Experiments on constructing seismic using generative adversarial network

Zhan Niu and Daniel Trad

Dec 2020

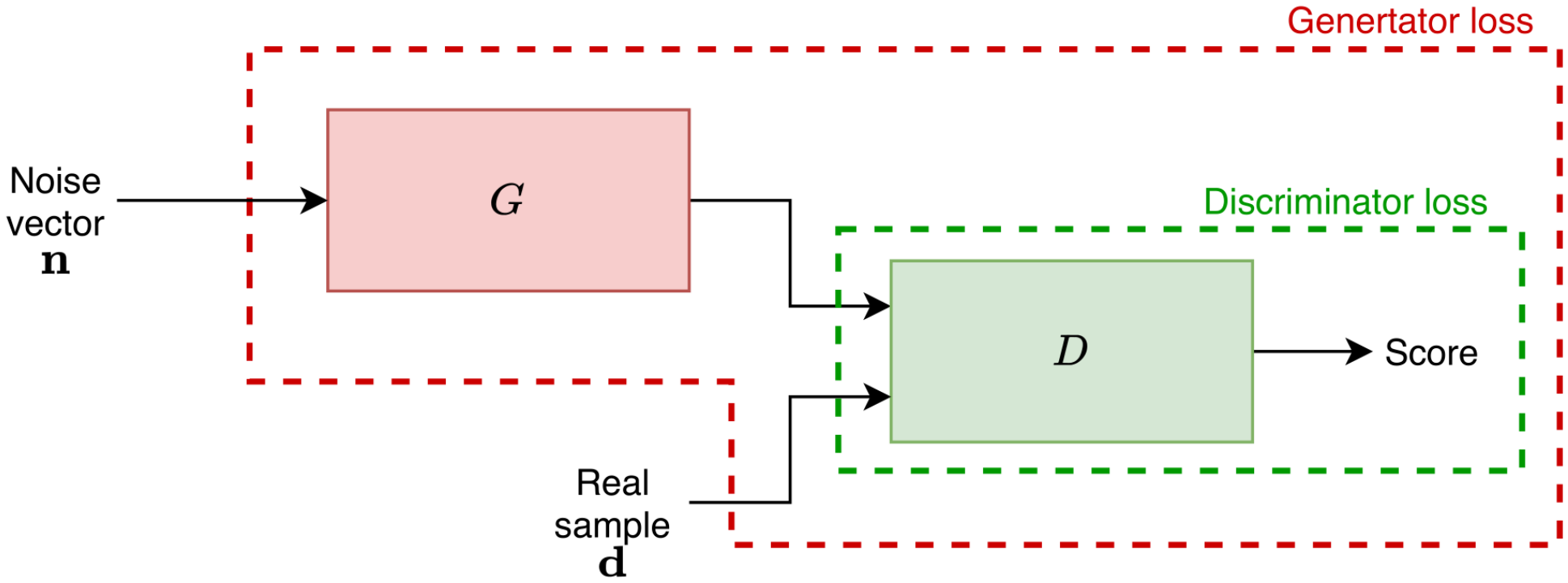
NSERC CRSNG
UNIVERSITY OF CALGARY
FACULTY OF SCIENCE
Department of Geoscience

Outline

- Theory
  - GAN
  - Wasserstein loss
  - Gradient penalty
- Architecture
  - Generator
  - Discriminator
- Training workflow
- Results
- Conclusion
- Acknowledgements

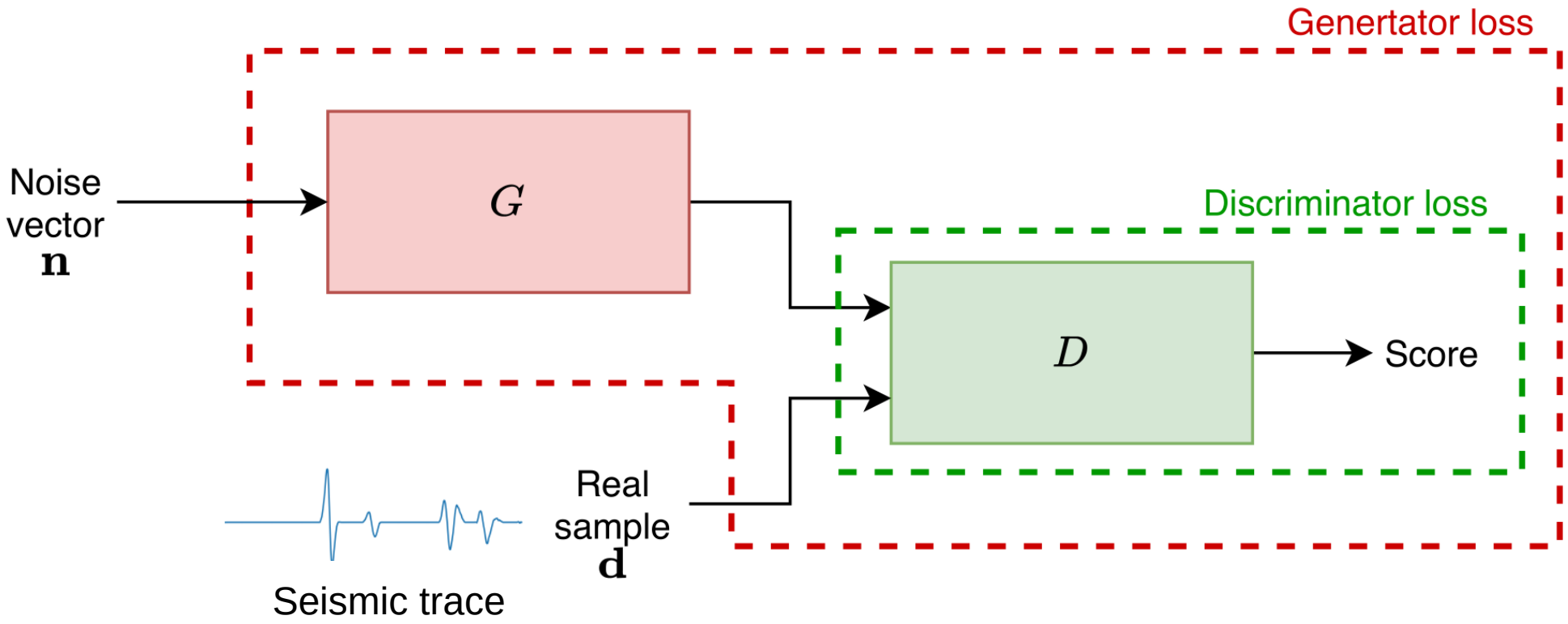
Generative adversarial network

- Forger vs Detective
- Unsupervised:
  - Goal: Transform from a random distribution to data distribution



Generative adversarial network

- Forger vs Detective
- Unsupervised:
  - Goal: Transform from a random distribution to data distribution



Wasserstein loss & gradient penalty

Score on real data

Score on generated sample

$$V_W = \min_G \max_D \mathbb{E}[D(\boldsymbol{d})] - \mathbb{E}\left[D(G(\boldsymbol{n}))\right]$$

Generated sample

(Arjovsky et al., 2017)

$$V_P = \left(\left\| \frac{\partial D(\boldsymbol{m})}{\partial \boldsymbol{m}} \right\|_2 - 1 \right)^2, \text{where } \boldsymbol{m} = \epsilon \boldsymbol{d} + (1-\epsilon) G(\boldsymbol{n}).$$

(Gulrajani et al. 2017)

Loss function

$$L_G = -\mathbb{E}\left[D(G(\boldsymbol{n}))\right],$$
$$L_D = \mathbb{E}\left[D(G(\boldsymbol{n}))\right] - \mathbb{E}[D(\boldsymbol{d})] + \lambda V_P$$

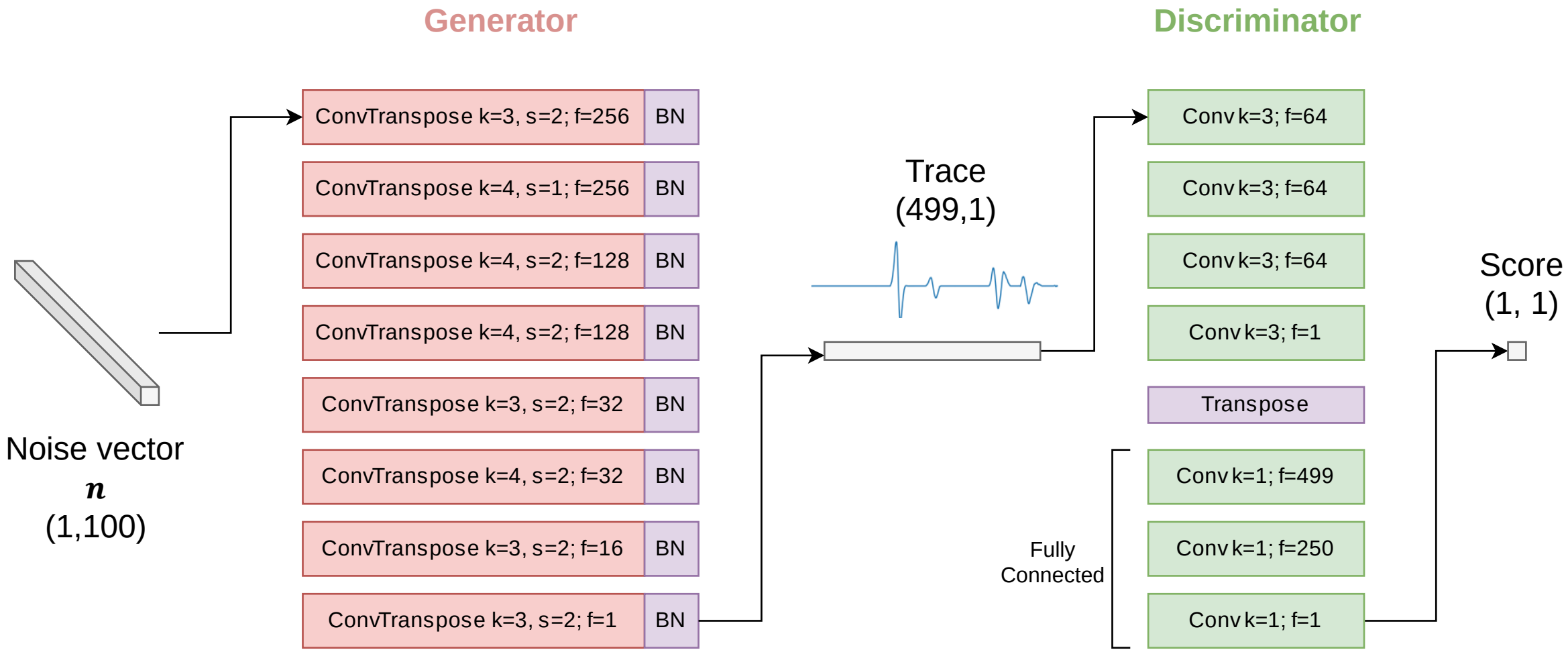
Optimizer:

Adam optimizer with $lr = 1 \times 10^{-4}, \beta_1 = 0.5$

Hyper-parameter:

lr_G, lr_D, λ and std for parameter initialization

Architecture



Training workflow

at each iteration:

Generator

for $n_d = 1, 2$:

$\boldsymbol{n} \sim N(0,1)$

$\widehat{\boldsymbol{d}} = G(\boldsymbol{n})$

$S_{fake} = \mathrm{D}(\widehat{\boldsymbol{d}})$

$S_{real} = \mathrm{D}(\boldsymbol{d})$

$\boldsymbol{m} = \epsilon \boldsymbol{d} + (1 - \epsilon)\widehat{\boldsymbol{d}}$, $\epsilon \sim U(0,1)$

Update D using S_{fake}, S_{real} and $\frac{\partial D(\boldsymbol{m})}{\partial \boldsymbol{m}}$

Discriminator

$\boldsymbol{n} \sim N(0,1)$

$\widehat{\boldsymbol{d}} = G(\boldsymbol{n})$

$S_{fake} = \mathrm{D}(\widehat{\boldsymbol{d}})$

Update G using S_{fake}

Training details:

$\lambda = 10.0$

$lr = 1 \times 10^{-4}$

$n_{\mathrm{epoch}} = 300$

$n_d = 2$

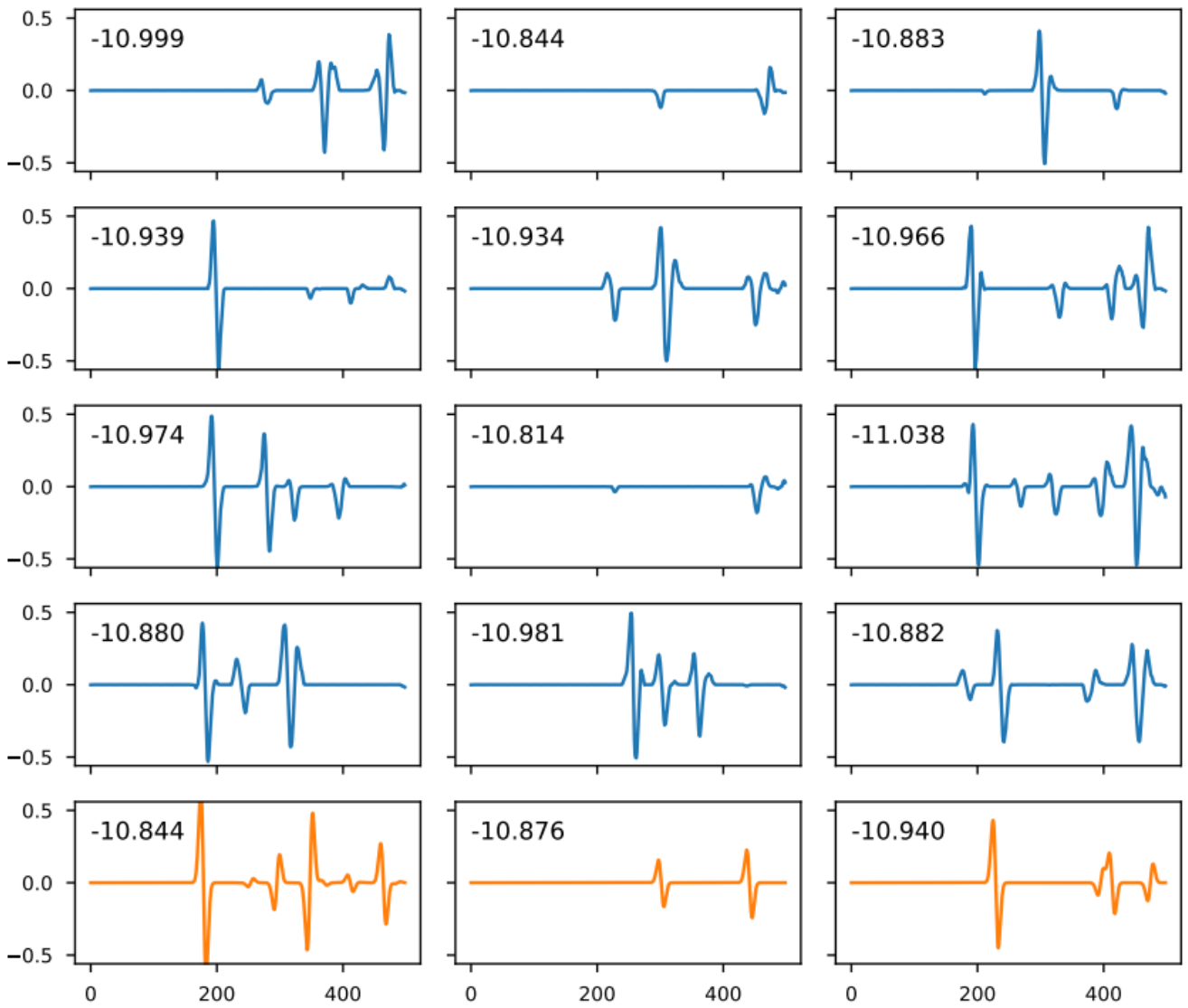
$\theta_G, \theta_D \sim N(0, 0.2)$

Data:

10,000 traces from FD modeling

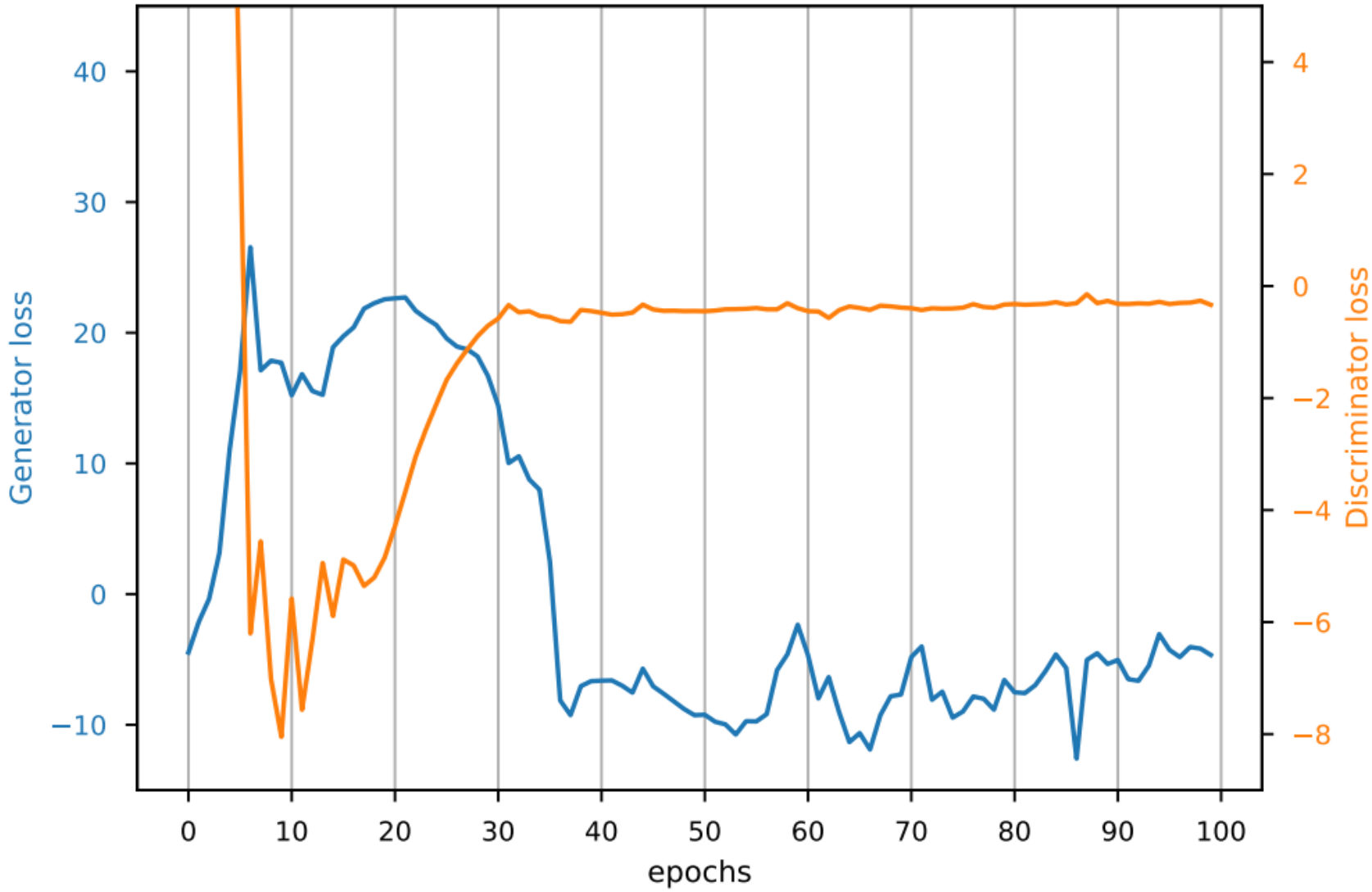
batchsize = 128

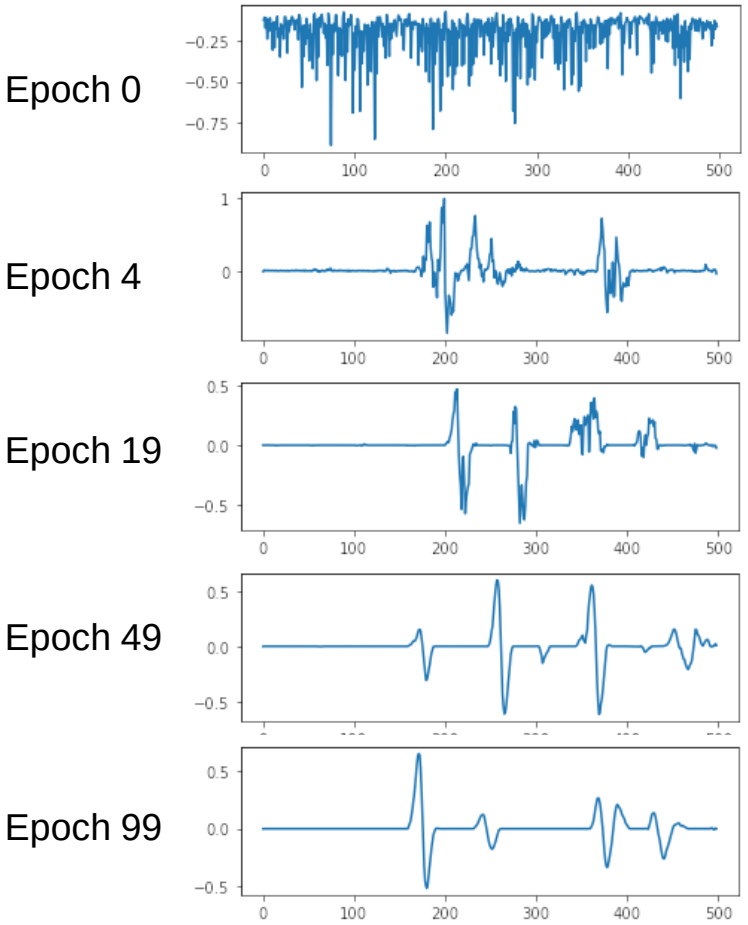
Results – Predictions



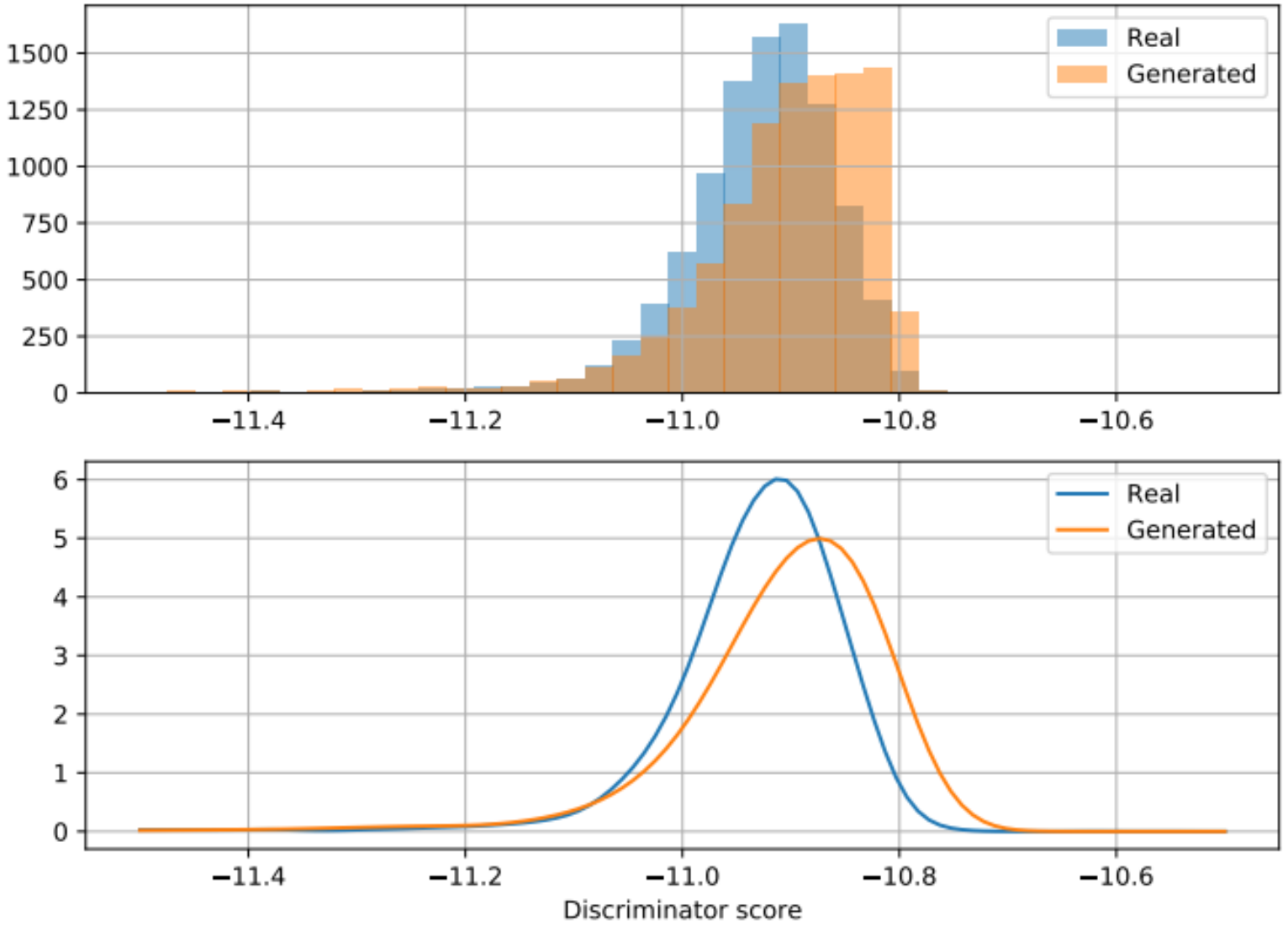
Real Data
Generated data

Results – Loss curves

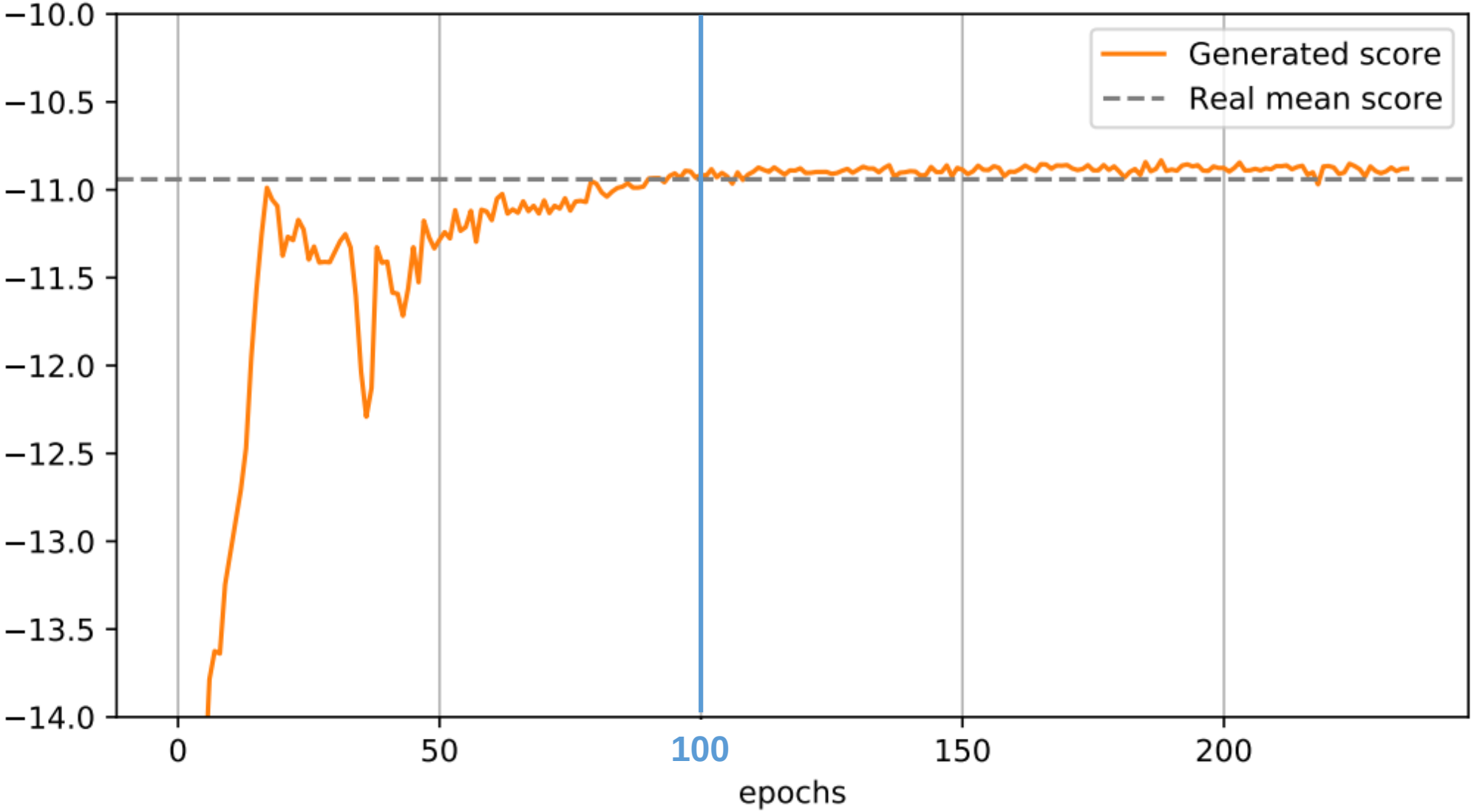




Results – Distributions



Results – Discriminator score



Conclusion

- We explore a way of generating 1D seismic traces using WGAN and the training workflow is established
- The two models reach equilibrium at around 100 epochs and hardly improve each other afterwards.
- The trained generator can successfully produce data in the same distribution of the real data, but there is still room for improvement

Future work:

- Expanding the same model architecture to 2D
- Gain more control over the generation process
- More advanced neural network structures

Acknowledgments

- All CREWES sponsors
- NSERC (CRDPJ 461179-13 and CRDPJ 543578-19)

Thank you!