Master of Science Thesis

Adaptation of Multi-Scale Hydraulic Graph Neural Networks for Pluvial Flood Modelling

Evaluation under Spatially Uniform and Spatially Varying Rainfall

Srijan Kumar Jha

Student number: 6168388

MSc Thesis in Environmental Engineering (Water Resources Engineering)
Delft University of Technology

Defence date: 26 October 2026

Thesis committee chair:	Dr. Riccardo Taormina
TU Delft supervisor:	Dr. Roberto Bentivoglio
TU Delft supervisor:	Prof. Markus Hrachowitz
Deltares supervisor:	Mr. Ruben Dahm
Deltares supervisor:	Dr. Willem Tromp

Faculty of Civil Engineering and Geosciences
Delft University of Technology
in collaboration with Deltares





Acknowledgements

I am very grateful to everyone who supported me in completing this thesis and my MSc journey.

First, I would like to thank my supervisors for their guidance throughout this project.

I am especially grateful to Dr. Riccardo Taormina, who has been my mentor for more than a year. His insights have helped me grow as a researcher, and I have learned a great deal from him.

I would like to thank Dr. Roberto Bentivoglio for being so generous with his time. I am truly grateful for our many discussions and for the support he gave me throughout the thesis.

I thank Prof. Markus Hrachowitz for being such a kind and approachable professor.

I am thankful to Mr. Ruben Dahm, who acted as a mentor to me. His regular update meetings and discussions kept my work on track and were very helpful.

I would also like to thank Dr. Willem Tromp for the many technical discussions, which helped me solve problems and improve my work.

I thank Deltares for giving me the opportunity to carry out my MSc thesis in collaboration with them. I am grateful for access to the institute and its resources, such as the HPC cluster, which were very valuable for completing this thesis.

I am also grateful to all the professors who taught me at TU Delft. I really enjoyed the lectures and coursework, which gave me the technical skills and knowledge that will help me in my future career.

I thank my friends and colleagues for their support and encouragement throughout my time in Delft.

Finally, I would like to thank my family for their constant support and help in completing my master's degree.

Thank you and kind regards,

Srijan Kumar Jha

Abstract

Pluvial flooding occurs when intense rainfall exceeds the drainage capacity and results in surface runoff. Physics-based flood models simulate it accurately, but they are too slow for early warning or large ensembles of scenarios. This thesis tests whether a multi-scale hydraulic graph neural network (mSWE-GNN) can act as a faster alternative for traditional physics-based flood models while keeping reasonable accuracy.

The mSWE-GNN, originally built for dike-breach flooding, was adapted for pluvial flooding. Rainfall was added as a time-varying input at each node, and the model predicts water depth only. It was trained on synthetic SFINCS simulations. These cover 46 DEMs with and without an added slope (base DEMs and sloped DEMs), under two kinds of rainfall: spatially uniform (SU) and spatially varying (SV). The SV events include stationary and moving storms. For SV rainfall, three rainfall input configurations were compared: current rainfall only, current plus past rainfall, and current plus future rainfall. Depth errors were measured using RMSE, MAE and spatial extent and agreement using CSI and E_{SP}. A simple bucket model with only local rainfall accumulation acted as a baseline. Generalization was tested on unseen domains, unseen rainfall, and both together.

For SU rainfall, the median RMSE ranged from 1.57 to 6.21 cm across splits, compared with 10.49 to 17.37 cm for the bucket model. For SV rainfall, adding future rainfall as input gave the best results, lowering the median RMSE from about 10–12 cm to 3–5 cm. The past-rainfall setup became unstable in training and performed worse than the baseline on RMSE. Performance dropped on unseen conditions and was weakest when both domain and rainfall were unseen. Sloped DEMs performed worse than their base counterparts. On a GPU, the model ran about 17 times faster than SFINCS on a CPU for SV events, and about 12–20 times faster for SU events. However, a preliminary CPU test showed the adapted mSWE-GNN model was slower than SFINCS on CPU.

The adapted mSWE-GNN emulated SFINCS flood dynamics well enough to be a promising fast surrogate. Future work should evaluate the model on realistic terrain, including infiltration, drainage networks, and infrastructure and if available benchmark against observed flood events. This would show whether the adapted mSWE-GNN model can be applied in operational flood forecasting.

Contents

Acknowledgements **i**

Abstract **ii**

List of Figures **vi**

List of Tables **x**

List of Abbreviations **xi**

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Statement 1

1.3 Aim and Research Questions 2

1.4 Report Outline 3

2 Background **4**

2.1 Physics-Based Flood Modelling 4

2.2 Reduced-Physics Based Flood Modelling 5

2.2.1 Super-Fast INundation of CoastS (SFINCS) 5

2.3 Data-Driven Surrogate Models for Flood Prediction 5

2.3.1 Early Machine Learning Approaches 5

2.3.2 Deep Learning for Flood Modelling 6

2.3.3 Graph Neural Networks for Flood Modelling 7

3 Methodology **9**

3.1 Synthetic Terrain and Rainfall 11

3.2 SFINCS Reference Simulations 11

3.3 mSWE-GNN Adaptation for Pluvial Flooding 11

3.3.1 Model Overview . . . 11
3.3.2 Encoders . . . 12
3.3.3 Processor: Hydraulic Message Passing . . . 13
3.3.4 Multiscale U-shaped Architecture . . . 13
3.3.5 Decoder . . . 14
3.4 Model Training and Loss Function . . . 15
3.5 Evaluation Framework . . . 15
3.5.1 Bucket Model . . . 16
3.5.2 Evaluation Metrics . . . 16

4 Synthetic Dataset **18**
4.1 Dataset Overview . . . 18
4.2 Terrain Library . . . 18
4.3 Rainfall Events . . . 19
4.4 Events and Data Splits . . . 21

5 Experimental Design **22**
5.1 Overview of Experiments . . . 22
5.2 Hyperparameter Optimization . . . 22
5.3 Rainfall Context Experiment . . . 23
5.4 Generalization Tests . . . 23
5.5 Runtime Evaluation . . . 23

6 Results and Discussion **25**
6.1 Selected Hyperparameter Configuration . . . 25
6.2 Performance under Spatially Uniform Rainfall . . . 26
6.2.1 Comparison with the Bucket Model . . . 26
6.2.2 Performance on Unseen Domain and Rainfall . . . 27
6.2.3 Effect of Added Slope . . . 27
6.2.4 Errors over Time . . . 29
6.2.5 Flood Extent and Depth Errors . . . 32
6.2.6 Relationship between Elevation and Prediction Errors . . . 33
6.3 Performance under Spatially Varying Rainfall . . . 35
6.3.1 Effect of Rainfall Context and Comparison with the Bucket Model . . . 35
6.3.2 Generalization to Unseen Domains and Rainfall . . . 36

6.3.3 Effect of Added Slope . . . 38
6.3.4 Errors over Time . . . 39
6.3.5 Spatial Errors and Flood Extent . . . 42
6.4 Computational Inference Runtime . . . 44
6.5 Limitations . . . 46

7 Conclusions, Future Work, and Societal Relevance 47
7.1 Conclusions . . . 47
7.1.1 Adaptation of mSWE-GNN for Pluvial Flood Modelling . . . 47
7.1.2 Synthetic SFINCS Dataset . . . 48
7.1.3 Performance of the Adapted Model and Computational Runtime . . . 48
7.1.4 Generalization Across Unseen Domains and Rainfall . . . 49
7.2 Future Work . . . 49
7.3 Societal Relevance . . . 50

References 51

Declaration 54

A Terrain, Rainfall and SFINCS Configuration 55
A.1 Terrain Selection and Slope Construction . . . 55
A.2 Spatially Varying Rainfall Generation . . . 57
A.3 SFINCS Configuration and Input/Output Specifications . . . 58

B Graph Construction, Data Normalization and Training Configuration 60
B.1 Multiscale Graph Construction . . . 60
B.2 Normalization . . . 60
B.3 Temporal Samples and Training Settings . . . 61

C Supplementary Results under Spatially Uniform Rainfall 63

D Supplementary Results under Spatially Varying Rainfall 70

List of Figures

3.1 Overview of the research workflow. 10

3.2 Overview of the three input encoders in the mSWE-GNN pluvial flooding model: the static encoder, the dynamic encoder, and the edge encoder. 12

3.3 Overview of hydraulics-inspired message passing from a source node j to a receiving node i. The message combines learned coefficients, the hydraulic-head gradient, and a wet-dry gate. 13

3.4 U-shaped architecture of the mSWE-GNN model for pluvial flooding. 14

4.1 Example DEMs used in this thesis: (left) base DEM, (middle) DEM with an additional 5 m slope, (right) DEM with an additional 10 m slope. 19

4.2 SU design hyetographs for the six return periods. 19

4.3 SV rainfall intensity at eight times for two events. 20

6.1 Pareto front of validation RMSE versus autoregressive rollout time per event. . . 25

6.2 GNN and bucket model performance under SU rainfall across the evaluation splits: (a) RMSE, (b) MAE, (c) CSI at 0.05 m and (d) CSI at 0.30 m. 27

6.3 Matched base and sloped terrains for unseen domains under RP2 rainfall: (a) RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m. 28

6.4 Matched base and sloped terrains for unseen domains under RP50 rainfall: (a) RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m. 28

6.5 Matched base and sloped terrains for familiar terrains under unseen RP100 rainfall: (a) RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m. 29

6.6 Temporal performance of the base and 10 m sloped versions of DEM 15004 under unseen RP100 rainfall with an unseen domain: (a) rainfall, (b) RMSE, (c) CSI at 0.05 m, (d) CSI at 0.30 m, and (e) domain-average signed error. 30

6.7 Temporal performance of the base and 5 m sloped versions of DEM 511 under unseen RP100 rainfall. Panels follow Figure 6.6. 31

6.8 Flood classification at 0.30 m for DEM 15004 under unseen RP100 rainfall with an unseen domain. The top row shows the base terrain and the bottom row the 10 m sloped terrain. CSI is above each map and the TP, FP, and FN counts are below. 32

6.9 Flood classification at 0.30 m for DEM 511 under unseen RP100 rainfall. The top row shows the base terrain and the bottom row the 5 m sloped terrain. 32

6.10 Mean signed depth error by elevation interval for DEM 41 and its 5 m sloped version at 45 h under RP10 rainfall (validation split). 33

6.11 Mean signed depth error by elevation interval for DEM 15004 and its 10 m sloped version at 45 h under RP10 rainfall (unseen domain). Panels follow Figure 6.10. . 34

6.12 Mean signed depth error by elevation interval for DEM 41 and its 5 m sloped version at 45 h under unseen RP100 rainfall. Panels follow Figure 6.10. 34

6.13 Overall performance under SV rainfall for No rainfall context, Past rainfall context, Future rainfall context and the bucket model: (a) RMSE, (b) CSI at 0.05 m and (c) CSI at 0.30 m. Bars show split means of event-average scores. The validation, unseen-domain, unseen-rainfall and both-unseen splits contain 150, 400, 150 and 80 events, respectively. 35

6.14 Event-level RMSE and CSI under SV rainfall for the No, Future and Past rainfall contexts across the four evaluation splits. 37

6.15 Base and sloped DEM pairs for unseen domains. Columns show event-average RMSE and CSI at 0.05 and 0.30 m. 38

6.16 Base and sloped DEM pairs for both unseen. 39

6.17 Temporal evolution of metrics for DEM 15003 and its 10 m sloped version for both unseen. Panels show domain-average rainfall intensity, spatial RMSE, CSI at 0.05 m and CSI at 0.30 m. 40

6.18 Temporal evolution of metrics for DEM 55 and its 10 m sloped version for unseen rainfall. Panels show domain-average rainfall intensity, spatial RMSE, CSI at 0.05 m and CSI at 0.30 m. 41

6.19 Spatial errors for DEM 55 and its 10 m sloped version under unseen rainfall at 45 h. 42

6.20 Spatial errors for DEM 15003 and its 10 m sloped version under both unseen at 45 h. 43

6.21 Runtime per event for SFINCS on the CPU, the GNN on the CPU and the GNN on the GPU across the SU evaluation splits. Each point is one event, and black lines show the medians. 44

6.22 GPU batching speedup relative to individual GPU inference for the SU splits. . . 45

6.23 Runtime per event for the 150 SV validation events: SFINCS on the CPU, the GNN on the CPU with the No rainfall context, and the GNN on the GPU with the No and Future rainfall contexts. 45

6.24 GPU runtime per event for the No and Future rainfall contexts across the three unseen SV splits. 46

C.1 Training and validation wet-weighted MSE on normalized water depths. 63

C.2 Base and sloped DEMs for unseen domain under RP5 rainfall: (a) event-average RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m. 64

C.3 Base and sloped DEMs for unseen domain under RP10 rainfall: (a) event-average RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m. 64

C.4 Base and sloped DEMs for unseen domain under RP20 rainfall: (a) event-average RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m. 65

C.5 Temporal performance of the base and 10 m sloped versions of DEM 15004 under RP10 rainfall in the unseen domain. 66

C.6 Temporal performance of the base and 10 m sloped versions of DEM 512 under RP100 rainfall in the unseen rainfall. 67

C.7 Mean signed depth error (GNN minus SFINCS) by elevation interval for DEM 15004 and its 10 m sloped version at 45 h under RP2 rainfall in the unseen domain split. 68

C.8 Mean signed depth error (GNN minus SFINCS) by elevation interval for DEM 15004 and its 10 m sloped version at 45 h under RP20 rainfall in the unseen domain split. 68

C.9 Mean signed depth error (GNN minus SFINCS) by elevation interval for DEM 15001 and its 5 m sloped version at 45 h under RP5 rainfall in the unseen domain split. 68

C.10 Mean signed depth error (GNN minus SFINCS) by elevation interval for DEM 15001 and its 5 m sloped version at 45 h under RP50 rainfall in the unseen domain split. 69

C.11 Event-average spatial pattern efficiency (E_{SP}) of the adapted mSWE-GNN under SU rainfall. Higher values indicate better agreement with SFINCS. 69

D.1 Training and validation wet-weighted MSE on normalized water depths for the SV model with No rainfall context. 70

D.2 Training and validation wet-weighted MSE on normalized water depths for the SV model with Past rainfall context. 71

D.3 Training and validation wet-weighted MSE on normalized water depths for the SV model with Future rainfall context. 71

D.4 Event-average RMSE across rainfall return periods in the SV unseen domain split, comparing No rainfall context and Future rainfall context. 72

D.5 Event-average CSI at 0.05 m across rainfall return periods in the SV unseen-domain split, comparing No rainfall context and Future rainfall context. 72

D.6 Event-average CSI at 0.30 m across rainfall return periods in the SV unseen-domain split, comparing No rainfall context and Future rainfall context. 73

D.7 Event-average RMSE across five storm-movement patterns under RP100 rainfall in the SV unseen rainfall split, comparing No rainfall context and Future rainfall context. 73

D.8 Event-average CSI at 0.05 m across five storm-movement patterns under RP100 rainfall in the SV unseen-rainfall split, comparing No rainfall context and Future rainfall context. 74

D.9 Event-average CSI at 0.30 m across five storm-movement patterns under RP100 rainfall in the SV unseen-rainfall split, comparing No rainfall context and Future rainfall context. 74

D.10 Event-average RMSE across five storm-movement patterns under RP100 rainfall in the SV both unseen split, comparing No rainfall context and Future rainfall context. 75

D.11 Event-average CSI at 0.05 m across five storm-movement patterns under RP100 rainfall in the SV both unseen split, comparing No rainfall context and Future rainfall context. 75

D.12 Event-average CSI at 0.30 m across five storm-movement patterns under RP100 rainfall in the SV both unseen split, comparing No rainfall context and Future rainfall context. 76

D.13 Spatial depth errors and flood classification for DEM 55 and its 10 m sloped version at 24 h under stationary RP100 rainfall in the SV unseen rainfall split. . 76

D.14 Spatial depth errors and flood classification for DEM 15003 and its 10 m sloped version at 24 h under stationary RP100 rainfall in the SV both unseen split. . . . 77

D.15 Event-average spatial pattern efficiency (E_{SP}) under SV rainfall for the No, Future and Past rainfall contexts across the evaluation splits. Higher values indicate better agreement with SFINCS. 77

List of Tables

2.1 Momentum terms retained and neglected by the full 2D SWE and its simplified forms. 5

4.1 Composition of the final DEM library. 18

4.2 Peak intensity and total depth of the SU design hyetographs. 19

4.3 Summary of training, validation, and testing dataset splits. 21

5.1 Overview of the experiments. 22

5.2 Dynamic inputs of the SV rainfall context configurations. 23

A.1 Relief of the selected base DEMs. 55

A.2 Selected base DEMs and slope settings for the SU dataset. A dash means that no slope was added. 56

A.3 Selected base DEMs and slope settings for the SV dataset. A dash means that no slope was added. 56

A.4 SFINCS settings 58

A.5 SFINCS input files. 58

A.6 Numerical parameters of the SFINCS model. 59

B.1 Graph sizes for the 6.4 km × 6.4 km domains. 60

B.2 Training settings 62

List of Abbreviations

SWE	Shallow Water Equations
SFINCS	Super-Fast INundation of CoastS
DL	Deep Learning
GNN	Graph Neural Network
mSWE-GNN	Multi-Scale Hydraulic Graph Neural Network
LIE	Local Inertial Equations
DWE	Diffusive Wave Equations
KWE	Kinematic Wave Equations
ANNs	Artificial Neural Networks
SVM	Support Vector Machines
CNN	Convolutional Neural Network
ConvLSTM	Convolutional Long Short-Term Memory
cGAN	Conditional Generative Adversarial Networks
RNN	Recurrent Neural Network
DEM	Digital Elevation Model
SU	Spatially Uniform
SV	Spatially Varying
MLP	Multi Layer Perceptron
GPU	Graphics Processing Unit
CPU	Central Processing Unit

Chapter 1

Introduction

1.1 Motivation

Pluvial flooding occurs when intense rainfall exceeds the capacity of drainage systems, leading to a high volume of surface runoff [1]. It causes damage to public infrastructure, homes, and businesses, and it disrupts public services such as transportation and other economic activities [2]. In developing countries, pluvial flooding may also lead to waterborne epidemics [3]. Urban areas are more vulnerable as they have reduced infiltration due to high imperviousness resulting in increased runoff [1].

Due to global warming, extreme precipitation is expected to become more intense and frequent. Together with urban expansion, which increases impervious areas, this will make pluvial flooding more severe and frequent [3]. Pluvial flooding can also occur far from any water body, and because there is no visible water source nearby, people often underestimate the risk and have a low perception of threat [1]. Despite these risks, pluvial flooding has received less attention than fluvial and coastal flooding. Many national and regional authorities still lack pluvial flood hazard and risk maps [1].

Flood models help assess this hazard by predicting water depth dynamics and flooding extent. These models assist in identifying vulnerable areas, appraisal of mitigation measures and preparing for flood events [4]. These models must be able to run multiple rainfall scenarios and terrain configurations, and therefore need to be not only accurate but also fast. However, traditional numerical flood models become computationally expensive and slow for larger domains, finer resolutions, or when running many scenarios [4]. This motivates investigating alternative modelling approaches that can simulate pluvial flood dynamics accurately and fast enough for many scenarios.

1.2 Problem Statement

Flood models based on the two-dimensional Shallow Water Equations (SWE), such as Delft3D and SOBEK, predict flood dynamics accurately. However, they are computationally intensive and slow [2], which can limit their use in early-warning systems and ensemble modelling, particularly

for large domains or fine spatial resolutions. Reduced-physics solvers such as SFINCS simplify the SWE to lower computational cost while preserving reasonable accuracy [5]. They make running simulations over larger domains feasible but speed remains a bottleneck when running ensembles or multiple scenarios. Deep learning (DL) surrogates have emerged as a promising alternative because they can provide substantial speed-ups compared with numerical models. They have simulated flood dynamics effectively for different flood types, such as fluvial, coastal, and pluvial flooding [4, 6].

In addition to faster inference times, these models are differentiable. This allows gradients with respect to model variables or parameters to be calculated accurately and efficiently, which enables the discovery of high-dimensional unknown relationships [7]. These gradients also show how changes in input affect the output, and they can be used in inverse optimization. Despite these advantages, DL flood models face several limitations. The most prominent limitation is the inability to generalize to unseen terrain, i.e. terrain not seen during training [4]. Such models need to be retrained for every new area, which undermines their speed advantage. Generalization to unseen terrain would allow application in data-scarce regions, where limited or no data is available for training. Other limitations of DL models are limited interpretability and physical consistency.

Graph neural networks (GNNs) represent spatial relationships through connected nodes and edges and can work with arbitrarily structured data. They show potential of generalizing across different terrain [4]. mSWE-GNN is a GNN-based model that exploits this ability to generalize to new case studies [8]. It is a multi-scale hydraulic GNN that models flooding at different resolutions and propagation speeds [8]. Its message-passing rule is inspired by finite-volume flux exchanges between neighbouring cells, introducing hydraulic principles into the learned interactions [9]. With only one fine-tuning simulation, the model generalized well to a realistic case study, Dike Ring 15 in the Netherlands. This case study had a larger area and wider range of boundary conditions than the training data [8].

Three research gaps remain. First, mSWE-GNN has only been applied to dike-breach flooding, and its application to pluvial flooding has not been investigated. Second, the large speed-ups reported for mSWE-GNN were measured against Delft3D-FM [8], so the speed-up over a reduced-physics solver such as SFINCS has not been quantified. Third, most DL surrogates for pluvial flooding use spatially uniform rainfall [10], whereas spatially varying rainfall is more realistic and closer to operational conditions.

This thesis addresses these gaps by adapting mSWE-GNN to pluvial flooding. It evaluates accuracy, inference runtime and generalization under spatially uniform and spatially varying rainfall. For varying rainfall, it also tests whether providing past or future rainfall context improves predictions compared with no rainfall context.

1.3 Aim and Research Questions

The aim of this thesis is to adapt the dike breach mSWE-GNN model for pluvial flood modelling and to evaluate its ability to reproduce SFINCS-simulated water depth dynamics and flood

extent. The evaluation considers spatially uniform and spatially varying rainfall, generalization to unseen terrain and rainfall, and computational performance relative to SFINCS.

The following research questions support this aim:

1. What adaptations are needed to make mSWE-GNN suitable for pluvial flood modelling?
2. How can a synthetic SFINCS dataset be created for training, validation, and testing the adapted model under spatially uniform and spatially varying rainfall?
3. How accurately does the adapted model reproduce SFINCS-simulated water depth dynamics and flood extent under spatially uniform and spatially varying rainfall, and how does its inference runtime compare with SFINCS?
4. How well does the adapted model generalize to unseen terrain, unseen rainfall, and their combination?

1.4 Report Outline

Chapter 2 provides background on numerical flood models, deep learning surrogates, and graph neural networks. Chapter 3 describes the adapted mSWE-GNN architecture, training procedure, and evaluation metrics. Chapter 4 presents a description of the synthetic terrain, rainfall events, and dataset splits. Chapter 5 explains the design of experiments used in this thesis. Chapter 6 presents and discusses the results and limitations. Finally, Chapter 7 provides the conclusions and directions for future work.

Chapter 2

Background

This chapter provides background on numerical flood models, deep learning surrogates for flood modelling, and graph neural networks for flood modelling.

2.1 Physics-Based Flood Modelling

The most prominent physics-based models for pluvial flood modelling are two-dimensional (2D) hydrodynamic models. These models simulate flood dynamics in the x and y directions [2]. They discretize the terrain elevation into a computational grid and apply rainfall as a forcing term to every grid cell. Using this forcing, the models predict the water depth and velocity in each grid cell at every time step [5].

2D hydrodynamic models use the shallow water equations (SWE) as their governing equations. The SWE are obtained by simplifying the Navier-Stokes equations in the vertical direction. The SWE consist of the conservation of mass and the conservation of momentum equations [2].

$$\underbrace{\frac{\partial h}{\partial t}}_{\text{depth change}} + \underbrace{\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y}}_{\text{net horizontal outflow}} = \underbrace{R - I}_{\text{net water input}} , \tag{2.1a}$$

$$\underbrace{\frac{\partial q_x}{\partial t}}_{\text{local acceleration}} + \underbrace{\frac{\partial (u q_x)}{\partial x} + \frac{\partial (v q_x)}{\partial y}}_{\text{advective acceleration}} = \underbrace{S_{bx}}_{\text{pressure and bed gradients}} + \underbrace{S_{fx}}_{\text{friction}} , \tag{2.1b}$$

$$\underbrace{\frac{\partial q_y}{\partial t}}_{\text{local acceleration}} + \underbrace{\frac{\partial (u q_y)}{\partial x} + \frac{\partial (v q_y)}{\partial y}}_{\text{advective acceleration}} = \underbrace{S_{by}}_{\text{pressure and bed gradients}} + \underbrace{S_{fy}}_{\text{friction}} . \tag{2.1c}$$

Equation (2.1a) expresses conservation of mass, and Equations (2.1b) and (2.1c) express conservation of momentum in the x and y directions respectively. Solving the SWE is computationally expensive and slow for large domains. Full 2D SWE models are therefore impractical for early warning systems and for running ensembles.

2.2 Reduced-Physics Based Flood Modelling

Full SWE models are computationally expensive. Simplified forms of the SWE reduce this cost by neglecting some terms of the momentum equations (Equations (2.1b) and (2.1c)). These equations contain four terms: local acceleration, advective acceleration, pressure and bed gradients, and friction. The kinematic wave equations (KWE), the diffusive wave equations (DWE) and the local inertial equations (LIE) each keep some of these terms and neglect the others. Table 2.1 summarises which terms each form retains and which it neglects.

Table 2.1: Momentum terms retained and neglected by the full 2D SWE and its simplified forms.

Equations	**Terms retained**	**Terms neglected**
Full 2D SWE	Local acceleration, advective acceleration, pressure and bed gradients, friction	None
LIE	Local acceleration, pressure and bed gradients, friction	Advective acceleration
DWE	Pressure and bed gradients, friction	Local acceleration, advective acceleration
KWE	Bed gradient, friction	Local acceleration, advective acceleration, pressure gradient

Models based on the DWE or LIE are known as reduced-physics models [5]. They reduce computational cost, although running large ensembles can still be demanding. This motivates investigating alternative approaches that could provide further speed-ups.

2.2.1 Super-Fast INundation of CoastS (SFINCS)

SFINCS is the reduced-physics solver used in this thesis. It is designed for compound flooding and solves a simplified form of the SWE on a regular grid. SFINCS offers two solver modes: SFINCS-LIE and SFINCS-SSWE. SFINCS-LIE is based on the LIE and neglects advective acceleration. SFINCS-SSWE extends the LIE formulation by adding advective acceleration [5]. This thesis uses SFINCS-SSWE, with advection enabled, to generate the simulations for training and testing the mSWE-GNN for pluvial flood modelling. The SFINCS model setup is described in Section 3.2 and Appendix A.3.

2.3 Data-Driven Surrogate Models for Flood Prediction

2.3.1 Early Machine Learning Approaches

This section describes early machine learning (ML) approaches applied to flood modelling. These approaches provided significant speed-ups and lower computational cost than numerical models.

Liu and Pender [11] developed linear regression and ν-support vector regression (ν-SVR) models to predict water depth and velocity series. They trained the models on fine-grid simulations. The models reached reasonable accuracy at substantially lower prediction times. Bermúdez et al. [12] developed a least-squares support vector machine (SVM) to predict maximum water depths and velocities quickly. The model learned the relationships between upstream discharge, downstream water levels and flood conditions at individual locations.

Artificial neural networks (ANNs) were also widely used as flood surrogates. Berkhahn et al. [13] used an ANN ensemble to predict maximum water depth maps rapidly during urban pluvial floods. The model took rainfall time series as input. However, the study was limited to spatially uniform, synthetic rainfall events, and the model could not predict the temporal evolution of flooding. Some early ML models could predict this evolution. Zahura et al. [14] used a random forest surrogate that learned relationships between topographic and environmental features to predict hourly water depths.

These studies demonstrated that early ML models can support flood modelling. However, several of them predicted only a single hydraulic variable and did not capture the spatio-temporal evolution of flooding. They also gave limited attention to transferability to unseen domains.

Traditional ML models often struggle with high-dimensional feature representation. Deep learning (DL) models offer another approach to flood prediction by learning spatial and temporal patterns from the input data. They extract complex spatio-temporal patterns automatically from large, multi-dimensional data on rainfall, land surface characteristics and hydraulic responses, which improves prediction accuracy [10]. Section 2.3.2 reviews these DL models.

2.3.2 Deep Learning for Flood Modelling

This section reviews DL surrogate models for flood prediction. It focuses on their accuracy, their speed and their ability to generalize to unseen terrain.

Kabir et al. [15] developed a deep convolutional neural network (CNN) for rapid prediction of fluvial flooding. The CNN emulated LISFLOOD-FP water depths with reasonable accuracy and much greater speed. It also outperformed SVR, a traditional ML approach. However, the study did not evaluate generalization to unseen terrain.

Two later studies addressed urban pluvial flooding. Löwe et al. [16] adapted the image segmentation architecture U-Net to predict 2D maximum water depths. They used geodata and rainfall as inputs. The authors evaluated it on natural rainfall events and on locations excluded from training. Testing at unseen locations was a key contribution of the study. However, the model did not account for spatial variability in rainfall. Liao et al. [10] addressed this variability with a Convolutional Long Short-Term Memory (ConvLSTM) model. Compared with a physics-based model, it predicted the evolution of inundation under spatially and temporally varying rainstorms with reasonable accuracy and substantial speed-ups. However, errors increased under extreme rainfall, and the authors did not investigate generalization to unseen areas.

Other studies focused on generalization. do Lago et al. [17] developed cGANs-Flood, based on conditional generative adversarial networks (cGANs), to generalize predictions across catchments

and boundary conditions. The model predicted maximum water depths successfully in testing areas. It also enforced catchment-scale mass conservation. However, it tended to underestimate flood depths in channels for small rainfall events. Cache et al. [6] developed a model that incorporates contextual terrain information, motivated by the limited generalizability of data-driven flood models. The model combined CNNs for terrain feature extraction, an attention mechanism linking local and surrounding terrain, and a recurrent neural network (RNN) for rainfall time series. It reproduced pluvial flood maps well for unseen rainfall events in the training domain. On unseen terrain, it broadly identified areas of water accumulation. However, the model still had overall poor generalization.

Two limitations therefore remain. First, generalization to unseen terrain is not consistently demonstrated. Second, the grid-based architectures reviewed here are not directly applicable to unstructured meshes. Graph neural networks (GNNs) are a potential way to address these limitations. Section 2.3.3 discusses them.

2.3.3 Graph Neural Networks for Flood Modelling

A graph $G = (V, E)$ consists of nodes V and edges E that connect them, and both can carry features. Graph neural networks (GNNs) typically encode these features as learned vectors, called embeddings, and update them through message passing [18]. In each step, a node gathers information from its connected neighbours, combines it, and uses it to update its own embedding. In flood modelling, computational cells are represented as nodes, neighbouring cells are connected by edges, and the features describe terrain and hydraulic conditions. Because GNNs can work with unstructured data, Bentivoglio et al. [4] highlighted their potential to generalize across case studies and to represent interactions between the natural and built environment.

Kazadi et al. [19] developed FloodGNN-GRU, which combines a GNN with a gated recurrent unit (GRU) to predict water depths and velocities using rainfall, terrain and previous hydraulic states as input. This model was trained and evaluated using LISFLOOD-FP simulation of Hurricane Harvey. The model achieved substantial speed-ups (approximately 1000×) with reasonable accuracy. FloodGNN-GRU was also tested on unseen subregions of the same event and domain. However, the model lacked explicit physical constraints. Acosta et al. [20] incorporated physical constraints in DUALFloodGNN through loss terms that promote mass conservation. Although, the study did not evaluate the model's generalization to an unseen case study.

Bentivoglio et al. [9] developed SWE-GNN, which contains message passing inspired by the exchange of fluxes between neighbouring finite-volume cells. It predicts water depth and unit discharge using terrain properties and previous hydraulic states. It generalized to unseen synthetic topographies, breach locations, larger domains and longer simulations, and it was substantially faster than Delft3D-FM. However, it used constant breach inflow, needed initialization by a numerical solver, and did not strictly enforce conservation laws.

Bentivoglio et al. [8] extended SWE-GNN into the multiscale mSWE-GNN, which passes information through progressively coarser graphs before returning to the finest resolution. This lets information travel farther with fewer message-passing steps. It also handles time-varying boundary conditions and generalized to unseen irregular meshes, topographies and inflow hydrographs.

Bentivoglio et al. [8] also demonstrated its generalization to an unseen realistic case study.

The mSWE-GNN study focused on dike-breach flooding [8]. Its performance under spatially uniform and spatially varying rainfall was therefore not tested in that study. Its speed-ups were also measured against Delft3D-FM, leaving the advantage over an efficient reduced-physics solver such as SFINCS unestablished. This thesis addresses these gaps by adapting mSWE-GNN for pluvial flooding and evaluating its accuracy, generalization and inference runtime against SFINCS.

Chapter 3

Methodology

This chapter describes the methods used to adapt and evaluate mSWE-GNN for pluvial flood modelling. Figure 3.1 summarizes the workflow, from dataset creation, to generating SFINCS reference simulations, graph construction, model training, and evaluation framework.

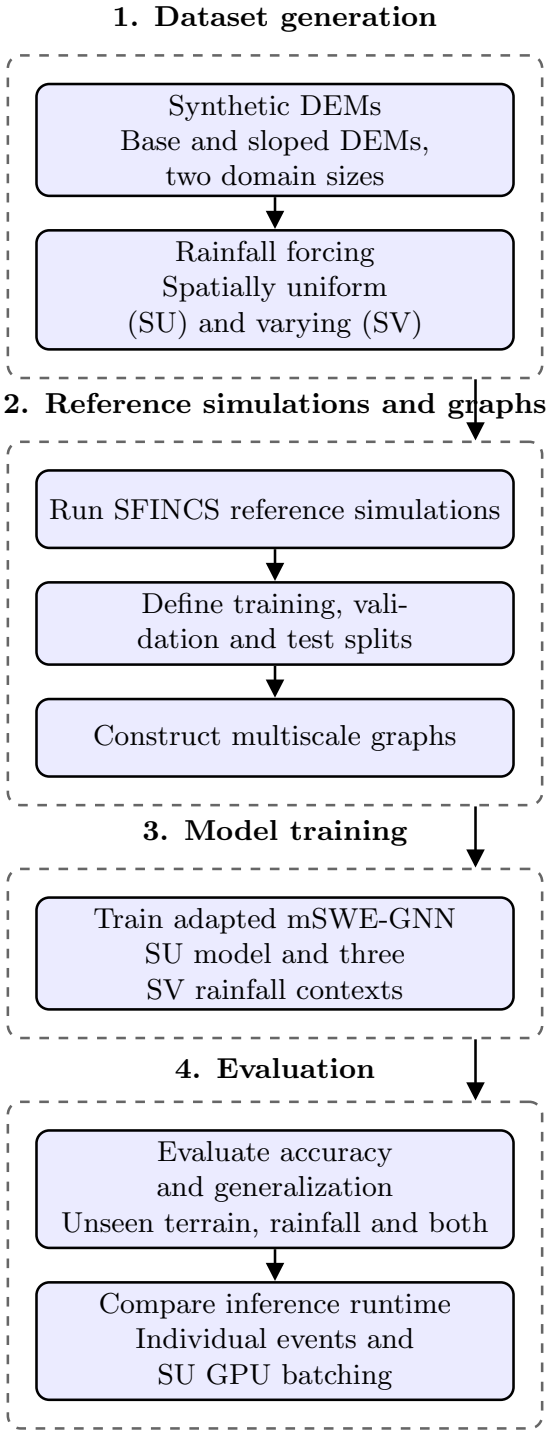


Figure 3.1: Overview of the research workflow.

3.1 Synthetic Terrain and Rainfall

Synthetic terrains and rainfall events are used to generate reference numerical simulations. The terrains are Perlin-noise-generated digital elevation models (DEMs) developed by Bentivoglio et al. [9]. The DEMs have two domain sizes: 6.4 km × 6.4 km and 12.8 km × 12.8 km. Both have cell size of 100 m × 100 m. These terrains are relatively flat and contain no built infrastructure or drainage networks. They are referred to as base DEMs in this thesis.

To help the model learn flow from uphill to downhill, a slope is added to some base DEMs in a random direction. The added elevation difference across the domain is either 5 m or 10 m. These modified DEMs are referred to as sloped DEMs.

Two rainfall types are used: spatially uniform (SU) and spatially varying (SV). SU rainfall applies the same intensity to every cell, and this intensity changes only over time. The SU events use design hyetographs for the city of Barcelona, provided by Deltares, with return periods of 2, 5, 10, 20, 50, and 100 years. Each hyetograph has fifteen 3-hour blocks, giving a total duration of 45 hours. The intensity is constant within each block.

SV rainfall combines these hyetographs with a Gaussian spatial field, following an approach adapted from Liao et al. [10]. Rainfall is highest near the center of the field and decreases with distance. Five storm movements are considered: stationary, left-to-right, right-to-left, bottom-to-top, and top-to-bottom. At each time step, the Gaussian field is divided by its spatial mean and then multiplied by the corresponding SU intensity. This keeps the domain-averaged intensity equal to that of the SU hyetograph. Further details about the DEMs and rainfall are provided in Chapter 4.

3.2 SFINCS Reference Simulations

SFINCS is used to generate the reference simulations for this thesis. The SFINCS-SSWE configuration is used with advection enabled. Rainfall is applied directly to the computational grid, and SFINCS simulates the water depth evolution. Each simulation starts with a dry domain and runs for 45 hours. Water-depth maps are stored every 30 minutes, giving 91 frames including the initial state.

A uniform Manning roughness coefficient of $0.04\,\mathrm{s\,m^{-1/3}}$ is used. All cells are active, and the domain boundaries are closed. Infiltration, evaporation, drainage networks, groundwater exchange, and built infrastructure are excluded. The detailed numerical settings and input-file specifications are provided in Appendix A.3.

3.3 mSWE-GNN Adaptation for Pluvial Flooding

3.3.1 Model Overview

The multi-scale hydraulic graph neural network (mSWE-GNN) developed by Bentivoglio et al. [8] represents flood dynamics through message passing on graphs at multiple spatial resolutions.

Fine graphs capture local detail, whereas coarser graphs enable information to travel over longer distances in a single message-passing step. The original model was developed for dike-breach flooding and predicts water depth and unit discharge.

This thesis retains the encoder-processor-decoder architecture and multiscale structure of mSWE-GNN. The model is adapted for pluvial flooding by introducing rainfall as an input and predicting water depth only. At each time step, the model uses terrain elevation, graph connectivity, previous water-depth fields, and rainfall to predict the water-depth field at the next time step.

The following subsections describe the input encoders, hydraulic message-passing processor, multiscale structure, and water-depth decoder. The main adaptations are the inclusion of rainfall in the dynamic encoder and modifications to hydraulic message passing. Water redistribution is based on hydraulic gradients, providing a more explicit hydraulic basis for message passing. A wet/dry gate suppresses messages along edges where the combined water depth of the two connected nodes is below a specified threshold.

3.3.2 Encoders

Encoders transform input into learned representations, referred to as embeddings. These embeddings are used during message passing. In mSWE-GNN, each encoder is implemented using a multilayer perceptron (MLP).

The model uses three encoders: a static encoder, a dynamic encoder, and an edge encoder, as illustrated in Figure 3.2. The static encoder receives terrain elevation as input. The dynamic encoder receives the three previous water-depth fields and current rainfall intensity as input. The edge encoder receives the distance between two connected nodes as input and produces an edge embedding. The original mSWE-GNN model by Bentivoglio et al. [8] can use Manning's roughness as an input, whereas the current adaptation does not.

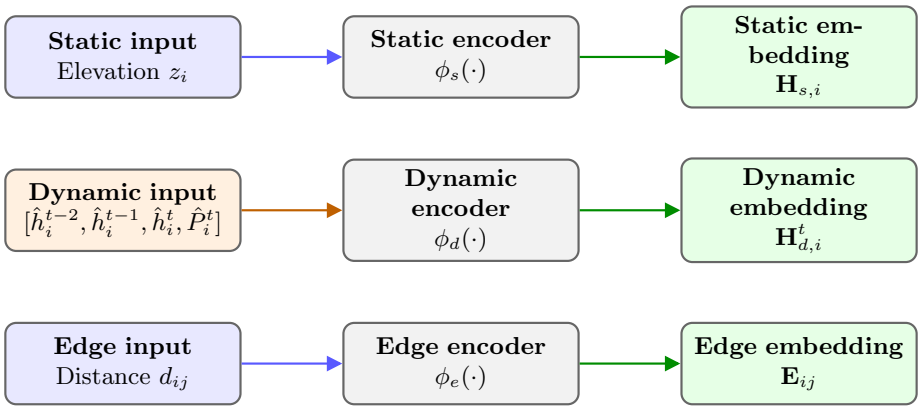


Figure 3.2: Overview of the three input encoders in the mSWE-GNN pluvial flooding model: the static encoder, the dynamic encoder, and the edge encoder.

3.3.3 Processor: Hydraulic Message Passing

The hydraulic processor exchanges information between neighbouring nodes after the encoders transform the inputs into embeddings. Each node represents a terrain grid cell. For a directed edge from source node j to receiving node i, the processor calculates a message using information from both nodes and their connecting edge.

Before calculating the message, rainfall intensity is converted into rainfall depth over the prediction interval and added to the current water depth. The hydraulic head is then calculated as terrain elevation plus rainfall adjusted water depth. The difference in hydraulic head between neighbouring nodes, divided by their distance, gives the hydraulic-head gradient.

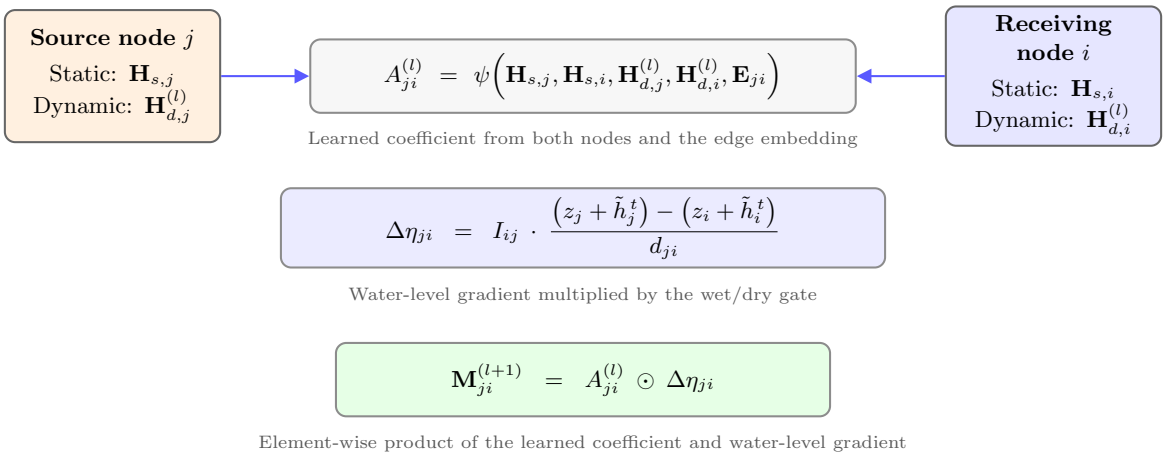


Figure 3.3: Overview of hydraulics-inspired message passing from a source node j to a receiving node i. The message combines learned coefficients, the hydraulic-head gradient, and a wet-dry gate.

An MLP uses the static and dynamic embeddings of both nodes, together with the edge embedding, to calculate learned coefficients for each directed edge. A wet-dry gate controls whether a message is passed along an edge. When the combined water depth of the two connected nodes is below a specified threshold, the message is set to zero. Otherwise, the message is calculated by multiplying the learned coefficients by the hydraulic-head gradient. Figure 3.3 illustrates the message calculation from source node j to receiving node i. It shows how the learned coefficients, hydraulic-head gradient, and wet-dry gate together determine the message passed along an edge.

Each node sums the messages received from its neighbours. The summed message is combined with the node's current dynamic embedding to update that embedding. The hydraulic processor repeats this process through several message-passing blocks. The decoder then converts the updated embeddings into a water-depth prediction, as described in Section 3.3.5.

3.3.4 Multiscale U-shaped Architecture

The adapted mSWE-GNN follows the multiscale U-shaped architecture proposed by Bentivoglio et al. [8]. As shown in Figure 3.4, the model processes the same domain across three graph resolutions: fine, medium, and coarse.

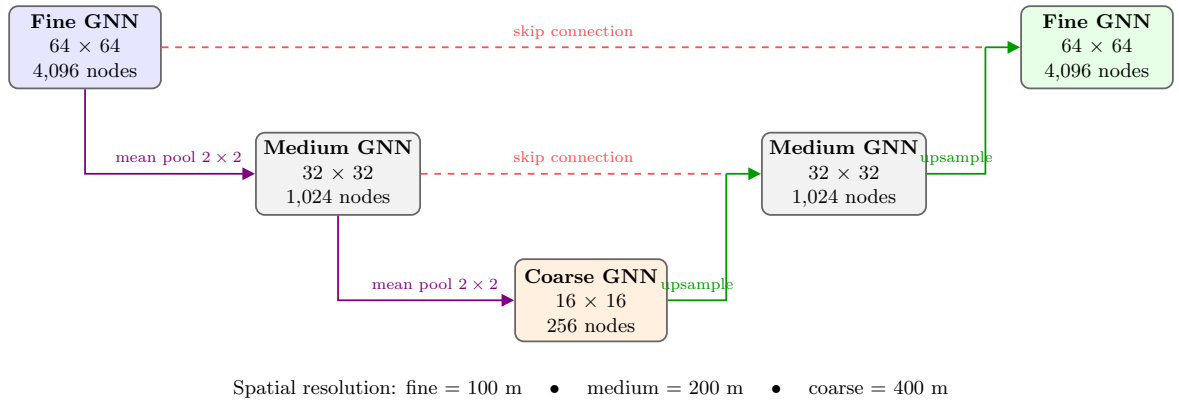


Figure 3.4: U-shaped architecture of the mSWE-GNN model for pluvial flooding.

The fine graph provides the most detailed representation of the domain. The medium and coarse graphs represent the same domain with fewer nodes and larger distances between neighbouring nodes. In this study, neighbouring nodes are separated by 100 m in the fine graph, 200 m in the medium graph, and 400 m in the coarse graph. Message passing at coarser resolutions therefore allows learned information to travel over larger physical distances.

The model transfers information from a finer graph to a coarser graph through 2×2 mean pooling. Each parent node represents four child nodes from the finer graph. The `fine_to_medium` and `medium_to_coarse` maps define these parent-child relationships.

During pooling, the model averages the dynamic embeddings and rainfall adjusted water depths of child nodes to obtain values at the coarser resolution. It then calculates the hydraulic head at the coarser resolution and performs message passing. Mean pooling smooths the dynamic embeddings and can remove local detail. To retain this detail, the model saves the dynamic embeddings at the fine and medium resolutions before pooling.

After processing information at the coarser scale, the model transfers it from the coarse graph to the medium graph and then from the medium graph to the fine graph through upsampling. At each stage, the embedding of a parent node is copied to its child nodes. The copied embedding is combined with the dynamic embedding saved before pooling through a skip connection. A fusion MLP then combines the two embeddings, allowing the model to retain local detail from the finer graph.

3.3.5 Decoder

The decoder predicts water depth at the next time step from the processed fine-scale embedding. It first calculates a correction using an MLP. This correction is added to a baseline prediction based on the three input water-depth fields and rainfall during the prediction interval. The final prediction is given by:

$$\hat{\bar{h}}_{i,t+1} = \mathrm{ReLU}\left(w_1 \bar{h}_{i,t-2} + w_2 \bar{h}_{i,t-1} + w_3 \bar{h}_{i,t} + \Delta \bar{h}_{i,t}^{\mathrm{rain}} + f_{\mathrm{dec}}\left(\mathbf{d}_{i,t}^{\mathrm{final}}\right)\right)$$

where $\hat{\bar{h}}_{i,t+1}$ is the predicted normalized water depth at node i in the next time step. The terms $\bar{h}_{i,t-2}$, $\bar{h}_{i,t-1}$, and $\bar{h}_{i,t}$ are the three input normalized water depths. The weights w_1, w_2, and w_3 determine the contribution of each input depth to the baseline prediction. The term $\Delta\bar{h}_{i,t}^{\text{rain}}$ is the normalized rainfall depth added during the prediction interval. The term $f_{\text{dec}}(\mathbf{d}_{i,t}^{\text{final}})$ is the correction predicted by the decoder from the final fine-scale embedding. The ReLU function sets negative predicted water depths to zero.

3.4 Model Training and Loss Function

Curriculum learning is used for training the model as it improves rollout stability[21]. Training starts with one-step prediction, and the training horizon H then increases gradually to six steps, following Bentivoglio et al. [8]. After each epoch, the model is validated on six-step autoregressive predictions, regardless of the current training horizon. The checkpoint with the lowest validation RMSE is retained. Appendix B.3 reports the optimizer settings and the curriculum schedule.

The adapted mSWE-GNN uses a wet-weighted mean squared error loss, adapted from D'Costa et al. [22], for training. Flood simulations often contain many dry cells, which can be easier to predict than wet cells. An unweighted loss could therefore give a low error mainly because of the dry cells. A wet-weighted loss counters this imbalance. This thesis uses a weight of seven for cells with a reference depth of at least 0.05 m and a weight of one for all other cells. This gives greater emphasis to errors in wet cells during training. The loss is calculated on normalized depths at the finest graph scale:

$$\mathcal{L} = \frac{1}{NH} \sum_{k=1}^{H} \sum_{i=1}^{N} w_{i,t+k} \left(\hat{\tilde{h}}_{i,t+k} - \tilde{h}_{i,t+k} \right)^2, \tag{3.1}$$

with

$$w_{i,t+k} = \begin{cases} 7, & h_{i,t+k} \geq 0.05\,\text{m}, \\ 1, & \text{otherwise.} \end{cases} \tag{3.2}$$

Here, N is the number of fine-scale nodes, H is the training prediction horizon, and $\hat{\tilde{h}}$ and $\tilde{h}$ are the predicted and reference normalized depths. The weights are based on the reference depth h in meters.

3.5 Evaluation Framework

The performance of the adapted mSWE-GNN model is evaluated to measure: error in water depth, agreement in flood extent, and agreement in spatial pattern. The SFINCS reference simulations serve as the benchmark.

Each SFINCS simulation stores 91 frames, including the initial state at 0 h. The model needs three previous water-depth fields as input. The SFINCS water depths at 0, 0.5, and 1 h are used as the initial input in training. After initialization, each predicted depth field becomes the input for the next prediction, so the model predicts the full event autoregressively. Therefore,

evaluation starts at the first prediction at 1.5 h and covers 88 frames up to 45 h.

3.5.1 Bucket Model

The adapted mSWE-GNN model is compared with a bucket model with only local ponding abilities. In the bucket model, rainfall accumulates in each cell with no lateral water movement and no water losses.

The bucket model starts from a dry domain. Its depth is updated as:

$$h_{i,t+1}^{\text{bucket}} = h_{i,t}^{\text{bucket}} + \frac{P_{i,t}\,\Delta t}{1000}, \tag{3.3}$$

where $P_{i,t}$ is the rainfall intensity in mm/h and Δt is the prediction interval in hours. The factor 1000 converts millimeters to meters.

3.5.2 Evaluation Metrics

The evaluation uses four metrics. For each predicted time frame the metric is calculated over all grid cells. Root mean square error (RMSE) and mean absolute error (MAE) measures depth errors, critical success index (CSI) measures agreement in flood extent, and spatial pattern agreement is measured using spatial pattern efficiency (E_{SP}) metric defined by Dembélé et al. [23].

$$\text{RMSE}_t = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{h}_{i,t} - h_{i,t}\right)^2}, \qquad \text{MAE}_t = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{h}_{i,t} - h_{i,t}\right|, \tag{3.4}$$

For RMSE and MAE, $\hat{h}_{i,t}$ and $h_{i,t}$ are the predicted and reference depths. Both metrics are in meters, and lower values indicate better agreement. RMSE gives more weight to large errors, whereas MAE is the average absolute error.

Flood-extent agreement is measured with CSI at thresholds of 0.05 m and 0.30 m. A cell is classified as flooded when its depth is at least the selected threshold. CSI is:

$$\text{CSI}_t = \frac{\text{TP}_t}{\text{TP}_t + \text{FP}_t + \text{FN}_t}. \tag{3.5}$$

True positives (TP) are cells that are flooded in both the reference and the prediction. False positives (FP) are cells that are predicted as flooded but are not flooded in the reference. False negatives (FN) are cells that are flooded in the reference but are not predicted as flooded. CSI ranges from 0 to 1, and 1 indicates perfect agreement.

E_{SP} combines three terms: the rank agreement of cell depths, the relative spatial variability, and the agreement of standardized depth patterns [23]. It is calculated as:

$$E_{\text{SP}} = 1 - \sqrt{(r_s - 1)^2 + (\gamma - 1)^2 + (\alpha - 1)^2}, \tag{3.6}$$

where r_s is the Spearman rank correlation between predicted and reference cell depths. The

variability ratio γ and the spatial-location term α are:

$$\gamma = \frac{\sigma_{\hat{h}}/\mu_{\hat{h}}}{\sigma_h/\mu_h}, \qquad \alpha = 1 - \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\hat{h}_i - \mu_{\hat{h}}}{\sigma_{\hat{h}}} - \frac{h_i - \mu_h}{\sigma_h}\right)^2}. \tag{3.7}$$

Here, μ and σ are the mean and standard deviation. Higher E_{SP} indicates better agreement. The maximum value is 1, and negative values are possible.

Chapter 4

Synthetic Dataset

4.1 Dataset Overview

This chapter describes the synthetic dataset used to train, validate, and test the adapted mSWE-GNN. It also describes the terrains, rainfall events and dataset splits. The SU and SV datasets each contain 46 digital elevation models (DEMs): 30 base DEMs and 16 sloped DEMs. The two datasets use different selections of smaller base DEMs and different sloped DEMs, while sharing the same ten larger base DEMs. Combining these terrains with the rainfall events gives 276 SU events and 1,380 SV events.

4.2 Terrain Library

The original terrain library contains 131 base DEMs: 121 DEMs of 6.4 km × 6.4 km and 10 DEMs of 12.8 km × 12.8 km [9]. All DEMs use a cell size of 100 m × 100 m, which gives grids of 64 × 64 and 128 × 128 cells [9]. Relief is defined as the difference between the maximum and minimum elevation of a DEM. The average relief is about 4.23 m for the smaller DEMs and about 3.78 m for the larger DEMs. A subset of 20 smaller base DEMs is used for this thesis. All ten larger base DEMs are retained.

Sloped DEMs are created from selected base DEMs, as described in Chapter 3. Ten of the smaller base DEMs are modified, five with an added elevation difference of 5 m and five with 10 m. Six of the larger base DEMs are modified, three with each value. Table 4.1 gives the composition of the terrain library. It contains 46 DEMs: 30 base DEMs and 16 sloped DEMs. Figure 4.1 shows one example of each DEM type.

Table 4.1: Composition of the final DEM library.

Domain size	Base	+5 m slope	+10 m slope	Total
6.4 km × 6.4 km	20	5	5	30
12.8 km × 12.8 km	10	3	3	16
Total	30	8	8	46

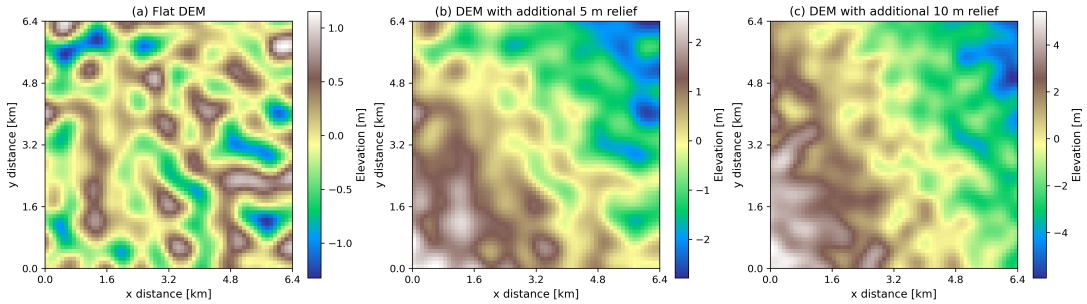


Figure 4.1: Example DEMs used in this thesis: (left) base DEM, (middle) DEM with an additional 5 m slope, (right) DEM with an additional 10 m slope.

4.3 Rainfall Events

The SU rainfall uses six design hyetographs for the city of Barcelona with return periods of 2, 5, 10, 20, 50, and 100 years. Each hyetograph has fifteen 3-hour blocks, which gives a duration of 45 hours. Table 4.2 gives the peak intensity and total rainfall depth of each hyetograph, and Figure 4.2 shows the hyetographs.

Table 4.2: Peak intensity and total depth of the SU design hyetographs.

Return period (years)	Peak intensity (mm/h)	Total depth (mm)
2	8.27	84.1
5	9.80	101.3
10	11.11	116.5
20	12.58	133.9
50	14.79	160.8
100	16.69	185.0

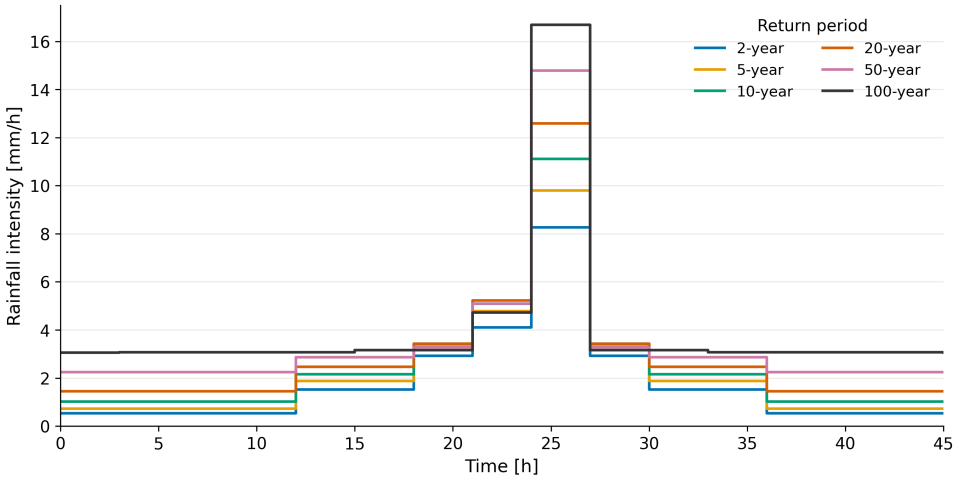
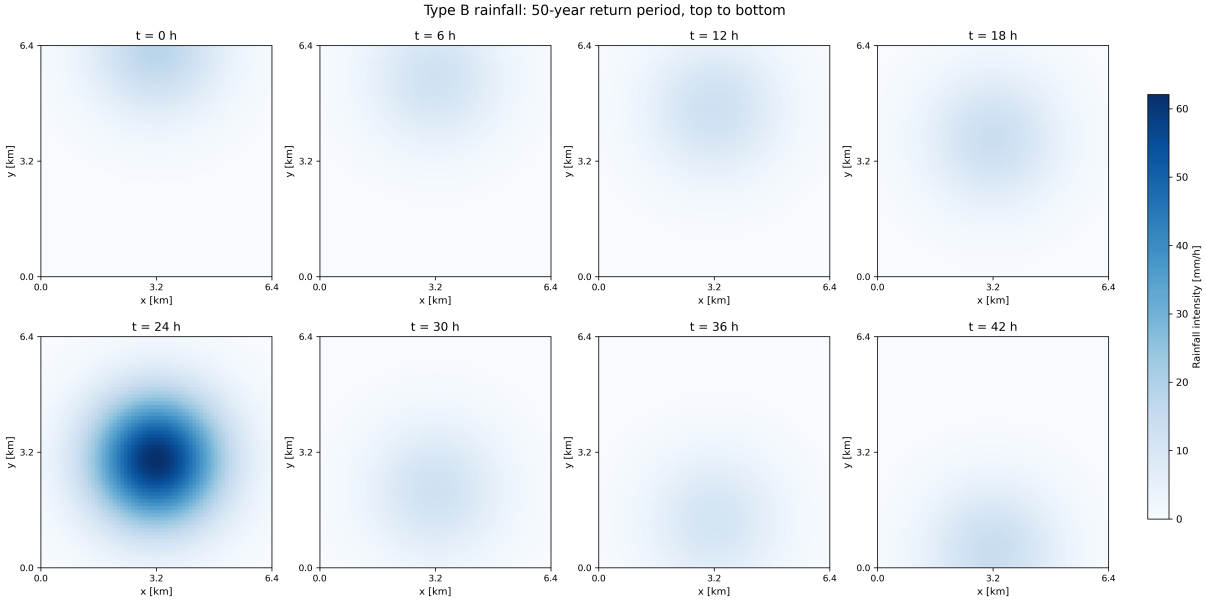
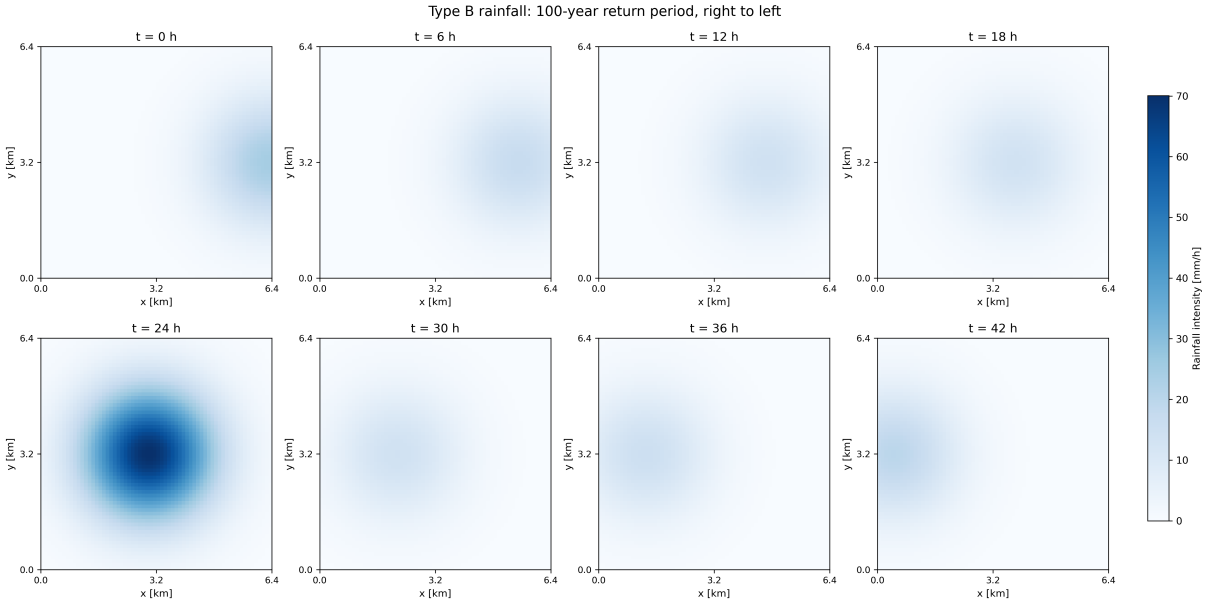


Figure 4.2: SU design hyetographs for the six return periods.

The SV rainfall combines the SU hyetographs with a moving Gaussian field, as described in Chapter 3. Each of the five storm movements is combined with each of the six hyetographs. This gives 30 SV rainfall forcings. Figure 4.3a and Figure 4.3b show two SV events.



(a) Top-to-bottom storm movement, 50-year return period.



(b) Right-to-left storm movement, 100-year return period.

Figure 4.3: SV rainfall intensity at eight times for two events.

4.4 Events and Data Splits

The events are divided by terrain and by return period. The 30 smaller DEMs are the seen DEMs, which are used in training. The 16 larger DEMs are the unseen DEMs, which are excluded from training and validation. Training uses the return periods 2, 5, 20, and 50 years. Validation uses the 10 year return period event. Testing for extreme rainfall uses 100 years return period. All five SV storm movement patterns appear in every split. The unseen DEMs are both new to the model and larger than the training DEMs. Table 4.3 summarizes the allocation.

Table 4.3: Summary of training, validation, and testing dataset splits.

Split	Terrain	Return periods (years)	SU events	SV events
Training	30 seen DEMs	2, 5, 20, 50	120	600
Validation	30 seen DEMs	10	30	150
Unseen domain	16 unseen DEMs	2, 5, 10, 20, 50	80	400
Unseen rainfall	30 seen DEMs	100	30	150
Both unseen	16 unseen DEMs	100	16	80

Chapter 5

Experimental Design

5.1 Overview of Experiments

This chapter describes the experiments used to select the model architecture and to evaluate the adapted mSWE-GNN. The experiments are summarized in Table 5.1. The architecture is selected first, using the SU dataset, and is reused for the SV dataset. For SV rainfall, three rainfall-context configurations are compared. The models are then compared with the SFINCS reference and the bucket model. This is followed by generalization assessment to unseen domain, unseen rainfall and both unseen domain and rainfall. Finally, computational inference runtime is compared between SFINCS and the mSWE-GNN.

Table 5.1: Overview of the experiments.

Experiment	Purpose	Data	Section
Hyperparameter search	Select the number of message-passing layers and hidden features that balance accuracy and inference time	SU	5.2
Rainfall context experiments	Compare configurations with current, past, and future rainfall as inputs	SV	5.3
Generalization tests	Evaluate performance in unseen domain, unseen rainfall, and both unseen	SU and SV	5.4
Runtime evaluation	Compare SFINCS and mSWE-GNN runtime, for individual and batched events	SU and SV	5.5

5.2 Hyperparameter Optimization

A grid search is used to select model architecture that balances accuracy and inference time. For simplicity, the hyperparameters considered were only the number of message-passing layers and the number of hidden features. The tested number of layers are: 2, 3, or 4, and the number of hidden features are: 32, 64, or 128. Their combination gave nine configurations.

Each configuration is trained on the SU training split for 100 epochs with curriculum learning.

The checkpoint with the lowest six-step validation RMSE is kept. Each kept checkpoint is then evaluated on the SU validation events with full autoregressive rollouts. The full-rollout RMSE and the inference time are recorded for every configuration. The configurations are compared by plotting RMSE against inference time. The selected architecture is reused by the model when it is working with SV events. In future, a separate hyperparameter search using SV events needs to be done.

5.3 Rainfall Context Experiment

SV rainfall varies in space and time. This experiment tests whether providing context about past or future rainfall improves the predictions of mSWE-GNN model for SV events. Three configurations are compared: current rainfall only, past rainfall, and future rainfall. All configurations use the same three water-depth inputs and the same terrain input. Table 5.2 summarizes the dynamic inputs used in each configuration.

Table 5.2: Dynamic inputs of the SV rainfall context configurations.

Configuration	Water-depth inputs	Rainfall inputs	Dynamic features per cell
Current rainfall only	$h(t-2), h(t-1), h(t)$	$P(t)$	4
Past rainfall	$h(t-2), h(t-1), h(t)$	$P(t-2), P(t-1), P(t)$	6
Future rainfall	$h(t-2), h(t-1), h(t)$	$P(t), P(t+1), P(t+2)$	6

The past and future configurations provide two additional rainfall intensities apart from the current rainfall. All configurations use the selected architecture and the same SV training, validation, and test splits.

5.4 Generalization Tests

To evaluate the generalizability of mSWE-GNN, three tests are conducted:

- Unseen domain: This is done using 16 unseen DEMs with return periods of 2, 5, 10, 20, and 50 years.
- Unseen rainfall: This is done using 30 seen DEMs with the 100-year return period.
- Both unseen: This is done using 16 large unseen DEMs with the 100-year return period.

5.5 Runtime Evaluation

The runtime of the mSWE-GNN is compared with the runtime of SFINCS for SU and SV events. The SFINCS simulations were run locally on an Intel Core i7-1355U CPU. GNN inference was performed on the Deltares HPC cluster, using an NVIDIA H100 GPU for GPU inference and an Intel Xeon Gold 6342 CPU for CPU inference. The comparisons are made for single events

and for batched events. SFINCS can also run on a GPU, but SFINCS GPU runtimes were not available. SFINCS on a CPU is therefore compared with the mSWE-GNN on a GPU and on a CPU. A comparison on the same GPU hardware is left for future work.

The speedup over SFINCS is calculated as:

$$\text{speedup} = \frac{T_{\text{SFINCS}}}{T_{\text{mSWE-GNN}}}, \tag{5.1}$$

where T_{SFINCS} is the SFINCS runtime and $T_{\text{mSWE-GNN}}$ is the mSWE-GNN runtime for the same event.

Chapter 6

Results and Discussion

6.1 Selected Hyperparameter Configuration

Nine combinations of message-passing layers ($L = 2, 3, 4$) and hidden features ($H = 32, 64, 128$) were tested. Figure 6.1 plots the validation RMSE from full autoregressive rollouts against the rollout time per event for each combination. Five combinations form the Pareto front: L2H32, L2H64, L2H128, L3H128, and L4H128. For these points, RMSE cannot be reduced without increasing rollout time. The other four combinations (L3H32, L3H64, L4H32, and L4H64) are dominated, because at least one other combination has a lower RMSE with a shorter rollout time. Rollout time increased mainly with the number of layers, from about 0.4 s for $L = 2$ to about 0.55 s for $L = 4$. The number of hidden features had a smaller effect on rollout time.

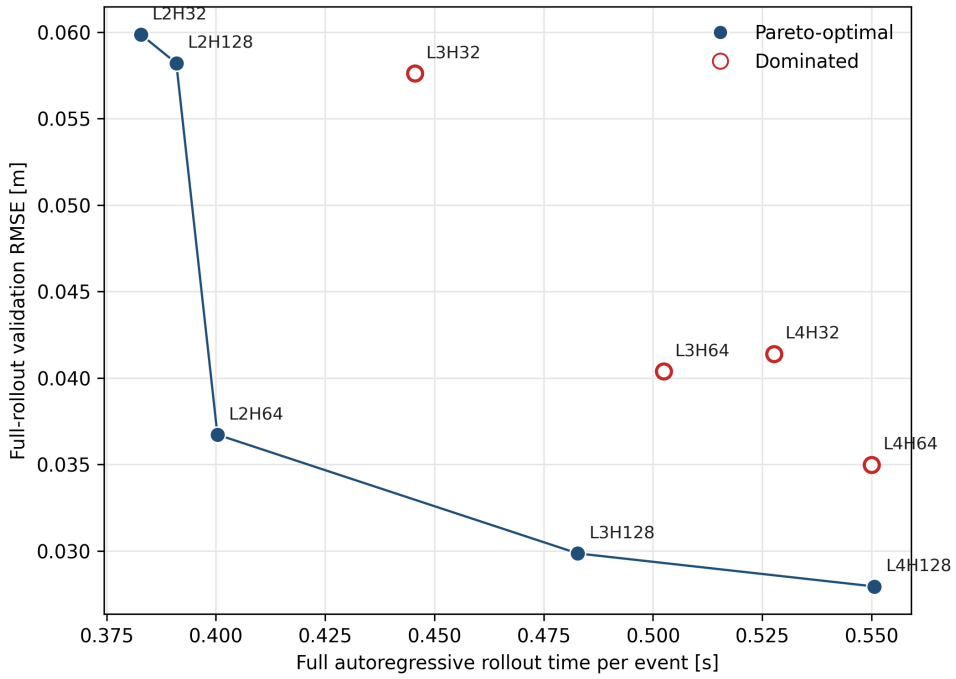


Figure 6.1: Pareto front of validation RMSE versus autoregressive rollout time per event.

Along the front, L2H64 reaches an RMSE of 0.037 m at 0.40 s per event. L3H128 lowers the RMSE to 0.030 m at 0.48 s per event, an increase of 0.08 s. L4H128 lowers it further to 0.028 m, but needs 0.55 s per event, an increase of 0.07 s. The RMSE gain from L3H128 to L4H128 is small compared with the added time. We therefore selected L3H128, with 3 message-passing layers and 128 hidden features. The search used spatially uniform events only, and the same selected hyperparameter configuration is reused for training and testing the model with spatially varying events.

6.2 Performance under Spatially Uniform Rainfall

This section evaluates the performance of the adapted mSWE-GNN under spatially uniform (SU) rainfall events. Firstly, it compares the model with a bucket model and then examines the generalizability under unseen domain and rainfall. Finally, it investigates the effect of additional slope and examines where and when errors occur. Computational performance is discussed in Section 6.4.

6.2.1 Comparison with the Bucket Model

The mSWE-GNN and the bucket model are evaluated against SFINCS over the same 88 predicted frames, from 1.5 to 45 h. For each event, RMSE, MAE and CSI are calculated between the model prediction and the SFINCS reference depth at every frame. These frame scores are then averaged to give one score per event. Figure 6.2 shows the distribution of these event scores for each split.

The mSWE-GNN is more accurate than the bucket model in every split and for every metric. Its median RMSE ranges from 1.57 to 6.21 cm, compared with 10.49 to 17.37 cm for the bucket model. At the 0.05 m threshold, the median CSI of the mSWE-GNN is between 0.765 and 0.850, compared with 0.168 to 0.240 for the bucket model. At 0.30 m, the bucket model has a CSI of zero in all splits, whereas the mSWE-GNN has a median CSI between 0.669 and 0.812.

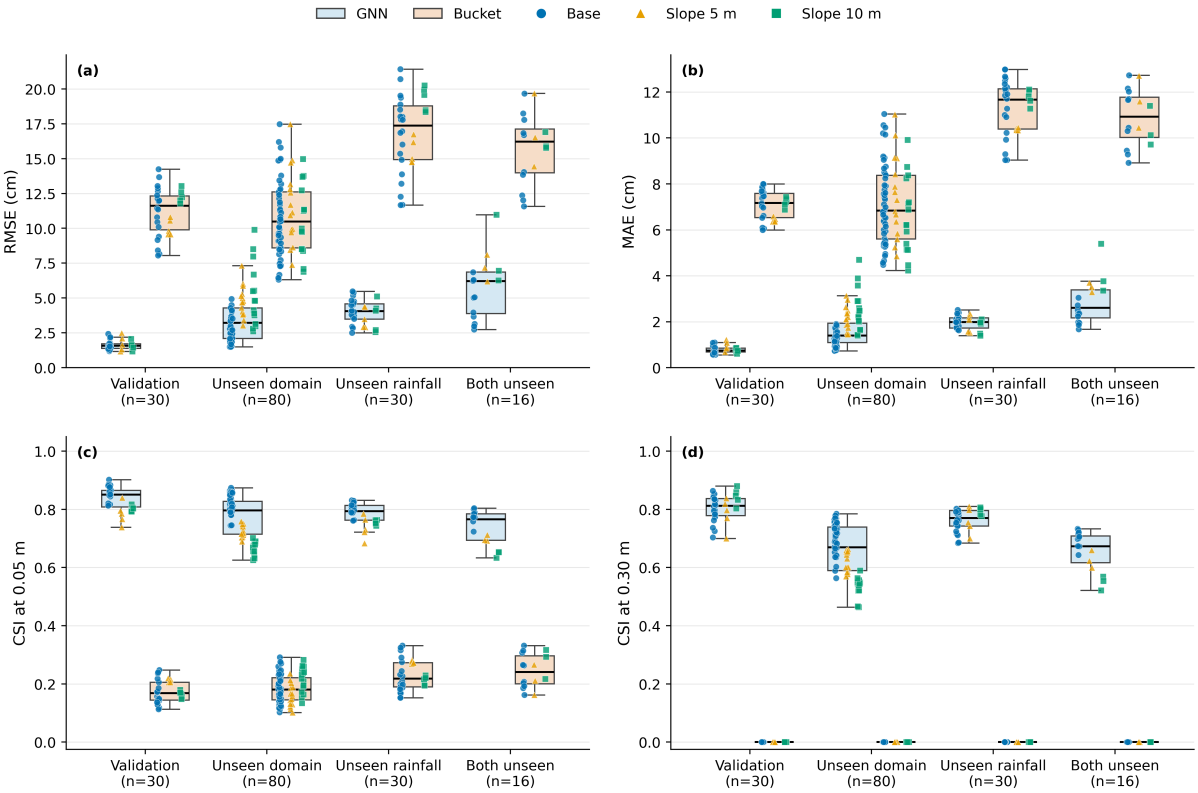


Figure 6.2: GNN and bucket model performance under SU rainfall across the evaluation splits: (a) RMSE, (b) MAE, (c) CSI at 0.05 m and (d) CSI at 0.30 m.

6.2.2 Performance on Unseen Domain and Rainfall

Figure 6.2 also shows how the mSWE-GNN performs on unseen domains, unseen rainfall and both unseen. These results indicate how well the model generalizes beyond its training conditions. Median RMSE increases from 3.22 cm for unseen domains to 4.05 cm for unseen rainfall and 6.21 cm for both unseen. CSI at 0.05 m is similar across unseen domain (0.796) and unseen rainfall (0.794) while being slightly lower for both unseen (0.765).

CSI at 0.30 m does not follow the same order. The median is 0.669 for unseen domains and 0.673 when both unseen, compared with 0.770 for unseen rainfall. Unseen rainfall therefore has a higher RMSE but a higher CSI than unseen domains. This is possible because a high CSI does not guarantee a low depth error: a cell is classified correctly whenever the predicted and reference depths are both above the threshold, even if the predicted depth differs from the reference.

The matched terrain comparisons suggest that the added slopes contribute to the lower CSI in the two splits with unseen domains. These domains are larger than any seen in training, and many of them include an added slope. The sloped terrains perform much worse than their base versions, which lowers the median. Section 6.2.3 examines this effect of slope.

6.2.3 Effect of Added Slope

Section 6.2.2 showed the poor performance of sloped DEMs compared to base DEMs in unseen domain dataset. To investigate this effect of slope, this section compares each base terrain with

its sloped version. Each pair has the same parent DEM, rainfall, and domain size.

The effect is clearest for unseen domains. Figures 6.3 and 6.4 show the pairs under the lowest (RP2) and highest (RP50) return periods. At both, adding slope increases the RMSE and lowers the CSI at both thresholds for all six pairs. The same pattern occurs at RP5, RP10 and RP20 (Appendix C).

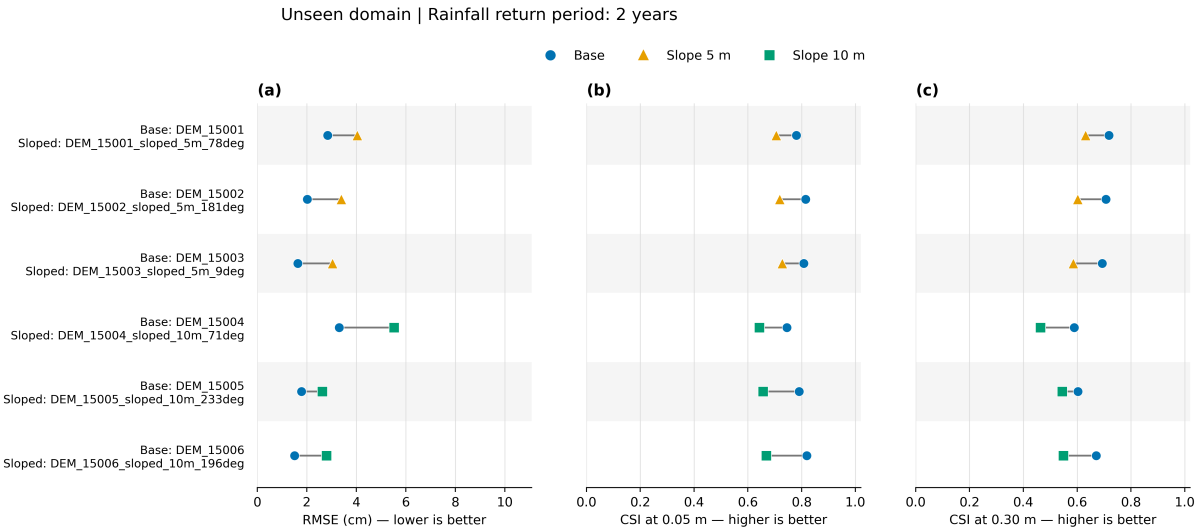


Figure 6.3: Matched base and sloped terrains for unseen domains under RP2 rainfall: (a) RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m.

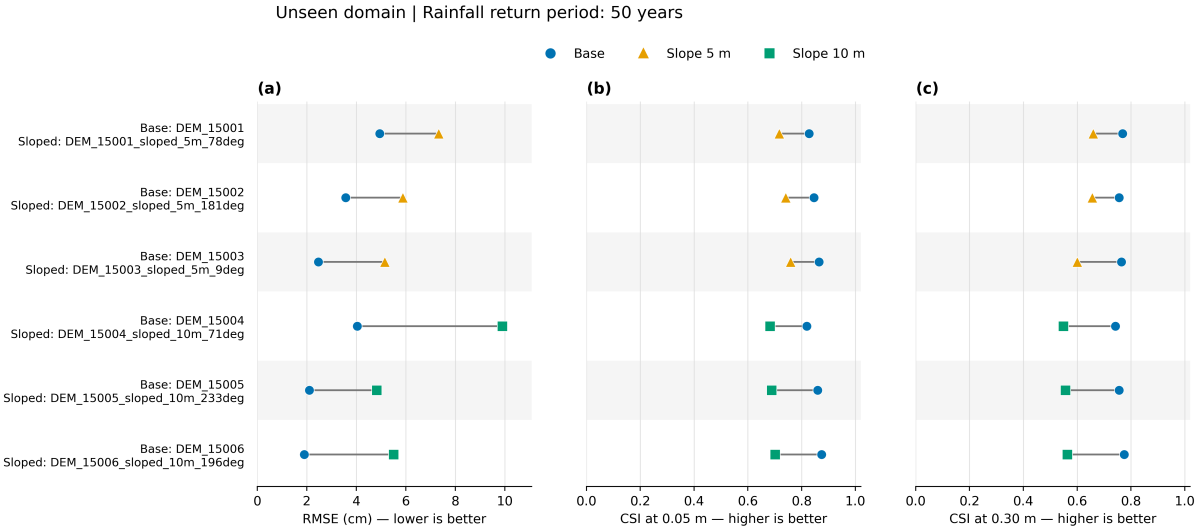


Figure 6.4: Matched base and sloped terrains for unseen domains under RP50 rainfall: (a) RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m.

Domains seen during training, when tested under unseen rainfall, show an anomaly (Figure 6.5). Adding slope increases RMSE in seven of the ten RP100 pairs and lowers CSI at 0.05 m in nine, which agrees with the unseen-domain results. However, CSI at 0.30 m improves in seven pairs. In three of them (DEM 47, 511, and 10002), the CSI improves even though the RMSE also increases. A higher CSI therefore does not mean a lower depth error, because a cell counts as correct whenever both depths exceed the threshold, even if the predicted depth is wrong.

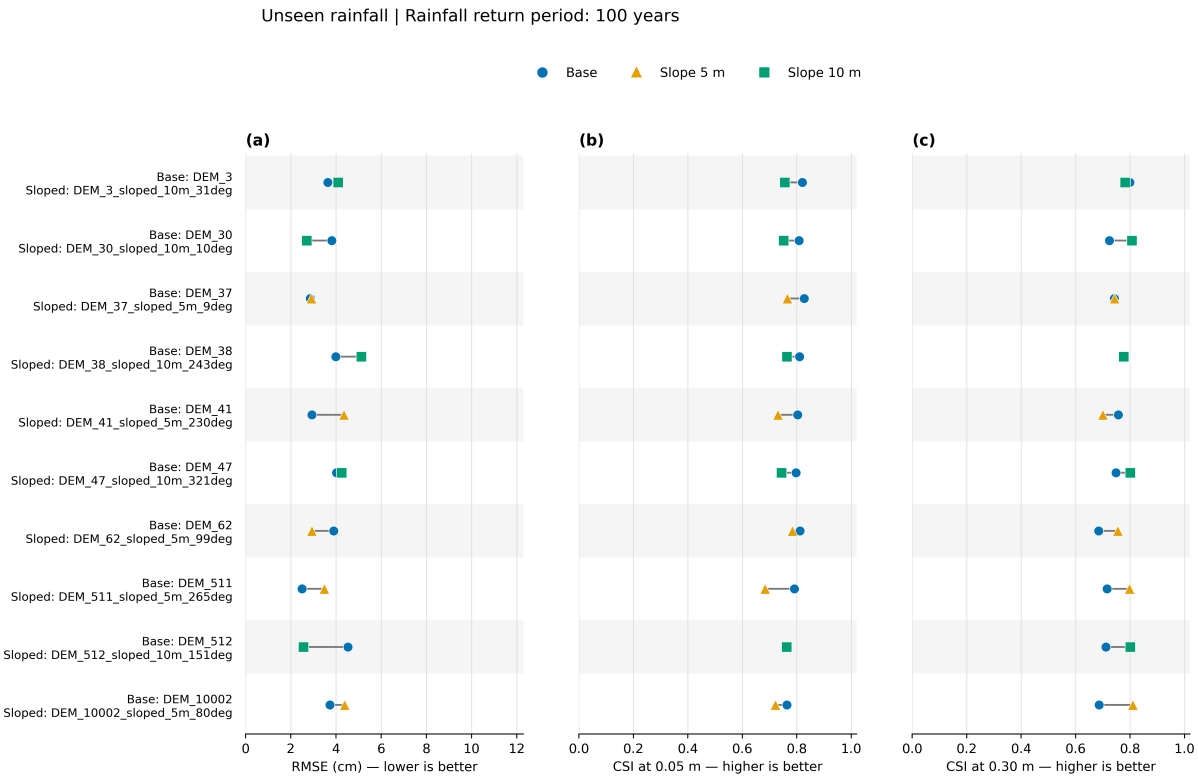


Figure 6.5: Matched base and sloped terrains for familiar terrains under unseen RP100 rainfall: (a) RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m.

6.2.4 Errors over Time

Event-average scores do not show when the differences between base and sloped terrains develop. Figures 6.6 and 6.7 therefore show the temporal evolution of metrics for two DEM pairs under unseen rainfall (RP 100). The performance of DEM 15004 (unseen DEM) deteriorates when slope is added, whereas DEM 511 (seen DEM) is one of the exceptions identified in Section 6.2.3. The pairs were chosen for their contrasting responses and do not represent all events. Two additional temporal comparisons are provided in Appendix C.

For DEM 15004, the sloped RMSE is higher from early in the event and increases sharply at the onset of the rainfall peak (Figure 6.6). At 45 h, it is 22.27 cm, compared with 14.15 cm for the base DEM. The CSI at 0.05 m is lower for the sloped DEM throughout. At 0.30 m, the base DEM is better until the two scores become nearly equal at the end of the event. The signed bias also differs: the base DEM underpredicts after the peak, whereas the sloped DEM overpredicts. The two DEMs therefore appear to fail in different ways, which Section 6.2.5 examines.

For DEM 511, adding slope increases the RMSE but improves the CSI at 0.30 m (Figure 6.7). The sloped RMSE peaks at about 6.5 cm after the rainfall peak. The CSI at 0.30 m is higher for the sloped terrain after about 25 h, whereas the CSI at 0.05 m remains lower throughout. This matches the event-average result in Figure 6.5.

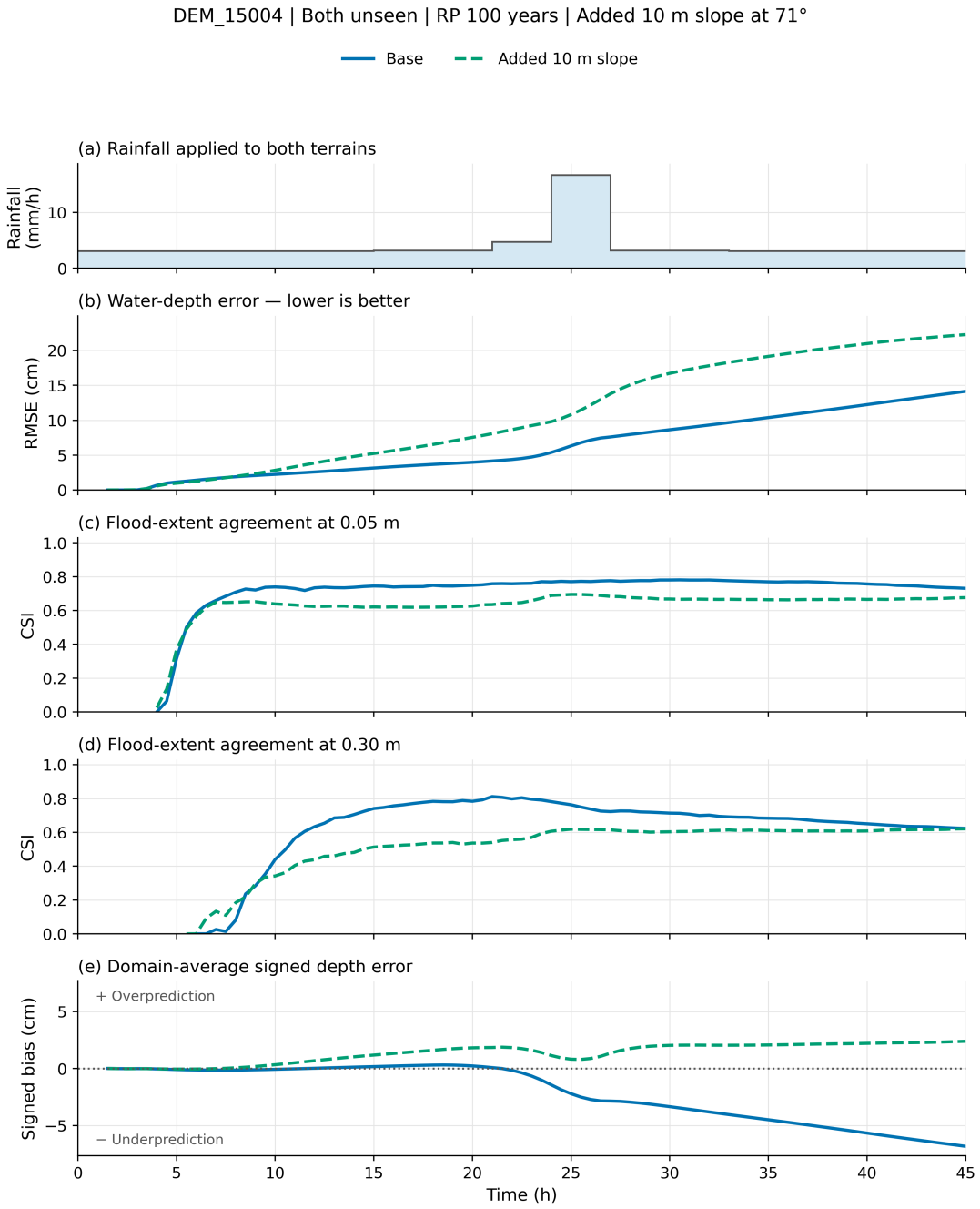


Figure 6.6: Temporal performance of the base and 10 m sloped versions of DEM 15004 under unseen RP100 rainfall with an unseen domain: (a) rainfall, (b) RMSE, (c) CSI at 0.05 m, (d) CSI at 0.30 m, and (e) domain-average signed error.

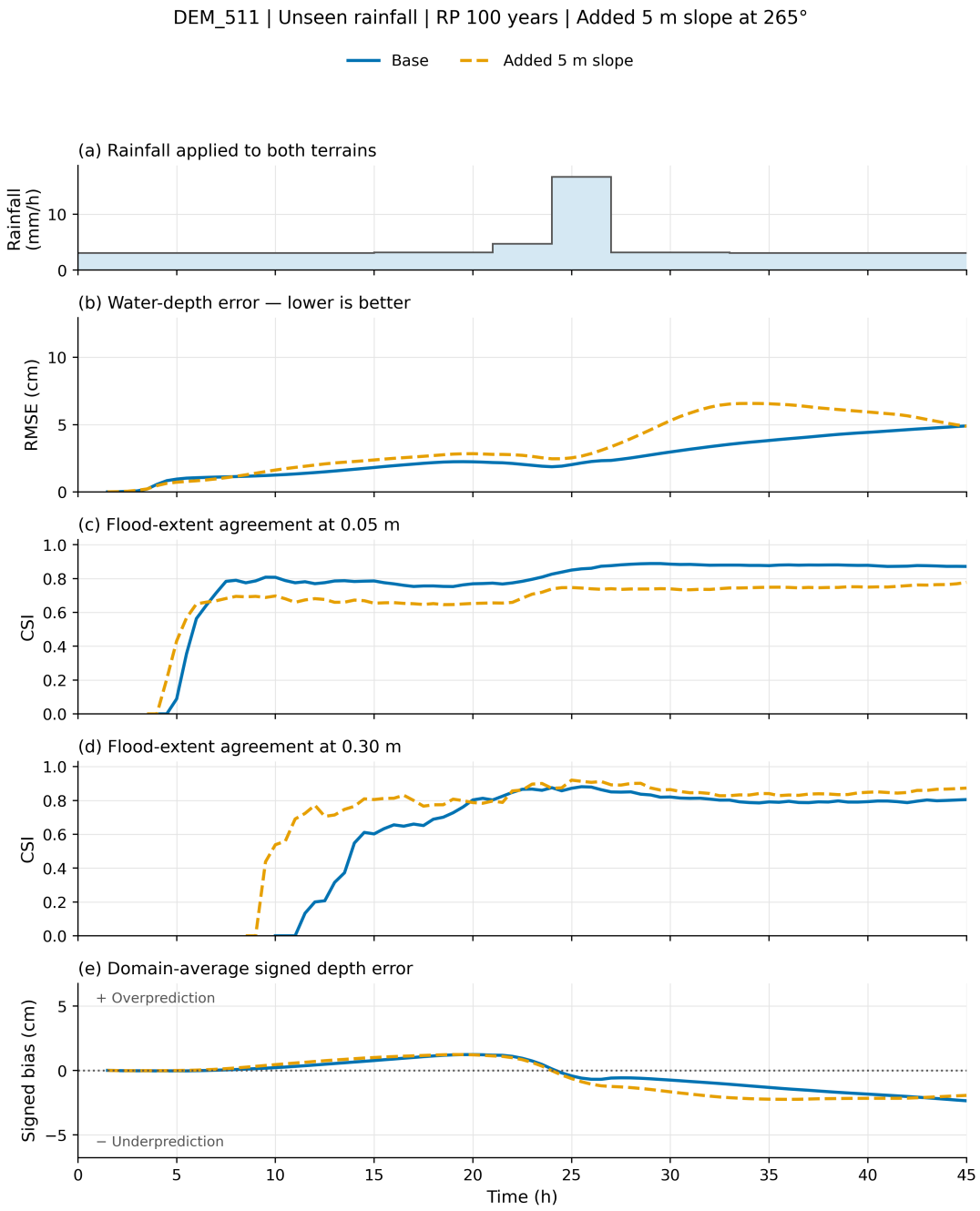


Figure 6.7: Temporal performance of the base and 5 m sloped versions of DEM 511 under unseen RP100 rainfall. Panels follow Figure 6.6.

6.2.5 Flood Extent and Depth Errors

The time curves show when the performance changes, but not where the errors occur. Figures 6.8 and 6.9 show the evolution of flooding, which helps identify the regions where the model prediction is poor.

For DEM 15004, the base DEM has more missed flooding (FN) than the sloped DEM (Figure 6.8). Its CSI peaks at 0.783 at 18 h and falls to 0.624 at 45 h, as missed flooding grows around the flooded patches (FN rises from 167 to 1,424, while FP stays below 100). The sloped DEM has more extra flooding (FP) than the base DEM. This extra flooding appears early on the low-lying side and grows to about 900 cells, and the CSI stays close to 0.60 from 24 h. At 45 h, the CSI is almost the same for both DEMs (0.624 and 0.621), but the errors differ: 20 FP and 1,424 FN for the base DEM, and 895 FP and 772 FN for the sloped DEM.

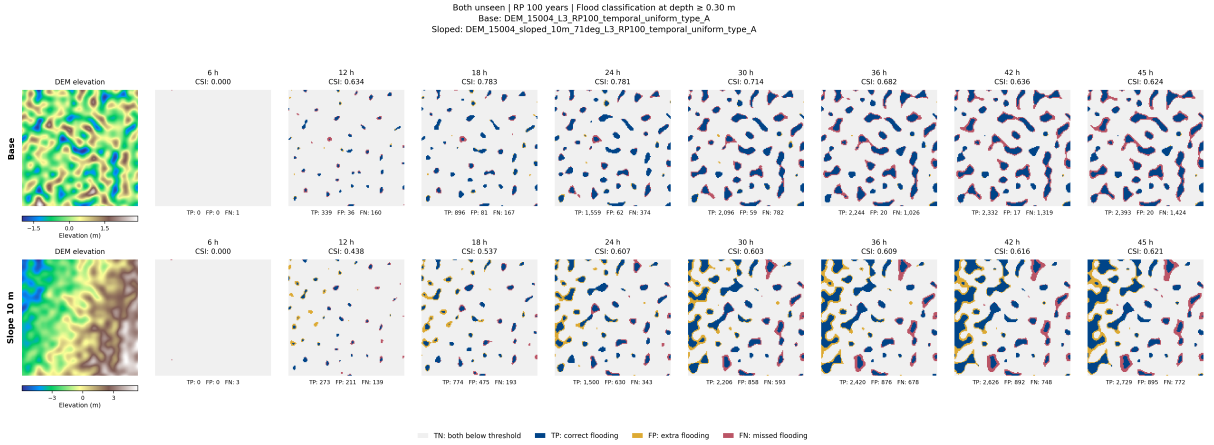


Figure 6.8: Flood classification at 0.30 m for DEM 15004 under unseen RP100 rainfall with an unseen domain. The top row shows the base terrain and the bottom row the 10 m sloped terrain. CSI is above each map and the TP, FP, and FN counts are below.

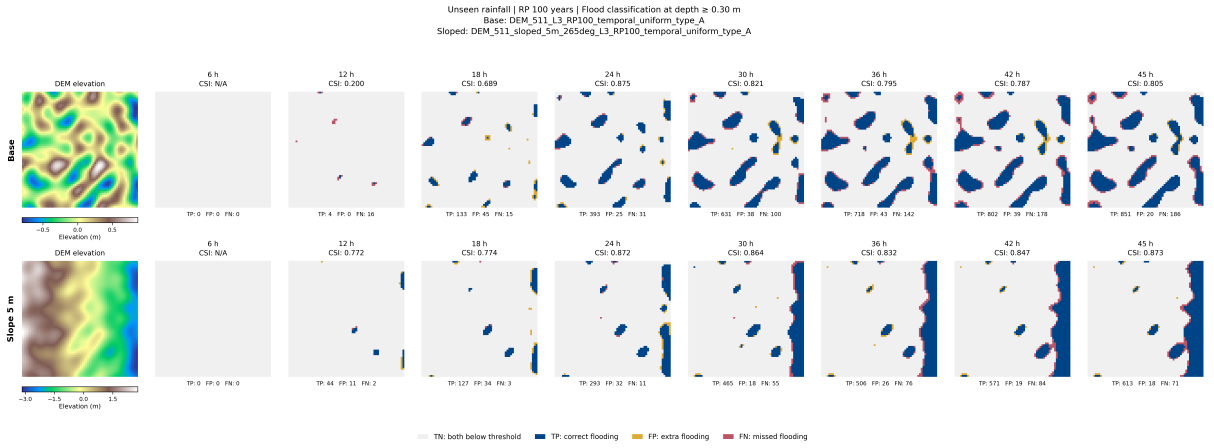


Figure 6.9: Flood classification at 0.30 m for DEM 511 under unseen RP100 rainfall. The top row shows the base terrain and the bottom row the 5 m sloped terrain.

For DEM 511, (Figure 6.9), the sloped DEM has a consistently higher CSI , except at 24 h when the two values are close (0.872 and 0.875). At 45 h, the CSI is 0.873 for sloped DEM and 0.805 for the base DEM. For both DEMs, the errors are thin patches at the edges of the flooded areas.

The base DEM has more missed flooding (FN rises from 31 at 24 h to 186 at 45 h). The sloped DEM has a smaller flooded area than the base DEM and fewer classification errors, with 71 FN and 18 FP at 45 h. Its RMSE is nevertheless larger (Figure 6.7), so a higher CSI does not mean a lower depth error.

In both cases, the flooding of the sloped DEM is concentrated on the low-lying side, so the next section examines how the errors vary with elevation.

6.2.6 Relationship between Elevation and Prediction Errors

The maps in Section 6.2.5 show that the sloped DEMs flood mainly on their low-lying side. To investigate this further, we split each DEM's elevation range into five equal-width intervals. For each interval, we calculate the mean signed error (prediction minus SFINCS) over all cells in each elevation interval at 45 h. Starting with DEM 41, which the model has seen during training, the base DEM slightly underpredicts in its lowest interval (−2.4 cm). The sloped DEM shows overprediction by 11.4 cm for the lowest interval, while the other intervals stay within 1 cm of zero. (Figure 6.10).

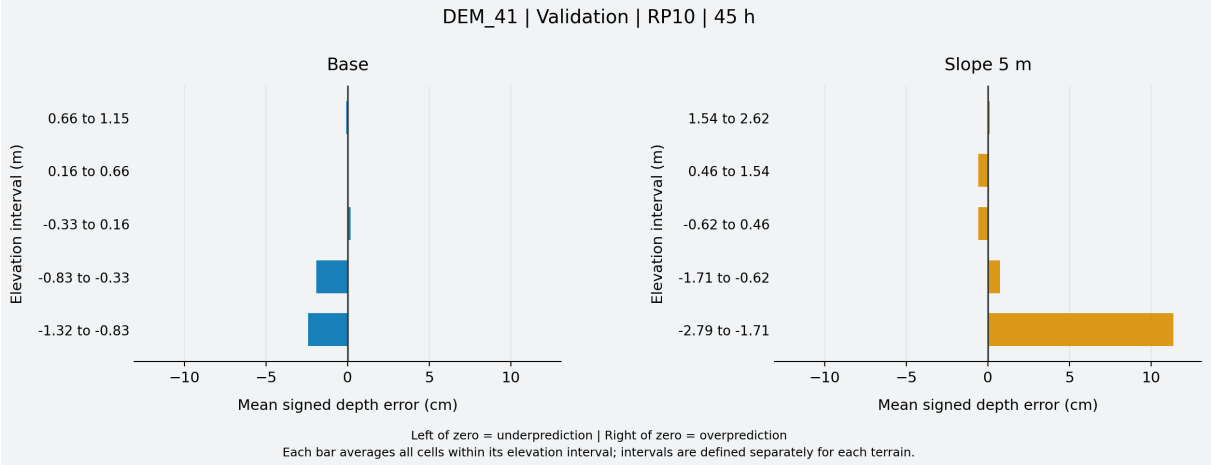


Figure 6.10: Mean signed depth error by elevation interval for DEM 41 and its 5 m sloped version at 45 h under RP10 rainfall (validation split).

The same pattern appears for unseen domains with seen return periods. For DEM 15004 under RP10, the base DEM underpredicts by 20 cm in its lowest interval, whereas the 10 m sloped DEM overpredicts by about 50 cm and underpredicts slightly (1 to 5 cm) in the upper intervals (Figure 6.11). The sloped DEMs also overpredict in the lowest interval for DEM 15004 at RP2 and RP20 (about 34 and 55 cm) and for DEM 15001 at RP5 and RP50 (about 17 and 13 cm), as shown in Appendix C. In these selected examples with rainfall return periods within the training range, the sloped DEMs show overprediction in the lowest elevation interval.

Under unseen RP100 rainfall, the pattern changes. For the same DEM 41 with the 5 m slope, the lowest interval shows underprediction (24 cm) instead of overprediction (11.4 cm) (Figure 6.12). The error sign therefore changes with the rainfall, which lies outside the training range.

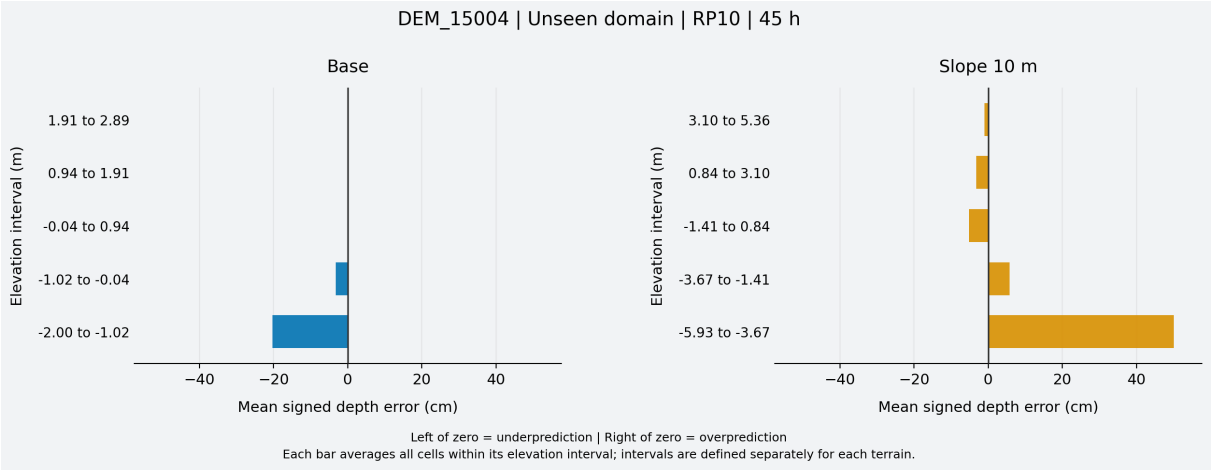


Figure 6.11: Mean signed depth error by elevation interval for DEM 15004 and its 10 m sloped version at 45 h under RP10 rainfall (unseen domain). Panels follow Figure 6.10.

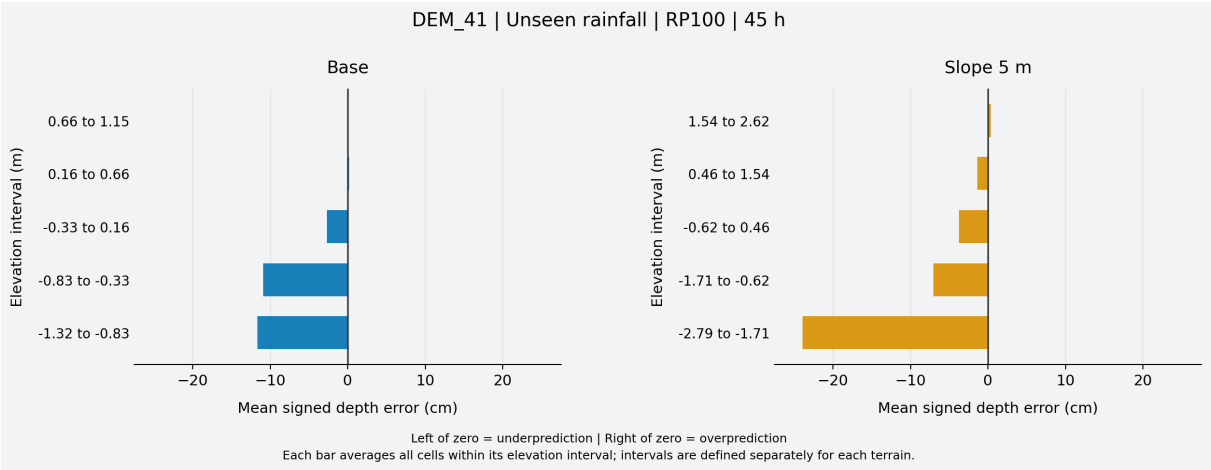


Figure 6.12: Mean signed depth error by elevation interval for DEM 41 and its 5 m sloped version at 45 h under unseen RP100 rainfall. Panels follow Figure 6.10.

6.3 Performance under Spatially Varying Rainfall

This section reports the performance of the adapted mSWE-GNN under SV rainfall. It also tests whether providing additional rainfall context improves the predictions. Three configurations are compared: No rainfall context, Past rainfall context and Future rainfall context. All three receive the current rainfall and three preceding water-depth fields. The Past and Future configurations also receive two past or two future rainfall fields, respectively.

The section first evaluates overall performance and generalization. It then looks at how added slope affects accuracy, how errors develop over time, and where the model over- and underpredicts flooding.

6.3.1 Effect of Rainfall Context and Comparison with the Bucket Model

Figure 6.13 compares the three configurations and the bucket model on the four evaluation splits. The Future rainfall context gives the lowest mean RMSE and the highest mean CSI at both thresholds in every split. On the validation split, it lowers the mean RMSE from 10.80 to 3.37 cm relative to No rainfall context. Mean CSI rises from 0.674 to 0.850 at 0.05 m and from 0.535 to 0.801 at 0.30 m. The Future rainfall context therefore improves both depth accuracy and flood-extent agreement, and the same trend holds in the unseen splits.

The No rainfall context also outperforms the bucket model on all three metrics in every split. This shows that the mSWE-GNN adds value even without extra rainfall context.

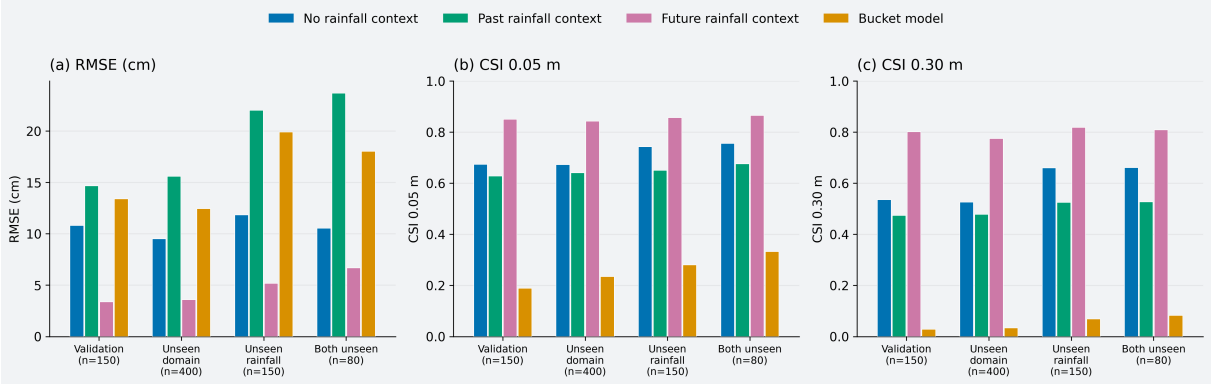


Figure 6.13: Overall performance under SV rainfall for No rainfall context, Past rainfall context, Future rainfall context and the bucket model: (a) RMSE, (b) CSI at 0.05 m and (c) CSI at 0.30 m. Bars show split means of event-average scores. The validation, unseen-domain, unseen-rainfall and both-unseen splits contain 150, 400, 150 and 80 events, respectively.

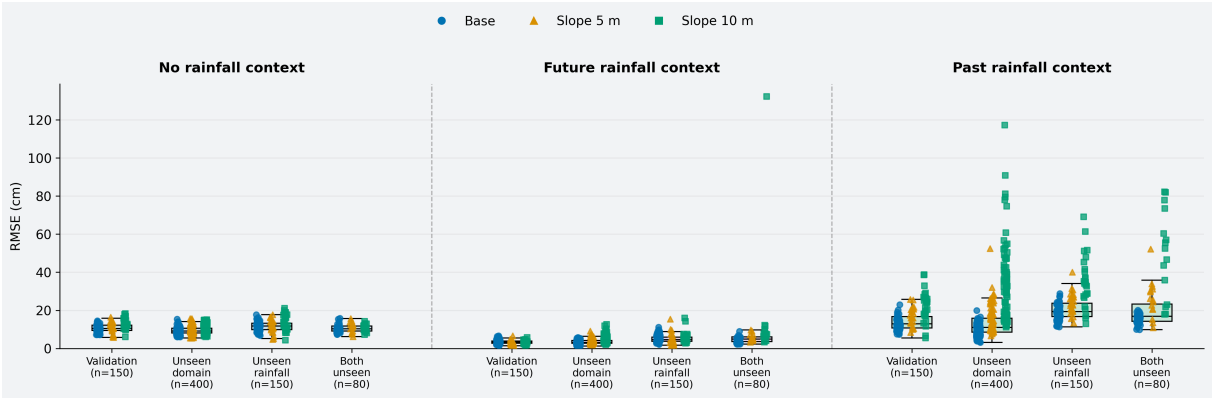
During the later epochs of training, the Past rainfall context began returning NaNs. The training loss was stable up to the retained checkpoint, so this checkpoint is used for the comparison (see Appendix D). The Past rainfall context performs worse than No rainfall context on all three metrics in Figure 6.13. Its mean RMSE is also higher than that of the bucket model in every split, although its CSI is much higher. Flooded cells can therefore be identified well while depths are still wrong. Section 6.3.2 examines this performance in more detail, for generalization to unseen conditions and for individual events.

6.3.2 Generalization to Unseen Domains and Rainfall

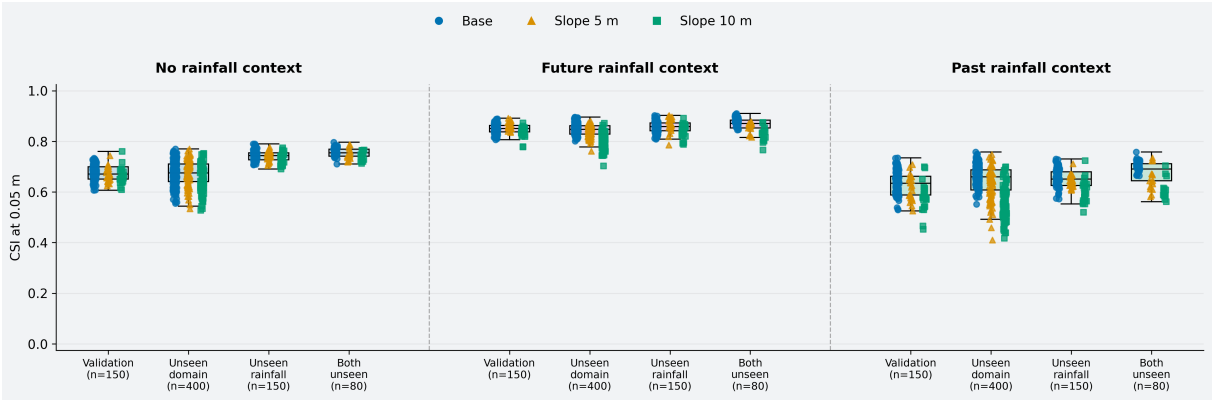
Figure 6.14 shows the event-level mean RMSE and CSI in each evaluation split. The Future rainfall context has the lowest RMSE and the highest CSI at both thresholds in every split. The median RMSE is 10–12 cm for the No rainfall context and 3–5 cm for the Future rainfall context.

For the Future rainfall context, performance deteriorates for unseen test conditions, particularly for sloped DEMs. In the unseen domain, unseen rainfall and both-unseen splits, adding slope increases RMSE and lowers CSI at both thresholds compared with the base DEMs. The mean CSI at 0.05 m nevertheless stays between 0.842 and 0.865 in all splits. For No rainfall context, CSI at 0.30 m threshold, model performs better for unseen rainfall than unseen domain and both unseen split. However, its RMSE is higher for unseen rainfall than both these other splits.

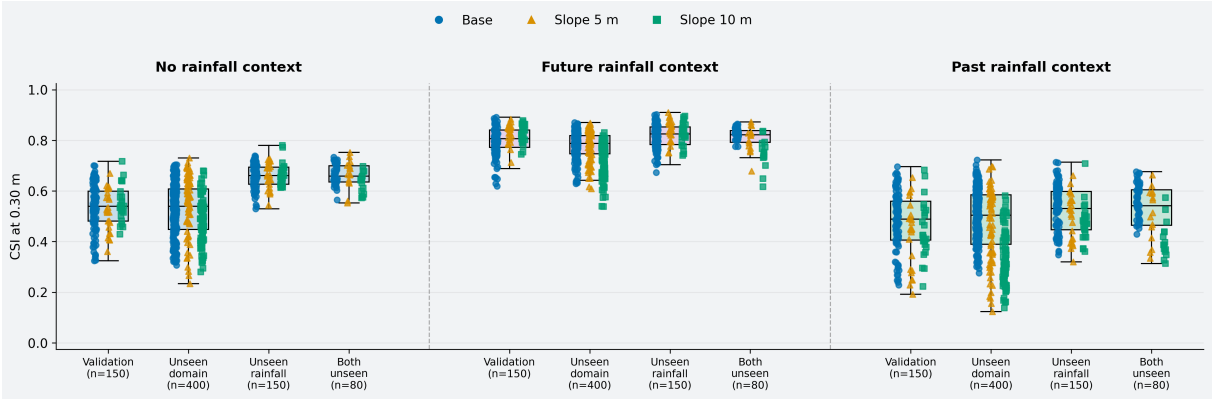
For the Past rainfall context, the sloped DEMs perform much worse than the base DEMs. This results in the poor aggregate performance seen in Figure 6.13. The Future rainfall context also holds advantage across return periods and storm movements (Appendix D). Section 6.3.3 examines the effect of slope in more detail.



(a) RMSE



(b) CSI at 0.05 m



(c) CSI at 0.30 m

Figure 6.14: Event-level RMSE and CSI under SV rainfall for the No, Future and Past rainfall contexts across the four evaluation splits.

6.3.3 Effect of Added Slope

To further investigate the poor performance of sloped DEMs, each is compared with its parent base DEM under the same storm movement, rainfall RP and domain size. (Figure 6.15) shows this comparison for the unseen domain dataset.

Adding slope generally increases RMSE for both the No rainfall context and Future rainfall context configurations, although exceptions occur. CSI generally decreases, mostly at 0.30 m. The Future rainfall context has lower errors on both base and sloped DEMs, but the effect of slope is more pronounced than No rainfall context. For DEM 15006, the RMSE of the future rainfall context rises from 1.5 to 4 cm. With the No rainfall context, the same DEM rises by only about 1 cm. DEM 15003 is the exception. For Future rainfall context, RMSE is almost unchanged, and its CSI at 0.30 m is slightly higher on the sloped DEM.

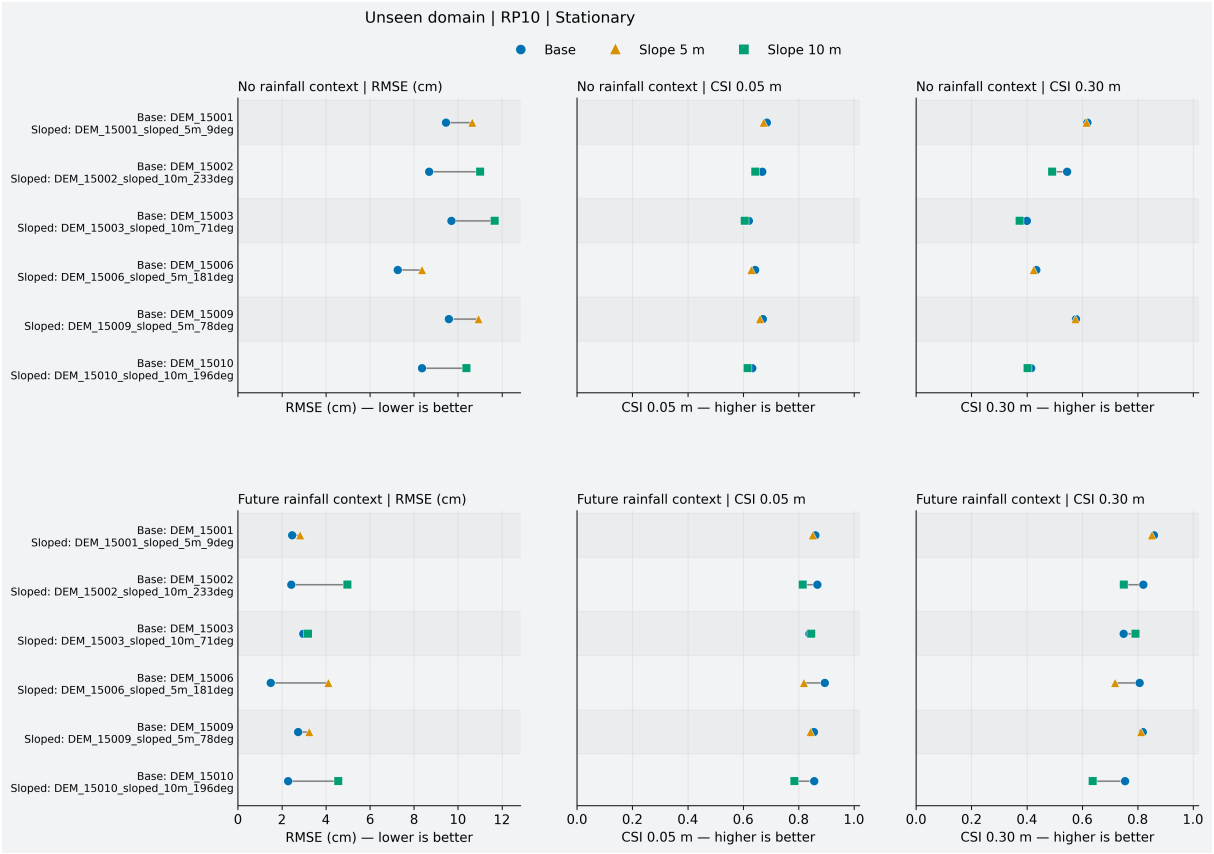


Figure 6.15: Base and sloped DEM pairs for unseen domains. Columns show event-average RMSE and CSI at 0.05 and 0.30 m.

The same comparison is repeated for both unseen split (Figure 6.16). Here the DEMs are larger and not seen during the training and the rainfall is also beyond the training range. For the Future rainfall context, most sloped DEMs have higher RMSE and lower CSI than their corresponding base DEMs, including DEMs 15002, 15006, and 15010.

Adding slope therefore increases the error for most DEMs with some exceptions to this trend. To investigate this further, the subsequent section 6.3.4 examines how the errors develop over time.

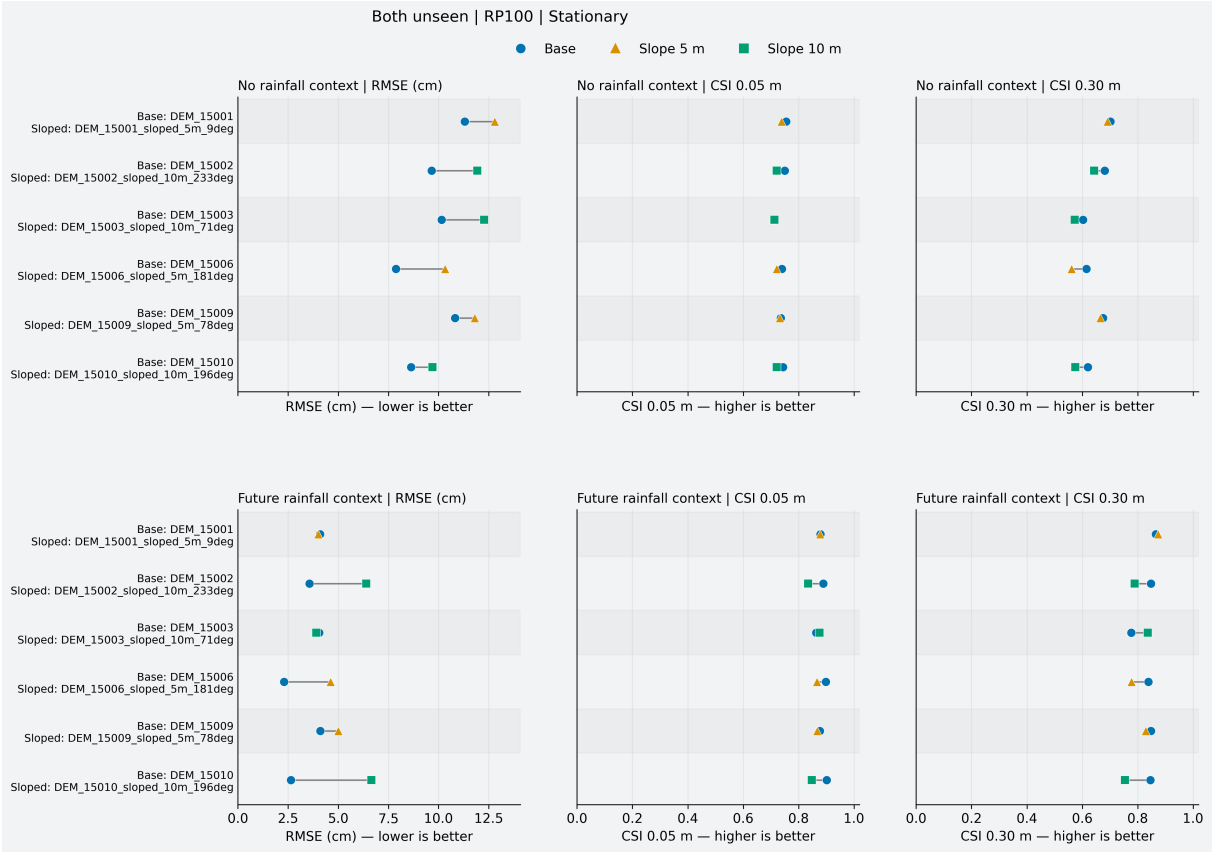


Figure 6.16: Base and sloped DEM pairs for both unseen.

6.3.4 Errors over Time

The event-average scores in Section 6.3.3 do not show the temporal evolution of the metrics. Figures 6.17 and 6.18 show how the RMSE and CSI develop over time for DEM 15003 (unseen DEM) and DEM 55 (seen DEM), both under stationary RP100 rainfall. DEM 15003 is the exception mentioned in the previous sections, and DEM 55 is the DEM where the effect of slope differs between the context configurations. For both DEMs, the RMSE grows gradually through the event for both configurations. The Future rainfall context performs better than No rainfall context.

For DEM 15003 with the No rainfall context, the sloped DEM has a higher RMSE than the base DEM for most of the event, and the base DEM has the higher RMSE at the end of the event. With the Future rainfall context, the sloped DEM has the lower RMSE until the rainfall peak ends at about 27 h, and a higher RMSE afterwards. The CSI at 0.05 m follows the base DEM closely, with the base slightly lower during the event and similar after the peak. The CSI at 0.30 m is higher for the sloped DEM for most of the rainfall event.

For DEM 55 with the No rainfall context, the base and sloped DEMs have a similar RMSE during the early part of the event. The sloped DEM then has the higher RMSE, and the difference grows after the rainfall peak. With the Future rainfall context, the sloped DEM has the lower RMSE and the higher CSI throughout, although the RMSE gap narrows towards the end. These figures show how the errors develop over time, but they do not show where in the domain the

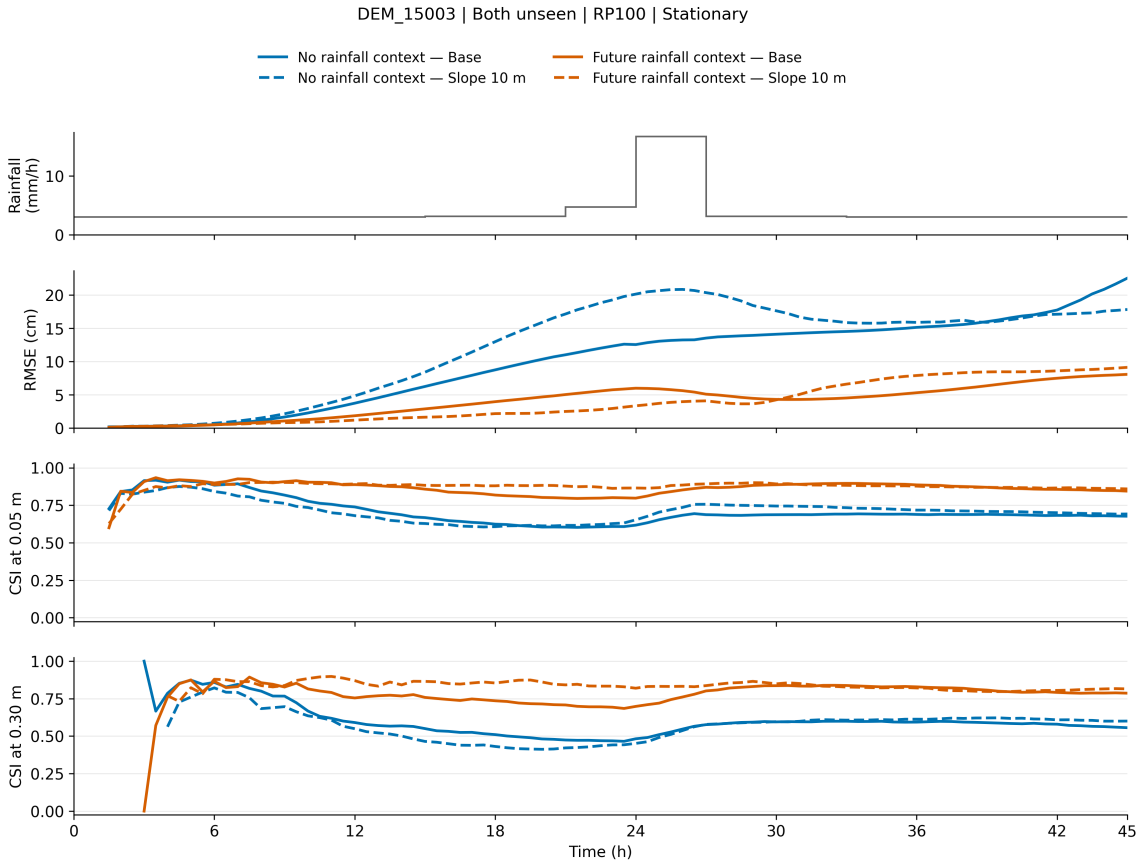


Figure 6.17: Temporal evolution of metrics for DEM 15003 and its 10 m sloped version for both unseen. Panels show domain-average rainfall intensity, spatial RMSE, CSI at 0.05 m and CSI at 0.30 m.

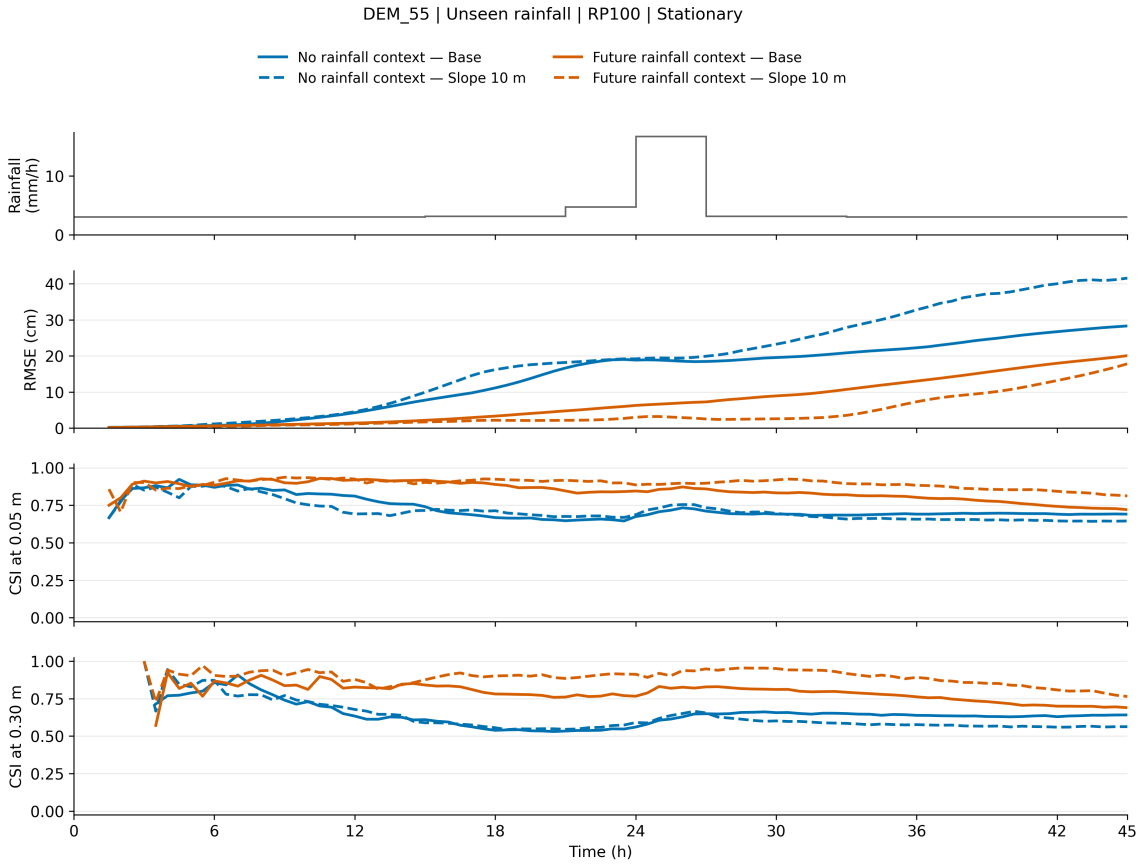


Figure 6.18: Temporal evolution of metrics for DEM 55 and its 10 m sloped version for unseen rainfall. Panels show domain-average rainfall intensity, spatial RMSE, CSI at 0.05 m and CSI at 0.30 m.

model performs poorly. The spatial pattern of the errors is also important, therefore the next Section 6.3.5 examines the flood maps.

6.3.5 Spatial Errors and Flood Extent

This section examines the spatial locations of error in the flood predictions. Figures 6.19 and 6.20 show the signed depth error and the flood classification at 0.05 and 0.30 m for DEM 55 and DEM 15003 at 45 h, the final water depth. The maps at 24 h, during the rainfall peak, are in Appendix D.

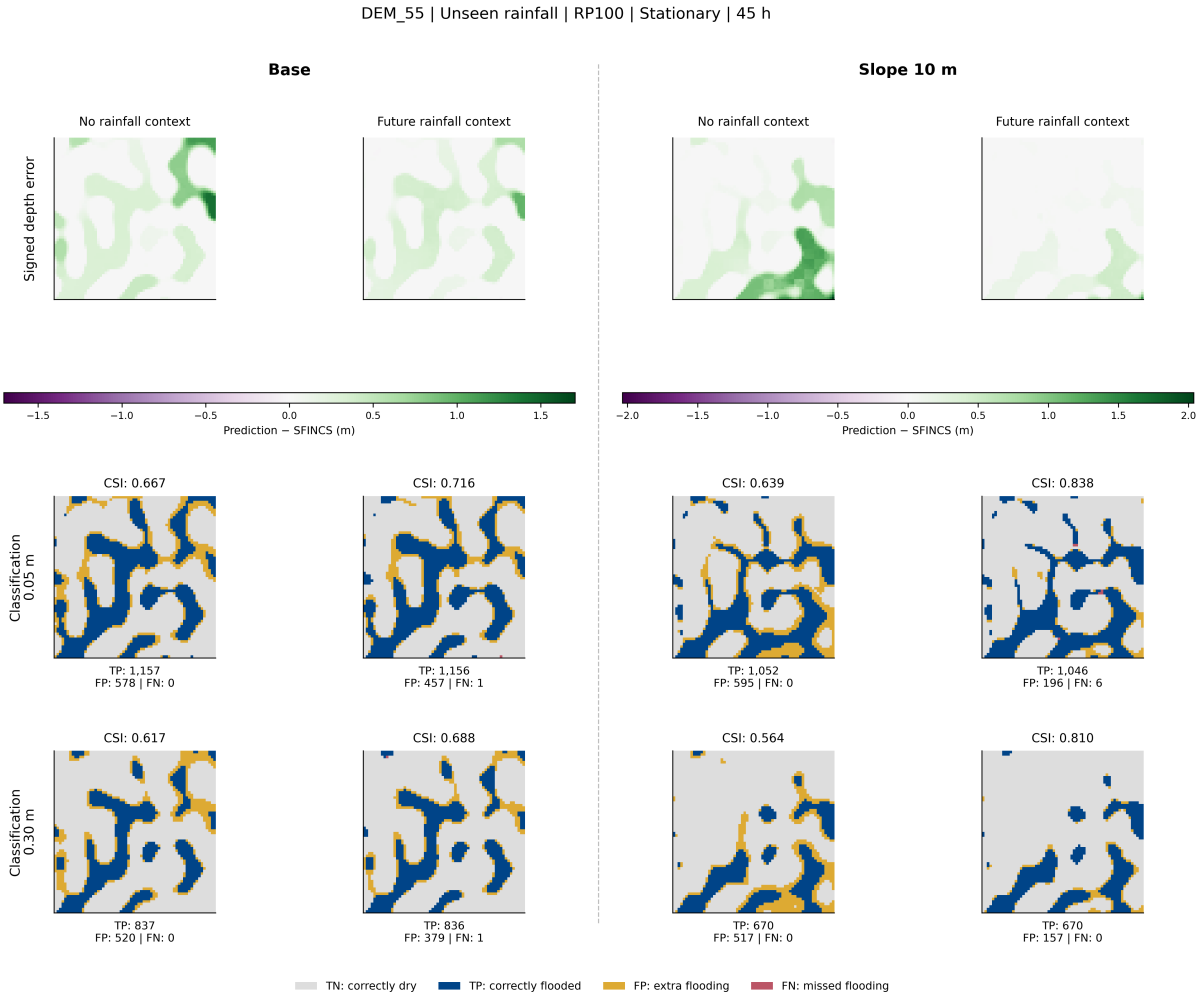


Figure 6.19: Spatial errors for DEM 55 and its 10 m sloped version under unseen rainfall at 45 h.

In both cases, the No rainfall context predicts more extra flooding than the Future rainfall context. At 0.30 m, the false positives fall from 520 to 379 cells for the base DEM 55 and from 517 to 157 for the sloped DEM 55. For DEM 15003, they fall from 3,335 to 944 cells for the base DEM and from 2,038 to 608 for the sloped DEM. The Future rainfall context has more missed flooding. For DEM 15003, the false negatives rise from 12 to 87 on the base DEM and from 13 to 137 on the sloped DEM, and for DEM 55 they change only from 0 to 1 on the base DEM and stay at 0 on the sloped DEM. In these examples, overprediction is concentrated in areas with lower elevation, resembling the pattern observed in several SU cases. The Future rainfall context

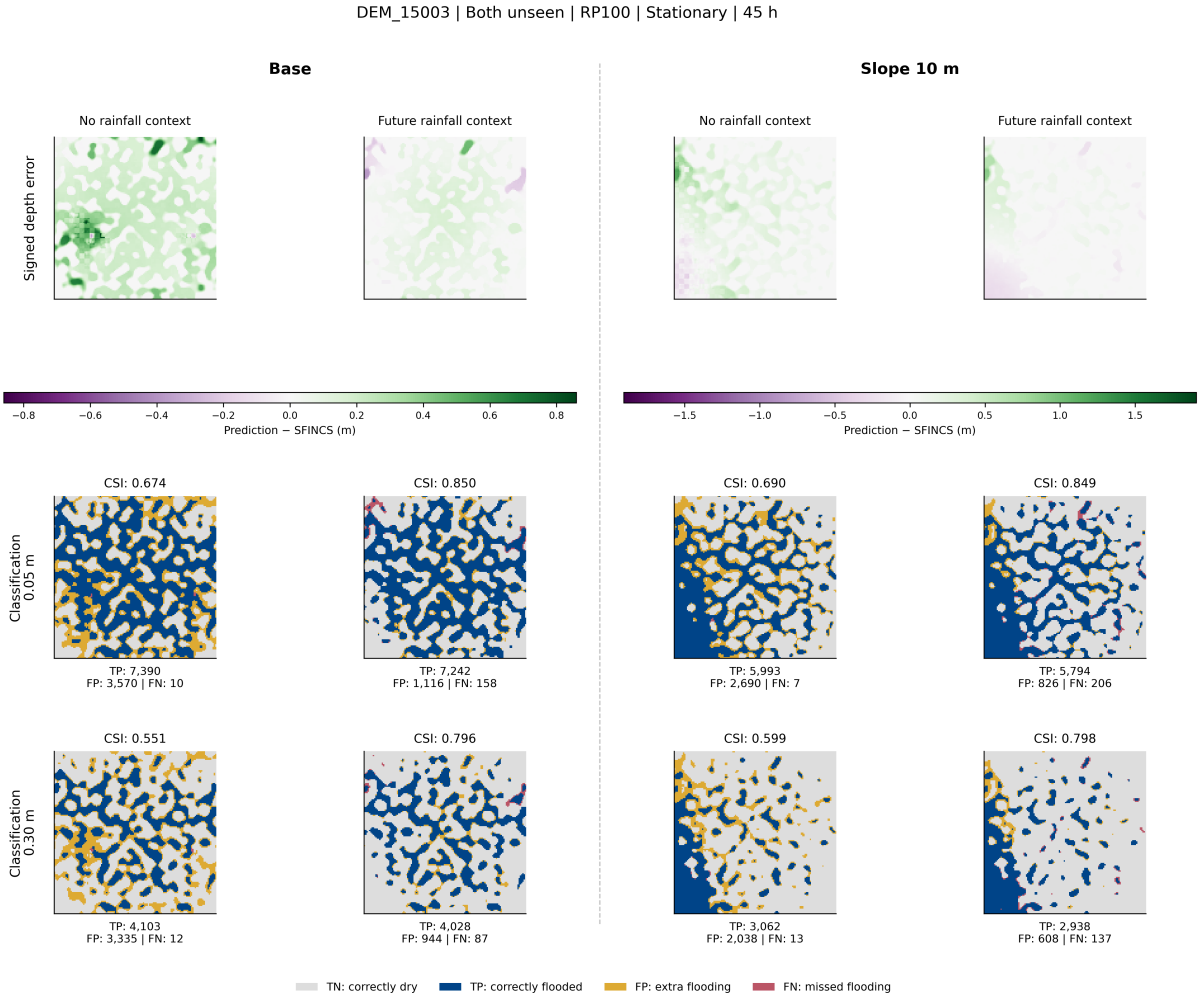


Figure 6.20: Spatial errors for DEM 15003 and its 10 m sloped version under both unseen at 45 h.

therefore improves the flood extent mainly by removing extra predicted flooding, sometimes at the cost of missed flooding. Because of this trade-off, a higher CSI does not mean that every part of the flood detection improves. CSI also ignores the depth inside correctly classified cells, which explains how CSI can rise while the RMSE stays large.

6.4 Computational Inference Runtime

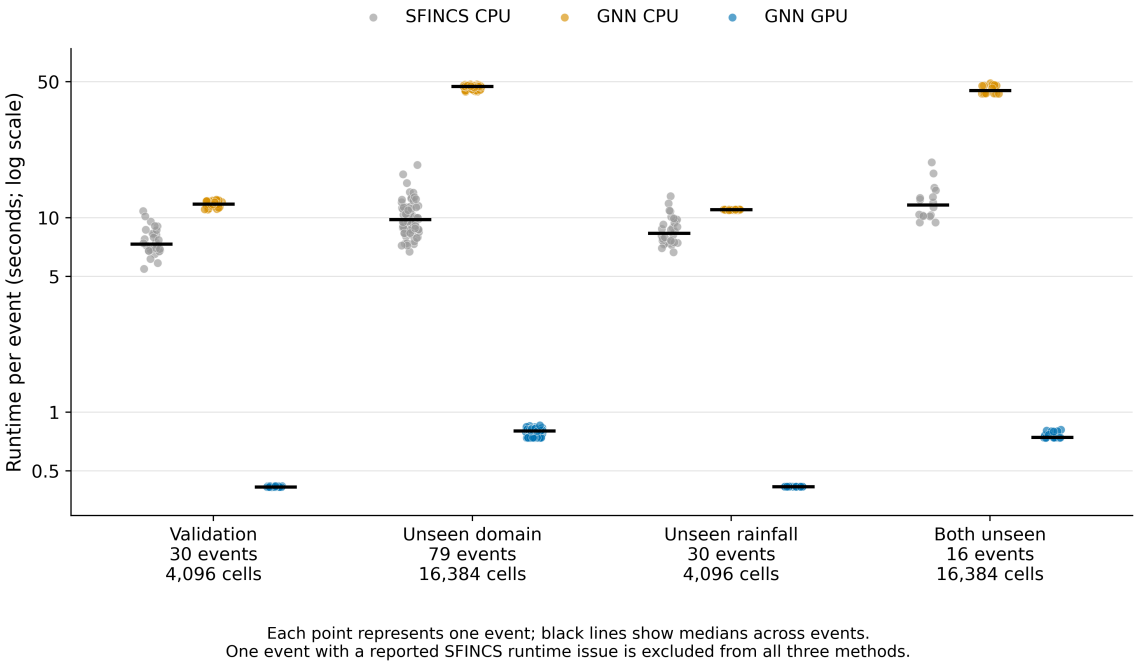


Figure 6.21: Runtime per event for SFINCS on the CPU, the GNN on the CPU and the GNN on the GPU across the SU evaluation splits. Each point is one event, and black lines show the medians.

SFINCS was run on a local Intel i7-1355U CPU, and the mSWE-GNN was run on an NVIDIA H100 GPU on an HPC cluster. For SU rainfall, Figure 6.21 shows the runtime per event across the evaluation splits. GNN GPU inference is faster than SFINCS CPU in every split, taking approximately 0.4 s for the seen DEMs (4,096-cells) and 0.8 s for the unseen DEMs (16,384-cells). One unseen domain event was removed from runtime analysis because its SFINCS runtime was anomalously high. In the tested hardware configurations, the mSWE-GNN was faster than SFINCS when run on the GPU, but slower when run on the CPU.

Batching events on the GPU further reduces computational runtime (Figure 6.22). For the validation split, processing all 30 events takes 12.43 s individually and 4.68 s in a batch, giving an additional speed-up of approximately 2.65× relative to individual GPU inference. For the both unseen split, which contains larger domains, the batching gain is approximately 1.18×. No batching results are reported for the unseen domain split because the attempted runs failed with out-of-memory errors.

For SV rainfall, Figure 6.23 shows the comparison for the 150 SV validation events. The mean SFINCS CPU runtime is 6.61 s per event, against 0.379 s for the mSWE-GNN GPU with the No

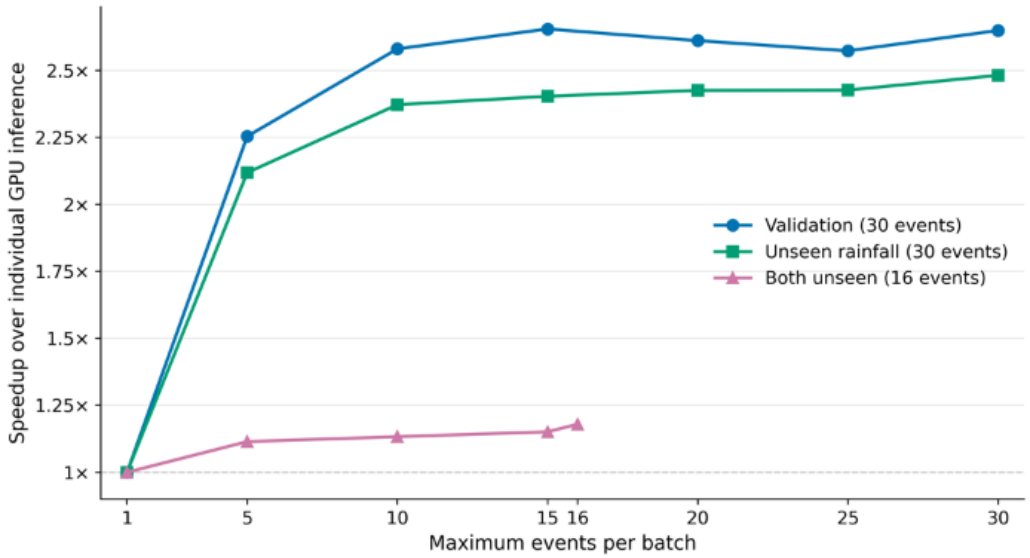


Figure 6.22: GPU batching speedup relative to individual GPU inference for the SU splits.

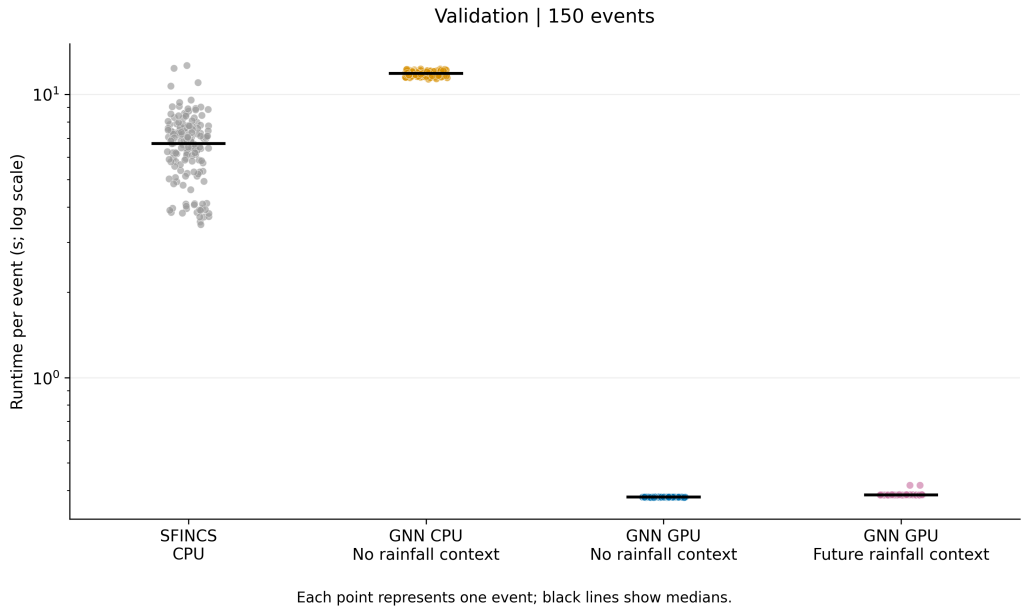


Figure 6.23: Runtime per event for the 150 SV validation events: SFINCS on the CPU, the GNN on the CPU with the No rainfall context, and the GNN on the GPU with the No and Future rainfall contexts.

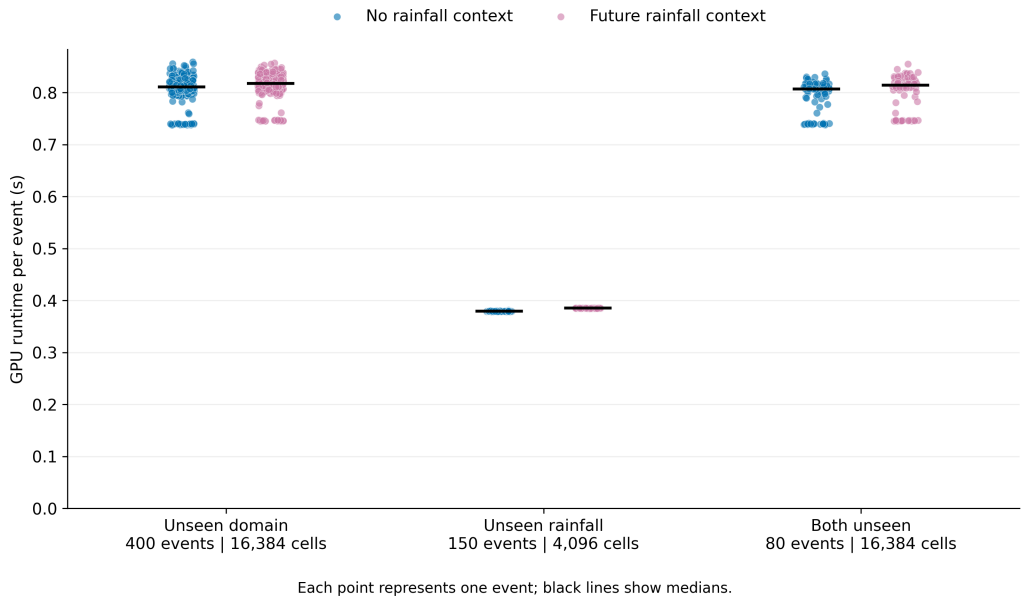


Figure 6.24: GPU runtime per event for the No and Future rainfall contexts across the three unseen SV splits.

rainfall context and 0.386 s with the Future rainfall context. The runtime ratio is about 17 for both configurations. For the unseen splits, only the GPU runtime was measured (Figure 6.24). It stays below one second per event and is about the same for both configurations.

6.5 Limitations

The dataset used for training and testing limit transferability of the model. The terrains are synthetic and idealized, with closed boundaries, uniform roughness and 100 m cells. The simulations also leave out hydrological processes such as infiltration, evaporation, drainage networks and buildings. The reference is SFINCS, a reduced-physics solver, and not observed flood events. The results therefore show how closely the mSWE-GNN reproduces this SFINCS setup, including its simplifications, and not how accurate it is against real floods. The conclusions apply to the simulated conditions, and performance on real catchments remains untested.

The model design also has limits. The hyperparameters were selected with SU data only, and SV data was not used for the selection. No ablation tests were conducted, so the improvements cannot be assigned to individual changes to adapted mSWE-GNN. Generalization was tested without fine-tuning on the unseen domain, so improvement with fine-tuning remains unexamined.

The runtime comparison also has limits. SFINCS ran on a local CPU, while the mSWE-GNN ran on an HPC GPU. SFINCS GPU runtimes were not available, and would provide a fairer comparison of the speed-ups. The mSWE-GNN CPU test was also run on different hardware from the SFINCS CPU test, so the CPU comparison is only preliminary. Running both on the same hardware would be needed for fairer results. These limitations define the future work in Chapter 7.

Chapter 7

Conclusions, Future Work, and Societal Relevance

This thesis adapted the multi-scale hydraulic graph neural network (mSWE-GNN) for pluvial flood modelling and evaluated its ability to emulate SFINCS-simulated water depths and flood extent. Performance was evaluated under spatially uniform (SU) and spatially varying (SV) rainfall. This chapter presents the main conclusions, discusses directions for future work, and emphasizes the societal relevance of the thesis.

7.1 Conclusions

7.1.1 Adaptation of mSWE-GNN for Pluvial Flood Modelling

The first research question investigated how mSWE-GNN can be adapted for pluvial flood modeling. The original multiscale mSWE-GNN architecture was modified to include time-varying rainfall as a dynamic input. The adapted model predicts water depth only, rather than both water depth and unit discharge. The adapted model also uses water-level gradients and a wet/dry gate in its message-passing. The wet/dry gate suppresses messages along edges where the combined water depth of the two connected nodes is below a specified threshold. These changes were intended to make the adapted model more hydraulically informed and interpretable. However, the model is still not fully interpretable as the message passing uses a learned coefficient generated by an MLP.

The adapted mSWE-GNN was able to reproduce SFINCS water-depth dynamics under both SU and SV rainfall. The individual effect of each adaptation was not evaluated because ablation experiments were outside the scope of this study due to lack of time. Future work should compare the adapted model with versions in which individual adaptations are removed to determine their contribution to model performance.

7.1.2 Synthetic SFINCS Dataset

The second research question focused on developing a synthetic dataset for training and evaluating the adapted mSWE-GNN model under spatially uniform (SU) and spatially varying (SV) rainfall. SFINCS was used as the numerical solver to generate this synthetic dataset. The dataset was created using synthetic DEMs, including base DEMs and DEMs with added slope, across two domain sizes. For SU events, time-varying hyetographs were applied uniformly across the domain. For SV events, the same hyetographs were multiplied by Gaussian spatial fields to create stationary or moving rainfall patterns while maintaining the same domain-average rainfall intensity. The resulting SFINCS simulations were converted into multiscale graphs and temporal prediction samples for model training and evaluation. These SFINCS simulations were divided into training, validation, unseen domain, unseen rainfall, and both unseen dataset splits.

The synthetic dataset has several limitations. It does not include hydrological processes such as infiltration and evaporation, or infrastructure such as buildings and drainage networks, and it uses closed domain boundaries. Future iterations could include these processes and features to create a more realistic dataset.

7.1.3 Performance of the Adapted Model and Computational Runtime

The third research question investigated how well the adapted mSWE-GNN emulates SFINCS simulated water-depth and flood extent, and compared its computational runtime relative to SFINCS. For SU rainfall events, the adapted mSWE-GNN achieved lower median RMSE than the bucket model across the validation and test splits (1.57-6.21 cm vs 10.49-17.37 cm). It also achieved higher median CSI values at both thresholds: 0.765-0.850 at the 0.05 m, compared with 0.168-0.240 for the bucket model, and 0.669-0.812 at the 0.30 m threshold, where the bucket model achieved a CSI of zero.

For SV rainfall events, three configurations were tested: current rainfall only (No rainfall context), current plus past rainfall (Past rainfall context), and current plus prescribed future rainfall (Future rainfall context). The Future rainfall context performed best, with median RMSE values of approximately 3–5 cm, compared with approximately 10–12 cm for the No rainfall context configuration, and higher CSI at both depth thresholds. Both the No rainfall context and Future rainfall context configurations outperformed the bucket model. In contrast, the Past rainfall context had higher RMSE than the bucket model and performed worse than the other two configurations across the evaluated metrics. Further exploration is needed to determine the cause of this poor performance of Past rainfall context. The SU model and the SV models with No and Future rainfall context outperformed the bucket model across the reported metrics, demonstrating predictive skill beyond local rainfall accumulation alone.

The adapted mSWE-GNN achieved substantially shorter GPU inference times than SFINCS on the CPU for both rainfall types. Under SU rainfall, inference required approximately 0.4 s per event for seen DEMs and 0.8 s for unseen DEMs, corresponding to an approximately 12–20 times faster runtime compared to SFINCS. Batching the 30 SU validation events provided an additional speed-up of approximately 2.65× relative to individual GPU inference. For SV validation events,

GPU inference took about 0.38 s per event versus 6.61 s for SFINCS on the CPU, corresponding to an approximately 17x speed-up. However, a preliminary CPU test found the mSWE-GNN to be slower than SFINCS on CPU.

7.1.4 Generalization Across Unseen Domains and Rainfall

The fourth research question explored the potential of the adapted mSWE-GNN to generalize to unseen domains, unseen rainfall, and both unseen. The model achieved reasonable accuracy across the test dataset.

Under SU rainfall, median RMSE increased from 1.57 cm for validation to 3.22 cm for unseen domains, 4.05 cm for unseen rainfall, and 6.21 cm for both unseen. Median CSI at 0.05 m ranged from 0.765 to 0.796 across these three test splits, while CSI at 0.30 m ranged from 0.669 to 0.770. Generally, sloped DEMs performed worse than their parent base DEMs, with higher RMSE and lower CSI. Investigating this further revealed that for some cases, the sloped DEMs overpredicted flooding in areas with lower-elevation, while the base DEMs showed more missed flooding.

Under SV rainfall, the Future rainfall context configuration generalized better than the No rainfall context configuration, with median RMSE values of approximately 3–5 cm compared with approximately 10–12 cm. The both unseen was the most difficult dataset split. Similar to SU rainfall, sloped DEMs generally performed worse than their parent base DEMs in matched comparisons. Further spatial analysis revealed that Future rainfall context reduced false positives (extra flooding) but, in some cases, increased false negatives (missed flooding) relative to the No rainfall context configuration.

7.2 Future Work

This thesis introduced several adaptations to the original mSWE-GNN. An ablation study is needed to quantify the contribution of these adaptations to model performance. In addition, hyperparameter optimization was conducted only for SU events. A separate optimization should therefore be performed for SV events.

The Past rainfall context should be investigated further. Its training became unstable, so further investigation is needed to determine whether the poor performance resulted from training irregularities or whether past rainfall information provides limited additional value to the model.

This thesis simplified the mSWE-GNN. Future work should incorporate spatially varying Manning's roughness as an input and also predict unit discharge together with water depth, as supported by the original model.

Finally, training and benchmarking should use more realistic simulations and, where available, observed flood events. This should include realistic terrain, infrastructure, infiltration, and open boundaries.

7.3 Societal Relevance

Pluvial flooding can disrupt communities and infrastructure, and timely flood information can support both preventive planning and emergency response. However, detailed flood modelling can become computationally expensive when large areas, high spatial resolution, or many rainfall scenarios must be assessed. Running an ensemble of simulations is particularly important for exploring uncertainty in rainfall and possible future conditions, but it increases this computational burden. In addition, setting up and validating models requires reliable local information on terrain, rainfall and flooding, which may be limited in data-scarce regions.

In data-scarce regions, deep-learning models may help reduce the need to build and run a new numerical model for every location, provided they can perform well beyond the conditions on which they were trained. This could make rapid flood-scenario assessment more accessible where local data and modelling resources are limited.

This thesis explored this potential by testing the adapted mSWE-GNN on unseen domain and rainfall. The model retained predictive skill across these unseen conditions, although performance varied with terrain and rainfall conditions. On GPU, it also achieved much shorter inference times than SFINCS, despite SFINCS already being a fast flood model.

References

[1] Luis Cea, Esteban Sañudo, Carlos Montalvo, Juan Farfán, Jerónimo Puertas, and Paolo Tamagnone. Recent advances and future challenges in urban pluvial flood modelling. *Urban Water Journal*, 22(2):149–173, 2025. doi: 10.1080/1573062X.2024.2446528. URL https://www.tandfonline.com/doi/full/10.1080/1573062X.2024.2446528.

[2] Kaihua Guo, Mingfu Guan, and Dapeng Yu. Urban surface water flood modelling—a comprehensive review of current models and future challenges. *Hydrology and Earth System Sciences*, 25:2843–2860, 2021. doi: 10.5194/hess-25-2843-2021. URL https://hess.copernicus.org/articles/25/2843/2021/.

[3] David Dodman, Bronwyn Hayward, Mark Pelling, Vanesa Castán Broto, Winston Chow, Eric Chu, Richard Dawson, Luna Khirfan, Timon McPhearson, Anjal Prakash, Yan Zheng, and Gina Ziervogel. Cities, settlements and key infrastructure. In *Climate Change 2022: Impacts, Adaptation and Vulnerability*, chapter 6, pages 907–1040. Cambridge University Press, Cambridge, UK and New York, NY, USA, 2022. doi: 10.1017/9781009325844.008. URL https://www.ipcc.ch/report/ar6/wg2/chapter/chapter-6/.

[4] Roberto Bentivoglio, Elvin Isufi, Sebastian Nicolaas Jonkman, and Riccardo Taormina. Deep learning methods for flood mapping: a review of existing applications and future research directions. *Hydrology and Earth System Sciences*, 26:4345–4378, 2022. doi: 10.5194/hess-26-4345-2022. URL https://hess.copernicus.org/articles/26/4345/2022/.

[5] Tim Leijnse, Maarten van Ormondt, Kees Nederhoff, and Ap van Dongeren. Modeling compound flooding in coastal systems using a computationally efficient reduced-physics solver: Including fluvial, pluvial, tidal, wind- and wave-driven processes. *Coastal Engineering*, 163:103796, 2021. doi: 10.1016/j.coastaleng.2020.103796. URL https://doi.org/10.1016/j.coastaleng.2020.103796.

[6] Tabea Cache, Milton Salvador Gomez, Tom Beucler, Jovan Blagojevic, João Paulo Leitao, and Nadav Peleg. Enhancing generalizability of data-driven urban flood models by incorporating contextual information. *Hydrology and Earth System Sciences*, 28:5443–5458, 2024. doi: 10.5194/hess-28-5443-2024. URL https://hess.copernicus.org/articles/28/5443/2024/.

[7] Chaopeng Shen, Alison P. Appling, Pierre Gentine, Toshiyuki Bandai, Hoshin Gupta, Alexandre Tartakovsky, Marco Baity-Jesi, Fabrizio Fenicia, Daniel Kifer, Li Li, Xiaofeng Liu, Wei Ren, Yi Zheng, Ciaran J. Harman, Martyn Clark, Matthew Farthing, Dapeng Feng, Praveen Kumar, Doaa Aboelyazeed, Farshid Rahmani, Yalan Song, Hylke E. Beck,

Tadd Bindas, Dipankar Dwivedi, Kuai Fang, Marvin Höge, Chris Rackauckas, Binayak Mohanty, Tirthankar Roy, Chonggang Xu, and Kathryn Lawson. Differentiable modelling to unify machine learning and physical models for geosciences. *Nature Reviews Earth & Environment*, 4(8):552–567, 2023. doi: 10.1038/s43017-023-00450-9. URL https://doi.org/10.1038/s43017-023-00450-9.

[8] Roberto Bentivoglio, Elvin Isufi, Sebastiaan Nicolas Jonkman, and Riccardo Taormina. Multi-scale hydraulic graph neural networks for flood modelling. *Natural Hazards and Earth System Sciences*, 25(1):335–351, 2025. doi: 10.5194/nhess-25-335-2025. URL https://nhess.copernicus.org/articles/25/335/2025/.

[9] Roberto Bentivoglio, Elvin Isufi, Sebastiaan Nicolas Jonkman, and Riccardo Taormina. Rapid spatio-temporal flood modelling via hydraulics-based graph neural networks. *Hydrology and Earth System Sciences*, 27:4227–4246, 2023. doi: 10.5194/hess-27-4227-2023. URL https://hess.copernicus.org/articles/27/4227/2023/.

[10] Yaoxing Liao, Zhaoli Wang, Haijun Yu, Weizhi Gao, Zhaoyang Zeng, Xuefang Li, and Chengguang Lai. Accelerating urban flood inundation simulation under spatio-temporally varying rainstorms using ConvLSTM deep learning model. *Water Resources Research*, 61 (8):e2025WR040433, 2025. doi: 10.1029/2025WR040433.

[11] Yang Liu and Gareth Pender. A flood inundation modelling using ν-support vector machine regression model. *Engineering Applications of Artificial Intelligence*, 46:223–231, 2015. doi: 10.1016/j.engappai.2015.09.014. URL https://doi.org/10.1016/j.engappai.2015.09.014.

[12] María Bermúdez, Luis Cea, and Jerónimo Puertas. A rapid flood inundation model for hazard mapping based on least squares support vector machine regression. *Journal of Flood Risk Management*, 12(S1):e12522, 2019. doi: 10.1111/jfr3.12522. URL https://doi.org/10.1111/jfr3.12522.

[13] Simon Berkhahn, Lothar Fuchs, and Insa Neuweiler. An ensemble neural network model for real-time prediction of urban floods. *Journal of Hydrology*, 575:743–754, 2019. doi: 10.1016/j.jhydrol.2019.05.066. URL https://doi.org/10.1016/j.jhydrol.2019.05.066.

[14] Faria T. Zahura, Jonathan L. Goodall, Jeffrey M. Sadler, Yawen Shen, Mohamed M. Morsy, and Madhur Behl. Training machine learning surrogate models from a high-fidelity physics-based model: Application for real-time street-scale flood prediction in an urban coastal community. *Water Resources Research*, 56(10):e2019WR027038, 2020. doi: 10.1029/2019WR027038. URL https://doi.org/10.1029/2019WR027038.

[15] Syed Kabir, Sandhya Patidar, Xilin Xia, Qiuhua Liang, Jeffrey Neal, and Gareth Pender. A deep convolutional neural network model for rapid prediction of fluvial flood inundation. *Journal of Hydrology*, 590:125481, 2020. doi: 10.1016/j.jhydrol.2020.125481. URL https://doi.org/10.1016/j.jhydrol.2020.125481.

[16] Roland Löwe, Julian Böhm, David Getreuer Jensen, Jorge Leandro, and Søren Højmark Rasmussen. U-FLOOD – topographic deep learning for predicting urban pluvial flood water

depth. *Journal of Hydrology*, 603:126898, 2021. doi: 10.1016/j.jhydrol.2021.126898. URL https://doi.org/10.1016/j.jhydrol.2021.126898.

[17] Cesar A. F. do Lago, Marcio H. Giacomoni, Roberto Bentivoglio, Riccardo Taormina, Marcus N. Gomes Junior, and Eduardo M. Mendiondo. Generalizing rapid flood predictions to unseen urban catchments with conditional generative adversarial networks. *Journal of Hydrology*, 618:129276, 2023. doi: 10.1016/j.jhydrol.2023.129276. URL https://doi.org/10.1016/j.jhydrol.2023.129276.

[18] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. doi: 10.48550/arXiv.2104.13478. URL https://arxiv.org/abs/2104.13478.

[19] Arnold Kazadi, James Doss-Gollin, Antonia Sebastian, and Arlei Silva. FloodGNN-GRU: A spatio-temporal graph neural network for flood prediction. *Environmental Data Science*, 3: e21, 2024. doi: 10.1017/eds.2024.19. URL https://doi.org/10.1017/eds.2024.19.

[20] Carlo Malapad Acosta, Herath Mudiyanselage Viraj Vidura Herath, Jia Yu Lim, Abhishek Saha, Sanka Rasnayaka, and Lucy Marshall. DUALFloodGNN: Physics-informed graph neural network for operational flood modeling. In Diego Calvanese, editor, *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-26*, pages 6408–6416. International Joint Conferences on Artificial Intelligence Organization, August 2026. doi: 10.24963/ijcai.2026/713. URL https://doi.org/10.24963/ijcai.2026/713.

[21] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48. Association for Computing Machinery, 2009. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

[22] Luc D'Costa, Yidi Wang, Jonathan L. Goodall, and Rohan Chandra. Physics-informed CNN-LSTM for street-scale urban flood prediction: Reconciling aggregate accuracy and street-level plausibility, 2026. URL https://arxiv.org/abs/2607.25148.

[23] Moctar Dembélé, Markus Hrachowitz, Hubert H. G. Savenije, Grégoire Mariéthoz, and Bettina Schaefli. Improving the predictive skill of a distributed hydrological model by calibration on spatial patterns with multiple satellite data sets. *Water Resources Research*, 56(1):e2019WR026085, 2020. doi: 10.1029/2019WR026085.

AI Declaration

During the preparation of this thesis report, I provided draft prose to AI tools (Claude) for assistance with restructuring, grammar correction and language refinement. I critically reviewed and revised the suggestions before incorporating them into the thesis. I remain responsible for the accuracy of the content, the interpretation of results and the final conclusions.

Codex and Copilot were used to assist with developing, debugging and refining programming code for modelling, data analysis and visualisation. I reviewed and tested the generated code and checked its outputs before using them in this thesis.

Appendix A

Terrain, Rainfall and SFINCS Configuration

A.1 Terrain Selection and Slope Construction

The base terrains come from the Perlin-noise library of [9]. The library contains 121 smaller DEMs (6.4 km × 6.4 km, 64 × 64 cells) and ten larger DEMs (12.8 km × 12.8 km, 128 × 128 cells). The cell size in all DEMs is 100 m x 100 m.

This study used 20 of the smaller DEMs and all ten larger DEMs. The larger DEMs are reserved for evaluation on unseen domains. Relief is the difference between the highest and lowest elevation in a DEM. Table A.1 summarizes the relief of the base DEMs used in this thesis.

Table A.1: Relief of the selected base DEMs.

Grid	Count	Minimum (m)	Maximum (m)	Mean (m)
64 × 64	20	1.67	7.54	4.38
128 × 128	10	1.60	6.50	3.78

Sloped terrains were created from 16 of the 30 base DEMs. Ten smaller DEMs received a slope (five with 5 m and five with 10 m), and six larger DEMs received a slope (three with 5 m and three with 10 m). A linear slope with an added elevation difference of 5 or 10 m, centered at zero, is added along a randomly selected direction.

To evaluate the effect of the added slope, both the sloped terrains and their parent base DEMs are retained. The final library consists of 46 DEMs. Table A.2 and Table A.3 summarizes the base DEMs, the added slope, and its direction in degrees.

Table A.2: Selected base DEMs and slope settings for the SU dataset. A dash means that no slope was added.

Grid	Parent DEM	Base relief (m)	H_{add} (m)	Direction (°)	Final relief (m)
64 × 64	DEM_41	2.48	5	230.19	5.41
64 × 64	DEM_37	1.83	5	9.00	5.45
64 × 64	DEM_62	2.25	5	99.01	6.42
64 × 64	DEM_10002	1.95	5	80.36	5.93
64 × 64	DEM_511	1.67	5	265.13	5.95
64 × 64	DEM_38	4.10	10	243.61	11.35
64 × 64	DEM_47	3.45	10	321.18	10.45
64 × 64	DEM_3	4.02	10	31.30	10.83
64 × 64	DEM_512	3.57	10	151.89	11.21
64 × 64	DEM_30	3.11	10	10.73	10.73
64 × 64	DEM_20	5.45	–	–	–
64 × 64	DEM_514	4.81	–	–	–
64 × 64	DEM_10012	5.26	–	–	–
64 × 64	DEM_56	4.91	–	–	–
64 × 64	DEM_10014	4.51	–	–	–
64 × 64	DEM_500	7.54	–	–	–
64 × 64	DEM_515	6.62	–	–	–
64 × 64	DEM_23	7.11	–	–	–
64 × 64	DEM_59	6.85	–	–	–
64 × 64	DEM_5	6.07	–	–	–
128 × 128	DEM_15001	6.50	5	78.71	9.83
128 × 128	DEM_15002	3.99	5	181.93	8.15
128 × 128	DEM_15003	2.76	5	9.55	6.89
128 × 128	DEM_15004	4.89	10	71.58	11.28
128 × 128	DEM_15005	1.60	10	233.96	10.06
128 × 128	DEM_15006	2.12	10	196.18	10.74
128 × 128	DEM_15007	6.13	–	–	–
128 × 128	DEM_15008	1.97	–	–	–
128 × 128	DEM_15009	4.74	–	–	–
128 × 128	DEM_15010	3.06	–	–	–

Table A.3: Selected base DEMs and slope settings for the SV dataset. A dash means that no slope was added.

Grid	Parent DEM	Base relief (m)	H_{add} (m)	Direction (°)	Final relief (m)
64 × 64	DEM_41	2.48	10	243.61	10.26

Continued on next page

Grid	Parent DEM	Base relief (m)	H_{add} (m)	Direction (°)	Final relief (m)
64×64	DEM_37	1.83	5	230.19	5.69
64×64	DEM_62	2.25	5	99.01	6.42
64×64	DEM_10002	1.95	–	–	–
64×64	DEM_511	1.67	–	–	–
64×64	DEM_51	4.10	5	80.36	8.12
64×64	DEM_39	3.48	10	31.30	11.22
64×64	DEM_75	4.05	–	–	–
64×64	DEM_77	3.61	–	–	–
64×64	DEM_70	3.14	–	–	–
64×64	DEM_12	5.46	–	–	–
64×64	DEM_71	4.85	5	265.13	7.92
64×64	DEM_13	5.30	–	–	–
64×64	DEM_10004	4.92	10	151.89	11.87
64×64	DEM_507	4.52	–	–	–
64×64	DEM_53	7.62	–	–	–
64×64	DEM_19	6.76	–	–	–
64×64	DEM_517	7.26	5	9.00	9.22
64×64	DEM_55	6.89	10	321.18	12.72
64×64	DEM_24	6.07	10	10.73	13.80
128×128	DEM_15001	6.50	5	9.55	10.14
128×128	DEM_15002	3.99	10	233.96	11.23
128×128	DEM_15003	2.76	10	71.58	11.21
128×128	DEM_15004	4.89	–	–	–
128×128	DEM_15005	1.60	–	–	–
128×128	DEM_15006	2.12	5	181.93	6.24
128×128	DEM_15007	6.13	–	–	–
128×128	DEM_15008	1.97	–	–	–
128×128	DEM_15009	4.74	5	78.71	8.04
128×128	DEM_15010	3.06	10	196.18	10.92

A.2 Spatially Varying Rainfall Generation

The method used to generate the SV events is adapted from Liao et al. [10]. It uses a Gaussian spatial field, which is highest at the center of a rain cloud and decays with distance from the center. For cell i with coordinates (x_i, y_i) and cloud center $(x_c(t), y_c(t))$, the field is

$$G_i(t) = \exp\left[-\frac{(x_i - x_c(t))^2 + (y_i - y_c(t))^2}{2\sigma^2}\right], \tag{A.1}$$

where σ controls the width of the cloud. The width σ is chosen to be 20% of the domain width for the thesis. The field is multiplied by the rainfall intensity to give the SV rainfall. It is scaled

at each timestamp so that its domain mean equals the SU intensity. In this thesis, the SU hyetographs provide the rainfall intensity at each timestamp.

The cloud center is defined for each timestamp. Five storm movements are considered. In the stationary case, the rain cloud stays at the center of the domain. In the four moving cases, it travels in a straight line across the domain from left to right, right to left, bottom to top, or top to bottom. The SV rainfall of each event is saved as a NetCDF file.

A.3 SFINCS Configuration and Input/Output Specifications

SFINCS-LIE is based on the local inertial equations, which neglect advection. SFINCS-SSWE extends the LIE formulation by adding advection [5]. This thesis uses the SSWE setting, with advection enabled for all simulations. Table A.4 lists the main settings.

Each simulation combines several input files, which are passed to the SFINCS solver to generate the outputs. Table A.5 describes the files. Table A.6 defines the numerical parameters of the model.

Table A.4: SFINCS settings

Item	Setting
Solver version	SFINCS 2.3.0
Interface	HydroMT-SFINCS
Equations	SSWE, advection enabled
Grid	64×64 or 128×128 cells of 100 m $\times$ 100 m
Event duration	45 h
Output interval	30 min (1,800 s)
Initial condition	Dry domain
Manning coefficient	0.04 s $m^{-1/3}$, uniform
Boundaries	Closed, all cells active
Infiltration	Set to zero

Table A.5: SFINCS input files.

File	Content
`sfincs.inp`	Parameters for model input and output
`sfincs.dep`	Elevation of the DEM (bathymetry and topography)
`sfincs.msk`	Status of each cell: inactive (0), active (1), boundary (2), or outflow boundary (3). All cells are active in this thesis.
`sfincs.prcp`	Spatially uniform rainfall
`precip*.nc`	Spatially varying rainfall as a NetCDF file

Table A.6: Numerical parameters of the SFINCS model.

Parameter	Value	Meaning
alpha	0.5 (default)	Time-step reduction factor for the CFL condition
theta	1 (default)	Smoothing factor for the momentum equation. A value of 1 applies no smoothing.
huthresh	–	Minimum water depth required to compute flow between neighboring cells
zsini	−999	Initial water level across the domain. This value is used to start the domain dry.

Appendix B

Graph Construction, Data Normalization and Training Configuration

B.1 Multiscale Graph Construction

Each SFINCS simulation is converted into one multiscale graph. Each grid cell is a node, and each node is connected to its four cardinal neighbors. The distance between connected nodes is stored as an edge feature.

The graph has three resolutions: fine, medium, and coarse. The fine graph uses the original grid, which is either 64×64 or 128×128 cells. The medium graph is created by 2×2 mean pooling of the fine graph, and the coarse graph by 2×2 mean pooling of the medium graph. Table B.1 gives the resulting sizes for the 6.4 km $\times$ 6.4 km domains. Pooling maps store which child nodes belong to each parent node in the next coarser graph.

Table B.1: Graph sizes for the 6.4 km $\times$ 6.4 km domains.

Level	Cell size (m)	Grid	Nodes
Fine	100	64×64	4,096
Medium	200	32×32	1,024
Coarse	400	16×16	256

Each graph stores the elevation at all three resolutions, the connectivity, the edge distances, the node coordinates, the pooling maps, the water depth frames, and the rainfall. The graphs are saved separately for the training, validation, and test splits.

B.2 Normalization

The graph contains variables with different units and magnitudes. They are normalized to unitless values, so that variables with large magnitudes do not dominate training. The variables

are water depth and elevation (m), rainfall intensity (mm/h), and edge distance (m).

The normalization parameters are computed from the training samples only, so that no information from the validation or test sets is used. Water depth, rainfall intensity, and edge distance are divided by their maximum value in the training samples. Dividing by the maximum preserves the zero as zero. Elevation is centered by subtracting its mean in the training samples and then divided by the water depth scale:

$$\hat{h} = \frac{h}{h_{\text{scale}}}, \qquad \hat{P} = \frac{P}{P_{\text{scale}}}, \qquad \hat{z} = \frac{z - z_{\text{mean}}}{h_{\text{scale}}}, \qquad \hat{d} = \frac{d}{d_{\text{scale}}}. \tag{B.1}$$

Here h, P, z, and d are the raw water depth, rainfall intensity, elevation, and edge distance, and the hat marks the normalized value. Elevation uses h_{scale} so that normalized elevation and normalized water depth share the same scale. This is needed because the two are added to give the water level.

B.3 Temporal Samples and Training Settings

Each graph contains 91 water depth frames. The graph is split along time into input-output samples. The input of one sample consists of three consecutive water depth frames and the current rainfall,

$$x_t = [h_{t-2}, h_{t-1}, h_t, P_t]. \tag{B.2}$$

The target contains the next six water depth frames,

$$y_t = [h_{t+1}, h_{t+2}, h_{t+3}, h_{t+4}, h_{t+5}, h_{t+6}]. \tag{B.3}$$

The rainfall series from P_t to P_{t+5} is also stored. Each forward pass predicts one water depth frame, and the six frames are predicted one after another. The six targets are stored for curriculum learning. Each graph gives 83 samples. Table B.2 lists the training settings.

Table B.2: Training settings

Item	Setting
Input history	3 depth fields
Dynamic input features	4 for current rainfall only; 6 for Past and Future rainfall context
Hidden features	128
Message-passing blocks per processor	3
Optimizer	Adam
Initial learning rate	0.001
Learning-rate schedule	StepLR, step size 20 epochs, factor 0.7
Batch size	8 temporal samples
Maximum epochs	200
Curriculum	One step every 15 epochs, maximum 6
Validation horizon	6 steps
Early-stopping patience	30 epochs, counted at the full horizon
Gradient clipping	Global norm 1.0
Mixed precision	Float16 autocast on GPU, training only
Random seed	42
Gate threshold	0.01 m
Loss weights	7 (reference depth $\geq$ 0.05 m), 1 otherwise

Appendix C

Supplementary Results under Spatially Uniform Rainfall

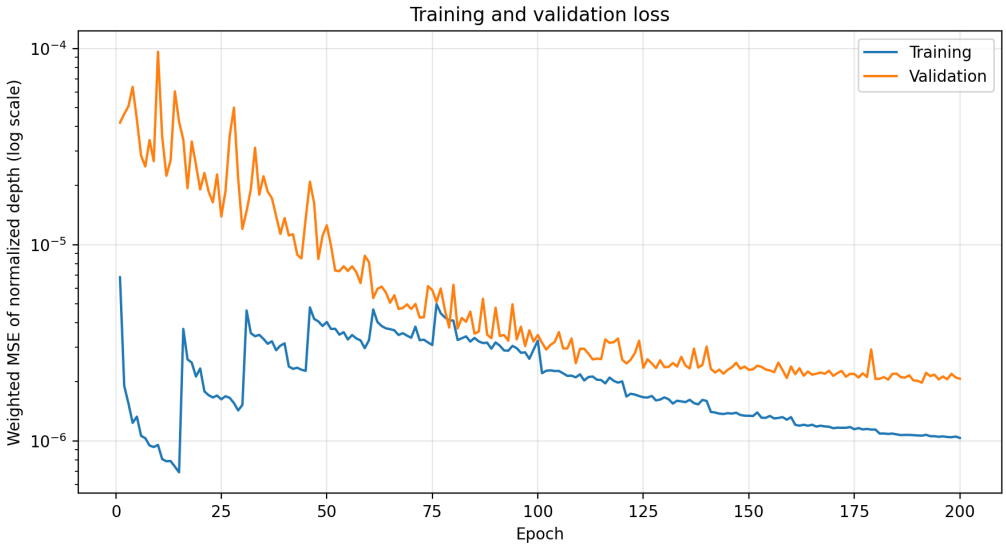


Figure C.1: Training and validation wet-weighted MSE on normalized water depths.

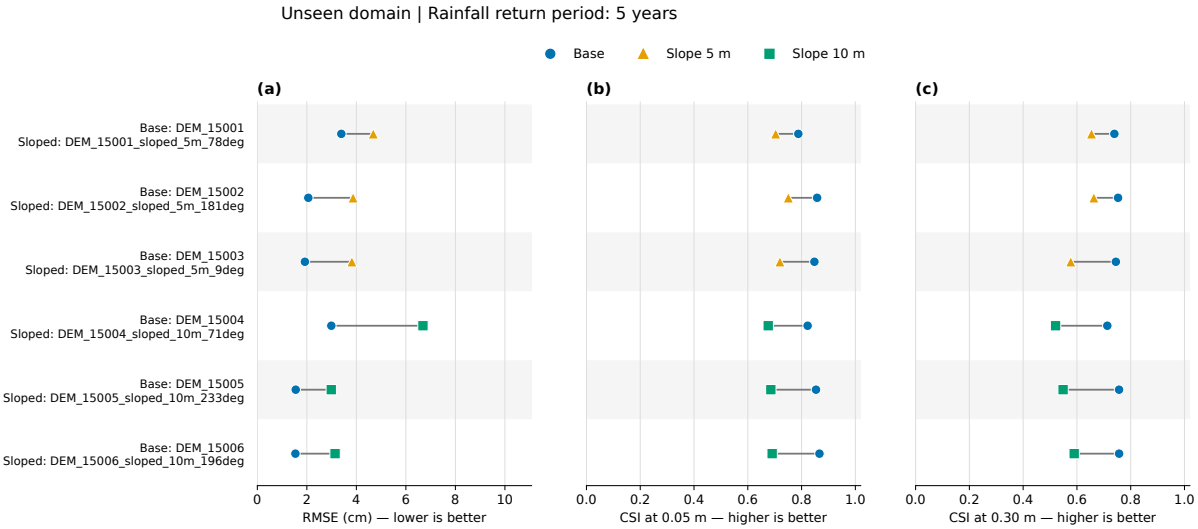


Figure C.2: Base and sloped DEMs for unseen domain under RP5 rainfall: (a) event-average RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m.

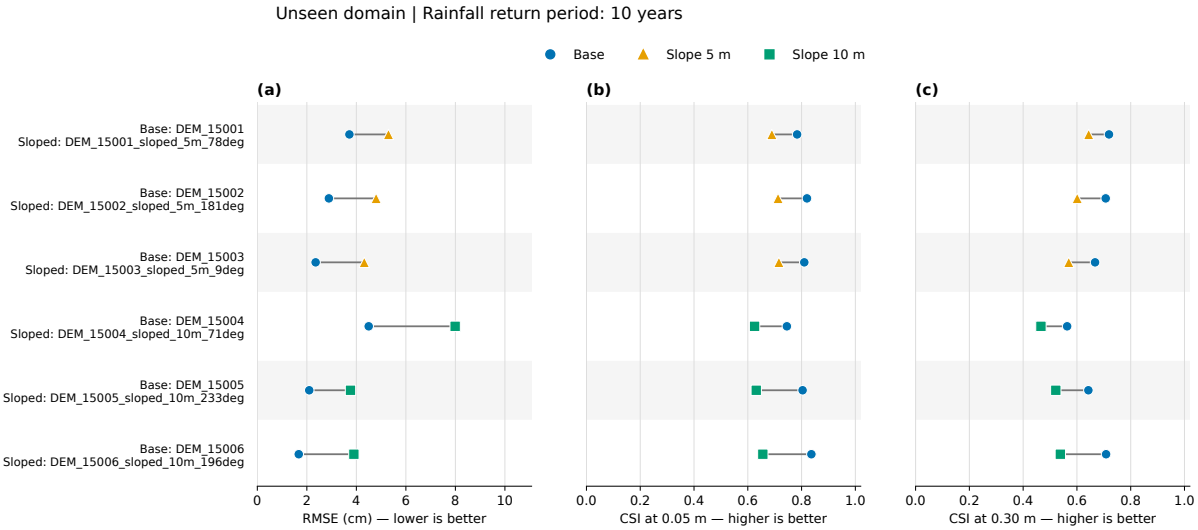


Figure C.3: Base and sloped DEMs for unseen domain under RP10 rainfall: (a) event-average RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m.

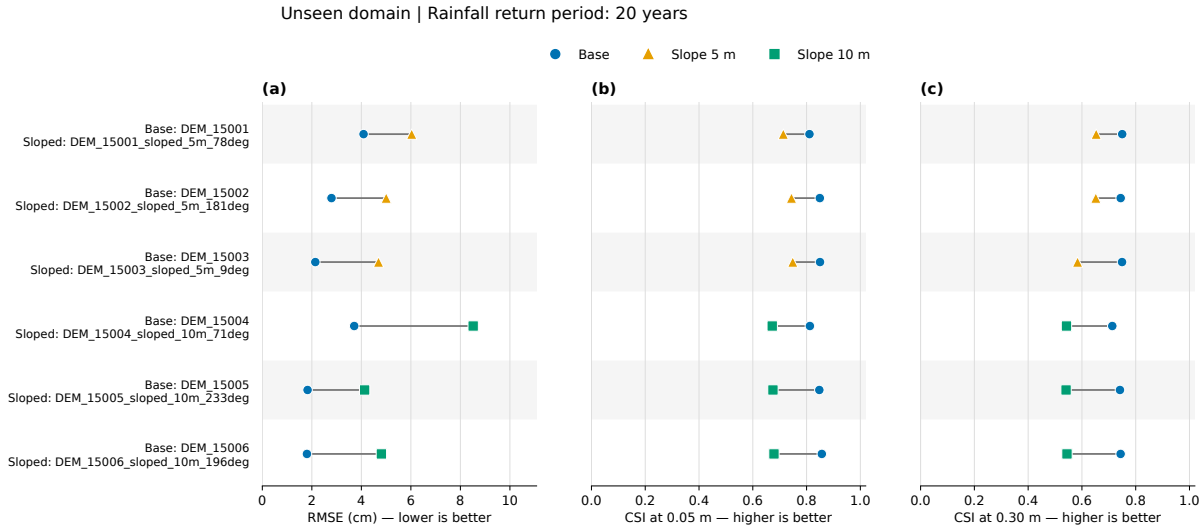


Figure C.4: Base and sloped DEMs for unseen domain under RP20 rainfall: (a) event-average RMSE, (b) CSI at 0.05 m, and (c) CSI at 0.30 m.

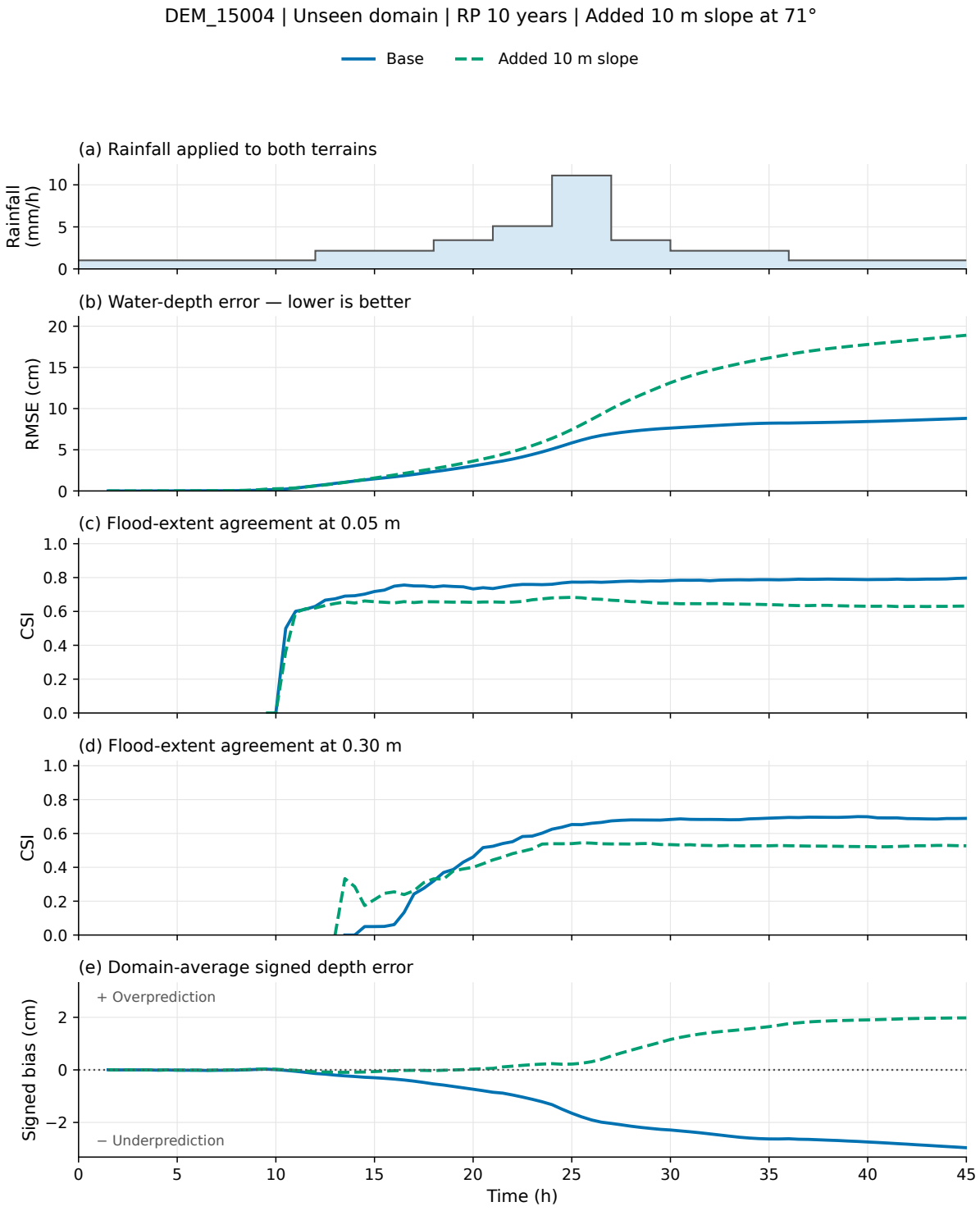


Figure C.5: Temporal performance of the base and 10 m sloped versions of DEM 15004 under RP10 rainfall in the unseen domain.

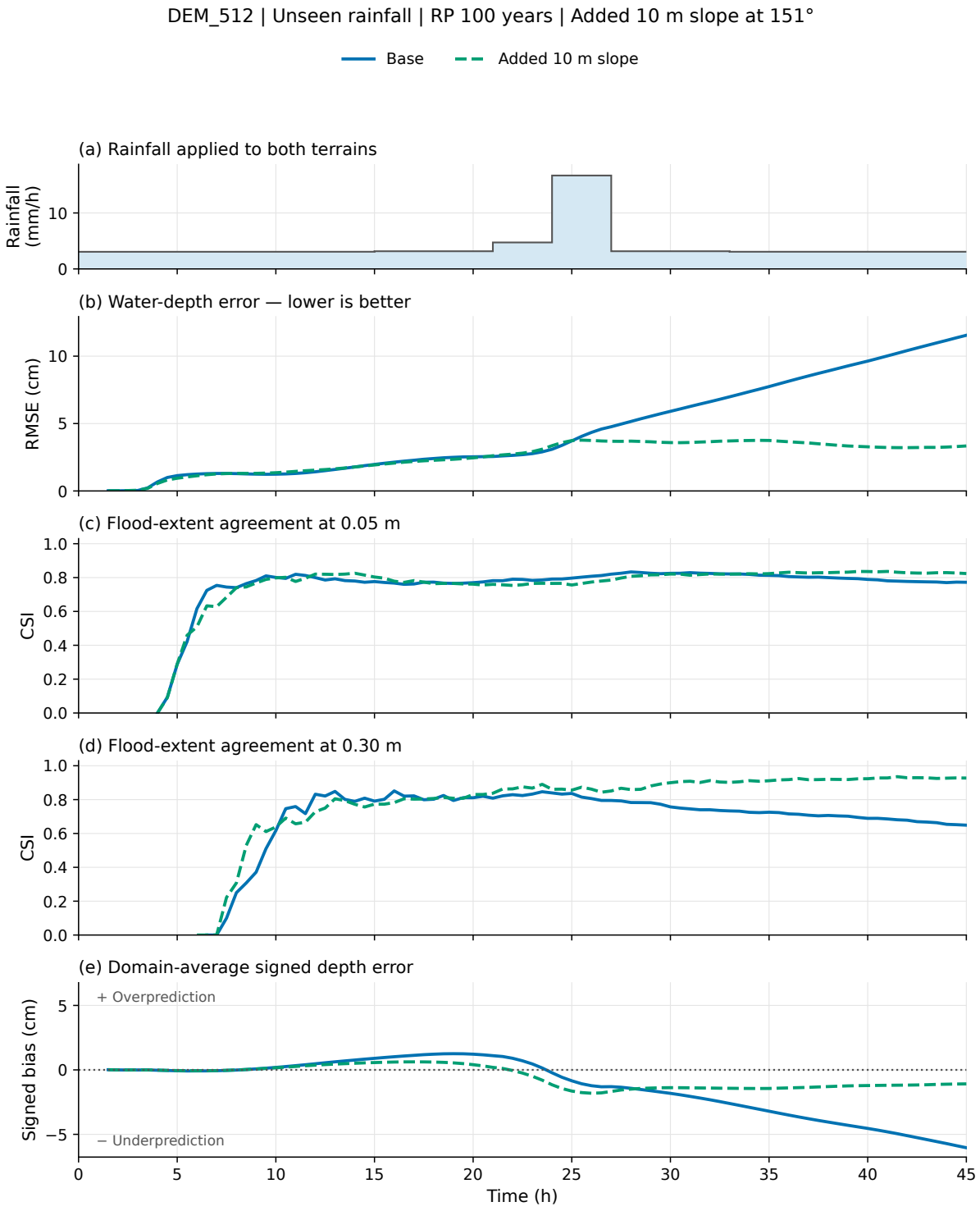


Figure C.6: Temporal performance of the base and 10 m sloped versions of DEM 512 under RP100 rainfall in the unseen rainfall.

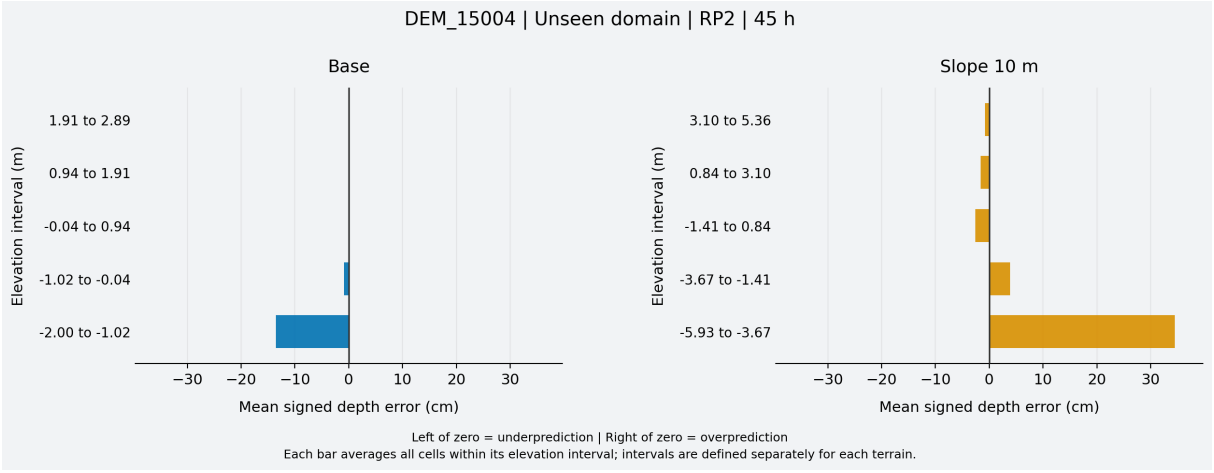


Figure C.7: Mean signed depth error (GNN minus SFINCS) by elevation interval for DEM 15004 and its 10 m sloped version at 45 h under RP2 rainfall in the unseen domain split.

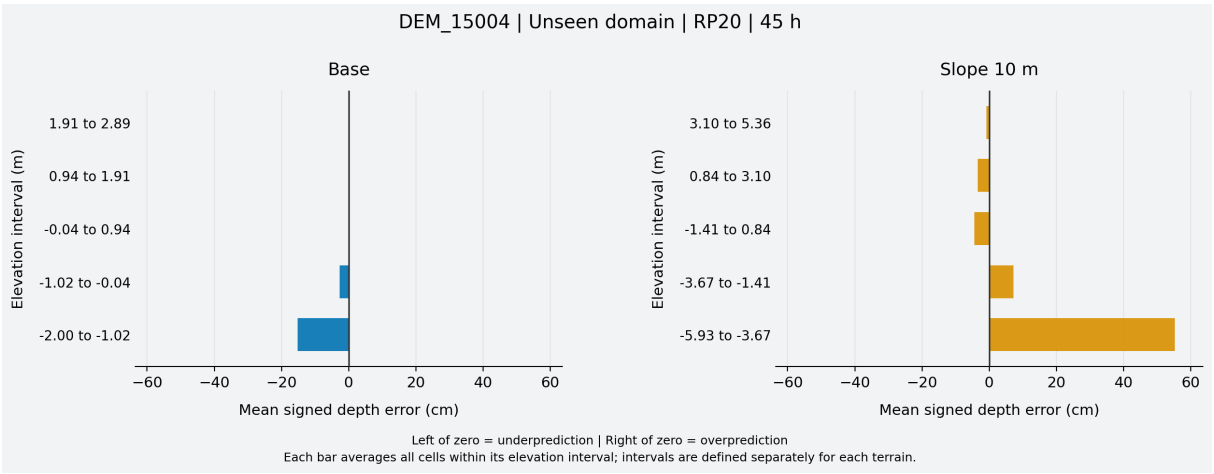


Figure C.8: Mean signed depth error (GNN minus SFINCS) by elevation interval for DEM 15004 and its 10 m sloped version at 45 h under RP20 rainfall in the unseen domain split.

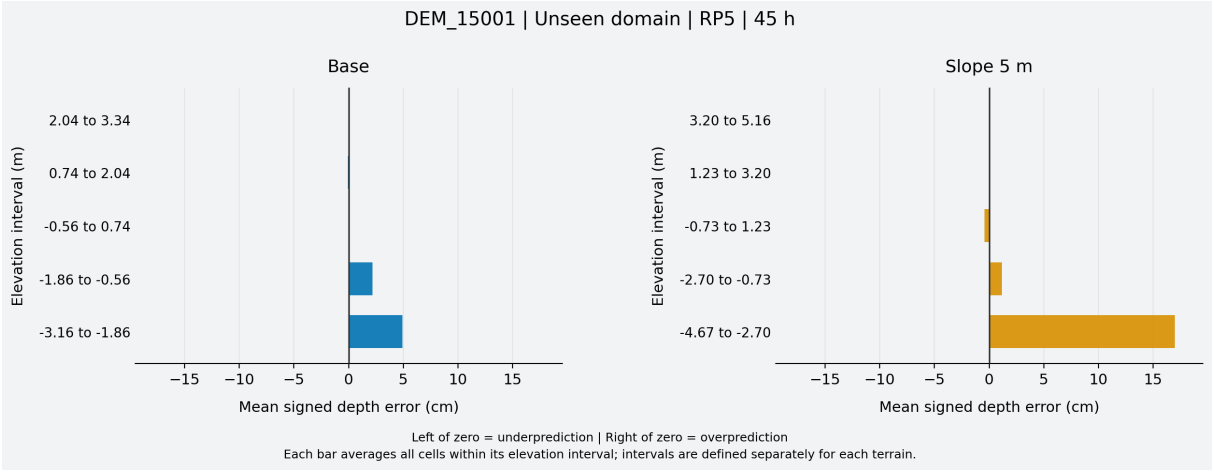


Figure C.9: Mean signed depth error (GNN minus SFINCS) by elevation interval for DEM 15001 and its 5 m sloped version at 45 h under RP5 rainfall in the unseen domain split.

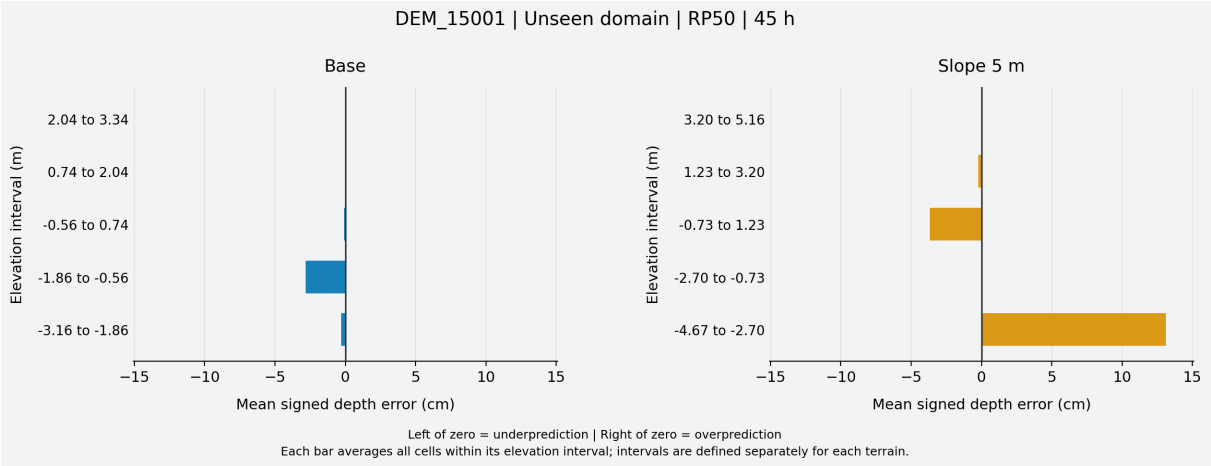


Figure C.10: Mean signed depth error (GNN minus SFINCS) by elevation interval for DEM 15001 and its 5 m sloped version at 45 h under RP50 rainfall in the unseen domain split.

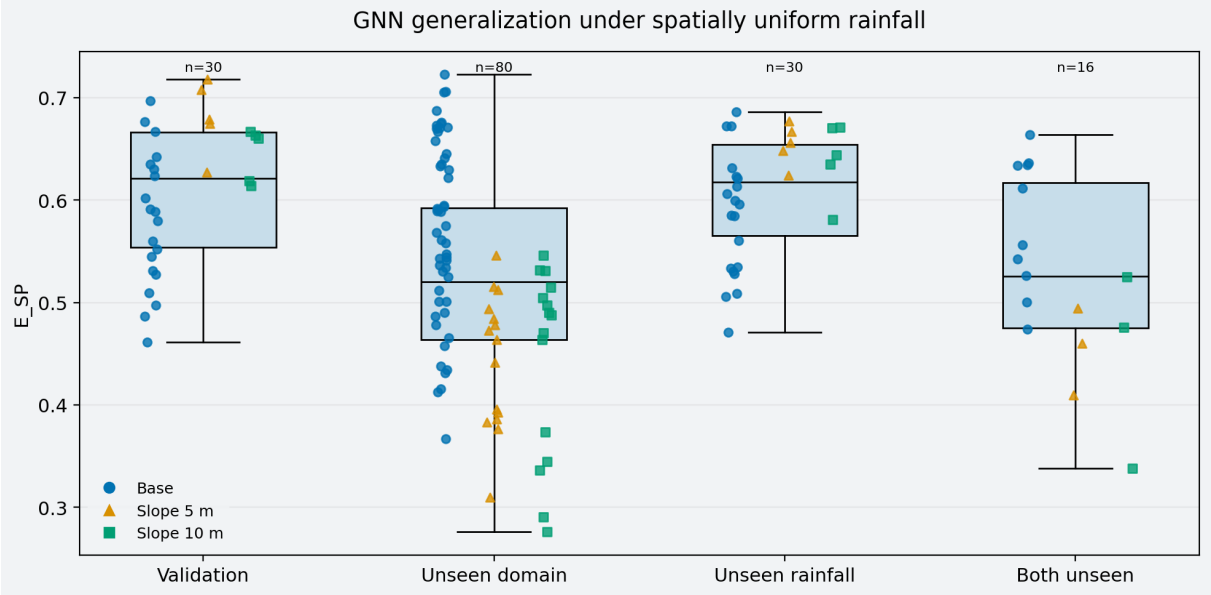


Figure C.11: Event-average spatial pattern efficiency (E_{SP}) of the adapted mSWE-GNN under SU rainfall. Higher values indicate better agreement with SFINCS.

Appendix D

Supplementary Results under Spatially Varying Rainfall

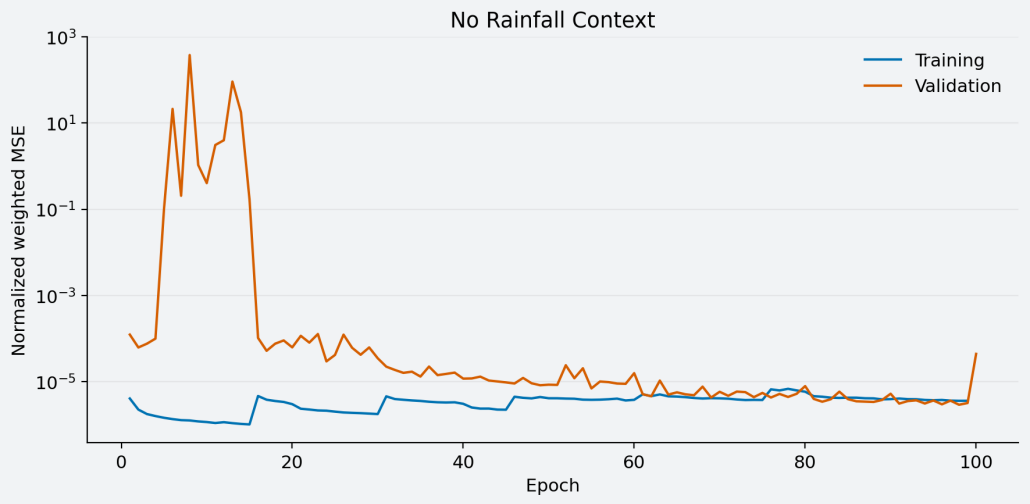


Figure D.1: Training and validation wet-weighted MSE on normalized water depths for the SV model with No rainfall context.

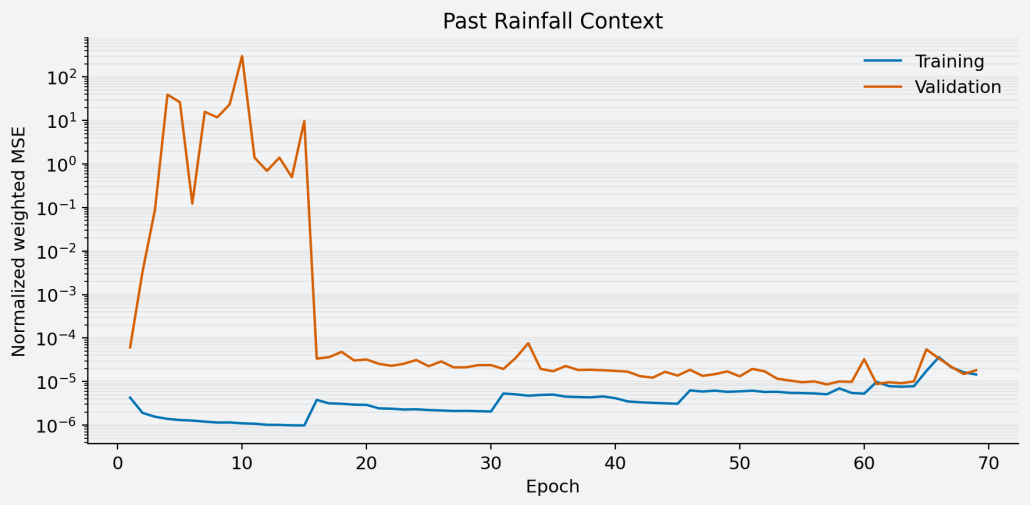


Figure D.2: Training and validation wet-weighted MSE on normalized water depths for the SV model with Past rainfall context.

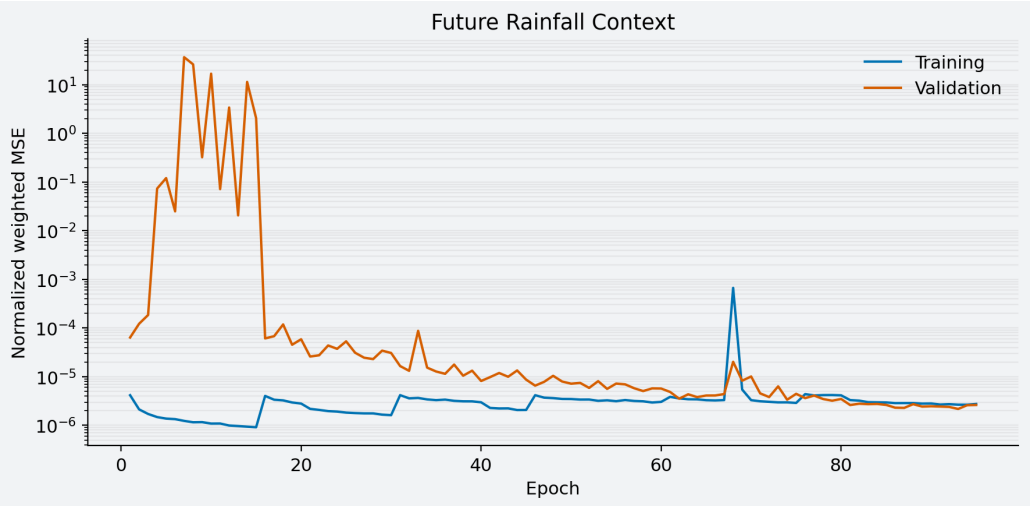


Figure D.3: Training and validation wet-weighted MSE on normalized water depths for the SV model with Future rainfall context.

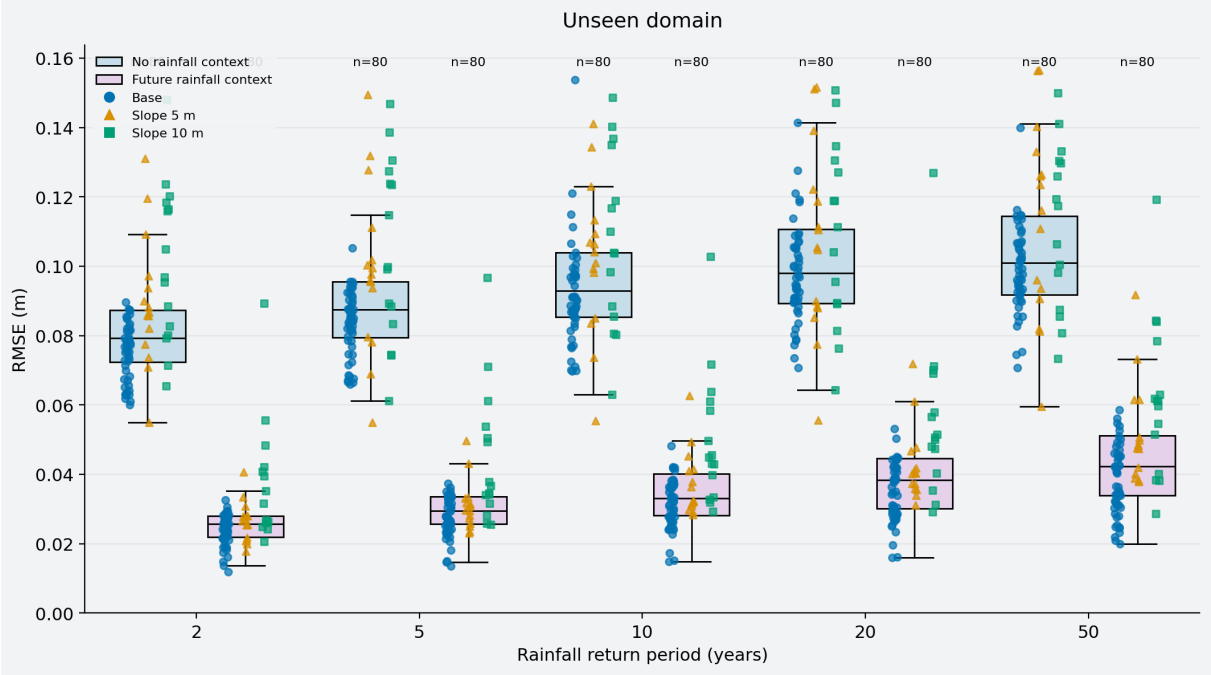


Figure D.4: Event-average RMSE across rainfall return periods in the SV unseen domain split, comparing No rainfall context and Future rainfall context.

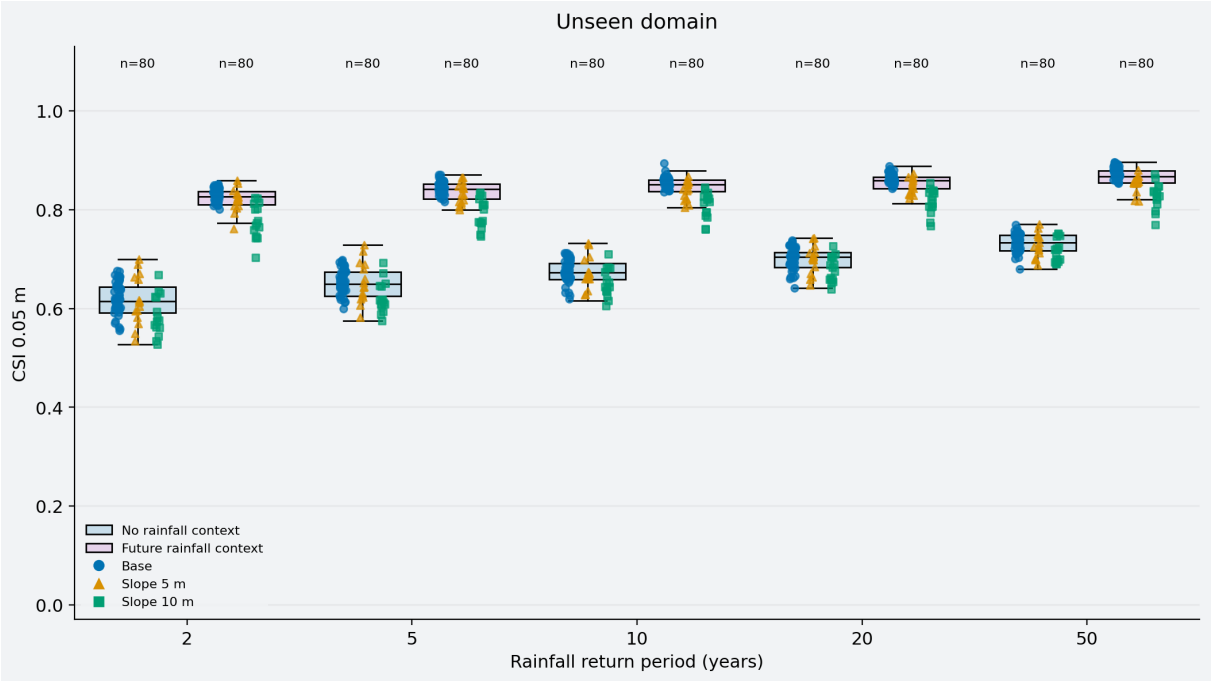


Figure D.5: Event-average CSI at 0.05 m across rainfall return periods in the SV unseen-domain split, comparing No rainfall context and Future rainfall context.

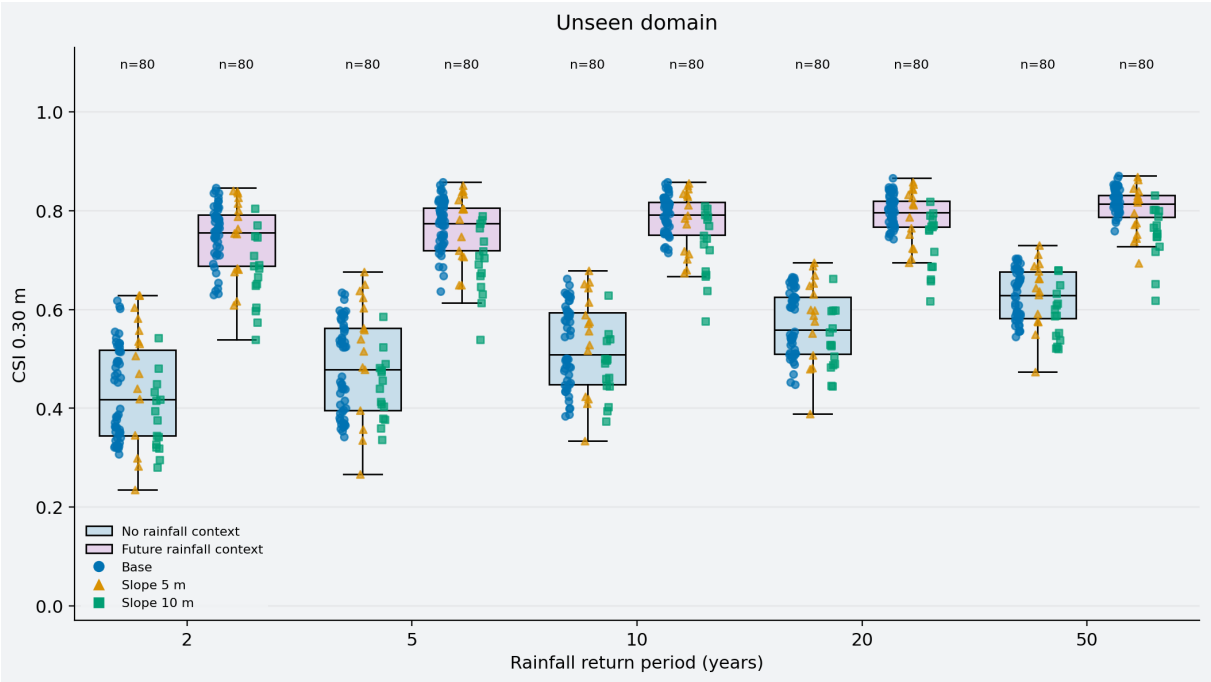


Figure D.6: Event-average CSI at 0.30 m across rainfall return periods in the SV unseen-domain split, comparing No rainfall context and Future rainfall context.

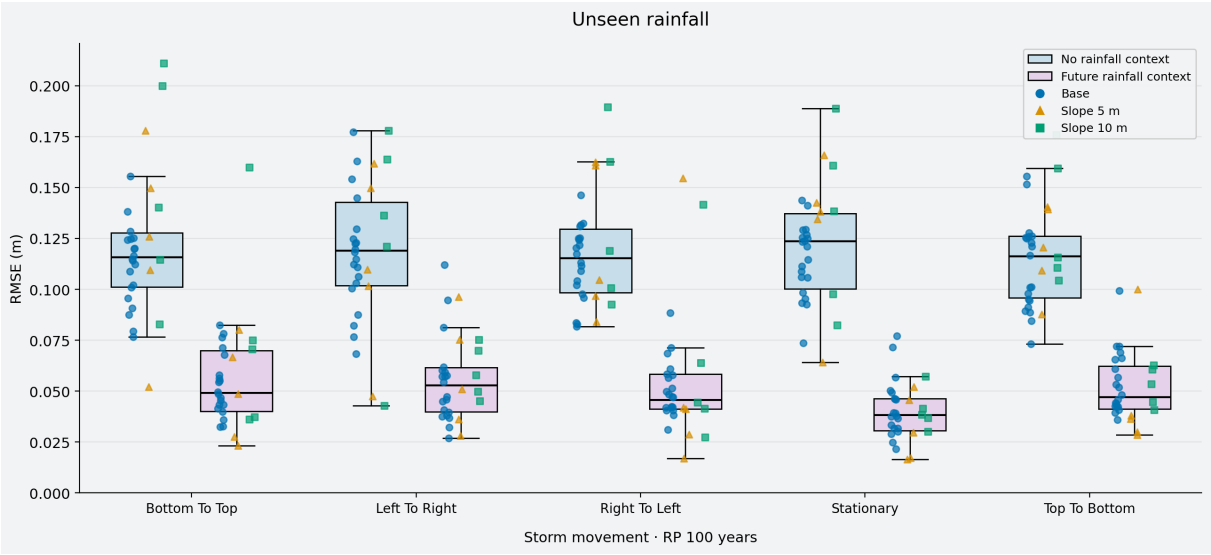


Figure D.7: Event-average RMSE across five storm-movement patterns under RP100 rainfall in the SV unseen rainfall split, comparing No rainfall context and Future rainfall context.

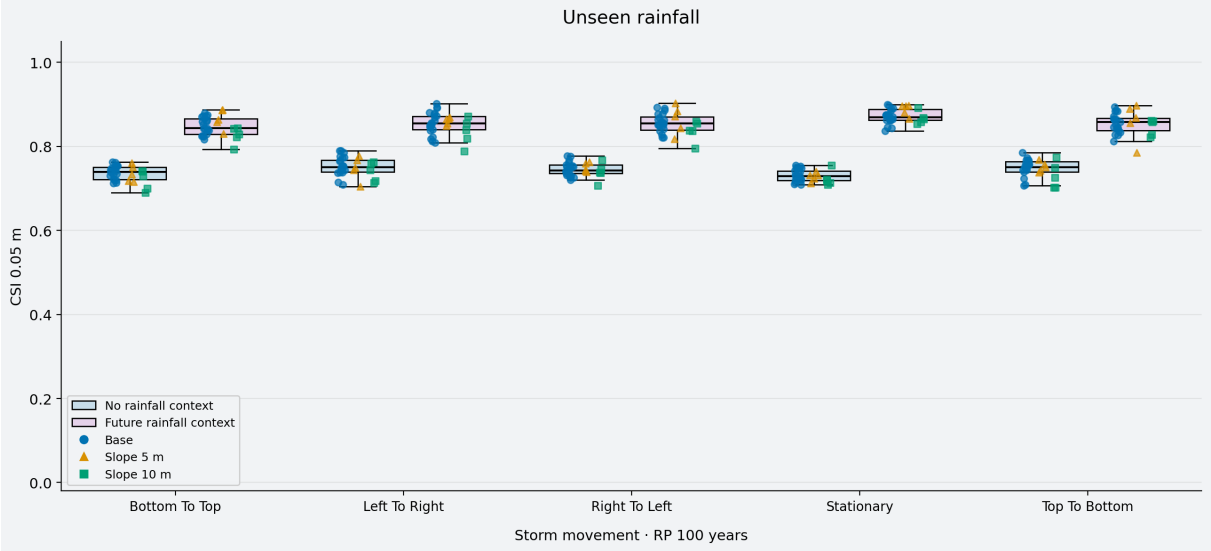


Figure D.8: Event-average CSI at 0.05 m across five storm-movement patterns under RP100 rainfall in the SV unseen-rainfall split, comparing No rainfall context and Future rainfall context.

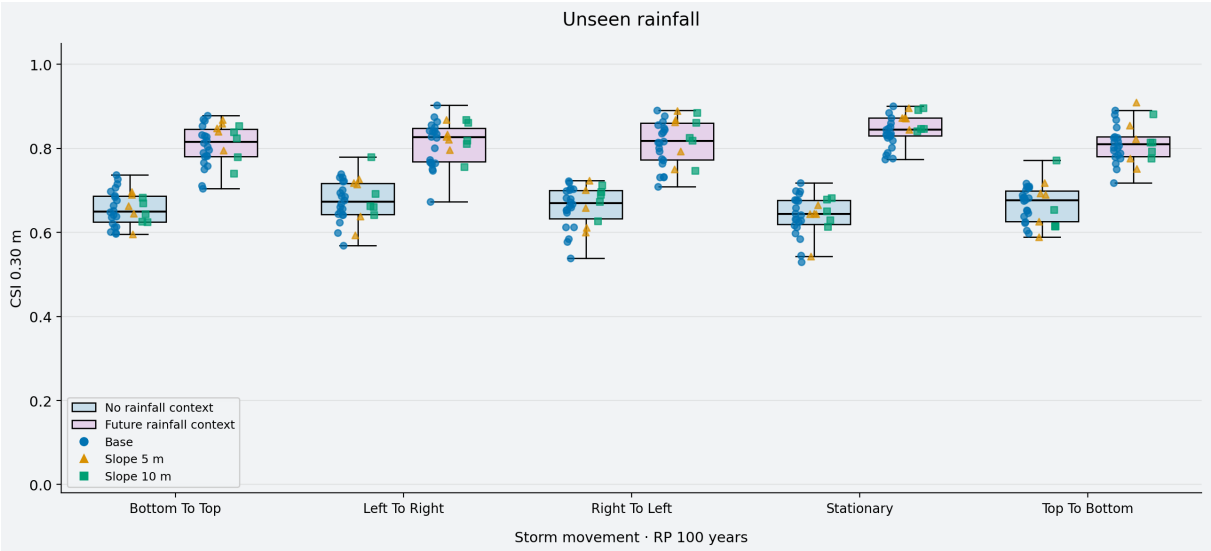


Figure D.9: Event-average CSI at 0.30 m across five storm-movement patterns under RP100 rainfall in the SV unseen-rainfall split, comparing No rainfall context and Future rainfall context.

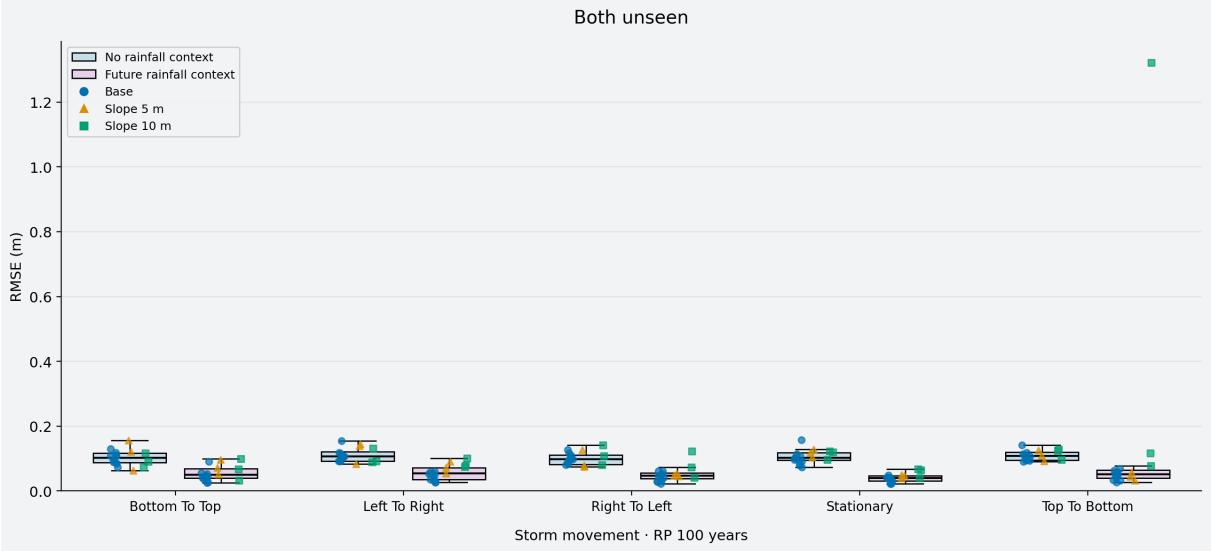


Figure D.10: Event-average RMSE across five storm-movement patterns under RP100 rainfall in the SV both unseen split, comparing No rainfall context and Future rainfall context.

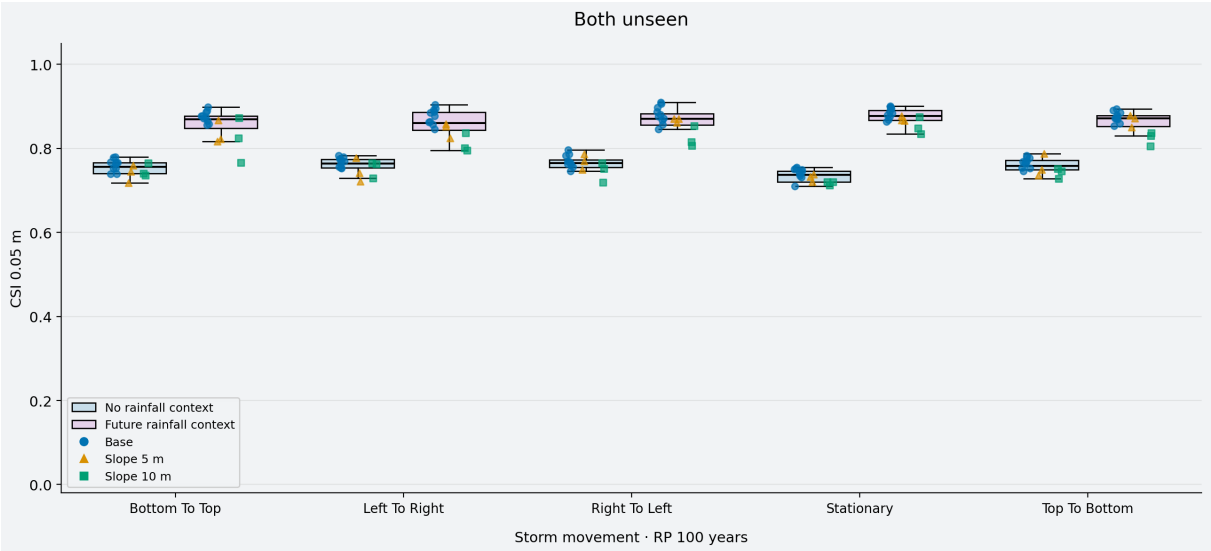


Figure D.11: Event-average CSI at 0.05 m across five storm-movement patterns under RP100 rainfall in the SV both unseen split, comparing No rainfall context and Future rainfall context.

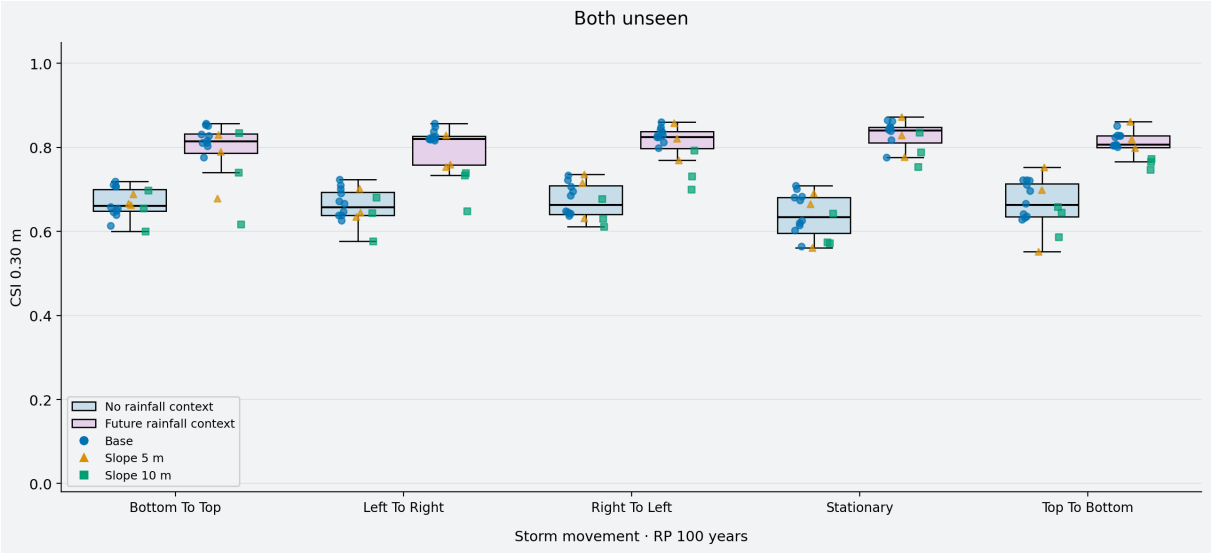


Figure D.12: Event-average CSI at 0.30 m across five storm-movement patterns under RP100 rainfall in the SV both unseen split, comparing No rainfall context and Future rainfall context.

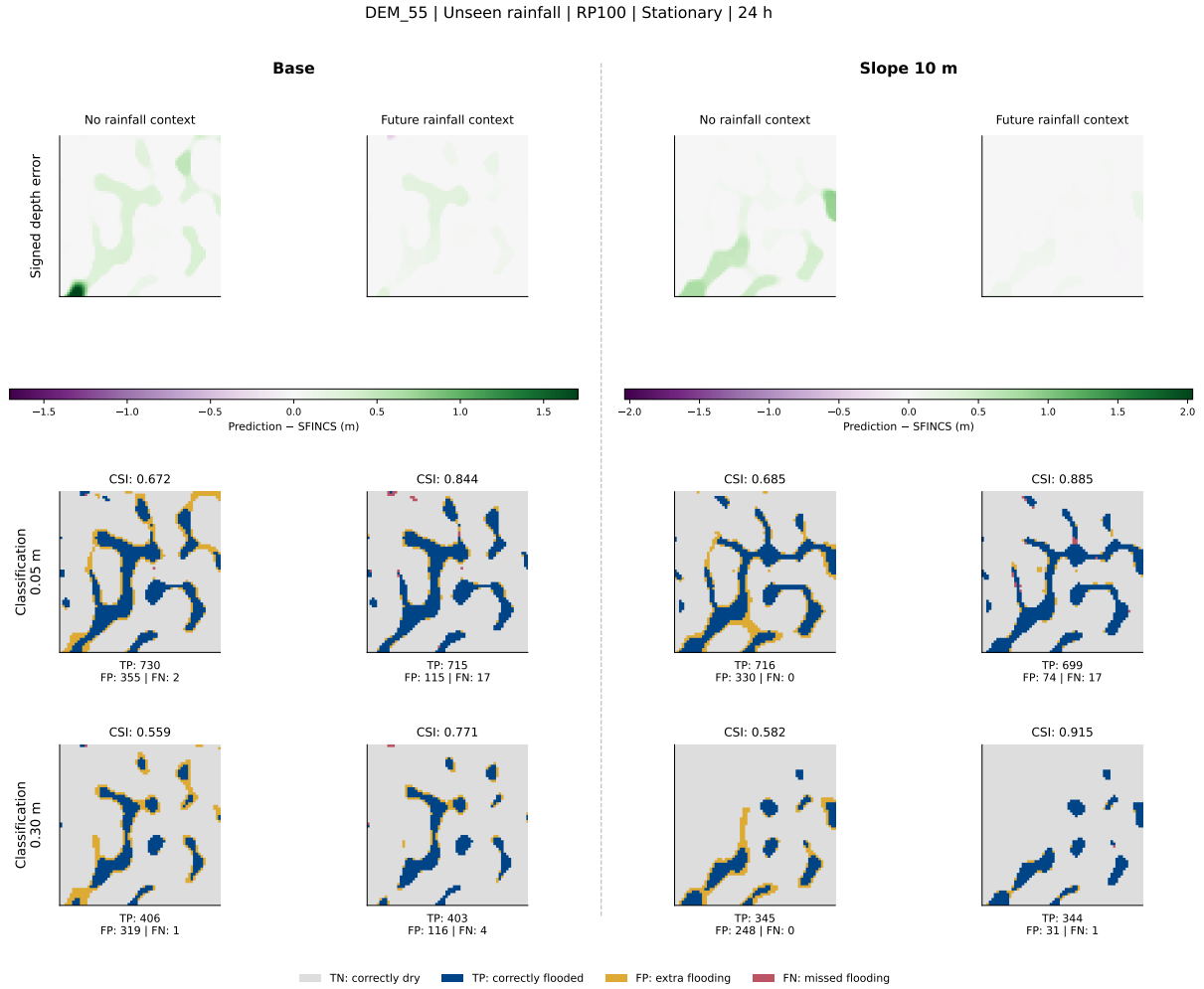


Figure D.13: Spatial depth errors and flood classification for DEM 55 and its 10 m sloped version at 24 h under stationary RP100 rainfall in the SV unseen rainfall split.



Figure D.14: Spatial depth errors and flood classification for DEM 15003 and its 10 m sloped version at 24 h under stationary RP100 rainfall in the SV both unseen split.

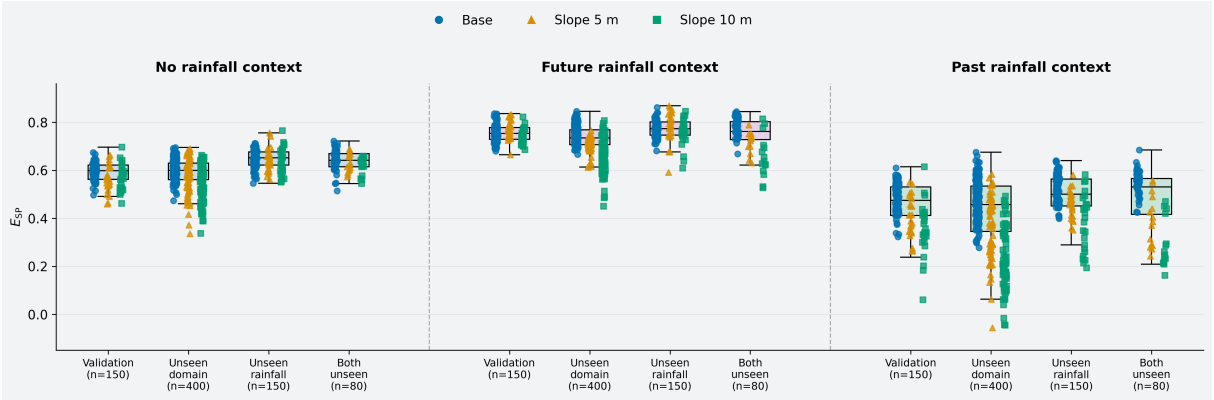


Figure D.15: Event-average spatial pattern efficiency (E_{SP}) under SV rainfall for the No, Future and Past rainfall contexts across the evaluation splits. Higher values indicate better agreement with SFINCS.