Flood Prediction with Graph Neural Networks

Arnold Kazadi[1], James Doss-Gollin[2], Antonia Sebastian[3], Arlei Silva[1]

[1] Computer Science, Rice University

[2]Civil and Environmental Engineering, Rice University

[3]Earth, Marine and Environmental Sciences, University of North Carolina at Chapel Hill

Introduction

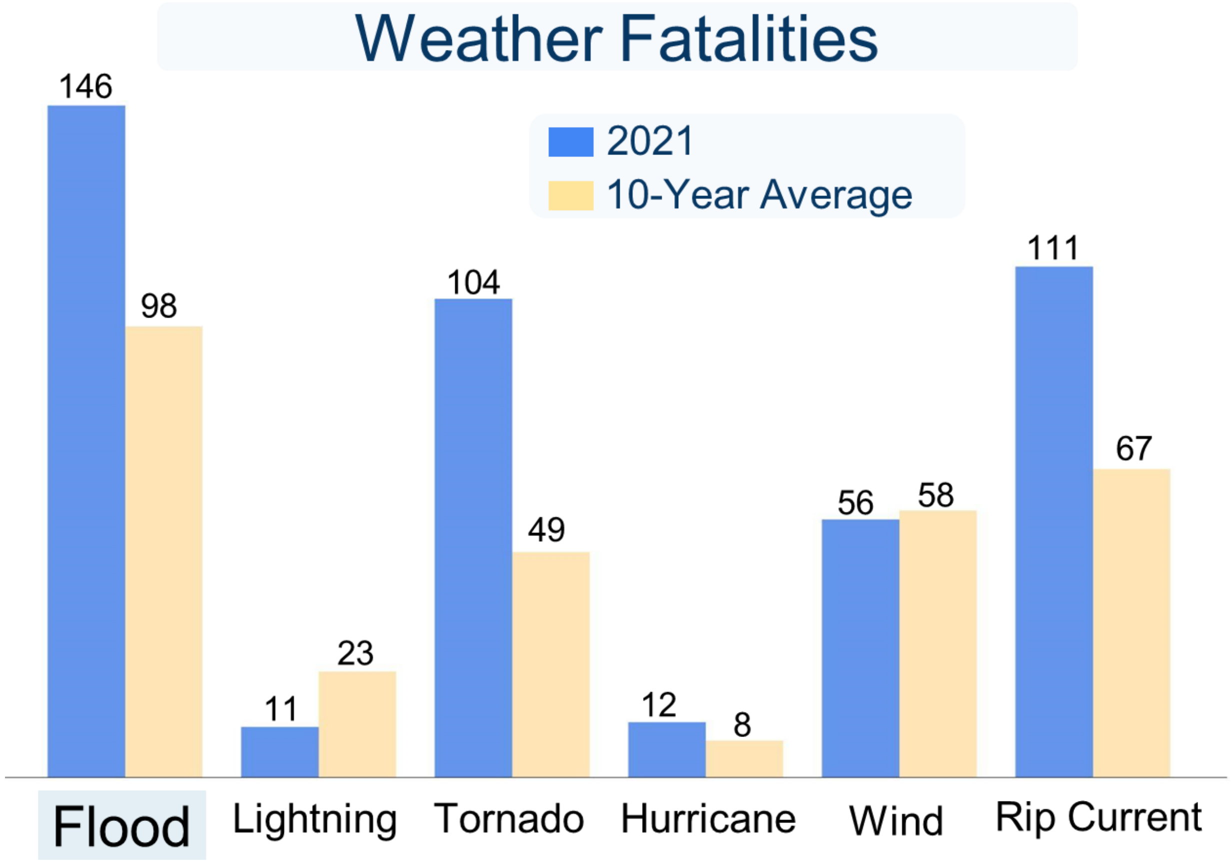
- Flooding: one of the most devastating hazards in the world.

- E.g., Hurricane Harvey's flooding led to **$125 billion** in losses, **30,000+ people displaced,** and **200,000+ damaged homes and businesses**

Introduction

- Flooding: one of the most devastating hazards in the world.
- Climate change ⇒ More precipitation ⇒ More flooding

- E.g., Hurricane Harvey's flooding led to **$125 billion** in losses, **30,000+ people displaced,** and **200,000+ damaged homes and businesses**



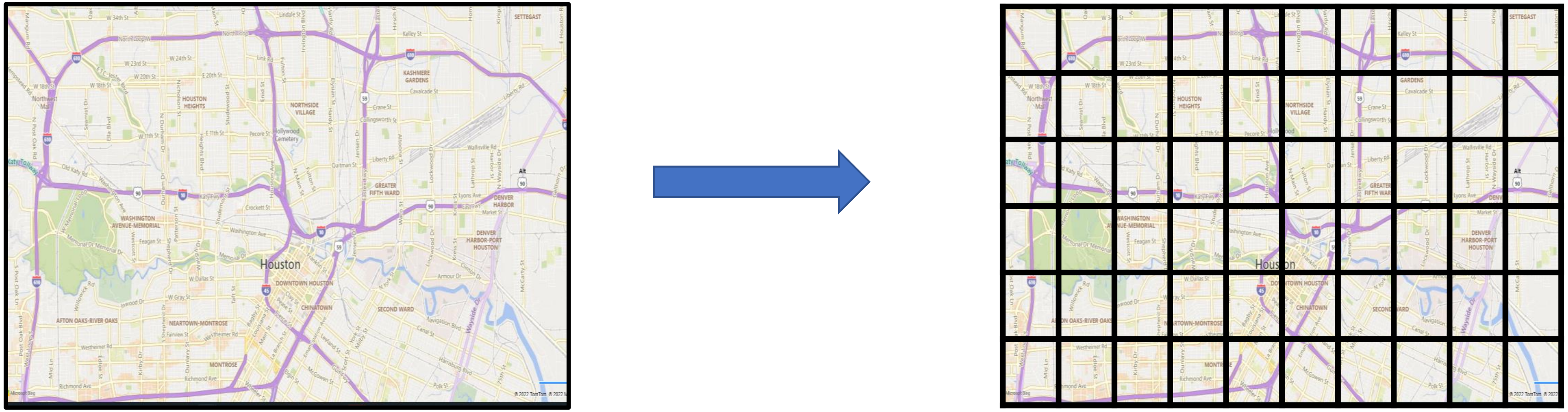
Introduction

- Popular approaches in practice simulate flooding by solving hydrodynamics (differential) equations (e.g., Saint-Venant equations)
  - Time consuming
  - Not scalable
- Our approach: FloodGNN flood prediction with graph neural networks.

Proposed Method

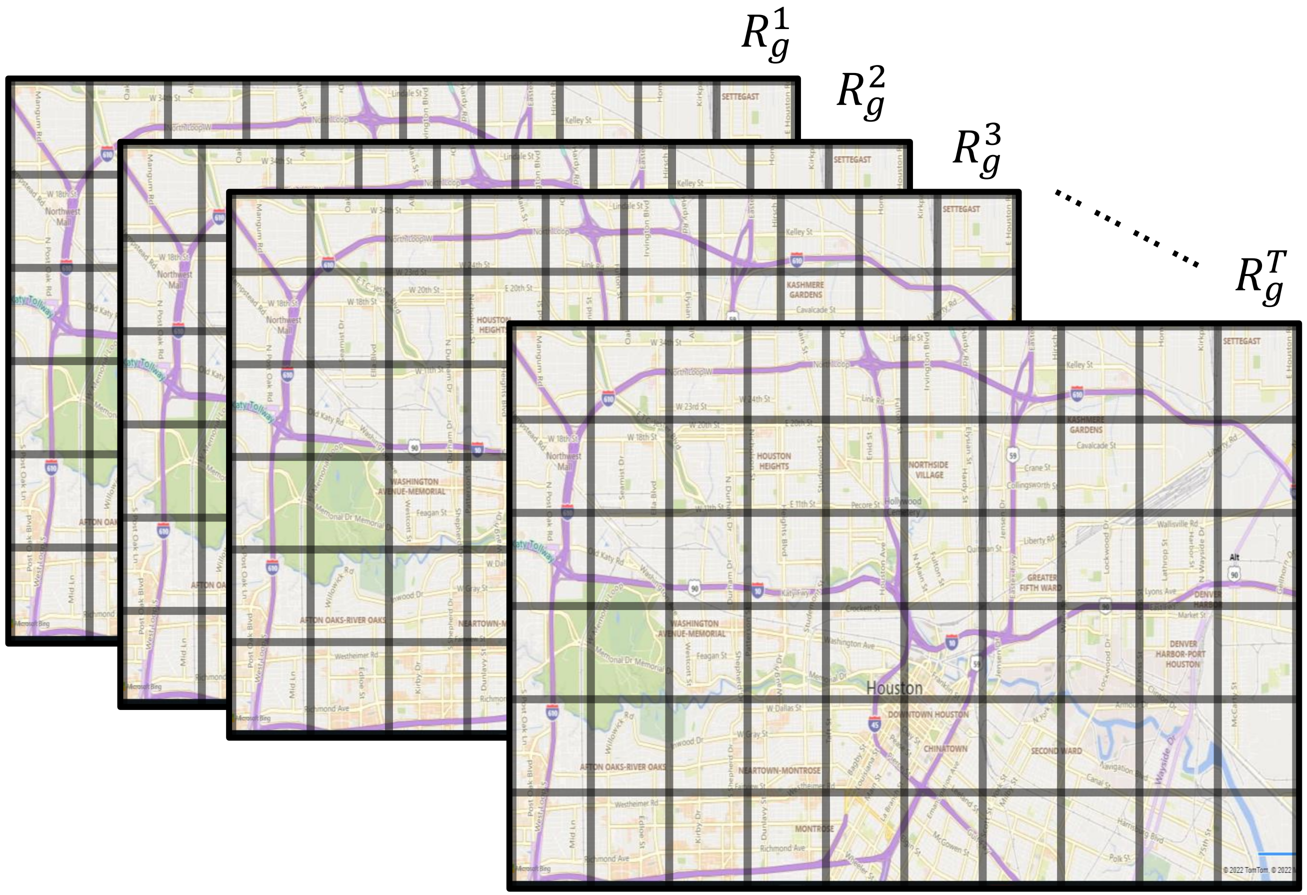
- A region R_g is represented as a mesh.

R_g



Proposed Method

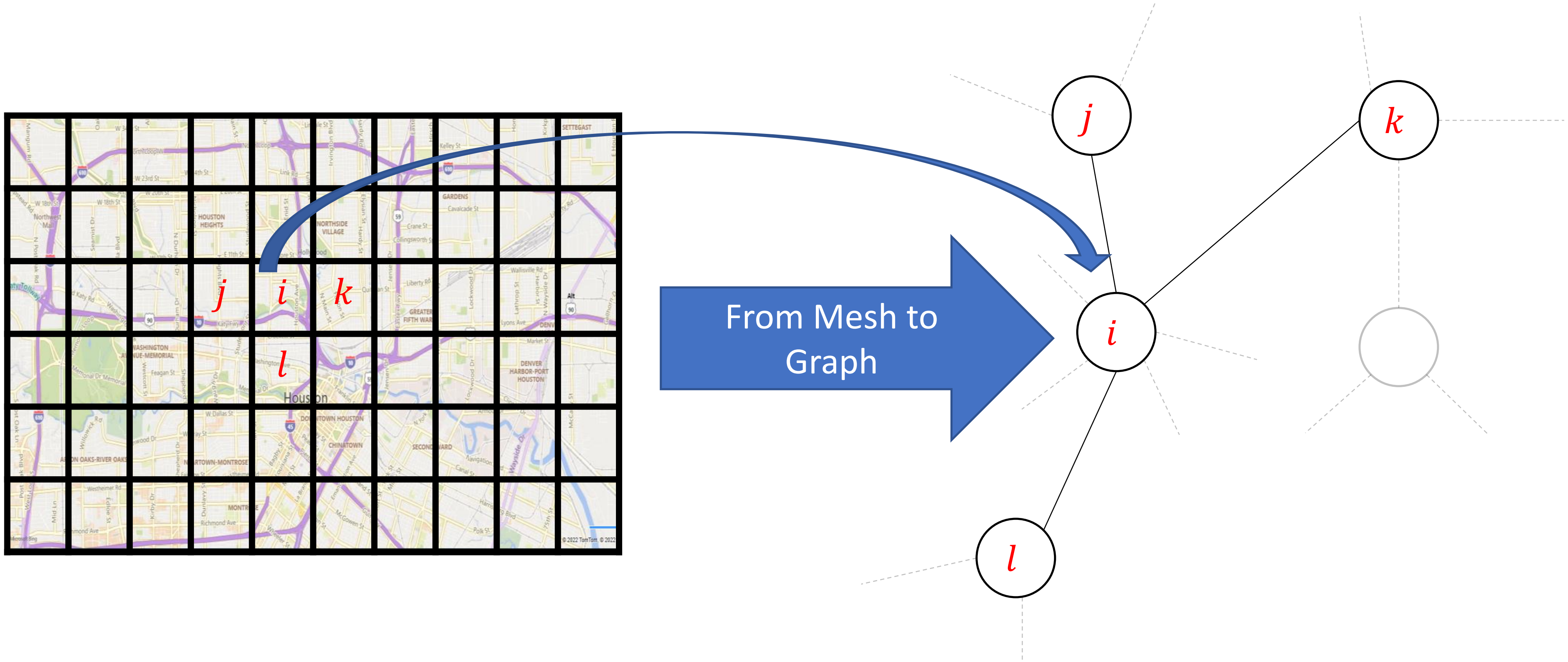
- Flooding events of a region R_g given as a time series $R_g^1, R_g^2, R_g^3 \dots, R_g^T$



Proposed Method

Graph representation from Mesh (GNN with mesh *).

- Cell i ➔ node i
- Neighboring cells of i ➔ edges

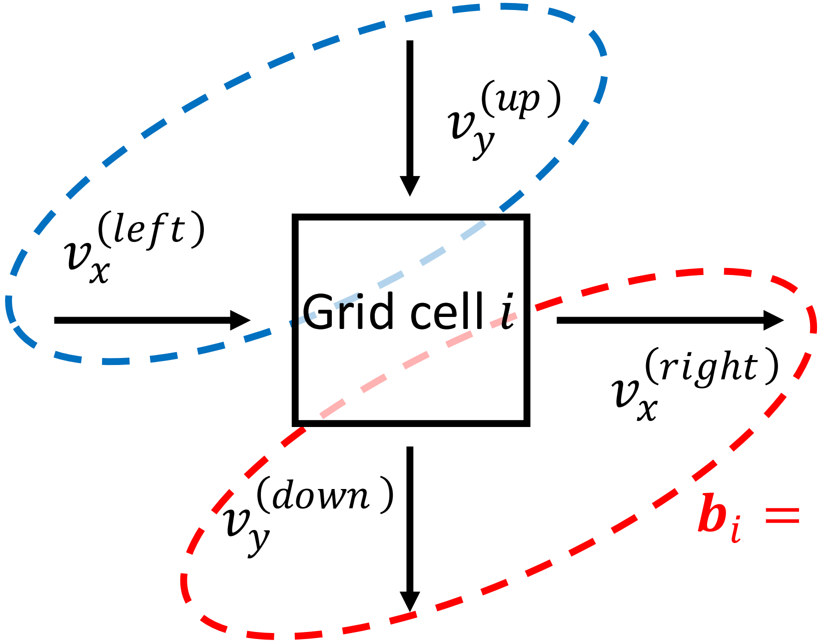


* Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. **Learning mesh-based simulation with graph network,** ICLR 2021

For each node $\boldsymbol{i}$

- Vector features $\boldsymbol{V}_i^t$ (velocities) = $\left[\frac{\boldsymbol{a}_i^t}{\|\boldsymbol{a}_i^t\|}, \frac{\boldsymbol{b}_i^t}{\|\boldsymbol{b}_i^t\|}\right]^T \in R^{2\times 2}$

$\boldsymbol{a}_i = (v_x^{(left)}, v_y^{(up)})$



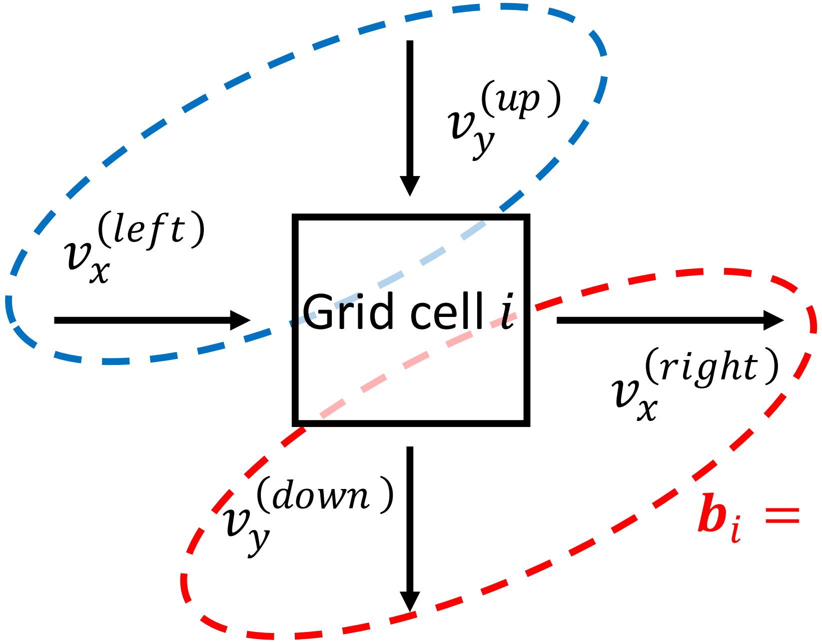
$\boldsymbol{b}_i = (v_x^{(right)}, v_y^{(down)})$

- Scalar feature $\boldsymbol{s}_i^t$
  - e_i: ground elevation ; n_i: friction; d_i: distance to stream; w_i^t: water-depth

For each node $\boldsymbol{i}$

- Vector features $\boldsymbol{V}_i^t$(velocities) $= \left[\frac{\boldsymbol{a}_i^t}{||\boldsymbol{a}_i^t||}, \frac{\boldsymbol{b}_i^t}{||\boldsymbol{b}_i^t||}\right] \in R^{2\times 2}$

$\boldsymbol{a}_i = (v_x^{(left)}, v_y^{(up)})$



$\boldsymbol{b}_i = (v_x^{(right)}, v_y^{(down)})$

Geometric vector perceptrons (GVP)*

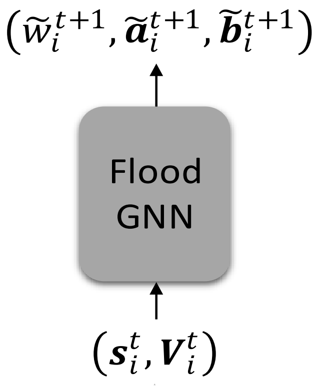
$$(s', V') = GVP(s, V)$$

Where $s', s \in R^n; V', V \in R^{m\times p}$

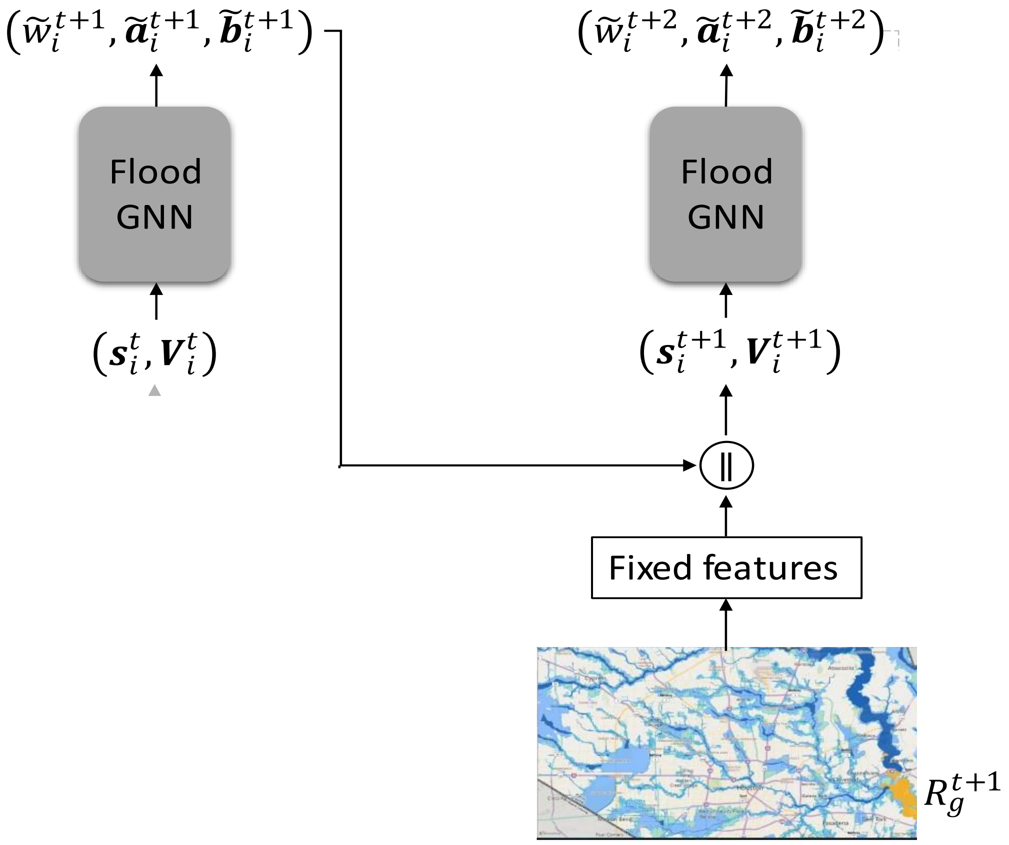
* Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. ***Learning from protein structure with geometric vector perceptrons***, ICLR 2021

- Scalar feature $\boldsymbol{s}_i^t$
  - e_i: ground elevation ; n_i: friction; d_i: distance to stream; w_i^t: water-depth

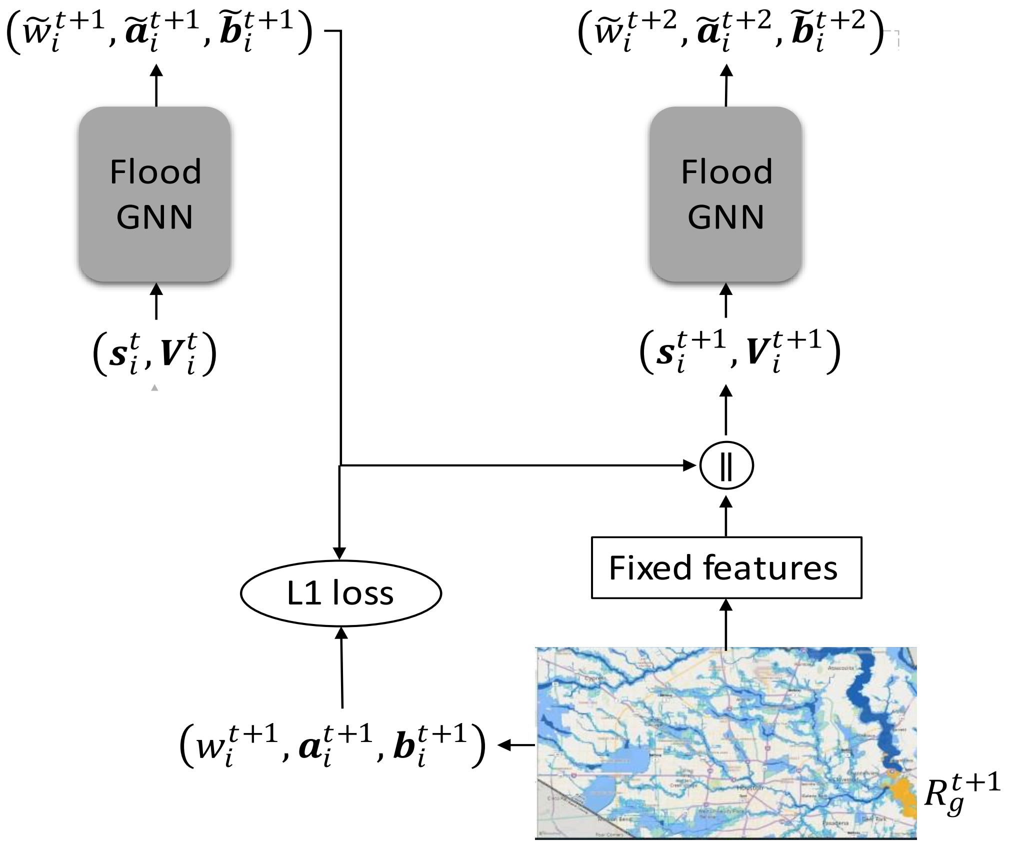
Proposed Method



Proposed Method



Proposed Method



Experiments

RMSE (lower better)

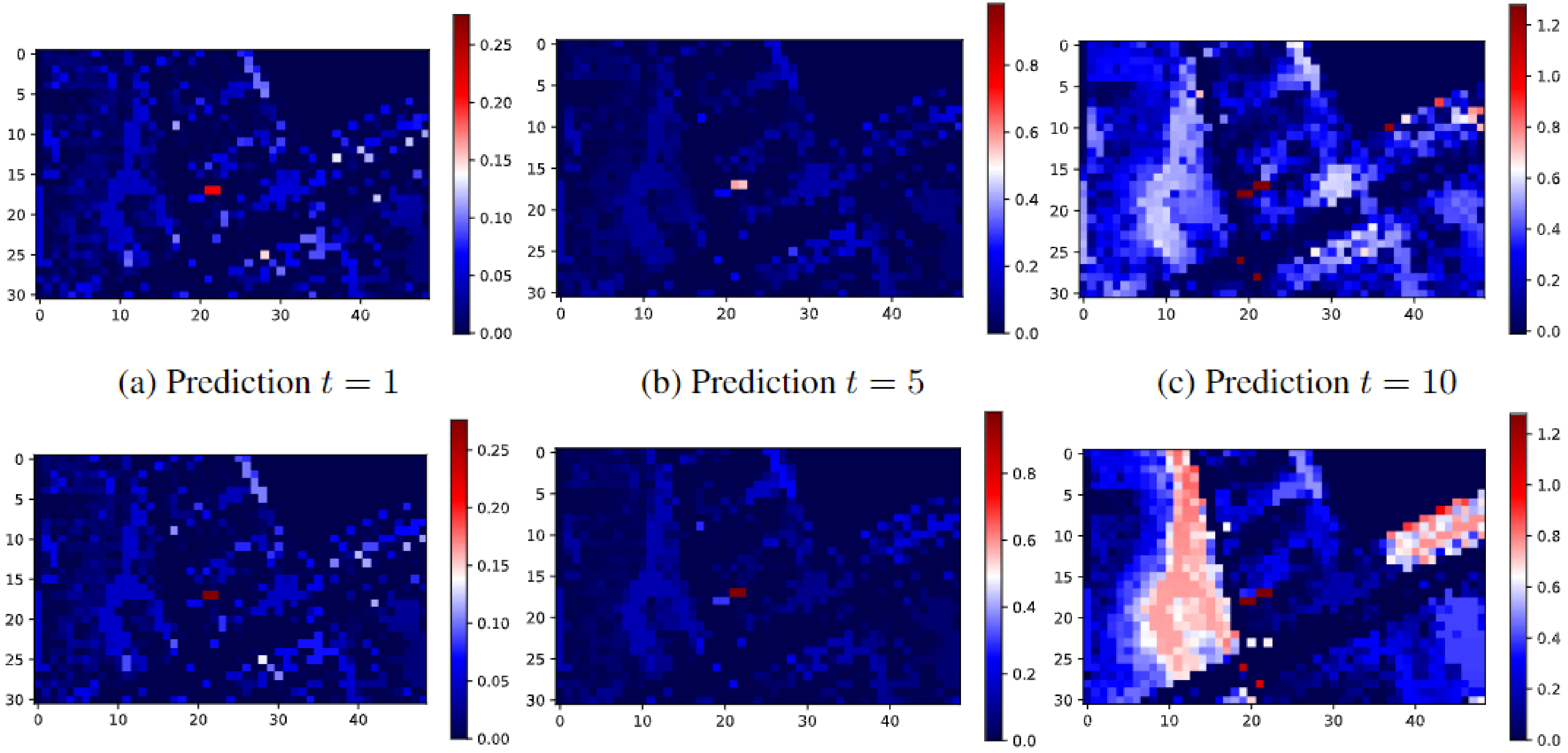
time-step t	1	2	3	4	5
FloodRNN	$.22 \pm .030$	$.33 \pm .030$	$.41 \pm .031$	$.48 \pm .034$	$.55 \pm .038$
FloodGNN-NoV	$.25 \pm .030$	$.40 \pm .031$	$.51 \pm .021$	$.60 \pm .007$	$.66 \pm .006$
FloodGNN (Ours)	$.17 \pm .031$	$.27 \pm .043$	$.34 \pm .040$	$.39 \pm .037$	$.44 \pm .036$

R^2 (higher better)

time-step t	1	2	3	4	5
FloodRNN	$.95 \pm .0160$	$.87 \pm .0390$	$.79 \pm .0540$	$.72 \pm .0620$	$.66 \pm .0660$
FloodGNN-NoV	$.95 \pm .0023$	$.88 \pm .0069$	$.80 \pm .0177$	$.71 \pm .0356$	$.63 \pm .0571$
FloodGNN (Ours)	$.98 \pm .0028$	$.93 \pm .0063$	$.89 \pm .0083$	$.85 \pm .0088$	$.80 \pm .0091$

- **FloodRNN**: RNN-based method
- **FloodGNN**: Our proposed method
- **FloodGNN-NoV**: FloodGNN variant, velocities as scalar features.

Experiments



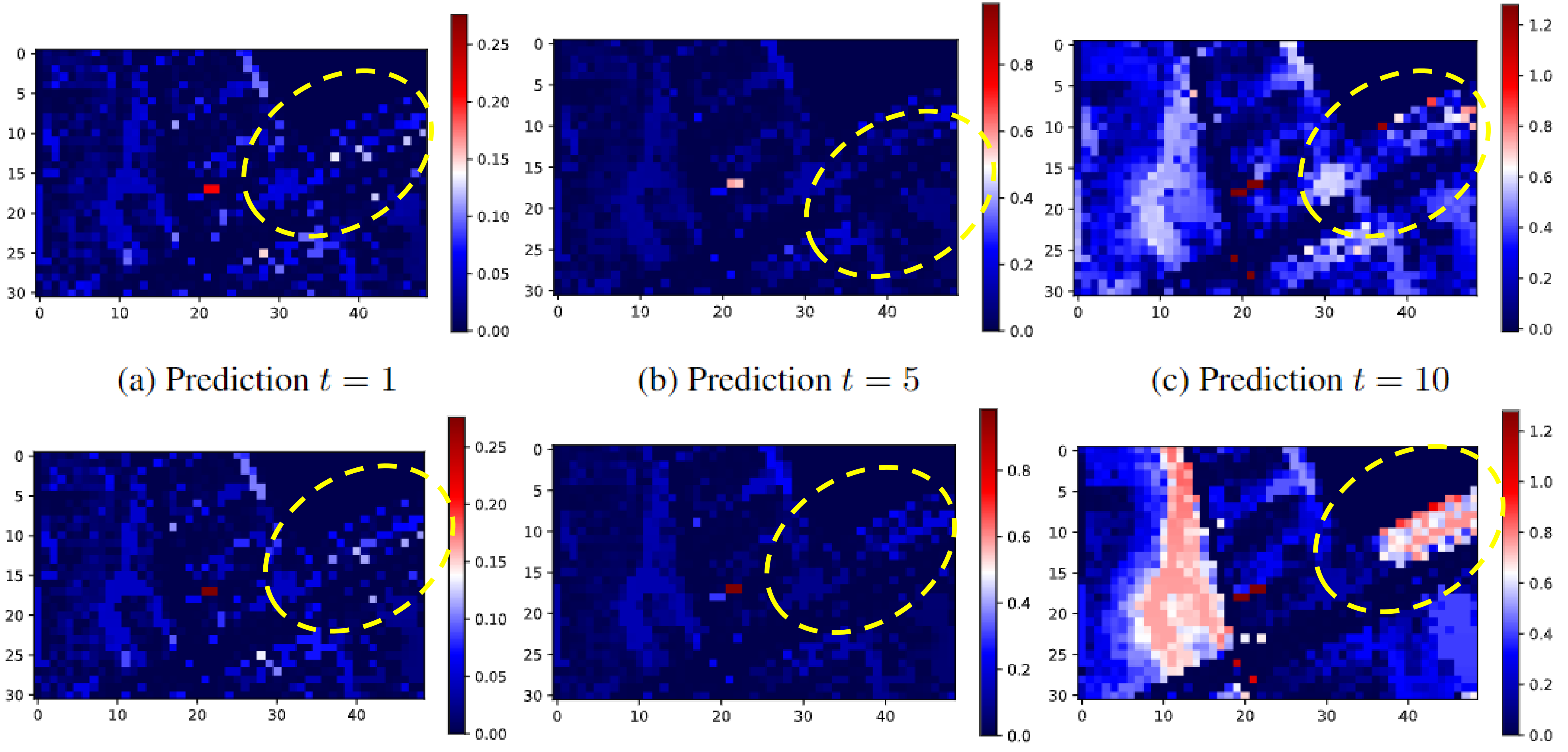
(a) Prediction $t = 1$ (b) Prediction $t = 5$ (c) Prediction $t = 10$

(d) Ground truth $t = 1$ (e) Ground truth $t = 5$ (f) Ground truth $t = 10$

Figure 3: Comparison between real data (bottom row) and predictions from our model (top row).

Experiments

Predictions follow trend, even though with underestimation.



(a) Prediction $t = 1$ (b) Prediction $t = 5$ (c) Prediction $t = 10$

(d) Ground truth $t = 1$ (e) Ground truth $t = 5$ (f) Ground truth $t = 10$

Figure 3: Comparison between real data (bottom row) and predictions from our model (top row).

Conclusion and ongoing work

- FloodGNN: spatio-temporal GNN for flood prediction.
- FloodGNN outperforms RNN-based model (no spatial relation).
- FloodGNN performs better with velocities as vector features.
- Future/ongoing work
  - Rainfall data input
  - Adaptive and irregular mesh representation
  - Physics-based constraints

For more information

- Email: akn7@rice.edu
- Website: https://kanz76.github.io/