US Army Corps of Engineers

# Fast prediction of riverine flow velocity using deep learning

Mojtaba Forghani[1], Yizhou Qian[2], Jonghyun Harry Lee[3], Matthew W. Farthing[4], Tyler Hesser[4], Peter K. Kitanidis[2,5], and Eric F. Darve[1,2]

[1]Mechanical Engineering, Stanford University, [2]Institute for Computational and Mathematical Engineering, Stanford University, [3]Civil and Environmental Engineering, University of Hawaii at Manoa, [4]US Army Corps of Engineers ERDC-CHL, [5]Civil and Environmental Engineering, Stanford University

## Overview

Fast and reliable prediction of river flow velocities is important in many applications, including flood risk management. The shallow water equations (SWEs) are commonly used for this purpose. However, traditional numerical solvers of the SWEs are computationally expensive and require high-resolution riverbed profile measurement (bathymetry).

In this work, we propose a two-stage process in which, first, using the principal component geostatistical approach (PCGA) we estimate the probability density function of the bathymetry from flow velocity measurements, and then use machine learning (ML) algorithms to obtain a fast solver for the SWEs. The fast solver uses realizations from the posterior bathymetry distribution and takes as input the prescribed range of boundary conditions (BCs). The first stage allows us to predict flow velocities without direct measurement of the bathymetry. Furthermore, we augment the bathymetry posterior distribution to a more general class of distributions before providing them as inputs to ML algorithm in the second stage. This allows the solver to incorporate future direct bathymetry measurements into the flow velocity prediction for improved accuracy, even if the bathymetry changes over time compared to its original indirect estimation.

We propose and benchmark three different solvers, referred to as PCA-DNN (principal component analysis-deep neural network), SE (supervised encoder), and SVE (supervised variational encoder), and validate them on the Savannah river near Augusta, GA. Our results show that the fast solvers are capable of predicting flow velocities for different bathymetry and BCs with good accuracy, at a computational cost that is significantly lower than the cost of solving the full boundary value problem with traditional methods.

## Savannah River, GA

In order to demonstrate the performance of the proposed methods, we applied them to flow velocity prediction of the Savannah river, GA (Figure 1).

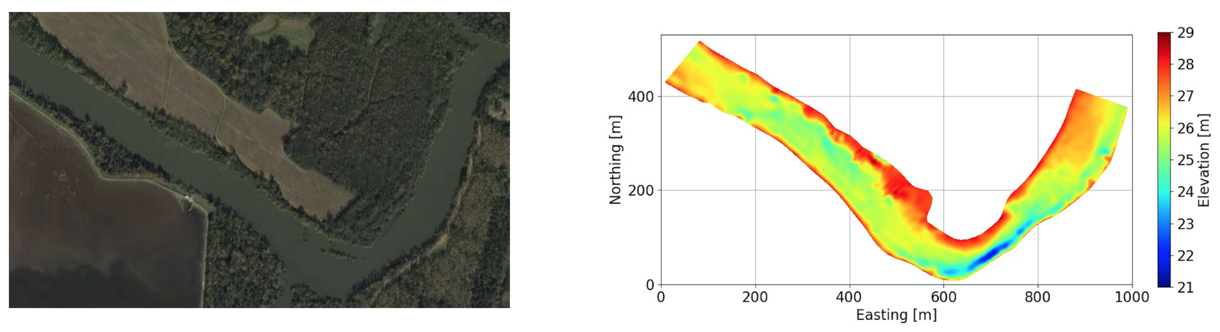


Figure 1: 1 mile reach of the Savannah River near Augusta, GA (left), and high-resolution bathymetry survey by U.S. Army Corps of Engineers (USACE) (right)

## Overview of the solver development

The stages to develop the fast solver is shown below. The DNN (deep neural network) block is the fast solver. ROM is the reduced order model.

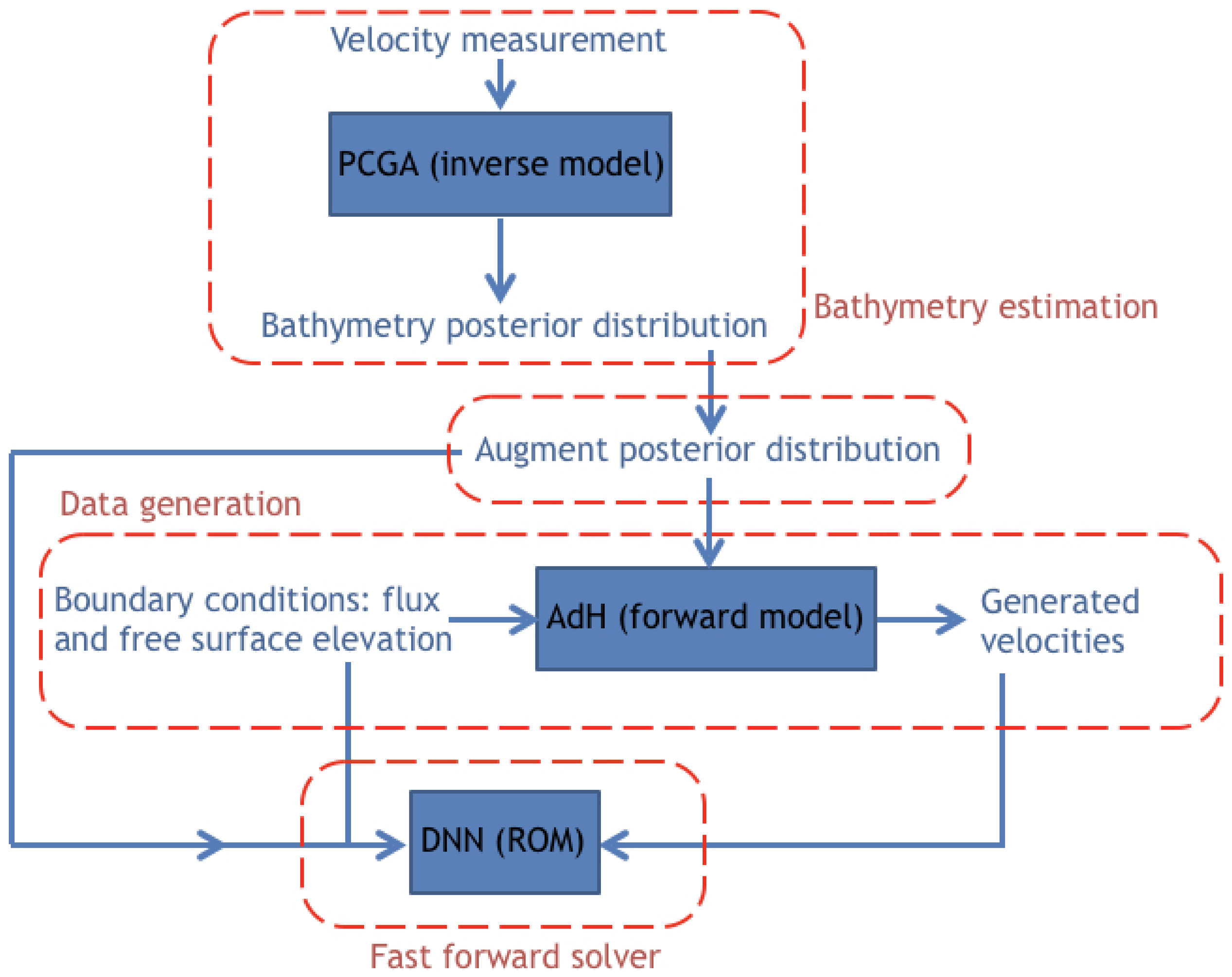
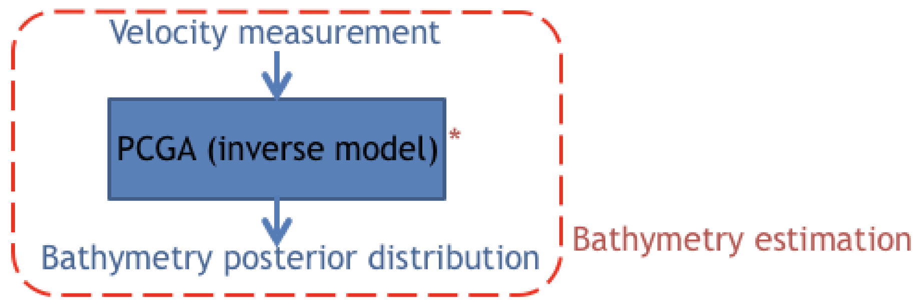



Figure 2: Stages to develop the DNN-based fast forward solver.

## PCGA (bathymetry estimation)

The first stage is the estimation of the river bathymetry from surface flow velocity measurement through PCGA.



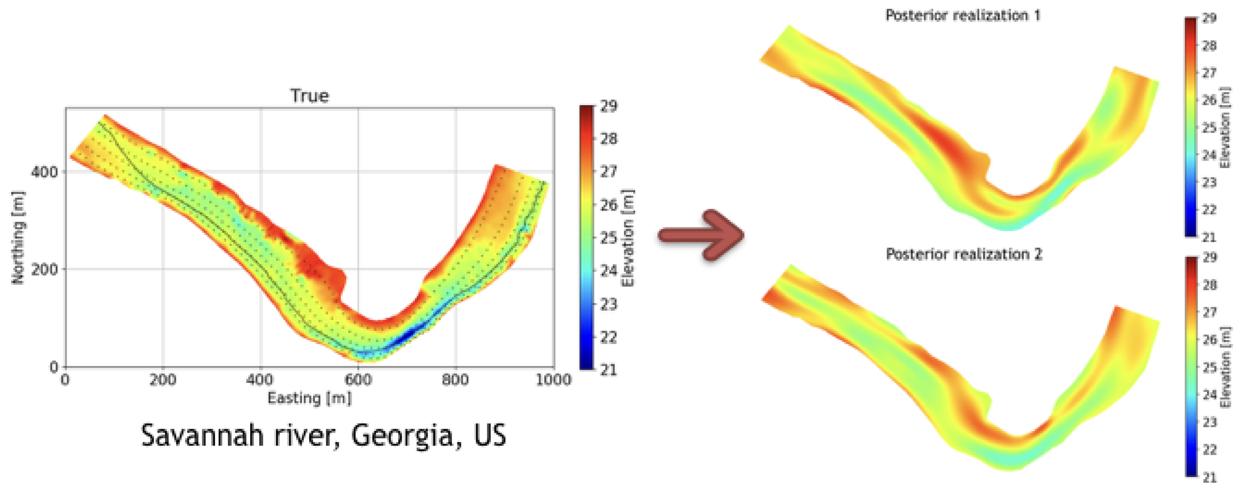




Figure 3: Posterior bathymetry distribution is estimated via PCGA. Left: actual bathymetry (assumed unknown). Right: examples of bathymetries sampled from the posterior.

## Augmenting posterior distribution

The second stage is posterior distribution augmentation. Its purpose is to broaden the range of bathymetries for which the fast solver is valid, e.g., when the bathymetry changes over time.

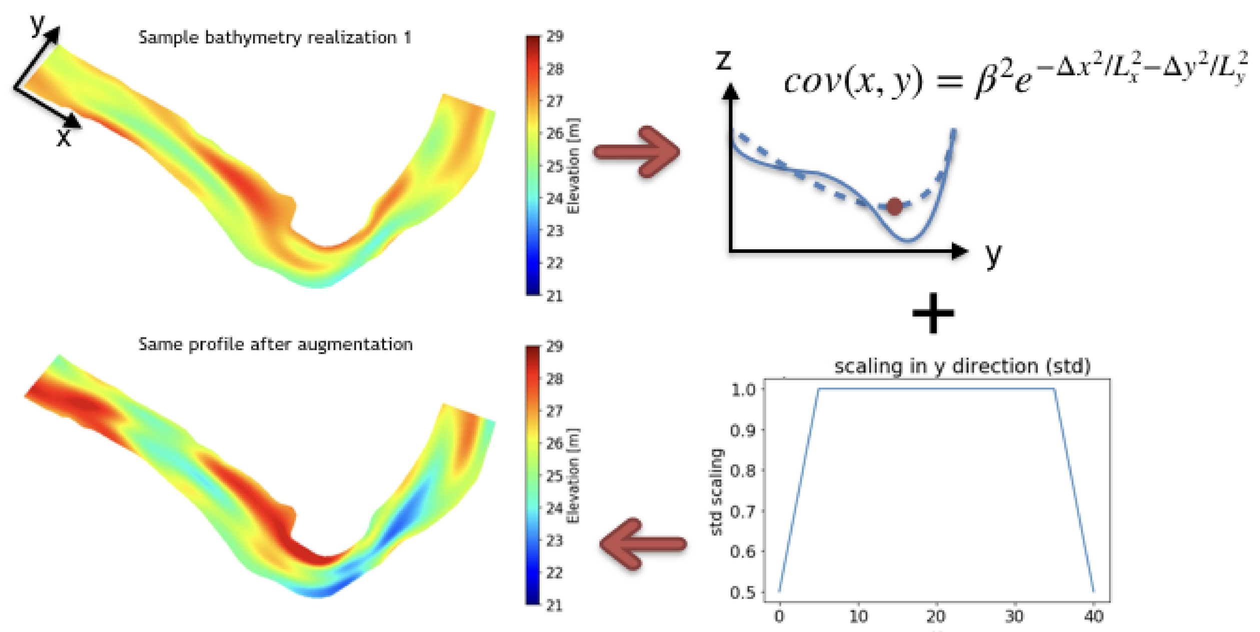



Figure 4: Augmentation happens via a Gaussian kernel and a scaling factor. The scaling factor captures the fact that the variations of the generated bathymetries near the shore are generally smaller than in the middle of the river.

## The DNN-based solvers

In the third stage, the data that are fed to DNNs are generated. The data include bathymetries, BCs, and flow velocities. In this process, the bathymetries sampled from the augmented posterior distribution and boundary conditions (BCs) are input to our numerical solver, AdH (Adaptive Hydraulics), to generate flow velocities. The fourth stage is the final stage where we train the solvers via the generated data. The solvers are shown below.

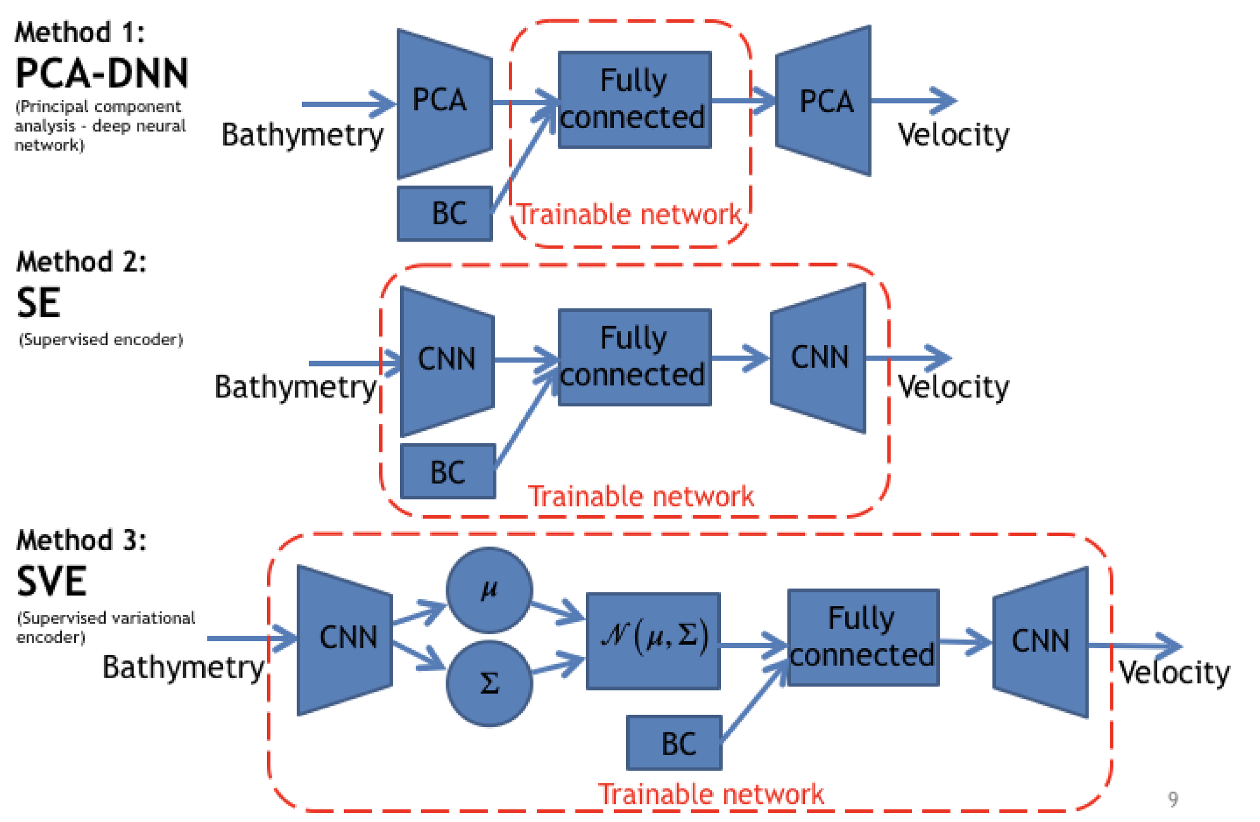



Figure 5: We have used three solvers: PCA-DNN, SE, and SVE.

## Performances

The train/validation/test sizes are 4000, 500, and 500, respectively. Their errors are shown below:

| RMSE of velocity magnitude [m/s] | Forward solver | | | |
|---|---|---|---|---|
| | PCA-DNN | PCA-DNN with linear map | SE | SVE |
| Train set | 0.0515 | 0.0514 | 0.0269 | 0.0286 |
| Validation set | 0.0570 | 0.0571 | 0.0374 | 0.0398 |
| Test set | 0.0546 | 0.0544 | 0.0381 | 0.0398 |

Figure 6: Performance of different solvers. SE/SVE perform significantly better. The PCA-DNN with linear map is similar to PCA-DNN except DNN has linear activation functions.

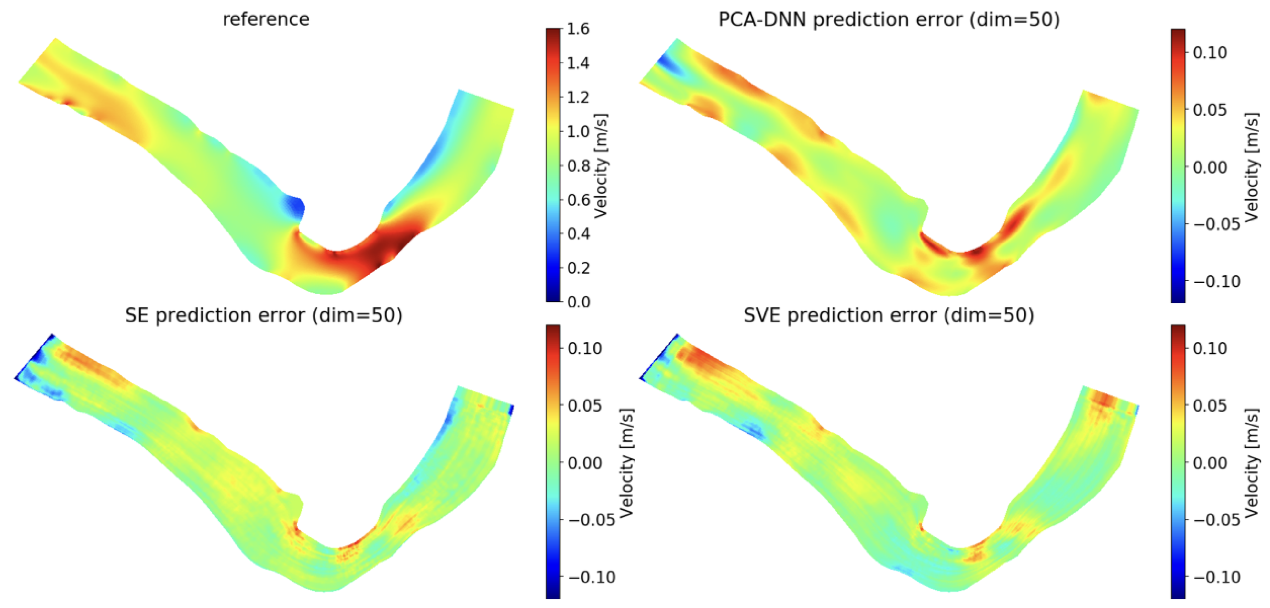



Figure 7: Example of prediction of different solvers for a test set datapoint. The error of SE/SVE is smaller.

## Performance with uncertain bathymetry

When bathymetry measurement is not available, we use posterior distribution for velocity prediction.

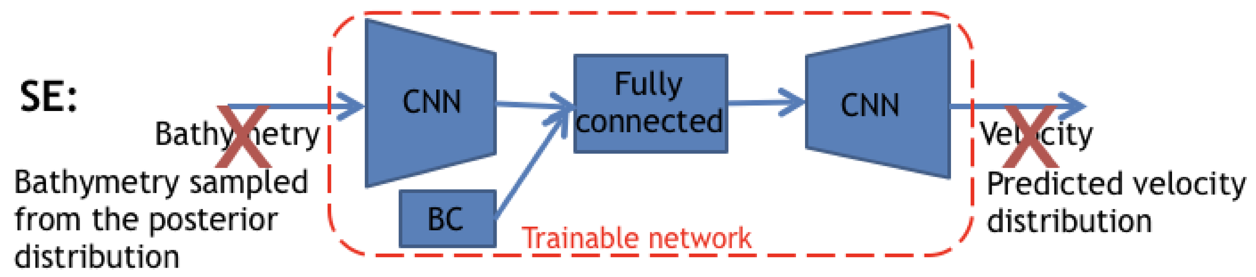



Figure 8: In the absence of bathymetry measurement, bathymetries sampled from the posterior distribution are input to DNNs and velocity distribution is obtained.

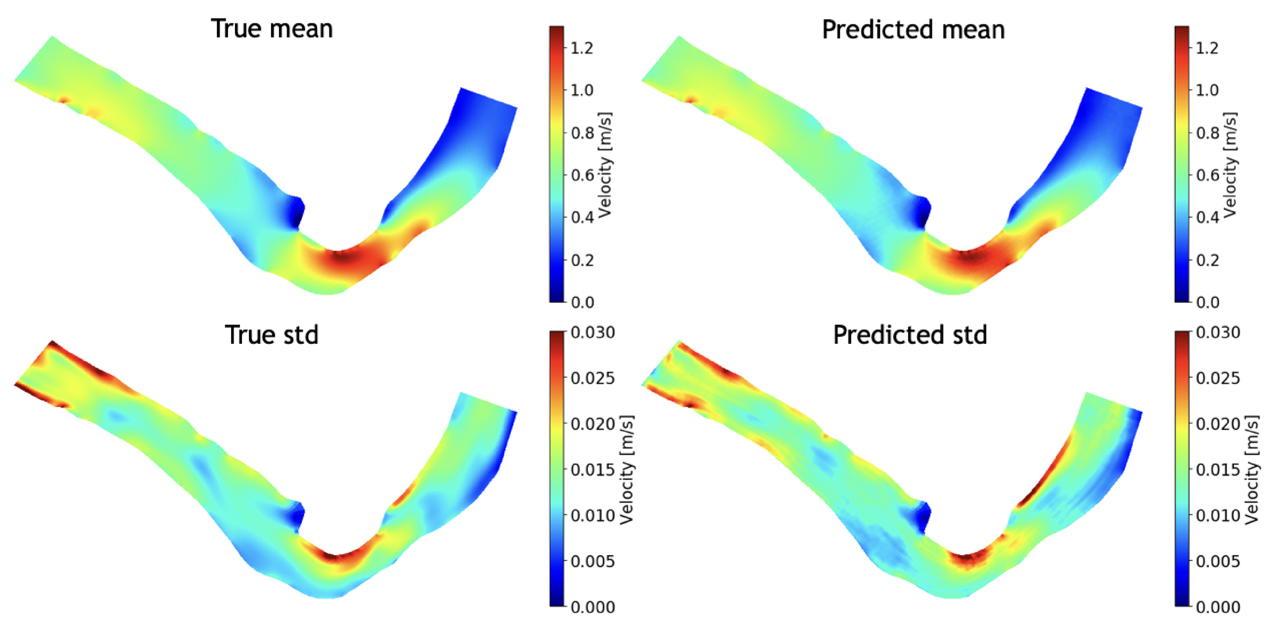



Figure 9: Example of true mean and standard deviation of velocity for a BC obtained from the AdH (left) and the predicted mean and standard deviation based on SE for the same BC.

## Conclusion

- All trainings can be done on personal CPU machines without access to GPU.
- The fast solvers are more than three orders of magnitude faster than numerical solvers (e.g., AdH).
- Direct bathymetry measurement not required when designing the solvers.
- Same solver can be used to predict velocity in the presence/absence of bathymetry measurement.

### References

- Forghani *et. al.* Application of deep learning to large scale riverine flow velocity estimation, *under review*
- Lee *et. al.*, Riverine bathymetry imaging with indirect observations, *WRR*, 2018
- Kitanidis and Lee, PCGA for large-dimensional inverse problem, *WRR*, 2014