EAGE
EUROPEAN ASSOCIATION OF GEOSCIENTISTS & ENGINEERS

INCLUDING SPE International EUROPEC

OPPORTUNITIES PRESENTED BY THE ENERGY TRANSITION

2018 EAGE ANNUAL

80TH CONFERENCE + EXHIBITION

COPENHAGEN | DENMARK

bp

REPSOL

11-14 JUNE 2018
WWW.EAGEANNUAL2018.ORG

EAGE ANNUAL 2018
80TH CONFERENCE + EXHIBITION
COPENHAGEN | DENMARK

Seismic data interpolation with Generative Adversarial Networks

Felix J. Herrmann

Seismic data interpolation with Generative Adversarial Networks

Ali Siahkoohi, Rajiv Kumar, and Felix J. Herrmann



SLIM
Georgia Institute of Technology

Herrmann, F.J. and Hennenfent, G. [2008] Non-parametric seismic data recovery with curvelet frames. Geophysical Journal International, 173(1), 233–248.
Kumar, R., Aravkin, A.Y., Mansour, H., Recht, B. and Herrmann, F.J. [2013] Seismic data interpolation and denoising using svd-free low-rank matrix factorization. In: 75th EAGE Conference & Exhibition incorporating SPE EUROPEC 2013.
Da Silva, C. and Herrmann, F.J. [2014] Low-rank Promoting Transformations and Tensor InterpolationApplications to Seismic Data Denoising. In: 76th EAGE Conference and Exhibition 2014.
Yarman, C.E., Kumar, R. and Rickett, J. [2017] A model based data driven dictionary learning for seismic data representation. Geophysical Prospecting.

SLIM

Interpolation

Interpolation schemes rely on prior information on the data to fill in missing traces

Previous approaches:

- use sparse transform domain
- rank minimization or tensor completion
- dictionary learning

Linear vs. Non-linear

Previous methods rely on some perhaps too simplifying assumptions on data

- linear mathematical models, i.e.,
- via superposition of prototype waveforms from a fixed or learned dictionary or in terms of a matrix factorizations

Nonlinear models outperform linear models in approximating a nonlinear real-world physical phenomenon

- they can capture the nonlinearities of observed data

Finding a probability distribution for data

Frequency slices, samples from a high-dimensional probability distribution

If data samples are IID, find a probability distribution that explains the data:

Model: A pre-defined probability function

Number of frequency slices

$$\max_{\theta} p(X;\theta) = \max_{\theta} \prod_{i=1}^{N} p(x^{(i)};\theta) = \max_{\theta} \frac{1}{N}\sum_{i=1}^{N} \log p(x^{(i)};\theta)$$

Matrix containing all the frequency slices

Model parameters

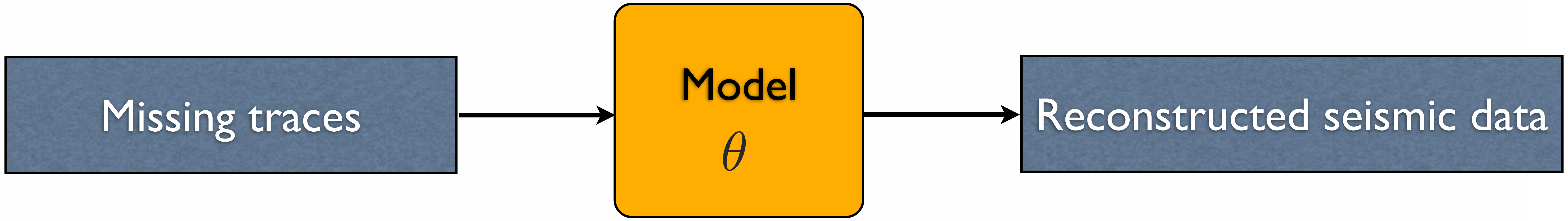
i^{th} frequency slice

Csáji, B.C., 2001. Approximation with artificial neural networks. Faculty of Sciences, Etvs Lornd University, Hungary, 24, p.48.

Neural networks as a model

Universal Approximation Theory: Neural networks can approximate any continuous function on compact subsets of $\mathbb{R}^n$ under some mild assumptions on their activation functions.

Goal: Learn a transformation mapping using neural networks as model via a probabilistic approach.



Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).
Goodfellow, I., 2016. NIPS 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160.

SLIM

Generative Adversarial Networks

Data driven nonlinear model, combined with insights from game theory

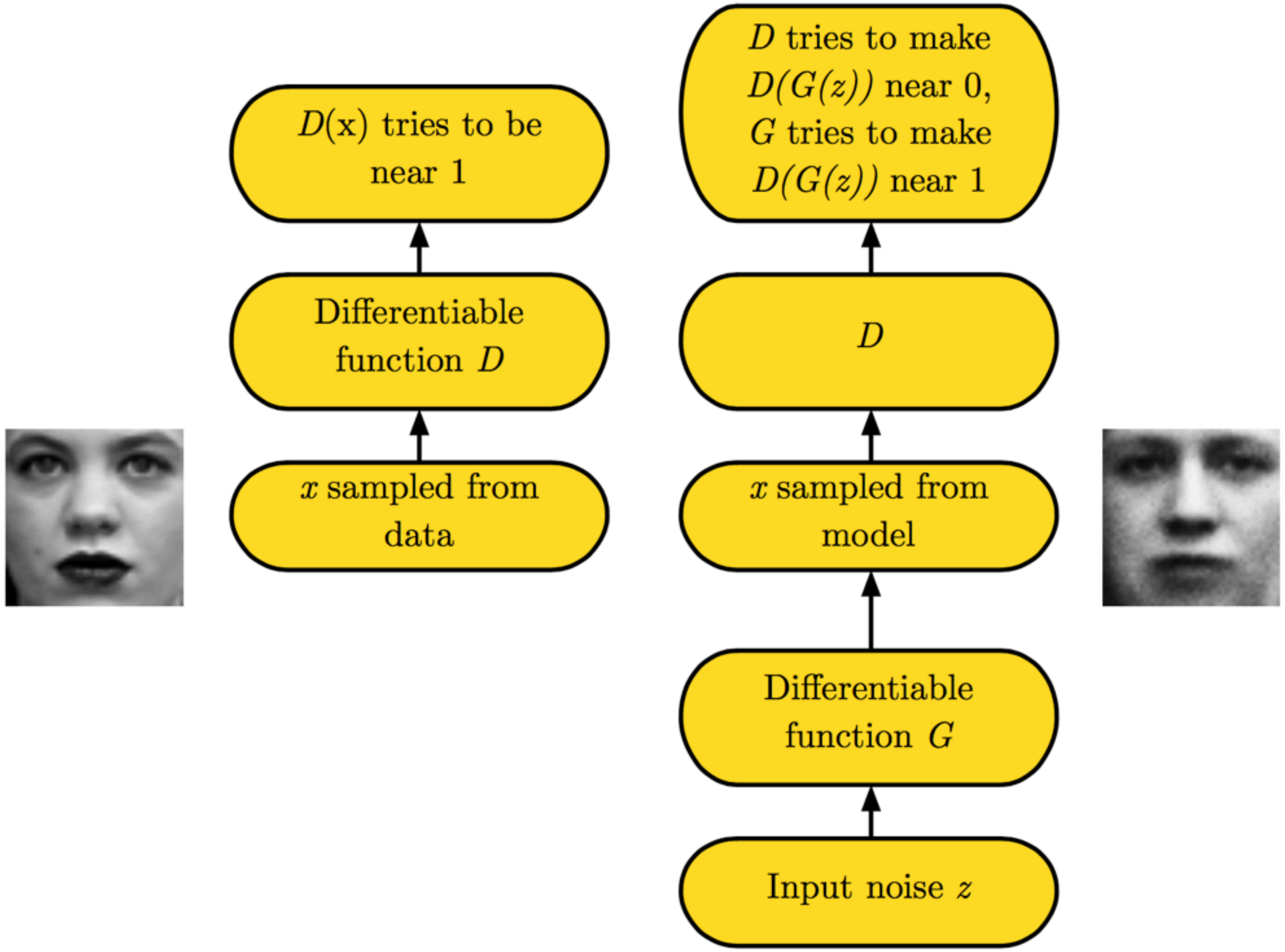
- implicitly models the complex probability distribution of the data
- How? By playing a game between two deep neural networks, **Generator** and **Discriminator**

GANs for data reconstruction:

- Outperform current methods for large percentages of traces missing, independent of type of sampling

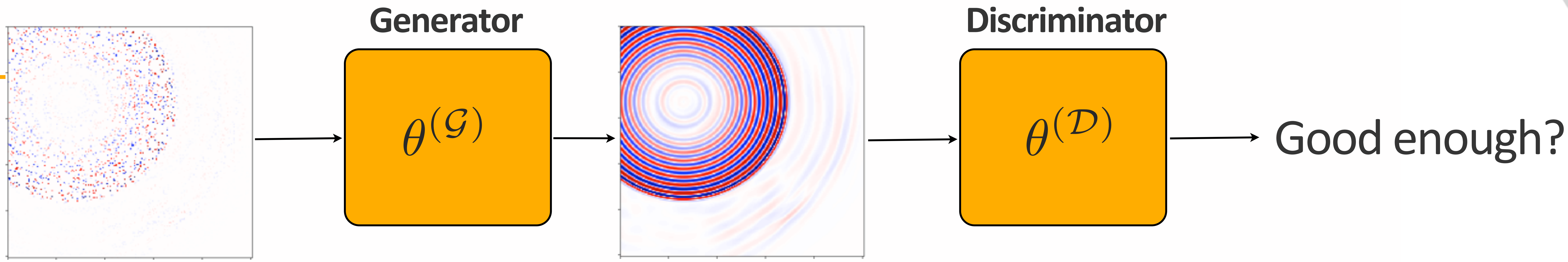
Goodfellow, I., 2016. NIPS 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160.

Vanilla GAN



GANs framework

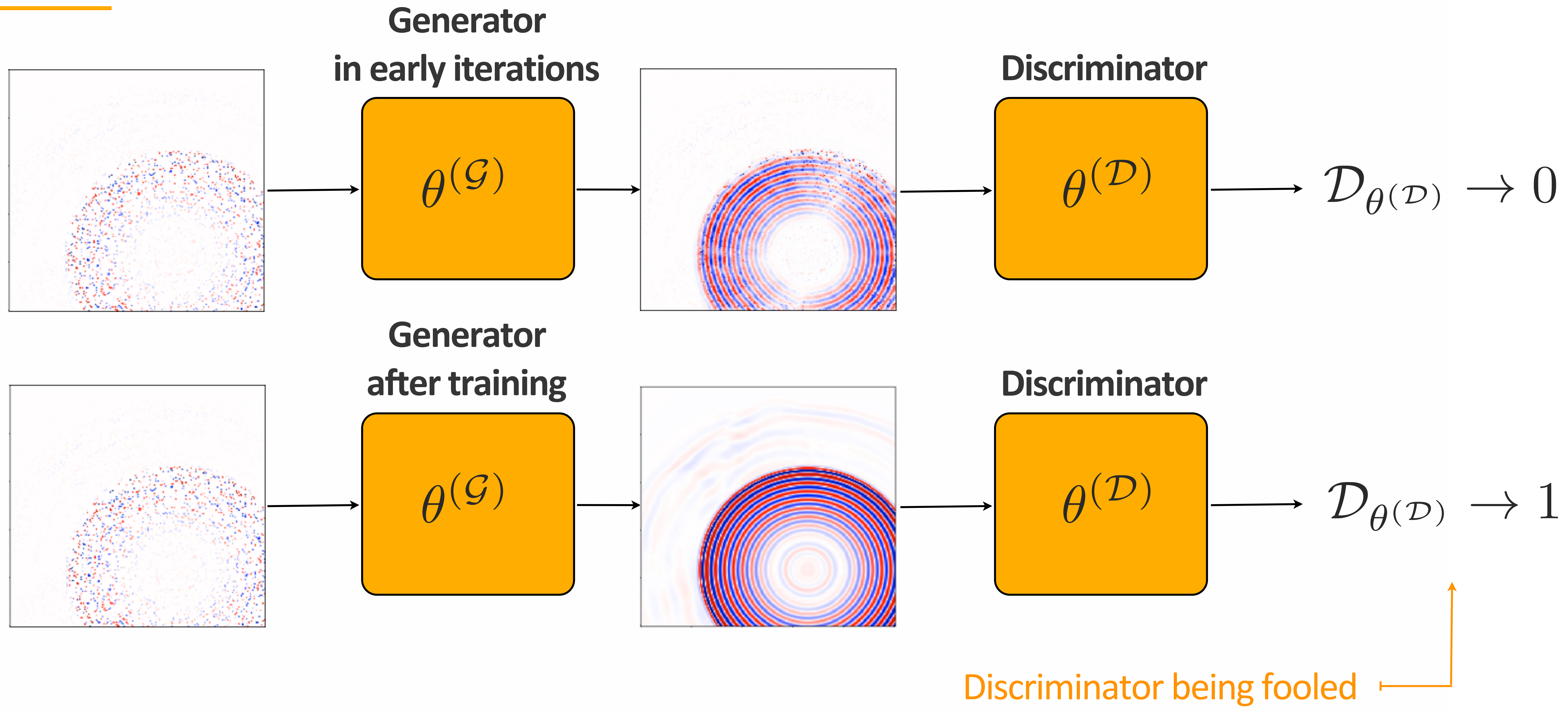
Generative Adversarial Networks



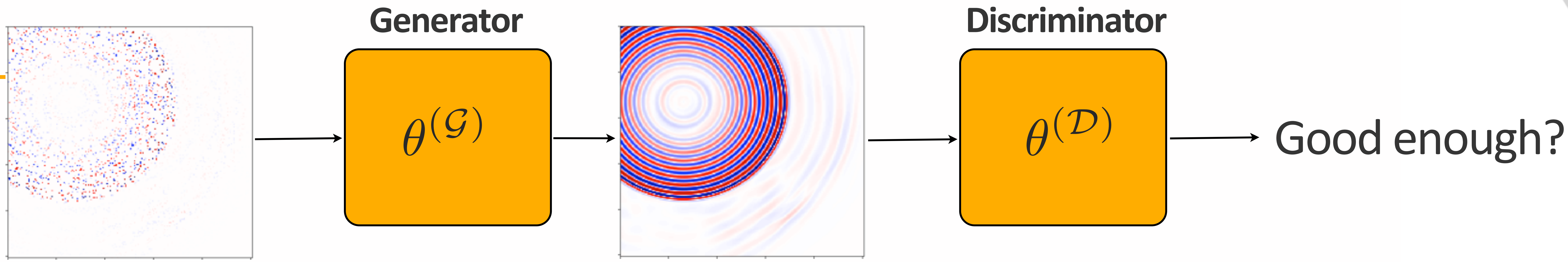
Discriminator neural network, $\mathcal{D}_{\theta^{(\mathcal{D})}}$, estimates the probability of it's input being:

- a frequency slice from the fully sampled data distribution ($\mathcal{D}_{\theta^{(\mathcal{D})}} \rightarrow 1$)
- or a frequency slice from output of the generator ($\mathcal{D}_{\theta^{(\mathcal{D})}} \rightarrow 0$)

Discriminator in pictures



Generative Adversarial Networks



Generator neural network, $\mathcal{G}_{\theta^{(\mathcal{G})}}$, takes partial measurements, $\mathbf{x}$, and outputs reconstructed frequency slices such that:

- reconstructions are hard for $\mathcal{D}_{\theta^{(\mathcal{D})}}$ to distinguish from fully sampled data,
- i.e., $\mathcal{D}_{\theta^{(\mathcal{D})}}\left(\mathcal{G}_{\theta^{(\mathcal{G})}}\left(\mathbf{x}\right)\right) \rightarrow 1$, although the input to $\mathcal{D}_{\theta^{(\mathcal{D})}}$ is **not** from the fully sampled data probability distribution.

Distribution of the fully sampled data

To train the discriminator, we need samples from the probability distribution of fully sampled data & measurements

We do not have access directly to the probability distribution of fully sampled seismic data, p_{data}.

We have a set of fully sampled frequency slices, $S_{\text{data}} = \{\mathbf{y}_i\}_{i=1}^{N}$, where N is number of samples.

Distribution of the measurements

We don’t have access to the probability distribution of the partial measurements, $p_{\text{measurements}}$.

Construct the set of incomplete frequency slices:

- $S_{\text{measurement}} = \{\mathbf{x}_i |\ \mathbf{x}_i = M \odot \mathbf{y}_i, \forall \mathbf{y}_i \in S_{\text{data}}\}$
- where M is the sampling mask

We will use set S_{data} and $S_{\text{measurement}}$ in order to train the discriminator network.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

SLIM

Theory

After training, generator implicitly defines a probability distribution, $p_{\text{reconstruction}}$

Given enough model (i.e., neural network) capacity and training time:

- for a fixed generator, and freq. slice, $\mathbf{x}$, optimal discriminator is:

$$\mathcal{D}^*(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\text{reconstruction}}(\mathbf{x})}$$

- ideally, if $p_{\text{reconstruction}} \approx p_{\text{data}}$, therefore $\mathcal{D}^*(\mathbf{x}) = \frac{1}{2}$.

Summarizing notation

p_{data}	: Probability distribution of fully sampled seismic data
$p_{\text{measurements}}$	: Probability distribution of the partial measurements
$\mathcal{D}_{\theta^{(\mathcal{D})}}$	: Discriminator neural network parameterized by $\theta^{(\mathcal{D})}$
$\mathcal{G}_{\theta^{(\mathcal{G})}}$	: Generator neural network parameterized by $\theta^{(\mathcal{G})}$
S_{data}	: Set of frequency slices from fully sampled seismic data
$S_{\text{measurement}}$	: Set of frequency slices from fully measurements
$S_{\text{data}} \cup S_{\text{measurement}}$	: Training data set
$\theta^{(\mathcal{D})}, \theta^{(\mathcal{G})}$	: Parameters to optimize
M	: Sampling mask

Zhu, J.Y., Park, T., Isola, P. and Efros, A.A. [2017] Unpaired image-to-image translation using cycleconsistent adversarial networks. arXiv preprint arXiv: 1703.10593
Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z. and Smolley, S.P. [2016] Least squares generative adversarial networks. arXiv preprint ArXiv:1611.04076.

SLIM

GAN in equations - Discriminator's loss

Discriminator's loss function:

$$\mathcal{L}_{\mathcal{D}} = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\mathrm{measurement}}(\mathbf{x})} \left[\left(\mathcal{D}_{\theta^{(\mathcal{D})}} \left(\mathcal{G}_{\theta^{(\mathcal{G})}}(\mathbf{x}) \right) \right)^2 \right] + \mathop{\mathbb{E}}_{\mathbf{y}\sim p_{\mathrm{data}}(\mathbf{y})} \left[\left(1 - \mathcal{D}_{\theta^{(\mathcal{D})}} \left(\mathbf{y} \right) \right)^2 \right],$$

Zhu, J.Y., Park, T., Isola, P. and Efros, A.A. [2017] Unpaired image-to-image translation using cycleconsistent adversarial networks. arXiv preprint arXiv: 1703.10593
Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z. and Smolley, S.P. [2016] Least squares generative adversarial networks. arXiv preprint ArXiv:1611.04076.

SLIM

GAN in equations - Discriminator's loss

Discriminator's loss function:

$$\mathcal{L}_{\mathcal{D}} = \underset{\mathbf{x}\sim p_{\text{measurement}}(\mathbf{x})}{\mathbb{E}} \left[\underbrace{\left(\mathcal{D}_{\theta^{(\mathcal{D})}} \left(\mathcal{G}_{\theta^{(\mathcal{G})}} (\mathbf{x}) \right) \right)^2}_{} \right] + \underset{\mathbf{y}\sim p_{\text{data}}(\mathbf{y})}{\mathbb{E}} \left[\left(1 - \underbrace{\mathcal{D}_{\theta^{(\mathcal{D})}} (\mathbf{y})}_{} \right)^2 \right],$$

Minimizing the loss will result in discriminating fully sampled data from reconstructions

Distribution of partial measurements

Distribution of fully sampled data

Discriminator's ability to recognize the reconstruction as being not fully sampled data by outputting 0

Discriminator's ability to recognize fully sampled data by outputting 1

Nowozin, S., Cseke, B. and Tomioka, R., 2016. f-gan: Training generative neural samplers using variational divergence minimization. In Advances in Neural Information Processing Systems (pp. 271-279).

SLIM

Dealing with $\underset{y \sim p_{\text{data}}(\mathbf{y})}{\mathbb{E}}[.]$ and $\underset{x \sim p_{\text{measurements}}(\mathbf{x})}{\mathbb{E}}[.]$

We approximate the expectation using batch size of m

- sample m data points, **without replacement** from S_{data} and $S_{\text{measurement}}$

$$\underset{\mathbf{x} \sim p_{\text{measurement}}(\mathbf{x})}{\mathbb{E}}\left[\left(\mathcal{D}_{\theta(\mathcal{D})}\left(\mathcal{G}_{\theta(\mathcal{G})}(\mathbf{x})\right)\right)^2\right] \simeq \frac{1}{m}\sum_{j=1}^{m}\left(\mathcal{D}_{\theta(\mathcal{D})}\left(\mathcal{G}_{\theta(\mathcal{G})}(\mathbf{x_j})\right)\right)^2$$

$$\underset{\mathbf{y} \sim p_{\text{data}}(\mathbf{y})}{\mathbb{E}}\left[\left(1 - \mathcal{D}_{\theta(\mathcal{D})}(\mathbf{y})\right)^2\right] \simeq \frac{1}{m}\sum_{j=1}^{m}\left(1 - \mathcal{D}_{\theta(\mathcal{D})}(\mathbf{y_j})\right)^2$$

Zhu, J.Y., Park, T., Isola, P. and Efros, A.A. [2017] Unpaired image-to-image translation using cycleconsistent adversarial networks. arXiv preprint arXiv: 1703.10593
Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z. and Smolley, S.P. [2016] Least squares generative adversarial networks. arXiv preprint ArXiv:1611.04076.

SLIM

GAN in equations - Generator's loss

Generator's loss function:

$$\mathcal{L}_{\mathcal{G}} = \mathbb{E}_{\mathbf{x} \sim p_{\mathrm{measurements}}(\mathbf{x})} \left[\left(1 - \mathcal{D}_{\theta^{(\mathcal{D})}} \left(\mathcal{G}_{\theta^{(\mathcal{G})}} (\mathbf{x}) \right) \right)^2 \right]$$

Zhu, J.Y., Park, T., Isola, P. and Efros, A.A. [2017] Unpaired image-to-image translation using cycleconsistent adversarial networks. arXiv preprint arXiv: 1703.10593
Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z. and Smolley, S.P. [2016] Least squares generative adversarial networks. arXiv preprint ArXiv:1611.04076.

SLIM

GAN in equations - Generator's loss

Generator's loss function:

$$\mathcal{L}_{\mathcal{G}} = \mathbb{E}_{\mathbf{x} \sim p_{\mathrm{measurements}}(\mathbf{x})} \left[\underbrace{\left(1 - \mathcal{D}_{\theta(\mathcal{D})} \left(\mathcal{G}_{\theta(\mathcal{G})} (\mathbf{x}) \right) \right)^2} \right]$$

Minimizing the loss will result in generating reconstructed data that is close to samples from distribution of fully sampled data

Distribution of partial measurements

Generator's ability to "fool" the discriminator by generating a reconstruction that makes the discriminator output 1, although the input to discriminator is not from p_{data}

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T. and Efros, A.A., 2016. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2536-2544).

Reconstruction loss

Reconstruction loss function:

$$\mathcal{L}_{\mathrm{reconstruction}} = \mathbb{E}_{\mathbf{x} \sim p_{\mathrm{measurements}}(\mathbf{x}),\ \mathbf{y} \sim p_{\mathrm{data}}(\mathbf{y})} \left[\left\| \mathcal{G}_{\theta(\mathcal{G})}(\mathbf{x}) - \mathbf{y} \right\|_1 \right],$$

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T. and Efros, A.A., 2016. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2536-2544).

Reconstruction loss

Reconstruction loss function:

$$\mathcal{L}_{\text{reconstruction}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{measurements}}(\mathbf{x}),\ \mathbf{y}\sim p_{\text{data}}(\mathbf{y})} \left[\underbrace{\|\mathcal{G}_{\theta(\mathcal{G})}(\mathbf{x}) - \mathbf{y}\|_1}_{} \right],$$

Enforcing the network to do reconstruction over paired measurements and fully sampled data in the training dataset, i.e., $\mathbf{x}_i = M \odot \mathbf{y}_i,\ \mathbf{x}_i \in S_{\text{data}},\ \mathbf{y}_i \in S_{\text{measurements}}$

The reconstruction loss is responsible for coherence with regards to the partial measurements.

Optimization problem

Solving via an alternating minimization scheme:

$$\min_{\theta(\mathcal{D})} \{\mathcal{L}_{\mathcal{D}}\},$$

$$\min_{\theta(\mathcal{G})} \{\mathcal{L}_{\mathcal{G}} + \lambda \mathcal{L}_{\text{reconstruction}}\}.$$

$\lambda = 100$: a hyper-parameter

Choosing λ is a trade off between getting a reconstruction that is:

coherent with measurements $\longleftrightarrow$ close to samples drawn from p_{data}

Training details

We used **Adam** optimizer with **batch size** 1

Learning rate for generator: 0.0002

Learning rate for discriminator: 0.0001

momentum: 0.5

Linearly decaying **learning rate** after 100 passes over dataset

Network architecture - Generator

- Input**: 202x202x2**, for **real** and **imaginary** parts
- Convolution layer with stride **1**, **7x7** filter **size**, and **64** filters, followed by batch normalization and ReLU.
- **2** convolution layers with stride **2** (down-sampling by factor of 2), **3x3** filter size**,** and **128** and **256** filters, respectively, followed by batch normalization and ReLU.
- **9** residual blocks with **256** filters
- **2** (de)convolution layers with stride **0.5** (up-sampling by factor of 2), **3x3** filter size, and **128** and **64** filters**,** respectively**,** followed by batch normalization and ReLU.
- Convolution layer with stride **1**, **7x7** filter size, and **2** filters, followed by batch normalization and ReLU.
- output**: 202x202x2** for **real** and **imaginary** part of the reconstruction

Network architecture - Discriminator

- Input**: 202x202x2**, for **real** and **imaginary** parts
- Convolution layer with stride **2**, **4x4** filter size, and **64** filters, followed by leaky ReLU.
- Convolution layer with stride **2**, **4x4** filter size, and **128** filters, followed by batch normalization and leaky ReLU.
- Convolution layer with stride **2**, **4x4** filter size, and **256** filters, followed by batch normalization and leaky ReLU.
- Convolution layer with stride **2**, **4x4** filter size, and **512** filters, followed by batch normalization and leaky ReLU.
- Convolution layer with stride **1**, **4x4** filter size, and **2** filters

Dataset

Seismic data is generated using finite difference method from the 3D Overthrust model.

- Number of receivers: 202x202
- Number of sources: 102x102
- Source/receiver sampling: 25 m

We extract the 5.31 Hz frequency slices for our experiment:

- Number of frequency slices in each frequency: 10404

Experiment 1

Reconstructing data with 90% missing values with **random** and **column-wise** sampling:

1. Measurements in training dataset ($S_{\text{measurement}}$): 250 frequency slices with 90% randomly missing entries (varying sampling mask) and 250 column-wise missing entries with the same sampling rate
2. Fully sampled data in training dataset (S_{data}): densely sampled frequency slices corresponding to measurements

Training data used: 500 out of 10404 frequency slices (**4.8%**)

Experiment 2

Showing the capability of our method to recover from large **contiguous areas** of missing data:

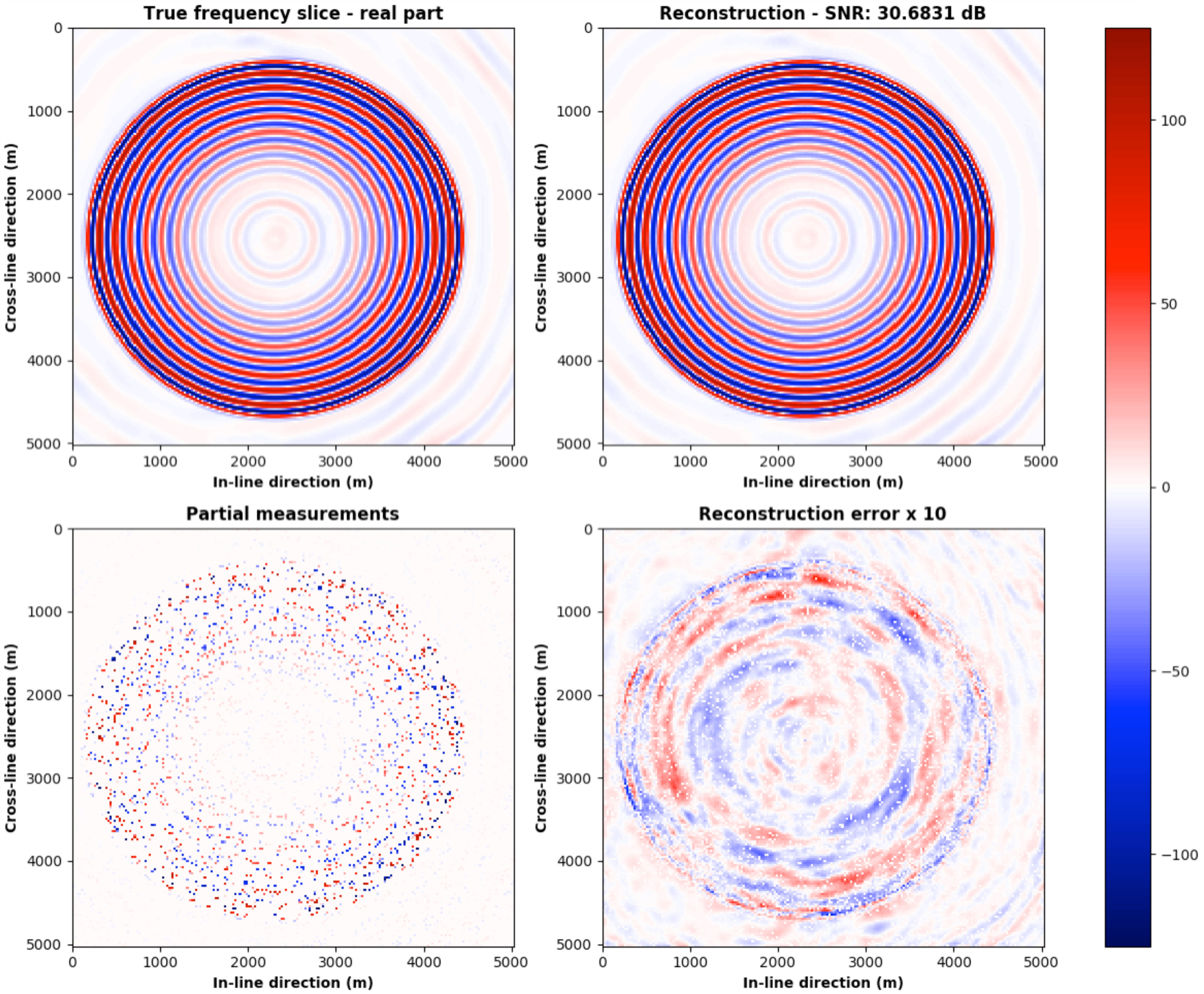
1. Measurements in training dataset ($S_{\text{measurement}}$): 2000 frequency slices with 50% missing entries as a square in the middle
2. Fully sampled data in training dataset (S_{data}): fully sampled frequency slices corresponding to measurements

Training data used: 2000 out of 10404 frequency slices (**19.2%**)

Random sampling

Sampling rate: 10%

Recovery SNR: 30.68 dB

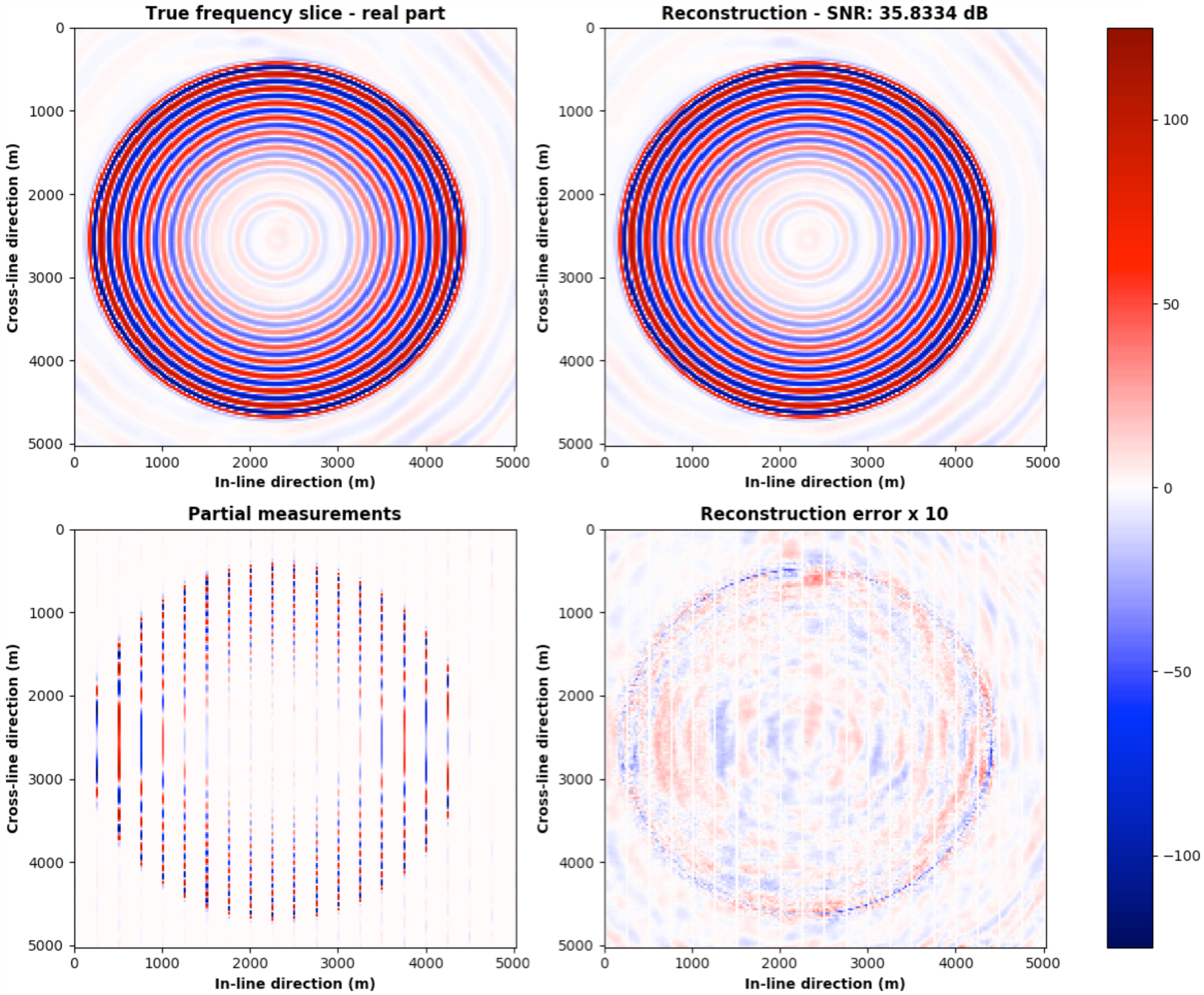


Top left: True frequency slice
Top right: Result
Bottom left: Measurements
Bottom right: Difference between true and result x10

Column-wise Sampling

Sampling rate: 10%

Recovery SNR: 35.83 dB

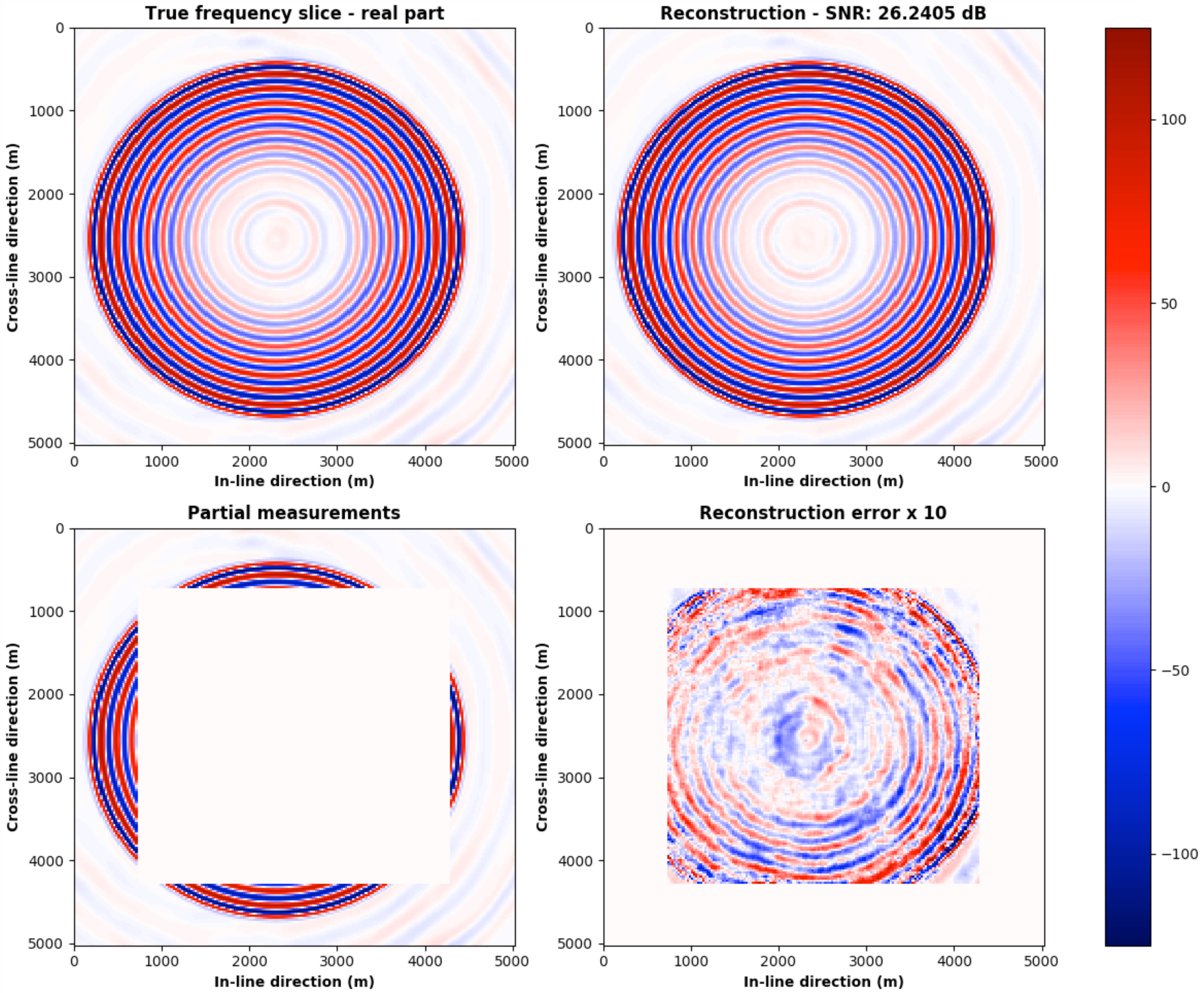


Top left: True frequency slice
Top right: Result
Bottom left: Measurements
Bottom right: Difference between true and result x10

Contiguous area of missing data

Sampling rate: 50%

Recovery SNR: 26.24 dB



Top left: True frequency slice
Top right: Result
Bottom left: Measurements
Bottom right: Difference between true and result x10

Discussion and conclusions

- Using a deep learning scheme we are able to reconstruct seismic data with **arbitrarily type of sampling** with **large percentages of missing values**.
- We assumed that training data is available, which requires having a small percentage (5% in first experiment) of shots fully-sampled.
- We observed that the quality of reconstruction is lower when we have large gaps in the observed data.

Missing type	Missing percentage	Average recovery SNR (dB)
Column-wise	90%	33.7072
Randomly	90%	29.6543
Square	50%	20.6053

Table 1: Average reconstruction SNR.

Future work

Designing an acquisition scheme to exploit our method, i.e.:

- recording only 5% of shots fully sampled
- record the rest of shots with 90% missing receivers

map from domain of data without multiples to domain of data with multiples, i.e. **multiple prediction**

SLIM

Acknowledgements

This research was carried out as part of the SINBAD project with the support of the member organizations of the SINBAD Consortium.

SINBAD

Thank you for your attention