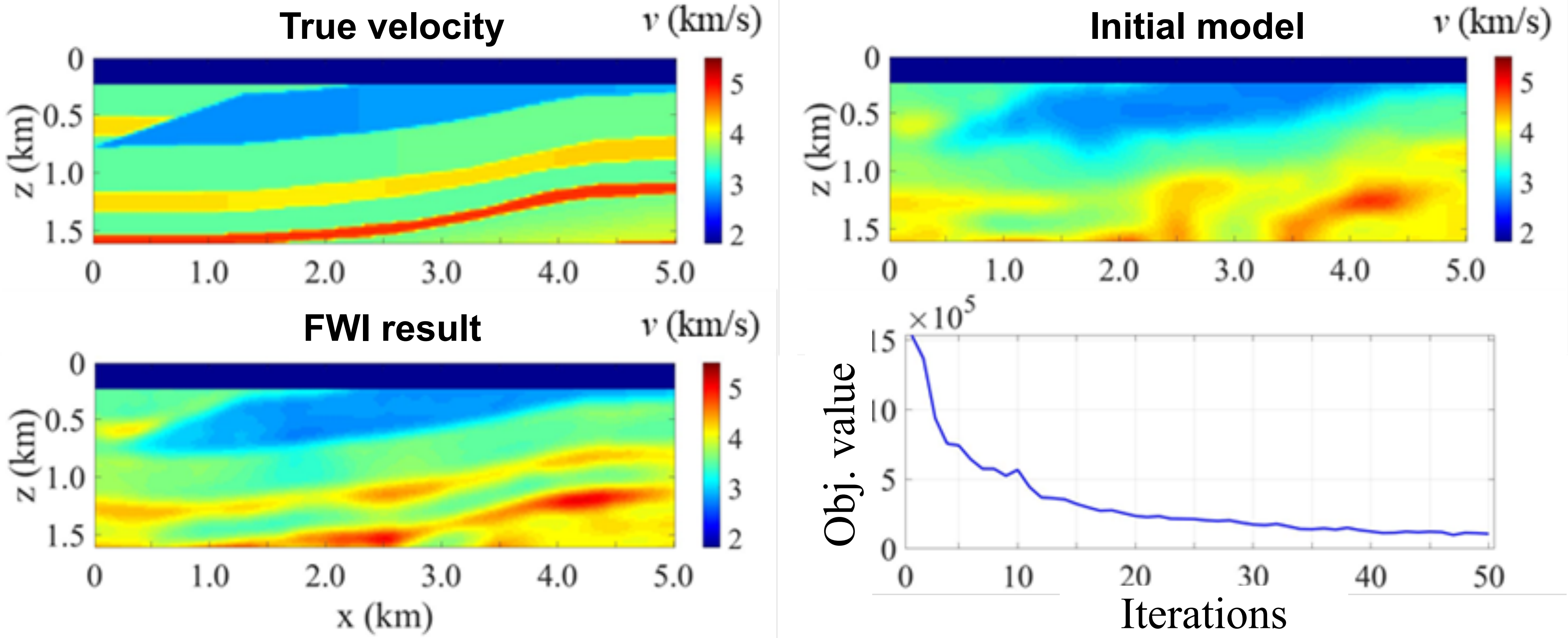
EXPT.3: Full waveform Inversion

- FWI using the **migration velocity** as initial model



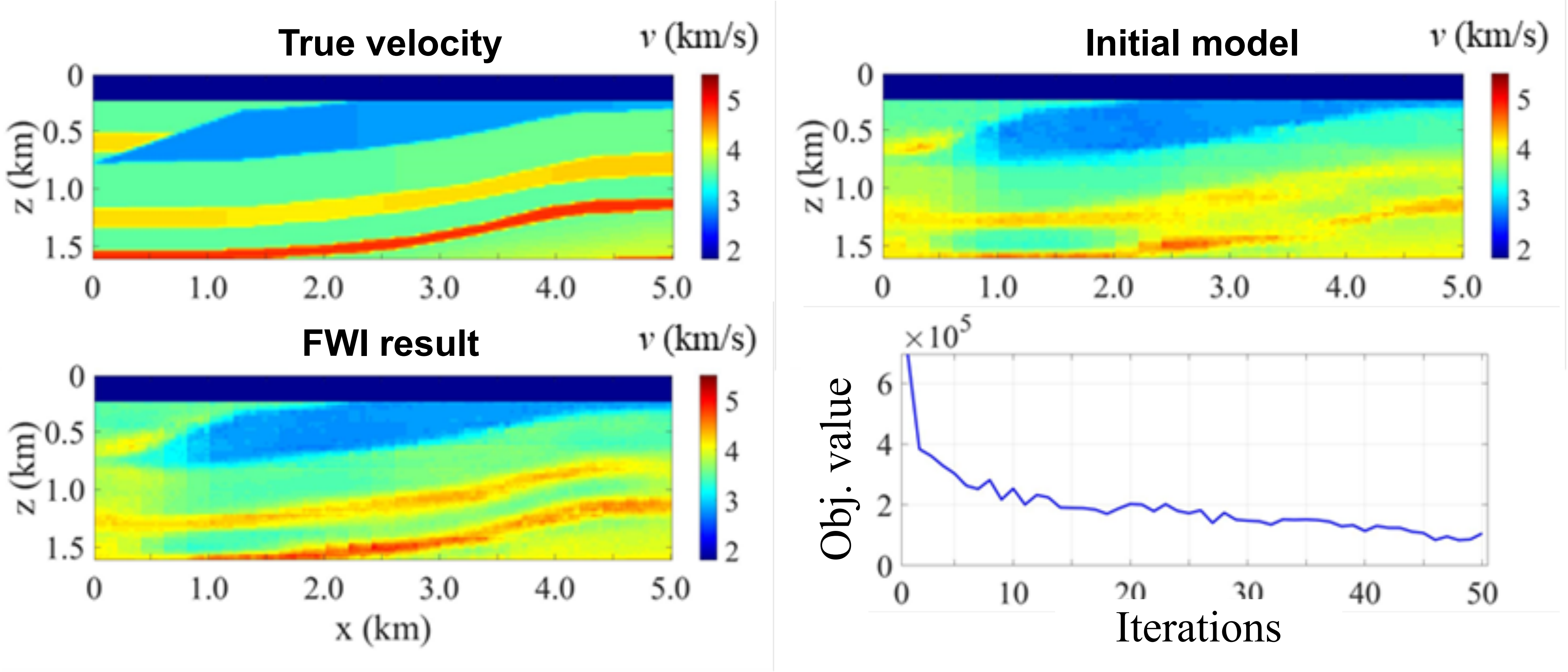
EXPT.3: Full waveform Inversion

- FWI using one of the **i-VelInvNet posterior samples** as initial model



EXPT.3: Full waveform Inversion

- FWI using the **i-VelInvNet posterior mean** as initial model



Conclusions

- CNF could provide a direct access to the conditional mean estimate and uncertainty assessment of the inverted results
- When applied in seismic velocity inversion problem, arbitrary number of inverted velocity models and its uncertainty quantification can be obtained via a trained i-VelInvNet.
- The inverted results, either the posterior samples or posterior mean, can be used as a good initial model in the subsequent FWI for a more accurate result.
- Improvement of the proposed i-VelInvNet can be made in terms of computation efficiency and inversion accuracy.

Acknowledgment

- Open-sourced Codes:
  https://github.com/slimgroup/JUDI.jl
  https://github.com/slimgroup/InvertibleNetworks.jl

- My greatest gratitude to Prof. Shucai Li and Prof. Bin Liu for the opportunities of visiting SLIM group in Gatech last year!
- My genuine thanks to Prof. Felix Herrmann and all the SLIM group members for a memorable time there!
- Please contact Dr. Yuxiao Ren (ryxchina@gmail.com) for any questions and suggestions!