Trustworthy DRL-Based Real-Time Control for Urban Stormwater Systems: High-Fidelity Surrogates, Explainability, and Natural Flow Regime Restoration

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Isidora Fletcher

May 2026

© by Isidora Fletcher, 2026
All Rights Reserved.

To the Graduate Council:
I am submitting herewith a dissertation written by Isidora Fletcher entitled "Trustworthy DRL-Based Real-Time Control for Urban Stormwater Systems: High-Fidelity Surrogates, Explainability, and Natural Flow Regime Restoration." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Data Science and Engineering.

Dr. Anahita Khojandi, Major Professor

We have read this dissertation
and recommend its acceptance:

Dr. Anahita Khojandi

Dr. Jon Hathaway

Dr. Branko Kerkez

Dr. Augustin Engman

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my major professor, Dr. Anahita Khojandi. Her invaluable guidance, unwavering support, and intellectual encouragement have been the cornerstone of my doctoral journey. Her mentorship has not only shaped this dissertation but has also profoundly influenced my development as a researcher.

I extend my sincere thanks to the members of my dissertation committee: Dr. Jon Hathaway, Dr. Branko Kerkez, and Dr. Augustin Engman. I am grateful for the time and expertise they have dedicated to my work. Their insightful feedback, critical questions, and diverse perspectives have been instrumental in strengthening this research and broadening my understanding of the field.

I am also incredibly grateful for my family. Their constant love, encouragement, and belief in me have been my greatest source of motivation. They have made countless sacrifices and provided endless support throughout this long process, and for that, I am eternally thankful.

To my friends and colleagues, thank you for the camaraderie, the stimulating discussions, and the much-needed moments of levity. Your friendship has made this challenging journey both manageable and memorable.

Finally, this research was supported by the National Science Foundation under award number 2206540.

Abstract

This dissertation explores the application of Deep Reinforcement Learning (DRL) for sustainable stormwater and pond management. Facing increasing strain on urban water systems due to rapid urbanization and climate change, Real-Time Control (RTC) emerges as a promising solution. However, its deployment is hindered by high computational costs, a lack of interpretability, and the ecological degradation of receiving streams. This research utilizes the U.S. EPA's Storm Water Management Model (SWMM) to establish a comprehensive framework bridging the gap between theoretical DRL agents and practical civil engineering deployment.

First, to address the computational bottleneck of high-fidelity hydraulic solvers, a Bidirectional Long Short-Term Memory (BiLSTM) surrogate model was developed (Chapter 2). This surrogate achieved a Nash-Sutcliffe Efficiency greater than 0.96 and enabled a 33-fold acceleration in DRL training. This efficiency allowed an Advantage Actor-Critic (A2C) agent to converge to a competent policy within a feasible timeframe, whereas direct simulation failed due to data starvation.

Second, the "black box" problem of DRL was addressed using a novel Surrogate-Accelerated Shapley Value Estimation (SVERL) framework (Chapter 3). This method moves beyond policy explanation to performance explanation, quantifying the "Value of Information" for sensor networks. The framework serves as a forensic tool, successfully diagnosing policy collapse and identifying structural conflicts between upstream storage and downstream erosion constraints.

Finally, a novel Eco-Mimicry control framework was evaluated to address the Urban Stream Syndrome (Chapter 4). By developing a Natural Proxy model—a modified simulation representing pre-development hydrological conditions—this research established a dynamic ecological baseline. Experimental evaluation reveals that while the computational architecture is sound, the efficacy of eco-mimicry is fundamentally bounded by a Hydraulic Capacity Gap. The results demonstrate that intelligent control can optimize release timing but cannot overcome the physical storage deficits of gray infrastructure during peak flow events.

Collectively, these studies provide a scalable, transparent, and ecologically conscious pathway for deploying autonomous control in sensitive urban watersheds.

Table of Contents

Chapter 1: Introduction **1**

1.1 Motivation: The Crisis in Urban Water Management 1

1.2 The Promise and Peril of Deep Reinforcement Learning 2

1.3 Research Objectives 3

1.4 Dissertation Organization 4

Chapter 2: Accelerating Decision-Making for Smart Stormwater Management Through SWMM Surrogate Modeling for Reinforcement Learning **5**

2.1 Introduction 7

2.2 Methods 10

2.2.1 Study Area Description 10

2.2.2 Data 11

2.2.3 Problem Formulation 12

2.2.4 Models and System Setup 14

2.2.5 Experiment Design and Evaluation Metrics 22

2.3 Computational Results 23

2.3.1 Surrogate Model Performance Validation 24

2.3.2 Agent Training Acceleration 29

2.4 Discussion 32

2.4.1 Limitations and Future Work 32

2.5 Conclusion 33

Chapter 3: Performance-Based Explainability for Deep Reinforcement Learning in Stormwater Control **35**

3.1 Introduction 37

3.2 Methodology 39

3.2.1 Virtual Watershed Environment and Surrogate Interface . . . 41

3.2.2 The Autonomous Operator: DRL Framework 43

3.2.3 The Interpretability Engine: SVERL-P 45

3.2.4 Experimental Protocol 47

3.3 Results and Discussion 49

3.3.1 Framework Validation: Ground-Truth Calibration via Logic Proxy 49

3.3.2 Failure Diagnosis: Identifying Policy Collapse in Long-Duration Training 51

3.3.3 Structural Analysis: Topological Conflicts at Conner Creek . . 53

3.3.4 Practical Implications 53

3.4 Conclusion 54

3.4.1 Limitations and Future Work 56

Chapter 4: Eco-Mimicry in Real-Time: Leveraging Deep Reinforcement Learning to Replicate Natural Flow Regimes **57**

4.1 Introduction 59

4.2 Methodology 61

4.2.1 Overview of the Eco-Mimicry Control Framework 61

4.2.2 Developing the Natural Proxy Model 62

4.2.3 High-Resolution Hydrologic Data Generation 62

4.2.4 Surrogate Modeling for Computational Efficiency 64

4.2.5 Markov Decision Process (MDP) Formulation 65

4.2.6 Reinforcement Learning Framework 66

4.3 Results and Discussion 66
4.3.1 Baseline Hydrologic Characterization 67
4.3.2 Surrogate Model Performance and Limitations 67
4.3.3 RL Agent Training and Convergence 69
4.3.4 Hydraulic Bottlenecks and the Flatline Phenomenon 71
4.4 Conclusion 72

Chapter 5: Conclusions **75**
5.1 Summary of Contributions 75
5.1.1 Surrogate Modeling for Accelerated Training 75
5.1.2 Performance-Based Explainability and Trust 76
5.1.3 Eco-Mimicry and Ecological Restoration 76
5.2 Future Research Directions 77
5.2.1 Integration of Quantitative Precipitation Forecasts (QPF) . . 77
5.2.2 From Soft Rewards to Hard Constraints 77
5.2.3 Sim-to-Real Transfer and Physical Deployment 78

Bibliography **79**

Vita **88**

List of Tables

2.1 Statistical Summary of Surrogate Model Performance on Unseen Historical Data (May 2013) across 10 runs. Values are shown as Mean ± Standard Deviation. 26
2.2 Statistical Summary of Surrogate Model Performance on an Unseen 12-hr, 10-yr Storm across 10 runs. Values are shown as Mean ± Standard Deviation. 27
2.3 Comparison of Agent Training Performance over a 5-Hour Time Budget. 29
3.1 Longitudinal audit of agent behavior during extreme design storms. The convergence toward an Average Valve setting of 2.00 (Fully Open) at 16–20 hours indicates a transition from reactive control to static policy collapse. 51
4.1 Performance metrics of the BiLSTM Surrogate Model. 69

List of Figures

2.1 Aerial view of the detention pond layout. 17

2.2 SWMM schematic of the Conner Creek watershed model. 17

2.3 The standard agent-environment feedback loop in Reinforcement Learning from Sutton & Barto (2018) Sutton and Barto (2018). . . . 18

2.4 The Reinforcement Learning framework applied to the Conner Creek stormwater system. 20

2.5 Precipitation data for the month of December 2013. 25

2.6 Precipitation data for the month of May 2013. 25

2.7 Comparison of surrogate model predictions against SWMM ground truth for a representative run on unseen historical data (May 2013). . 26

2.8 Precipitation data for the 12-hour 10-year storm. 27

2.9 Comparison of surrogate model predictions against SWMM ground truth for a representative run on an unseen 12-hour, 10-year design storm. 28

2.10 Impact of computational speed on learning within a fixed 5-hour budget. Agent S (Blue) leverages the surrogate to complete 1,191 episodes, allowing for policy convergence. In contrast, Agent W (Red) is limited to 36 episodes by the SWMM simulator, effectively stalling the learning process. 30

2.11 Detailed view of learning dynamics. Top (A): Agent S explores the environment and steadily optimizes the policy, as evidenced by the rising rolling average. Bottom (B): Agent W remains trapped in the initial exploration phase with high variance and no discernible improvement. 31
3.1 Overview of the Surrogate-Accelerated SVERL-P framework. The pipeline consists of: (1) a Virtual Watershed Testbed where a BiLSTM surrogate emulates physical SWMM dynamics; (2) a Control Loop where the DRL agent executes multi-objective actions; and (3) the SVERL-P Engine, which utilizes deterministic state restoration to perform thousands of parallel counterfactual rollouts for performance-based diagnostic auditing. 40
3.2 Schematic of the Conner Creek watershed control system. The agent monitors rainfall and pond levels to modulate four valves, managing the trade-off between upstream storage utilization and downstream channel protection. 41
3.3 Design of the stress test dataset. Synthetic 10-year and 100-year design storms are stitched after historical data, separated by 48-hour dry periods. This design protocol is critical as it forces the agent to not only manage the storm peak but also actively drain the storage ponds during the cooldown recovery period to restore capacity for subsequent events. 47
3.4 SVERL-P validation results for the Logic Agent proxy. The analysis correctly attributes the positive reward contribution to Pond Depth (the trigger for opening valves) while identifying Rainfall as a negative environmental stressor. Irrelevant sensors, such as Creek Depth, are assigned zero value, confirming the engine's ability to filter stochastic noise. 50

3.5 Visualizing Policy Collapse. The 12-hour agent (A) actively throttles valves during the storm peak to protect the creek. The 20-hour agent (B) adopts a static policy, locking valves fully open to prioritize flood safety, resulting in zero Shapley attribution across all sensors. 52

4.1 Conner Creek SWMM model structure. **Left:** The full urban watershed network. **Right:** Zoomed-in view of the four detention ponds (Ponds 1–4). For the Natural Proxy, these storage units are converted to junctions, and the outlet links are replaced with natural trapezoidal channels to generate the reference flow target $Q_{natural}$. . . 63

4.2 Schematic of the Eco-Mimicry Reinforcement Learning Architecture. The Natural Proxy Model generates the target reference flow (Q_{nat}), which is fed into both the Reward calculation (for error minimization) and the Ponds Control Policy π (as a state input). This allows the agent to actively track the pre-development hydrograph. 66

4.3 Baseline hydrologic comparison for December 2015. The **Red Dashed Line** indicates the engineered urban system with fully open valves, showing sharp peaks. The green solid line indicates the Natural Proxy target ($Q_{natural}$). The significant gap between the two demonstrates the loss of natural retention and the necessity for active control. . . . 68

4.4 Early training trajectory of the Eco-Mimicry Agent. The blue line represents the raw episodic reward, and the red line represents the moving average. The volatility indicates the agent is in the exploration phase, learning the causal relationship between valve actions, storage depth, and tracking error. 70

4.5 Evaluation of the Eco-Mimicry Agent under 10-year storm conditions. The massive gap between the Natural Target (peak ≈ 50 cfs) and the urban scenarios (Random Agent and 50% Static Baseline, both peaking ≈ 2 cfs) illustrates the hydraulic capacity limit of the infrastructure. Regardless of the control strategy, the existing pipe network cannot convey the natural flow magnitude. 74

List of Algorithms

2.1 Cross-Validation and Hyperparameter Tuning for Surrogate Model . . 19
2.2 A2C Training Protocol for One Episode 22
3.1 Surrogate-Accelerated SVERL-P Calculation 46
3.2 Diagnostic State Collection Workflow 48
4.1 Stochastic Rainfall Generation via Cubic Spline Interpolation 64
4.2 Eco-Mimicry Reference Tracking (A2C) 70

Introduction

1.1 Motivation: The Crisis in Urban Water Management

Urban drainage systems (UDS) are facing a converging crisis. Rapid urbanization has increased impervious surface areas, stripping watersheds of their natural ability to absorb rainfall. Simultaneously, climate change is altering precipitation patterns, leading to storms of increasing intensity and frequency (Sadler et al., 2020; Mullapudi et al., 2020). Conventional "gray" infrastructure—static networks of pipes and passive detention ponds—was designed under the assumption of stationarity, a statistical concept that implies past weather patterns reliably predict the future. This assumption no longer holds. As a result, cities globally are grappling with frequent combined sewer overflows (CSOs), catastrophic urban flooding, and the ecological degradation of receiving water bodies due to hydromodification (Akin, 2021; Bowes et al., 2022).

Addressing these challenges via traditional capacity expansion (e.g., digging larger tunnels or building new ponds) is often prohibitively expensive and disruptive to the urban fabric. Consequently, water utilities are turning toward Real-Time Control (RTC) as a cost-effective alternative. RTC transforms passive infrastructure into active systems by using sensors and actuated valves to dynamically regulate flow, maximizing the utility of existing storage capacity (Kerkez et al., 2016).

1.2 The Promise and Peril of Deep Reinforcement Learning

Among RTC methodologies, Deep Reinforcement Learning (DRL) has emerged as a particularly promising approach. Unlike Rule-Based Control (RBC), which relies on static "if-then" logic, or Model Predictive Control (MPC), which requires computationally expensive optimization during operation, DRL agents learn flexible, adaptive control policies through trial-and-error interaction with a simulation environment (Wang et al., 2020; Saliba et al., 2020).

However, despite its theoretical potential, the deployment of DRL in real-world water infrastructure remains limited. Three fundamental barriers hinder the transition from research to practice:

1. **The Computational Bottleneck:** DRL requires millions of interaction steps to learn a viable policy. High-fidelity hydraulic solvers like the US EPA's Storm Water Management Model (SWMM) are computationally intensive. Training an agent directly on SWMM can take weeks or months, making iterative research and hyperparameter tuning impractical (Tian et al., 2022a).

2. **The "Black Box" Problem:** DRL agents operate via deep neural networks, which are notoriously opaque. Civil engineers and utility operators operate in a safety-critical domain where liability is paramount. They cannot adopt a control system that makes decisions without an interpretable physical justification. The inability to explain *why* an agent opened a valve prevents trust and regulatory approval (Tian et al., 2024a).

3. **Robustness Under Uncertainty:** Most DRL research trains and tests agents on historical rainfall data. However, given the non-stationarity of the climate, an agent optimized for the weather of 2010 may fail catastrophically under the extreme weather patterns projected for 2050. Ensuring that learned policies are robust to out-of-distribution events is a prerequisite for deployment.

1.3 Research Objectives

The primary objective of this dissertation is to develop a comprehensive framework for enabling scalable, trustworthy, and resilient autonomous control in urban stormwater systems. This work addresses the three barriers identified above through the following specific research aims:

Objective 1: Accelerating Training via Surrogate Modeling

To resolve the computational bottleneck, this research develops a high-fidelity surrogate model using Bidirectional Long Short-Term Memory (BiLSTM) networks. This data-driven emulator mimics the hydraulic dynamics of SWMM but executes orders of magnitude faster, enabling rapid DRL training without sacrificing physical accuracy.

Objective 2: Establishing Trust via Explainable AI

To address the "Black Box" problem, this research adapts game-theoretic interpretability methods—specifically Shapley Values—to the domain of hydrological control. We propose a framework that moves beyond explaining the agent's *policy* to explaining its *performance*, quantifying the marginal contribution of specific sensors to flood mitigation and erosion control.

Objective 3: Eco-Mimicry for Hydrologic Restoration

To address the ecological impacts of urbanization, this research evaluates the feasibility and physical limits of using DRL to restore natural flow regimes in altered watersheds. This objective involves developing a Natural Proxy model to generate pre-development flow targets and investigating the extent to which software optimization can compensate for the hardware deficits of aging infrastructure to mitigate the Urban Stream Syndrome.

1.4 Dissertation Organization

This dissertation follows a manuscript style format, where the main chapters correspond to distinct research contributions:

Chapter 2 presents the development of the BiLSTM surrogate model and demonstrates its efficacy in accelerating the training of an Advantage Actor-Critic (A2C) agent for the Conner Creek watershed in Knoxville, TN.

Chapter 3 introduces the "Surrogate-Accelerated SVERL" framework. This chapter focuses on interpretability, using Shapley values to audit the agent's decision-making logic and quantify the value of information from various sensors.

Chapter 4 proposes the Eco-Mimicry framework. This chapter presents the experimental evaluation of a reference tracking control policy designed to replicate natural hydrology. It identifies the Hydraulic Capacity Gap and the limitations of surrogate modeling for variable-aperture control, providing a quantitative assessment of the boundaries between algorithmic intelligence and physical infrastructure constraints.

Chapter 5 summarizes the contributions of this work and outlines future directions for research in autonomous water systems.

Chapter 2

Accelerating Decision-Making for Smart Stormwater Management Through SWMM Surrogate Modeling for Reinforcement Learning

Disclosure

Author Contributions and Roles: Isidora Fletcher was the primary investigator and lead author for this study. She was responsible for the conceptualization and development of the BiLSTM surrogate model architecture, the end-to-end implementation of the Reinforcement Learning framework, and the primary drafting of the manuscript.

Anahita Khojandi supervised the project throughout its duration. She was instrumental in the design of the methodological framework, providing critical analytical guidance and strategic editorial refinement of the research findings.

Ghada Diab contributed the foundational SWMM model used in this study and drafted the technical description of the SWMM model included in the Methods section.

Jon Hathaway and Theodoros Kyriakopoulos provided essential technical expertise and extensive troubleshooting for the high-fidelity SWMM model, ensuring its accurate refinement and robustness for the simulation environment. Jon Hathaway also provided domain expertise in urban drainage systems.

Abstract

The large computational cost of high-fidelity simulators like the Storm Water Management Model (SWMM) limits the application of Deep Reinforcement Learning (DRL) for Real-Time Control (RTC) in urban stormwater management. This study presents a surrogate-accelerated framework applied to a four-pond network within the Conner Creek watershed (Knoxville, TN). We developed a Bidirectional Long Short-Term Memory (BiLSTM) surrogate to replicate system hydraulics, achieving a mean Nash-Sutcliffe Efficiency (NSE) greater than 0.99 on historical data, and

maintaining high fidelity during design storms with an NSE over 0.99 for ponds and 0.96 for the stream. This high-fidelity surrogate resolved a major computational bottleneck, providing a 33-fold acceleration in training DRL agents compared with direct SWMM simulation. While the agent trained directly on SWMM failed to progress, completing only 36 episodes, the surrogate-trained agent completed 1,191 episodes and successfully converged to a policy that balances complex hydraulic objectives. These results demonstrate that surrogate modeling offers a robust and feasible pathway for developing adaptive RTC strategies.

2.1 Introduction

Urban drainage systems (UDS) globally are grappling with escalating pressures from rapid urbanization and increasing hydrological extremes driven by climate change Mullapudi et al. (2020); Saliba et al. (2020); Wang et al. (2021). These pressures strain existing infrastructure, resulting in urban flooding and compromising the quality of receiving waters Akin (2021); Mullapudi and Kerkez (2018); Bowes et al. (2022). To enhance resilience and dynamically mitigate these risks, the integration of Real-Time Control (RTC) and smart stormwater systems has emerged as a promising strategy, allowing existing, conventionally passive infrastructure to dynamically adapt its operation Mullapudi et al. (2020); Akin (2021); Wong and Kerkez (2018). Effective RTC demands a system-level control approach that coordinates management actions across an entire urban watershed to ensure local actions contribute positively to the overall system performance Bowes et al. (2022); Kerkez et al. (2016).

Advanced stormwater control strategies rely on accurate and swift simulation capabilities. The Storm Water Management Model (SWMM), an established, high-fidelity physics-based model, is widely used to simulate complex hydrologic and hydraulic dynamics in urban catchments Mullapudi et al. (2020); Wang et al. (2021). However, because SWMM solves detailed, non-linear equations, its iterative application is computationally demanding Mullapudi et al. (2020); Tian et al. (2022a);

Baumann et al. (2022a). This high computational cost represents a major barrier to its utility in optimization-based frameworks and sophisticated RTC applications that require thousands of high-frequency model evaluations in a practical time frame Tian et al. (2022a); He et al. (2025).

To overcome this persistent trade-off between model fidelity and computational speed, researchers increasingly utilize surrogate modeling (also referred to as meta-modeling or emulation) Razavi et al. (2012); Keating et al. (2010); Behzadian et al. (2009). Surrogate models employ data-driven methods to rapidly approximate the input-output relationship of complex, computationally expensive models, accelerating processes such as sensitivity analysis, calibration, and optimization Razavi et al. (2012); Majeske et al. (2022). In the context of hydrological systems, Deep Learning (DL) architectures, particularly Recurrent Neural Networks (RNNs) like Long Short-Term Memory (LSTM) networks, have proven exceptionally suitable for capturing the complex, time-dependent, and non-linear dynamics inherent in time-series data Huff et al. (2023); Le et al. (2019); Girihagama et al. (2022). Furthermore, the utilization of Bidirectional LSTM (BiLSTM) networks, which process temporal sequences in both forward and backward directions, often leads to superior predictive accuracy and robust generalization performance, a significant advantage in approximating highly coupled hydrological systems Ghasemlounia et al. (2021); Ge et al. (2023); Siami-Namini et al. (2019). Prior work demonstrates the viability of using LSTM-based models as surrogates for high-fidelity models like SWMM for flood forecasting and runoff prediction Wood-Ponce et al. (2024); Zhao et al. (2023).

In parallel, Deep Reinforcement Learning (DRL) offers a potent, adaptive, and autonomous framework for RTC in UDS Mullapudi et al. (2020); Saliba et al. (2020); Tian et al. (2024a). An RL agent learns an optimal control policy by maximizing a user-defined cumulative reward through extensive, iterative interaction with its environment Mullapudi and Kerkez (2018); Sutton and Barto (2018). Previous research has successfully applied various DRL algorithms, including Deep Q-Networks (DQN), Deep Deterministic Policy Gradient (DDPG), and Proximal

Policy Optimization (PPO), to address core stormwater challenges such as flood and Combined Sewer Overflow (CSO) mitigation Saliba et al. (2020); Tian et al. (2024a); Bowes et al. (2020). However, the DRL paradigm requires generating vast amounts of training data through numerous simulation episodes Mullapudi et al. (2020); He et al. (2025); Tian et al. (2022b), confirming that the computational burden imposed by the underlying physics-based simulator (i.e., SWMM) remains a critical obstacle to large-scale RL adoption Wang et al. (2021); Tian et al. (2022a); He et al. (2025). While some studies explored acceleration by replacing the UDS model with simpler emulators (e.g., those based on the Koopman operator), limitations sometimes arose concerning modeling accuracy due to linear simplification Tian et al. (2022a).

This study presents a novel and accelerated RL training framework by integrating an efficient BiLSTM-based surrogate model to function as a fast, differentiable proxy for the complex SWMM environment. This methodology significantly reduces the computational overhead necessary for policy optimization, enabling the Advantage Actor-Critic (A2C) agent to learn a competent control policy within a practical time budget Wang et al. (2022); Fuente and Guerra (2024). Our control strategy addresses a multi-objective problem by designing a complex reward function that guides the agent toward: (1) keeping the detention ponds' depths below the elevations of their control structure to mitigate excessive uncontrolled overflow, and (2) maintaining the stream's shear stress, below a critical threshold downstream of where the ponds discharge to prevent erosion and channel degradation Akin (2021); Mullapudi and Kerkez (2018).

We acknowledge that developing and tuning a surrogate model requires an initial investment of computational time. However, once trained, the surrogate provides rapid inference capabilities that can be utilized repeatedly for extensive RL training sessions, ultimately yielding a net positive return on computational resources.

The primary contribution of this research is twofold: (1) the rigorous development and validation of a BiLSTM surrogate model capable of accurately emulating a high-fidelity SWMM simulation of a multi-pond stormwater network, and (2) a

demonstration of the computational feasibility of training an A2C control policy on this surrogate. This research quantifies the significant acceleration achieved by training an agent on the fast surrogate, a task that would be computationally prohibitive on the original SWMM model. This innovation directly addresses the computational bottleneck in DRL training for RTC, offering a validated methodology and a practical foundation for future work in deploying sophisticated, adaptive control policies for urban water systems.

2.2 Methods

This section details the comprehensive methodology used to develop, validate, and test a surrogate-accelerated DRL framework for stormwater control. The methodology begins in **Section 2.2.1** with a description of the physical study area. This is followed by the data sources and generation process in **Section 2.2.2**, the formal problem formulation in **Section 2.2.3**, and the technical details of the models in **Section 2.2.4**. The section concludes with the experimental design and evaluation metrics in **Section 2.2.5**.

2.2.1 Study Area Description

The study is grounded in the Conner Creek watershed, a 362-hectare tributary of the Clinch River located in Knoxville, TN. The drainage area includes four actively managed detention ponds that are used to control stormwater runoff from three nearby schools. To provide a geographical context for the simulation environment, **Figure 2.1** provides an aerial view of the site, grounding the model in the actual layout of the detention ponds.

2.2.2 Data

The foundation of the surrogate model is a rich dataset generated from the primary SWMM environment. The simulations were driven by both historical and synthetic precipitation data. The historical record was sourced from the McGee Tyson Airport in Knoxville, TN, while the synthetic data represented specific design storms.

Data Sources and Generation The foundation for the surrogate model is a rich and diverse dataset generated by running the high-fidelity SWMM environment under a wide range of conditions. Data generation was managed programmatically using the `pyswmm` library McDonnell et al. (2020). To ensure the data captured a wide variety of system responses, a random discrete action policy was applied to the four controllable valves during each simulation. The dataset comprises two distinct types of simulation scenarios to ensure the model is robust under both operational and extreme conditions:

1. **Historical Data:** The historical precipitation record was sourced from the NOAA dataset for McGee Tyson Airport in Knoxville, TN. The year 2013 was selected as it represents the rainiest year in the available record, providing a rigorous training set that captures system behavior under high antecedent soil moisture conditions. To generate a large and varied training set, nine independent simulations were run using the rainfall data from three representative months (wet, average, and dry) of that year. Crucially, a unique sequence of random actions was applied to the four controllable valves during each of these nine simulations. This process ensures that even with repeated rainfall patterns, the resulting system states (pond depths) are different in each run. Each of the nine simulations was then partitioned by month, resulting in a final dataset of 27 distinct training segments (9 runs $\times$ 3 months). This methodology creates a rich dataset where the surrogate is exposed to a

wide range of state-action-response scenarios, essential for learning the system dynamics.

2. **Synthetic Storm Data:** To ensure the model is proficient at capturing high-impact, nonlinear events, multiple simulations were run using two synthetic 24-hour design storms: a 10-year and a 25-year return period event. These storms were generated using the Huff distribution method, scaled by NOAA precipitation frequency estimates for the Knoxville area. To ensure the surrogate model learned system recovery dynamics, four days of zero rainfall were appended to the end of each synthetic storm event.

Data Preprocessing and Balancing Before training, the data underwent several preprocessing steps. The complete dataset, combining both historical and storm-event data, was balanced by oversampling the storm data. This critical step prevents the surrogate model from becoming biased towards the more frequent, low-flow historical conditions and improves its performance during extreme events. Furthermore, all state variables were normalized using a Min-Max scaler. The final artifacts from this process—the trained model and the corresponding data scaler were saved for later use in the RL environment.

2.2.3 Problem Formulation

To enable autonomous control, the stormwater management problem is formalized as a Markov Decision Process (MDP), defined by the tuple $\langle S, A, T, r, \gamma \rangle$.

Time Step Control decisions are made at a fixed time step of $t = 1$ hour. This interval was primarily selected to align with the temporal resolution of the available historical precipitation records. However, from a control perspective, this frequency is sufficient to capture the filling and draining dynamics of the detention ponds.

State Space (S) The state vector at time t, denoted as $S_t \in \mathbb{R}^{10}$, represents the agent's observation of the environment. Prioritizing robust, cost-effective implementation, we selected a state space based primarily on depth measurements of the four ponds and the stream, rather than flow rates. This design allows for real-world deployment using standard level sensors without the need for more resource intensive flow monitoring. The state vector consists of:

- *5 Depths:* Normalized water levels in Pond 1, Pond 2, Pond 3, Pond 4, and the creek conduit at a downstream point of interest.
- *4 Previous Actions:* The valve settings from the previous timestep $(t-1)$ for each of the four ponds.
- *1 Exogenous Variable:* The current rainfall intensity.

Action Space (A) The action at time t, A_t, is defined by a discrete space `MultiDiscrete([3, 3, 3, 3])`. Each of the four ponds can be set to one of three valve positions: 0 (Fully Closed), 1 (Half Open), or 2 (Fully Open). This discretization ensures the system dynamics remain learnable for the surrogate while reflecting the operational constraints of standard actuators.

Reward Function (r) The reward signal r_t guides the agent toward the competing objectives of flood mitigation and erosion control. It is calculated as the negative sum of three weighted penalty terms, based on extensive tuning to promote a balanced control policy:

$$r_t = -(W_{\text{depth}} \cdot P_{\text{depth},t} + W_{\text{erosion}} \cdot P_{\text{erosion},t} + P_{\text{flood},t}) \tag{2.1}$$

The large difference in the magnitude of these weights is intentional; it is necessary to normalize the penalties, as the raw values for squared depth, squared excess shear stress, and the discrete flood penalty operate on vastly different numerical scales. The individual penalties and their tuned weights are:

- *Depth Penalty ($P_{depth,t}$):* The sum of squared depths for all four ponds ($\sum_{i=1}^{4} \text{depth}_i^2$). This term is weighted by a small factor, $W_{\text{depth}} = 0.02$, which provides a gentle incentive for the agent to drain the ponds when it is safe to do so.

- *Erosion Penalty ($P_{erosion,t}$):* This penalty is designed to mitigate stream bank erosion and is applied when the shear stress (τ_t) in the creek exceeds a critical threshold ($\tau_{\text{crit}} = 0.024$). This threshold corresponds to values established for normal soil moisture in the Conner Creek watershed Akin (2021); Vermont Department of Environmental Conservation (nd). Shear stress is calculated dynamically as:

$$\tau_t = \gamma \cdot R_{h,t} \cdot S_o \tag{2.2}$$

  where γ is the specific weight of water ($62.4\ \text{lb/ft}^3$), S_o is the fixed channel slope (0.00281), and $R_{h,t}$ is the hydraulic radius. The penalty is calculated as $(\max(0, \tau_t - \tau_{\text{crit}}))^2$. Because the raw values of this penalty are numerically very small, this term is scaled by a very large weight, $W_{\text{erosion}} = 250,000$, to ensure that even minor violations of the shear stress threshold are significant enough to force the agent to learn a modulated release strategy.

- *Flooding Penalty ($P_{flood,t}$):* A discrete penalty of 100 is applied for each pond where the depth exceeds the flood limit (defined as 0.5 ft below the weir/bypass elevation). This serves as a hard constraint to prioritize the fundamental safety objective of flood prevention.

2.2.4 Models and System Setup

This study integrates three distinct models: the physical environment simulator, the data-driven surrogate emulator, and the DRL control agent.

Environment Model: The Conner Creek SWMM Model

The high-fidelity simulation environment is a detailed SWMM model of the Conner Creek watershed, the study area described in Section 2.1. SWMM simulates hydrology and hydraulics using a dynamic wave approach for flow routing, modeling the watershed as a system of subcatchments, junctions, and links Rossman (2015). A schematic of the model is shown in **Figure 2.2**.

The model was developed by discretizing the watershed into 128 subcatchments and integrating several high-resolution national datasets for parameterization: geometric properties were delineated from 1-meter USGS topographic maps; Green-Ampt infiltration parameters Green and Ampt (1911) were derived from the NRCS SSURGO database; and land cover was computed from the NLCD. Each of the four detention ponds was modeled as a storage node with an assigned stage-area curve, and their outlet structures were modeled as links.

To calibrate the model, on-site data was collected. Junction and conduit spatial data were provided by Knox County Government, while open channel cross-sections were delineated using transects from a digital elevation model enhanced by local surveying. Precipitation data were recorded on-site using a tipping bucket rain gage, and flow was monitored within the creek using an established stage-discharge relationship paired with an ultrasonic stage recorder Akin (2021). The model was calibrated utilizing the Sensitivity-based Radio Tuning Calibration (SRTC) tool, comparing daily model output against this on-site flow data, which was recorded from January to July 2021. The calibration achieved a Nash-Sutcliffe Efficiency (NSE) of 0.7, which is considered a good fit Skaggs et al. (2012); Moriasi et al. (2007).

All programmatic interaction with the SWMM simulator is managed via the `pyswmm` Python library McDonnell et al. (2020). To overcome low-level library conflicts between TensorFlow and the C-bindings used by `pyswmm`, a `subprocess`-based environment wrapper was implemented. This isolates the SWMM simulation

in a separate process, following established patterns for robust scientific scripting Langtangen (2004), ensuring stable, long-duration training runs.

Surrogate Model: BiLSTM Emulator

The core of the acceleration strategy is a surrogate model developed to serve as a fast and accurate proxy for the SWMM environment.

Model Architecture A Bidirectional Long Short-Term Memory (BiLSTM) network was selected for this task. LSTMs are a specialized type of Recurrent Neural Network (RNN) designed to overcome the long-term dependency problem in sequential data Hochreiter and Schmidhuber (1997). A BiLSTM enhances the standard LSTM by processing the input sequence in both the forward and backward directions, allowing the model to leverage information from both past and future contexts for any given time step, which significantly improves predictive accuracy in complex hydrological systems Siami-Namini et al. (2019).

The architecture, implemented using the Keras Sequential API, consists of two stacked BiLSTM layers followed by a hidden Dense layer and a final linear output layer. It was designed to process input sequences of six timesteps (representing six hours), mapping the 10-variable RL state (5 depths, 4 previous actions, and rainfall intensity) to a vector of 5 predicted depths for the subsequent hour. The first BiLSTM layer was configured to pass its full output to the second layer, and the hidden Dense layer utilized a ReLU activation function. The model was compiled using the Adam optimizer and trained to minimize Mean Squared Error (MSE) loss, with key hyperparameters such as the number of units in the BiLSTM layers and the learning rate being optimized via Keras Tuner.

Hyperparameter Tuning and Validation A rigorous 5-Fold Cross-Validation strategy was used to obtain an unbiased estimate of the model's generalization performance. This specific fold count was selected to balance the high computational

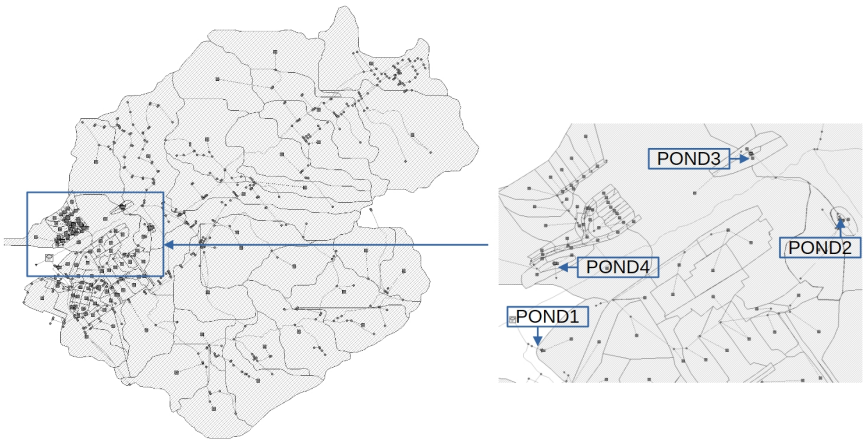


Figure 2.1: Aerial view of the detention pond layout.

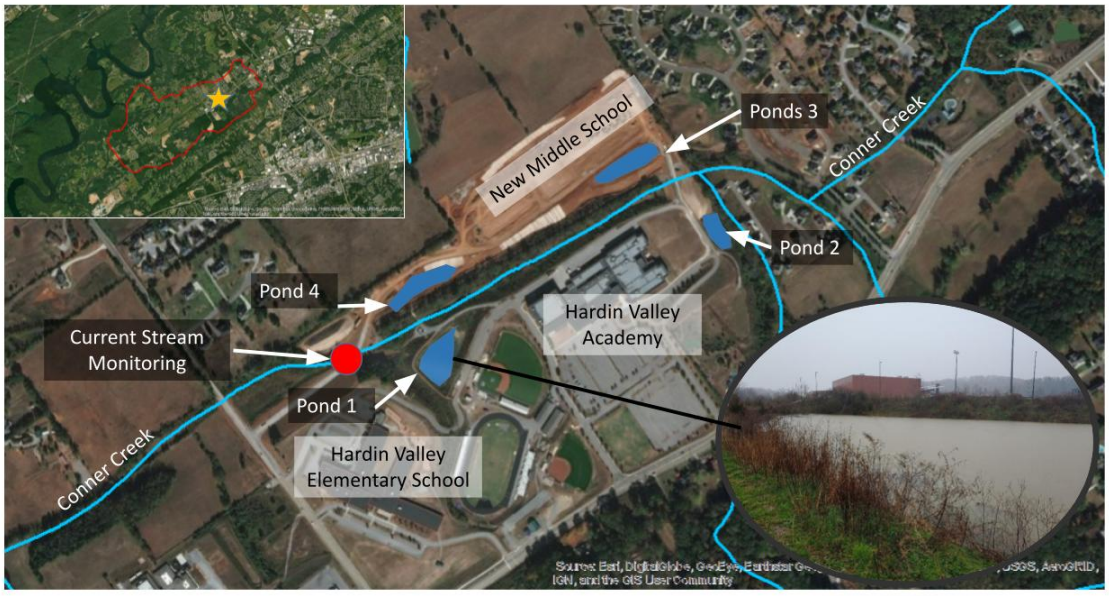


Figure 2.2: SWMM schematic of the Conner Creek watershed model.

cost of iterative deep learning training with the need for robust statistical verification, following established methodologies in hydrological deep learning Huang et al. (2024). Within each fold, Keras Tuner (RandomSearch) was employed to automatically discover the optimal hyperparameters for the architecture. The tuning process focused on identifying the optimal number of units for each of the two BiLSTM layers, as well as the learning rate for the Adam optimizer. The single best-performing set of hyperparameters from these validation folds was then used to train the final, definitive surrogate model on the entire balanced dataset. The detailed procedure for this combined cross-validation and hyperparameter tuning workflow is presented in **Algorithm 2.1**.

Deep Reinforcement Learning Agent Formulation

Reinforcement Learning (RL) is a paradigm in machine learning where an agent learns to make optimal decisions by interacting with an environment. The agent takes actions and receives feedback in the form of numerical rewards or punishments, with the ultimate objective of learning a policy—a mapping from states to actions—that maximizes a cumulative reward over time Sutton and Barto (2018). A generalized depiction of this agent-environment interaction loop is shown in **Figure 2.3**.

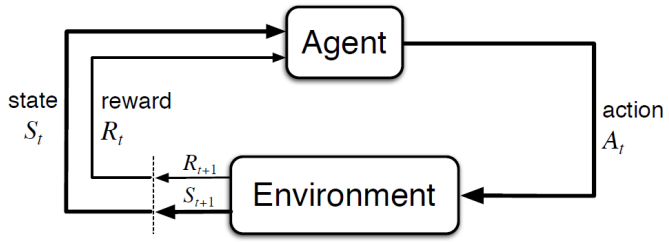


Figure 2.3: The standard agent-environment feedback loop in Reinforcement Learning from Sutton & Barto (2018) Sutton and Barto (2018).

Building on the formal MDP definition provided in Section 2.3, our specific implementation for the Conner Creek watershed is illustrated in **Figure 2.4**. This

Algorithm 2.1 Cross-Validation and Hyperparameter Tuning for Surrogate Model

procedure CrossValidateModel(D_{hist}, D_{storm}, K)
    Initialize: Empty list for fold performance: $L_{test_losses} \leftarrow []$
    ▷ Combine and balance the full dataset
    $f_{os} \leftarrow \text{count}(D_{hist})/\text{count}(D_{storm})$
    $D_{all} \leftarrow D_{hist} \cup \text{Oversample}(D_{storm}, f_{os})$
    Shuffle D_{all}
    ▷ Begin K-Fold Cross-Validation
    for $k \leftarrow 1$ to K **do**
        $D_{train}, D_{test} \leftarrow \text{GetFold}(D_{all}, k)$
        ▷ Fit scaler ONLY on the training data for the fold
        $S_k \leftarrow \text{InitializeMinMaxScaler}()$
        Fit S_k on D_{train}
        ▷ Apply scaler and create input sequences
        $(X_{train}, y_{train}) \leftarrow \text{CreateSequences}(\text{Transform}(D_{train}, S_k))$
        $(X_{test}, y_{test}) \leftarrow \text{CreateSequences}(\text{Transform}(D_{test}, S_k))$
        ▷ Hyperparameter search using Keras Tuner
        $T_k \leftarrow \text{InitializeRandomSearchTuner}(\text{BiLSTM_Hypermodel})$
        $T_k.\text{search}(X_{train}, y_{train})$
        $hp_{best} \leftarrow T_k.\text{get_best_hyperparameters}()$
        ▷ Train and evaluate the final model for the fold
        $M_k \leftarrow \text{BuildModel}(hp_{best})$
        Train M_k on (X_{train}, y_{train}) with early stopping
        $loss_k \leftarrow M_k.\text{evaluate}(X_{test}, y_{test})$
        Append $loss_k$ to L_{test_losses}
    end for
    ▷ Calculate final summary statistics
    $\mu_{loss} \leftarrow \text{mean}(L_{test_losses})$
    $\sigma_{loss} \leftarrow \text{std}(L_{test_losses})$
    return $\mu_{loss}, \sigma_{loss}$
end procedure

diagram shows how the A2C agent receives the state vector S_t, selects an action A_t, and receives a reward r_t from the SWMM or surrogate environment.

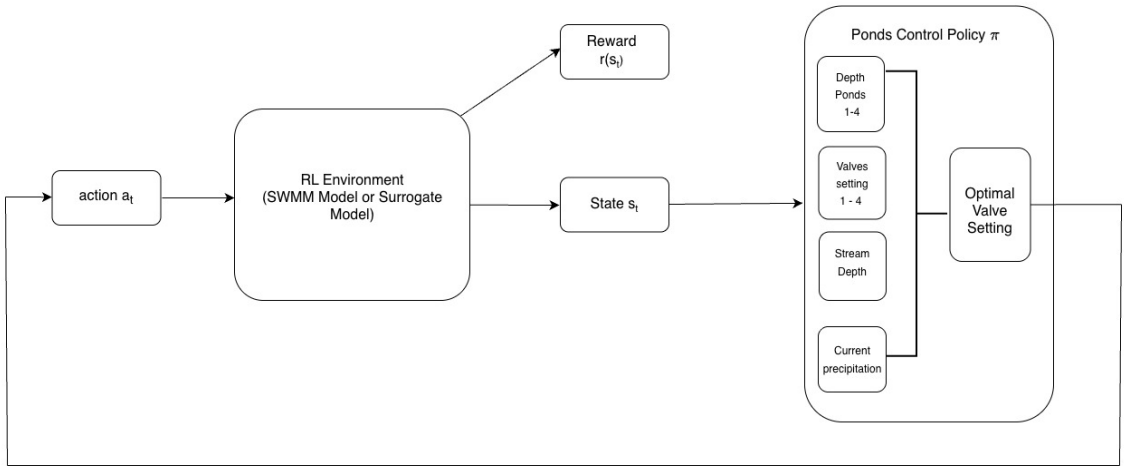


Figure 2.4: The Reinforcement Learning framework applied to the Conner Creek stormwater system.

Learning Algorithm (A2C) An A2C agent was implemented using TensorFlow 2. A2C is an on-policy, policy gradient method that combines elements of both actor-critic and advantage estimation frameworks Fuente and Guerra (2024); Wang et al. (2022). Algorithms in this family are designed to simultaneously learn both a policy function (the "actor") and a value function (the "critic"). This approach combines the strengths of purely policy-based and value-based methods, differentiating it from either approach alone Tian et al. (2022b).

The actor is responsible for learning the control policy, which is a parameterized stochastic policy, $\pi_\theta(a|s)$, that maps states to a probability distribution over actions. The critic learns a state-value function, $V^\pi(s)$, which estimates the expected cumulative reward from a given state. The critic is updated using the Temporal Difference (TD) error, δ, which measures the discrepancy between the observed reward and the critic's estimate, given by:

$$\delta = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t) \tag{2.3}$$

where r_t is the reward received at time t, and γ is the discount factor. The critic's primary role is to provide this low-variance baseline to guide the actor's updates. This is accomplished by using the advantage function, $A^\pi(s, a)$, which measures how much better a specific action a is compared to the average value of being in state s. The advantage function is formally defined as the difference between the action-value function, $Q^\pi(s, a)$, and the state-value function:

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s) \tag{2.4}$$

By using this advantage signal to scale the policy gradient, the agent learns more efficiently and stably.

Our implementation of this A2C agent uses a single neural network with a shared body and separate output heads. The shared body consists of two fully-connected hidden layers (Dense) with ReLU activation which processes the 10-dimensional state vector. The output heads are:

- **The Actor:** Consists of four separate Dense layers, one for each of the four ponds. Each head outputs the raw logits for the three possible actions (Closed, Half, Open) for its corresponding pond. A final action is determined by sampling from the resulting probability distribution for each head.

- **The Critic:** A single Dense layer with a linear activation function that outputs a scalar value, representing the agent's estimate of the state-value function $V^\pi(s)$.

The agent is trained to minimize a composite loss function. The critic's loss is calculated using the Huber loss between its predicted state-values and the standardized, discounted returns calculated from the episode (which serve as an empirical estimate of $Q^\pi(s, a)$). The actor's loss is calculated as the negative sum of the log-probabilities of the taken actions, weighted by the advantage (the difference between the discounted returns and the critic's value predictions). The total loss is

the sum of the actor and critic losses, and the network weights are updated using the Adam optimizer. The detailed, step-by-step training process is presented in **Algorithm 2.2**.

Algorithm 2.2 A2C Training Protocol for One Episode

procedure TRAINEPISODE($A_\theta, Env, Opt, \gamma, T_{max}$)
  Initialize: Empty trajectory buffers L_{log_probs}, V, and R.
  $s \leftarrow Env.reset()$
  ▷ Collect a trajectory by interacting with the environment
  for $t = 0$ to $T_{max} - 1$ **do**
    $[\text{logits}_1, ..., \text{logits}_4], v \leftarrow A_\theta(s)$
    for each pond $i = 1$ to 4 **do**
      Sample action a_i from softmax distribution of logits_i
      Calculate log-probability $\log \pi(a_i|s)$
    end for
    Append $\sum_{i=1}^{4} \log \pi(a_i|s)$ to L_{log_probs}
    Append v to V
    $s', r, d \leftarrow Env.step([a_1, ..., a_4])$
    Append r to R
    $s \leftarrow s'$
    if d is True **then break**
    end if
  end for
  ▷ Compute loss and update network weights
  $G \leftarrow$ Calculate standardized discounted returns from R
  $Adv \leftarrow G - V$
  $Loss_{critic} \leftarrow \mathscr{L}_{Huber}(V, G)$
  $Loss_{actor} \leftarrow -\sum(L_{log_probs} \cdot \text{stop_gradient}(Adv))$
  $Loss_{total} \leftarrow Loss_{actor} + Loss_{critic}$
  $\nabla_\theta L \leftarrow$ Compute gradients of $Loss_{total}$ w.r.t. θ
  $\theta \leftarrow Opt(\theta, \nabla_\theta L)$
  return Total episode reward $\sum(R)$
end procedure

2.2.5 Experiment Design and Evaluation Metrics

To quantify the utility of the surrogate model, a direct comparative experiment was designed to measure the acceleration in training efficiency.

The Comparative Experiment. Two agents were trained under identical conditions but in different environments:

- *Agent S (Surrogate-Trained):* Trained using the `ConnerCreekEnv` in `surrogate` mode, interacting with the final, tuned BiLSTM model.
- *Agent W (SWMM-Trained):* Trained using the `ConnerCreekEnv` in `swmm` mode, interacting directly with the high-fidelity physics-based simulator.

Controlled Conditions for Fair Comparison. To ensure a fair comparison, the training protocol was strictly controlled:

- Both experiments utilized an identical training period, driven by rainfall data from December 2013. The precipitation data for December 2013 is shown in **Figure 2.5**.
- Both agent training sessions were run for the exact same wall-clock time budget of 5 hours.
- Both environments used a deterministic `reset()` method, ensuring identical starting conditions for every training episode.

The final results, including the learning curves (reward vs. episode) and total episodes completed by each agent, were saved for analysis.

Evaluation Metrics. The predictive accuracy of the surrogate model was quantified using Nash-Sutcliffe Efficiency (NSE) and Root Mean Squared Error (RMSE). The primary metric for the agent training experiment was the total number of training episodes completed within the fixed time budget, which serves as a direct measure of computational acceleration.

2.3 Computational Results

This section presents the performance evaluation of the final, tuned Bidirectional LSTM surrogate model and discusses the implications of these findings. First,

in Section 3.1, the model's predictive accuracy is quantified on unseen historical and storm data. Second, Section 3.2 demonstrates the significant computational advantage the surrogate provides for Reinforcement Learning (RL) agent training. Finally, Section 4 interprets the overall results and their implications for applying DRL to real-time stormwater control.

2.3.1 Surrogate Model Performance Validation

To assess the generalization capability of the trained surrogate model, a multi-run evaluation was conducted against 10 different simulation runs for each test case. This approach provides a robust statistical summary of the model's performance.

Evaluation on Unseen Historical Data

The model was first evaluated on data from May 2013, a month not included in the training set, to test its performance under non-storm, operational conditions. The precipitation data for May 2013 is shown in **Figure 2.6**. The results, presented visually in **Figure 2.7** and quantitatively in **Table 2.1**, demonstrate a very high level of accuracy. The surrogate's predictions closely track the ground-truth values from SWMM, successfully capturing the magnitude and timing of multiple rainfall-driven events. Quantitatively, the mean NSE across all 10 runs exceeds 0.98 for all variables. This indicates that the surrogate model explains over 98% of the variance in the true system state for historical conditions, confirming its high fidelity and reliability for operational forecasting.

Evaluation on Unseen Storm Data

The model was next subjected to a more challenging evaluation using a 12-hour, 10-year design storm. The precipitation data for this storm is shown in **Figure 2.8**. As shown in **Figure 2.9**, the surrogate again performs admirably, successfully capturing the overall hydrograph shape, including the rising limb, peak, and recession

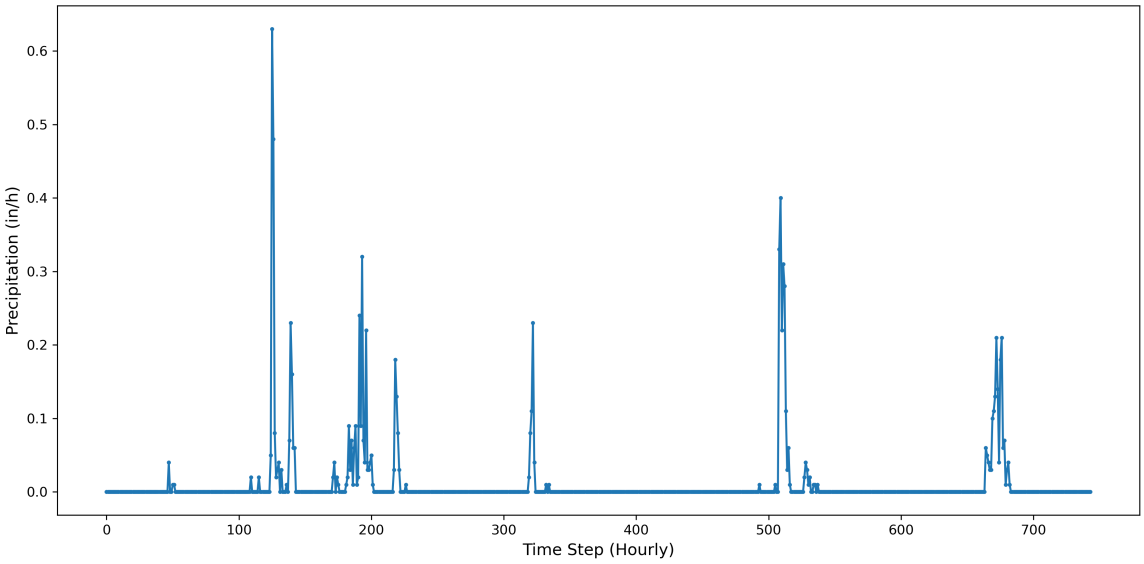


Figure 2.5: Precipitation data for the month of December 2013.

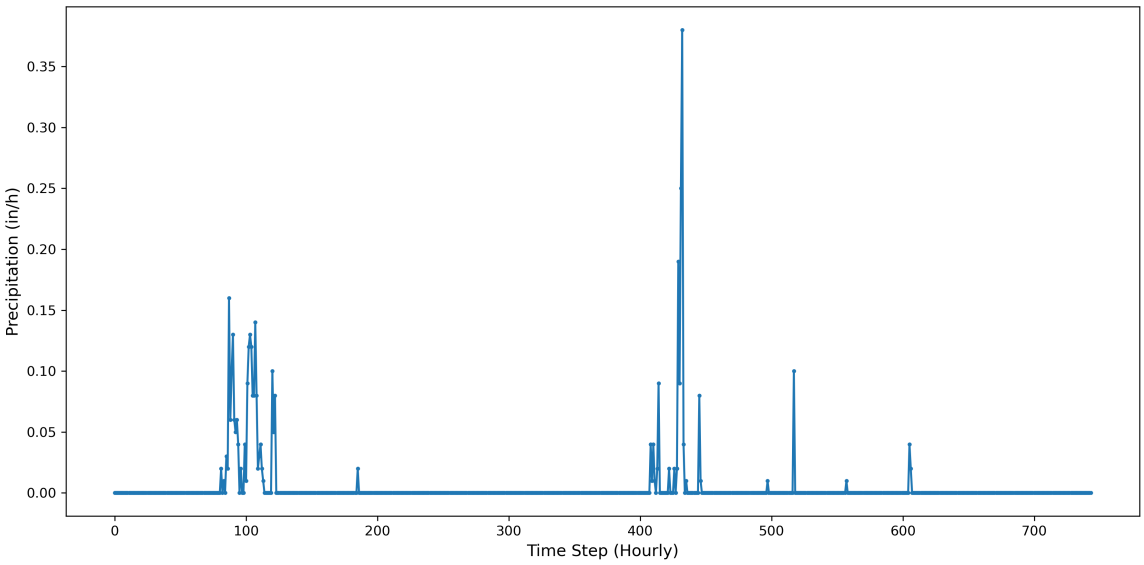


Figure 2.6: Precipitation data for the month of May 2013.

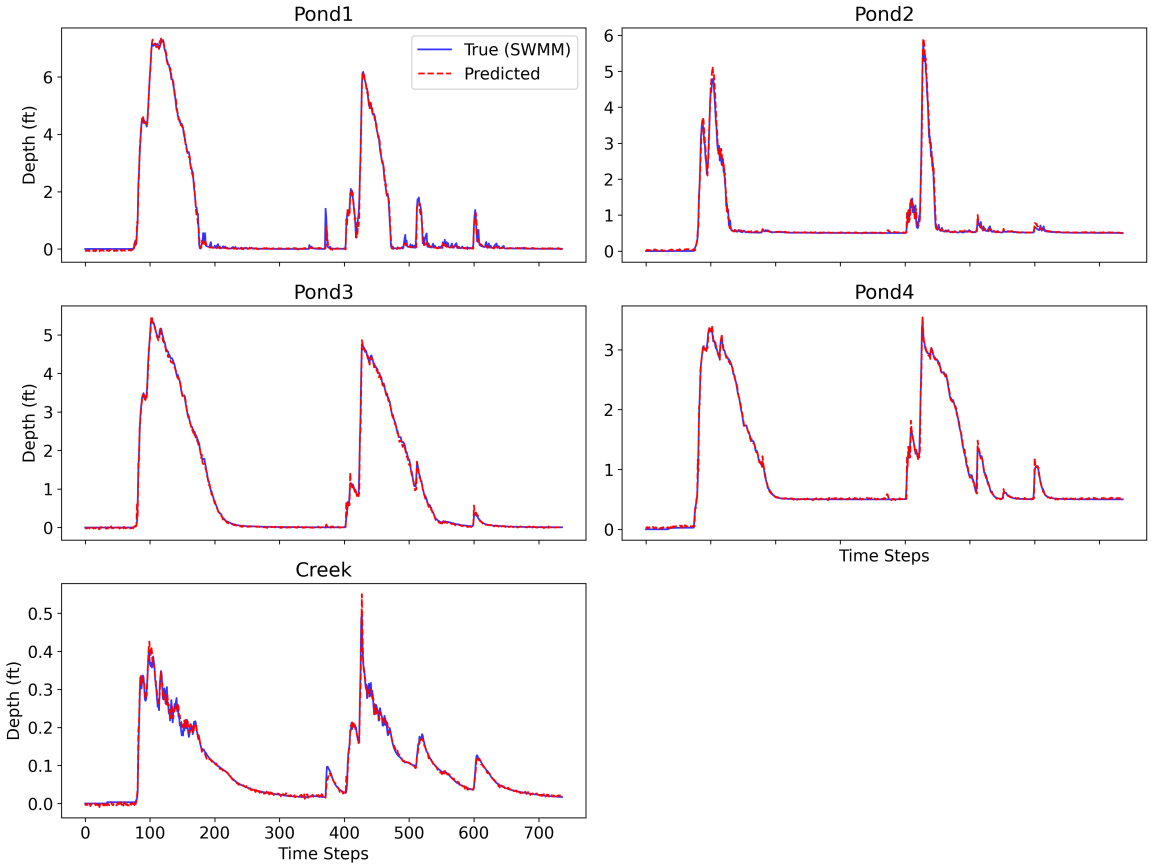


Figure 2.7: Comparison of surrogate model predictions against SWMM ground truth for a representative run on unseen historical data (May 2013).

Table 2.1: Statistical Summary of Surrogate Model Performance on Unseen Historical Data (May 2013) across 10 runs. Values are shown as Mean ± Standard Deviation.

Variable	**RMSE (ft)**	**NSE**
depth_stream	0.0110 ± 0.0033	0.9875 ± 0.0065
depth_pond1	0.2186 ± 0.2090	0.9905 ± 0.0111
depth_pond2	0.0885 ± 0.0055	0.9919 ± 0.0017
depth_pond3	0.0750 ± 0.0212	0.9976 ± 0.0014
depth_pond4	0.0464 ± 0.0052	0.9972 ± 0.0008

of the flood wave for all ponds. The quantitative metrics in **Table 2.2** confirm this strong performance, with mean NSE values for the pond depths remaining high (0.96 and over). The model's ability to capture the general shape of storm hydrographs is particularly significant, as these dynamic periods are often the most critical for control. Overall, these results validate the strategy of training on a balanced dataset of both historical and storm events to create a robust model.

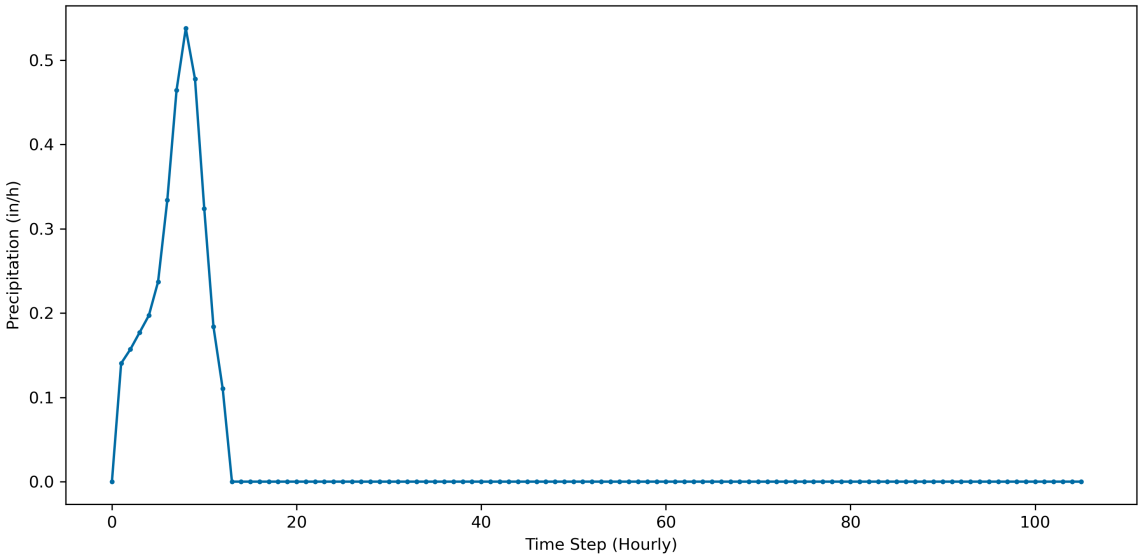


Figure 2.8: Precipitation data for the 12-hour 10-year storm.

Table 2.2: Statistical Summary of Surrogate Model Performance on an Unseen 12-hr, 10-yr Storm across 10 runs. Values are shown as Mean ± Standard Deviation.

Variable	RMSE (ft)	NSE
depth_stream	0.0247 ± 0.0013	0.9635 ± 0.0055
depth_pond1	0.1034 ± 0.0049	0.9986 ± 0.0002
depth_pond2	0.1546 ± 0.0114	0.9983 ± 0.0003
depth_pond3	0.0555 ± 0.0044	0.9988 ± 0.0003
depth_pond4	0.0398 ± 0.0012	0.9985 ± 0.0001

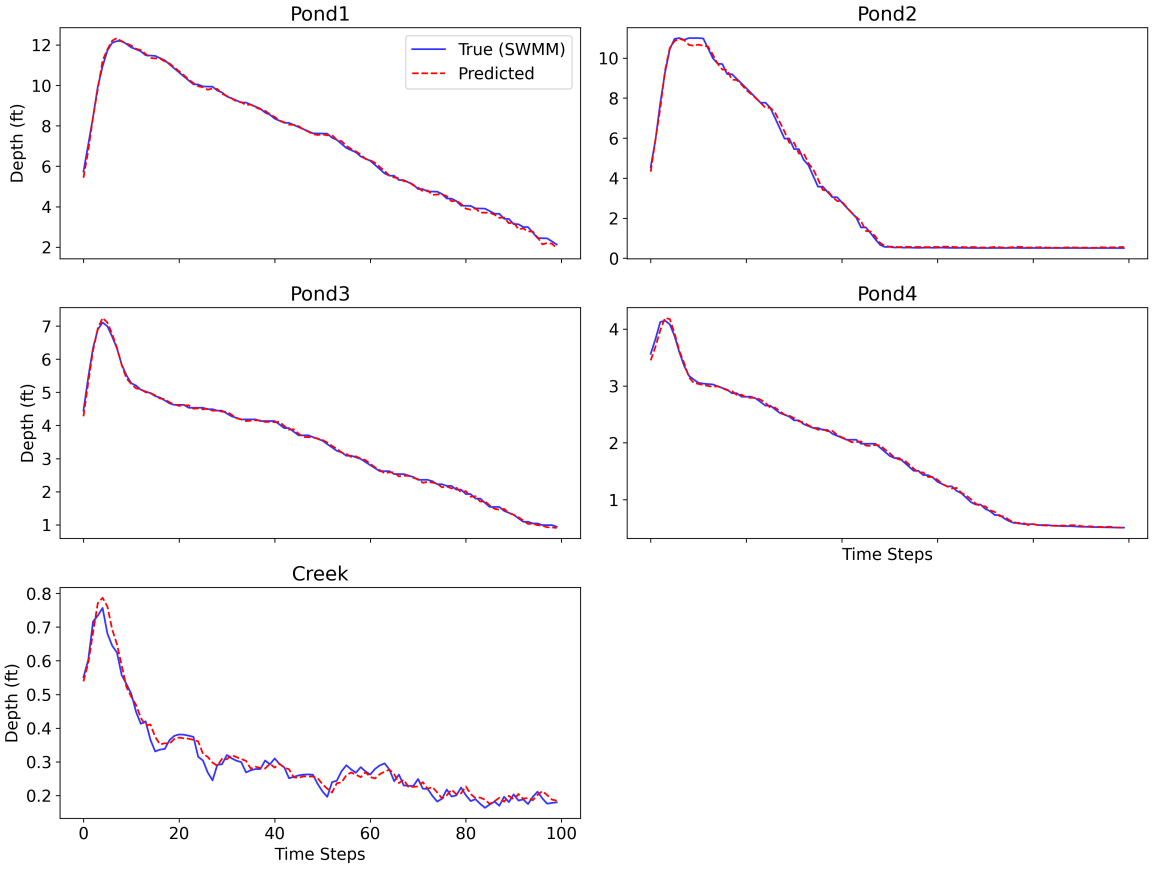


Figure 2.9: Comparison of surrogate model predictions against SWMM ground truth for a representative run on an unseen 12-hour, 10-year design storm.

2.3.2 Agent Training Acceleration

The most critical finding of this research is the dramatic acceleration in DRL agent training enabled by the surrogate model. As summarized in **Table 2.3**, two A2C agents were trained under identical conditions for a fixed 5-hour time budget. The surrogate-trained agent (Agent S) completed 1,191 full episodes, whereas the agent trained directly on SWMM (Agent W) completed only 36 episodes—a 33-fold increase in learning experiences. This result empirically validates the surrogate modeling approach as a solution to the computational difficulty of direct DRL training on high-fidelity simulators.

The learning curves provide a stark visual representation of the surrogate's impact. **Figure 2.10** illustrates the global difference in training volume. In the same 5-hour window, Agent S (Blue) traverses over 1,000 episodes, effectively exploring the state space and optimizing its reward. Agent W (Red), constrained by the high computational cost of physics-based simulation, appears as a brief fragment at the start of the timeline, unable to gather sufficient experience to learn.

Figure 2.11 offers a closer inspection of the policy evolution. Agent S (Panel A) demonstrates a clear learning trajectory, with the rolling average reward improving from -40,000 to approximately -29,000. Conversely, Agent W (Panel B) shows characteristic signs of early-stage training: high variance and a flat trend line. The agent simply lacks the iteration count required to approximate the policy gradient, confirming that direct SWMM training is infeasible for this task under practical time constraints.

Table 2.3: Comparison of Agent Training Performance over a 5-Hour Time Budget.

Metric	**Agent S (Surrogate)**	**Agent W (SWMM)**
Total Episodes Completed	1,191	36
Mean Reward per Episode	-34,082.2	-42,236.9
Best (Max) Reward	-28,449.2	-39,968.4
Mean Duration per Episode (sec)	15.1	501.9

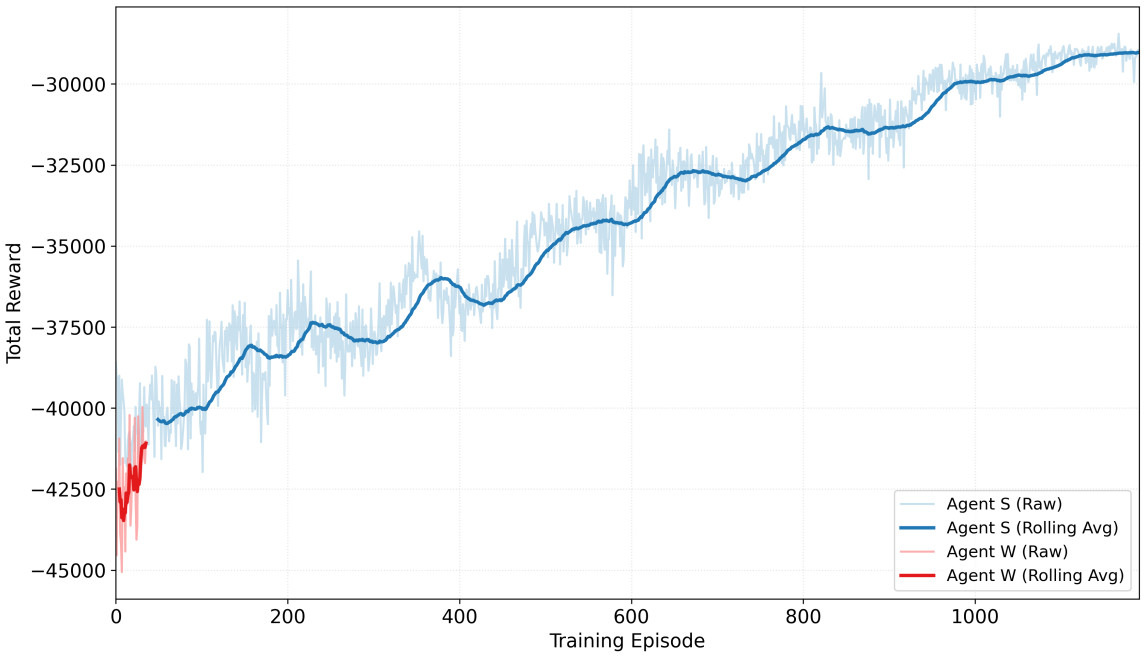


Figure 2.10: Impact of computational speed on learning within a fixed 5-hour budget. Agent S (Blue) leverages the surrogate to complete 1,191 episodes, allowing for policy convergence. In contrast, Agent W (Red) is limited to 36 episodes by the SWMM simulator, effectively stalling the learning process.

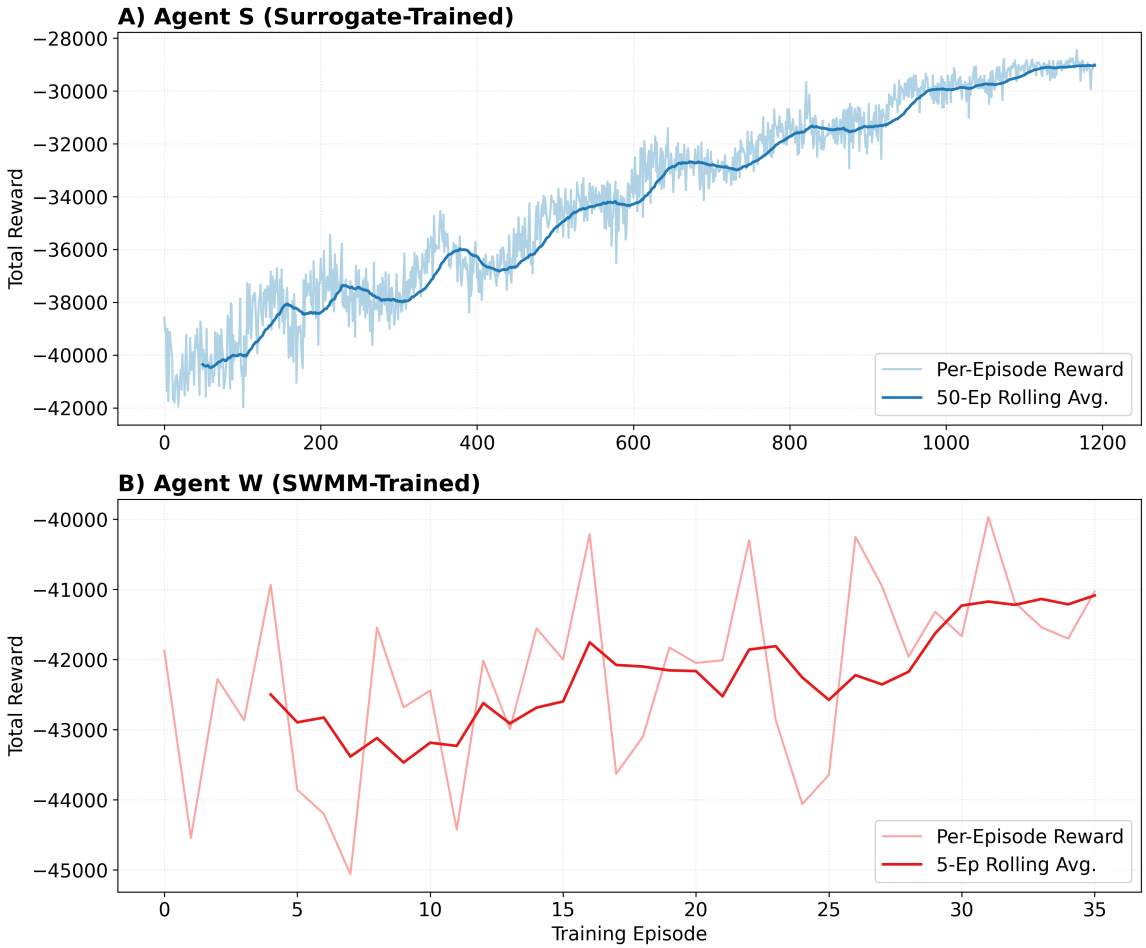


Figure 2.11: Detailed view of learning dynamics. Top (A): Agent S explores the environment and steadily optimizes the policy, as evidenced by the rising rolling average. Bottom (B): Agent W remains trapped in the initial exploration phase with high variance and no discernible improvement.

2.4 Discussion

The results confirm the central hypothesis of this research: a well-validated BiLSTM surrogate model can accurately emulate the non-linear dynamics of a SWMM system and dramatically accelerate the training of a DRL agent.

The surrogate's ability to replicate hydraulic behavior, particularly in the creek conduit (stream) where it achieved an NSE of 0.96 during design storms, is a critical success factor. Since the erosion penalty in the reward function is calculated based on shear stress within this specific conduit, the agent's ability to learn erosion mitigation depends entirely on the surrogate's precision in predicting channel depth. The high fidelity observed suggests that the BiLSTM architecture successfully captured the complex routing relationship between the pond discharges and the creek water levels, providing the agent with a reliable feedback signal for policy optimization.

The comparative training experiment highlights a fundamental challenge in applying DRL to physical simulations: sample efficiency. The failure of Agent W (SWMM-trained) to converge is not a reflection of the algorithm's capability, but rather a consequence of data starvation. With only 36 episodes completed in the 5-hour budget, the agent simply did not encounter enough state-action pairs to approximate the gradient of the reward function. In contrast, Agent S, leveraging the 33-fold acceleration provided by the surrogate, was able to traverse the exploration phase and exploit the learned policy within a practical timeframe. This confirms that surrogate modeling is not merely a convenience, but a necessity for applying model-free DRL algorithms to computationally expensive hydraulic simulations.

2.4.1 Limitations and Future Work

While this study successfully demonstrates the feasibility of this training framework, specific limitations define the path for future research. First, the generalizability of the surrogate model is strictly bound by its training data distribution. While the dataset included the rainiest historical year (2013) and high-intensity design

storms (10-year and 25-year), the surrogate's performance on events significantly outside these parameters remains unverified. Future iterations should expand the training dataset with a wider diversity of synthetic storm profiles to further enhance robustness.

Second, this research focused on validating the training methodology. We evaluated the agent's performance within the surrogate environment. The critical next step is "Sim-to-Real" transfer—validating the policy learned by Agent S by deploying it back into the high-fidelity SWMM environment (or the real world). This step is necessary to quantify potential performance degradation caused by minor epistemic errors in the surrogate's predictions.

Finally, the state space was intentionally simplified to rely on depth measurements rather than flow rates. While this enhances practical deployability, it may omit hydraulic information that could further optimize control. Future work should explore the trade-off between state-space complexity and agent performance, particularly determining if the inference of flow dynamics from depth sequences is sufficient for more complex topology control.

2.5 Conclusion

This research successfully addressed the primary computational barrier limiting the application of Deep Reinforcement Learning (DRL) to high-fidelity stormwater system modeling. By replacing the computationally expensive SWMM simulator with a well-validated BiLSTM surrogate (mean NSE > 0.96), we achieved a 33-fold acceleration in training speed. This efficiency allowed an A2C agent to complete 1,191 episodes and converge to a competent multi-objective control policy within a fixed 5-hour budget, whereas an agent trained directly on the physics-based model completed only 36 episodes and failed to learn. These results empirically demonstrate that surrogate modeling provides a robust and scalable pathway for deploying advanced, adaptive control strategies in urban water systems.

Software & Data Availability

The code for the SWMM and surrogate environments, along with the SWMM model and rainfall data used in this study, are available at https://github.com/isifletcher/Accelerating-Decision-Making-for-Smart-Stormwater-Management-Through-SWMM-Surrogate-Modeling-for-RL.

Chapter 3

Performance-Based Explainability for Deep Reinforcement Learning in Stormwater Control

Disclosure

Author Contributions and Roles: Isidora Fletcher was the primary investigator and lead author for this study. She was responsible for the conceptual adaptation of the framework, the implementation of the diagnostic workflow, the performance of the forensic analysis, and the drafting of the manuscript.

Anahita Khojandi and Jon Hathaway supervised the research. They provided methodological guidance, supervised the experimental design, and provided critical review of the manuscript drafts.

Abstract

Deep Reinforcement Learning (DRL) demonstrates significant potential for the Real-Time Control (RTC) of stormwater infrastructure, yet its adoption is hindered by the "Black Box" problem. Operators require transparent, performance-based justifications for autonomous decisions, particularly when managing conflicting flood and erosion risks. This study presents a novel framework, Surrogate-Accelerated SVERL-P (Shapley Value Estimation for RL-Performance), which utilizes a high-fidelity Bidirectional Long Short-Term Memory (BiLSTM) surrogate model to enable detailed post-hoc explainability. Unlike standard interpretability methods that map inputs to actions, our proposed methodology explains agent performance by quantifying the marginal contribution of each input to the obtained reward. We validate our proposed framework using a Logic Agent proxy, demonstrating that the engine correctly decouples causal triggers from environmental stressors. The results show that our proposed methodology allows engineers to quantify the Value of Information, identifying redundant sensors to optimize monitoring networks. Furthermore, the framework serves as a forensic tool: it successfully identifies the

root cause of policy collapse during training and pinpoints a structural topological conflict between upstream storage and downstream erosion constraints in a Conner Creek (Knoxville, TN) testbed. Our proposed methodology provides the necessary transparency to audit DRL failures and transition from theoretical simulation to trusted deployment in sensitive urban watersheds.

3.1 Introduction

Urban drainage systems (UDSs) globally face escalating challenges due to increasing urbanization and the intensifying effects of climate change, which accelerate the frequency and severity of urban flooding and receiving water degradation Sadler et al. (2020); Moriasi et al. (2007); Skaggs et al. (2012). These systems are vital for managing stormwater, yet conventional static infrastructure is often insufficient to cope with the stochastic and non-linear dynamics of modern storm events Mullapudi et al. (2020); Saliba et al. (2020). To address these limitations, the transformation of UDSs into adaptive systems through Real-Time Control (RTC) has emerged as a necessary paradigm shift. RTC leverages smart sensors and actuators to dynamically govern flows Sadler et al. (2020); Kerkez et al. (2016); Baumann et al. (2022b), enhancing system resilience and mitigating ecological damage such as downstream channel erosion Bowes et al. (2022); Akin (2021).

In recent years, Deep Reinforcement Learning (DRL) has emerged as a state-of-the-art methodology for RTC in hydrology Mnih et al. (2015); Sutton and Barto (2018); He et al. (2025). DRL trains an autonomous agent to learn an optimal control policy by interacting with a simulation environment, allowing it to handle non-linear hydrodynamics and manage complex multi-objective trade-offs—such as balancing flood prevention with erosion control—more effectively than static rule-based control (RBC) or linear control methods Mullapudi et al. (2020); Bowes et al. (2020); Wang et al. (2020); Tian et al. (2022a). Unlike Model Predictive Control (MPC), which requires computationally expensive optimization during operation, DRL shifts the

computational burden to the training phase, allowing for rapid decision-making during critical storm events Saliba et al. (2020); Fuente and Guerra (2024); Fletcher et al. (2026).

However, the widespread deployment of DRL in safety-critical infrastructure is currently stalled by the "Black Box" problem. While DRL offers superior control capabilities, its decision-making logic—encoded in the weights of deep neural networks—is inherently opaque He et al. (2025); Tian et al. (2024b). This lack of transparency poses a fundamental barrier to adoption in the water sector, where utility managers and engineers require interpretable justifications to ensure safety and regulatory compliance Aderemi et al. (2025). Recent surveys of the water industry indicate that "lack of trust" and "opaqueness" are among the primary impediments to AI integration ASCE Task Committee (2023). Without a transparent mechanism to explain *why* an agent prioritizes specific actions, engineers cannot distinguish between a robust control strategy and a reward-hacking failure. For example, an agent might learn to oscillate valves rapidly to marginally improve a numerical score (chattering), physically damaging the actuators, or it might sacrifice a specific upstream pond to maximize a global objective, leading to localized flooding liabilities Mullapudi et al. (2020); Kerkez et al. (2016).

To bridge this gap, recent hydrological studies have begun to adopt game-theoretic approaches, such as Shapley Additive exPlanations (SHAP), to interpret "Black Box" models. These techniques have been successfully applied to debug large-scale hydrologic models Husic et al. (2025) and to investigate the hydro-meteorological drivers of reservoir releases Fan et al. (2023). However, these applications have largely been limited to supervised learning tasks (prediction). Applying interpretability to DRL (control) presents a distinct challenge: engineers do not just need to know which inputs triggered an action (policy explanation); they need to quantify how those inputs contributed to the physical outcome (performance explanation). For instance, simply knowing that the rainfall input caused a valve to open is insufficient; an

operator needs to know if that decision successfully prevented a downstream erosion violation or if it was an unnecessary release that wasted storage capacity.

This study addresses this challenge by applying our proposed Surrogate-Accelerated SVERL-P (Shapley Value Estimation for Reinforcement Learning-Performance) framework Beechey et al. (2023) to stormwater control. SVERL-P offers a rigorous game-theoretic method to quantify the marginal contribution of each state feature—such as specific pond depths or rainfall intensity—to the total cumulative reward. However, calculating Shapley values for RL requires running thousands of counterfactual simulations to marginalize features, a task that is computationally prohibitive when coupled with high-fidelity physics solvers like SWMM which take seconds or minutes per timestep Bowes et al. (2020); Tian et al. (2022a); Razavi et al. (2012). To resolve this computational bottleneck, we leverage a high-fidelity Bidirectional LSTM (BiLSTM) surrogate model—developed and validated in our prior work Fletcher et al. (2026)—to enable the intensive sampling required for explanation. Our proposed framework addresses three critical needs identified in the recent literature: (1) it provides a diagnostic tool to detect reward loopholes; (2) it quantifies the "Value of Information" for sensor optimization; and (3) it demonstrates that surrogate modeling is a prerequisite for rigorous, real-time explainability in hydrology. We validate this approach in the Conner Creek watershed testbed using a Logic Agent proxy to demonstrate that the interpretability metric aligns with physical ground truths.

3.2 Methodology

Our proposed methodology establishes a rigorous framework for auditing and establishing trust in autonomous water systems. We propose Surrogate-Accelerated SVERL, a diagnostic pipeline that integrates the high-speed predictive power of neural emulators with the axiomatic rigor of game-theoretic attribution. As illustrated in **Figure 3.1**, the framework is organized into three primary modules: (1) the Virtual

Watershed Testbed, which serves as a high-fidelity simulation environment; (2) the Autonomous Operator, consisting of a DRL agent optimized for multi-objective control; and (3) the SVERL-P Interpretability Engine, which performs the forensic audit of the agent's decision-making logic.

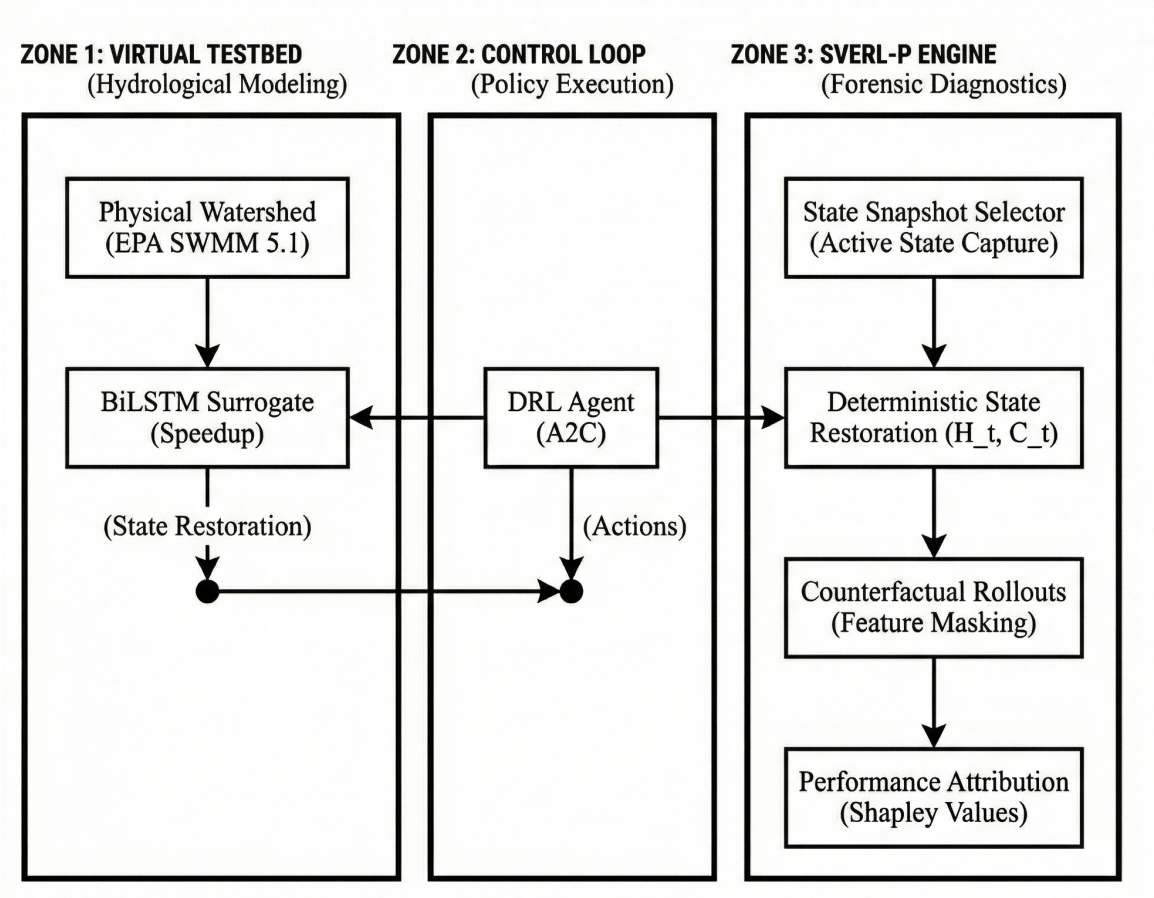


Figure 3.1: Overview of the Surrogate-Accelerated SVERL-P framework. The pipeline consists of: (1) a Virtual Watershed Testbed where a BiLSTM surrogate emulates physical SWMM dynamics; (2) a Control Loop where the DRL agent executes multi-objective actions; and (3) the SVERL-P Engine, which utilizes deterministic state restoration to perform thousands of parallel counterfactual rollouts for performance-based diagnostic auditing.

3.2.1 Virtual Watershed Environment and Surrogate Interface

Physical System: Conner Creek Watershed

The experimental testbed is modeled after the Conner Creek watershed in Knoxville, TN. The system is represented by a calibrated EPA SWMM 5.1 model, which serves as the physical ground-truth for generating high-fidelity training data Rossman (2015). The network, as illustrated in **Figure 3.2**, consists of four retention ponds connected in a mixed series/parallel configuration, ultimately discharging into a natural creek channel (stream).

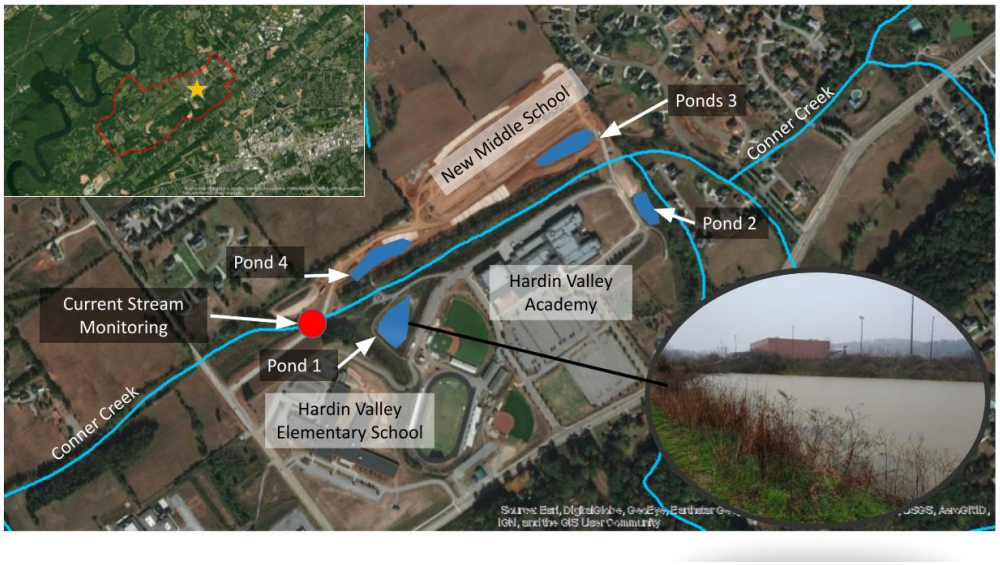


Figure 3.2: Schematic of the Conner Creek watershed control system. The agent monitors rainfall and pond levels to modulate four valves, managing the trade-off between upstream storage utilization and downstream channel protection.

The geomorphology of Conner Creek makes it a high-priority site for hydromodification management, as the natural banks are sensitive to sustained high-shear events.

The BiLSTM Surrogate: Computational Engine for Explainability

To computationally enable the intensive sampling required for game-theoretic explainability, our proposed framework utilizes a high-fidelity surrogate model. This model utilizes a BiLSTM architecture, which is uniquely suited for hydrological time-series due to its ability to process sequences in both forward and backward directions, capturing long-term dependencies in rainfall-runoff patterns. As established in our prior work Fletcher et al. (2026), the surrogate consists of two recurrent layers with 128 units each, trained to emulate hydraulic behavior with a Nash-Sutcliffe Efficiency (NSE) > 0.95.

The 33× execution speedup provided by the BiLSTM is a structural prerequisite for SVERL-P. While a single SWMM simulation may require several minutes to process a 24-hour storm, the BiLSTM allows for the thousands of parallel lookahead rollouts necessary to calculate Shapley values with mathematical precision. To maintain the integrity of the experimental boundary, the DRL agent is trained and audited entirely within this surrogate-enabled environment, which serves as the authoritative representation of the underlying physics for this study.

Deterministic State Restoration and Markovian Consistency

To support rigorous explainability, we implemented a deterministic state restoration mechanism. In recurrent architectures like the BiLSTM, the simulation state is not merely a function of physical variables (e.g., pond depth) but also the hidden state (H_t) and cell state (C_t) vectors that encode hydrological memory. Simply resetting water levels would violate Markovian consistency, leading to inaccurate performance attributions. Our proposed interface serializes these memory vectors at every timestep, allowing the SVERL-P engine to "checkpoint" the simulation and replay counterfactual trajectories from the exact same internal neural state.

3.2.2 The Autonomous Operator: DRL Framework

Agent Architecture

To handle the continuous state-space of the virtual watershed and provide stable control updates, we employ an Advantage Actor-Critic (A2C) agent. A2C is a robust reinforcement learning algorithm that utilizes a parallelized framework to manage the stochastic nature of storm-driven environments Mnih et al. (2016); Fletcher et al. (2026). The agent functions as an autonomous operator using a shared-backbone neural network architecture.

The network processes a 10-dimensional input vector (normalized to $[0, 1]$) through a shared encoder consisting of two fully connected layers with 128 units each, utilizing ReLU activation functions. This shared body extracts high-level hydraulic features which are then bifurcated into separate output heads:

- **Actor Heads:** The network splits into four independent actor heads, one for each controllable valve. Each head outputs a Softmax probability distribution over the discrete action space.
- **Critic Head:** A single linear output head estimates the Value Function $V(s)$, predicting the expected cumulative future reward to guide policy optimization.

Action Space Definition

To prevent erratic behavior and mimic the mechanical constraints of physical actuators, we discretize the control outputs. For each of the four valves, the agent selects an action a_i from the following permissible states:

$$a_i \in \begin{cases} 0\% & \text{(Closed)} \\ 50\% & \text{(Throttled)} \\ 100\% & \text{(Fully Open)} \end{cases}$$

This results in a MultiDiscrete action space of dimension $[3, 3, 3, 3]$, where the agent simultaneously determines the status of all valves at each timestep.

Reward Function Design and Ecological Rationale

The reward function is the mathematical representation of the regulatory and ecological constraints the agent must satisfy. We identify that the formulation of these rules dictates the success of the multi-objective balance. We evaluate two distinct reward configurations to understand the agent's behavior:

1. **Constraint-Heavy Baseline:** This configuration weights the penalty for downstream erosion ($w_{erosion} = 15,000$) significantly higher than upstream flood safety ($w_{flood} = 5,000$). This represents a conservative ecological stance where hydromodification is the primary concern.

2. **Optimized Multi-Objective Formulation:** To prevent policy collapse—where the agent adopts a static locked-open policy to minimize reward variance—we incorporate an Action Penalty (Δa). By penalizing the agent slightly for holding valves in extreme positions without justification, we encourage reactive control behavior.

The final reward function R_t used for the optimized agent is:

$$R_t = -\left(w_f \cdot N_{flood} + w_e \cdot \tau_{excess}^2 + w_o \cdot N_{overflow} + w_a \cdot \Delta a + w_d \cdot \sum d_{pond}^2\right) \quad (3.1)$$

where N_{flood} is a binary flag for weir overtopping, and τ_{excess}^2 is the squared shear stress above the critical limit. The inclusion of the action penalty ($w_a = 5.0$) proved to be the necessary regularizer to maintain policy stability during long-duration training.

3.2.3 The Interpretability Engine: SVERL-P

We adapt the Shapley Value Estimation for Reinforcement Learning (SVERL) framework Beechey et al. (2023) to quantify the Value of Information for sensor optimization.

SVERL-Performance (SVERL-P)

Standard interpretability methods often focus on policy explanation (identifying which inputs triggered an action). However, hydrological operators require a performance explanation to quantify how those inputs contributed to the ultimate physical outcome. We utilize SVERL-P, which calculates the marginal contribution of each sensor measurement (e.g., rain gauge, depth sensor) to the total cumulative future reward over a lookahead horizon ($H = 6$ hours). This horizon was selected to capture the relevant hydraulic time-of-concentration for the Conner Creek watershed.

Marginalization through Feature Masking

To isolate the performance contribution of a specific gauge, we simulate a state of feature masking. For example, to determine the value of a specific depth sensor, we run rollouts where that data point is hidden and replaced with background noise sampled from a historical dataset D_{bg}. By comparing the agent's performance with and without specific sensor data across all possible feature coalitions, we quantify the actual physical utility of the information. The computational procedure for SVERL-P, including the handling of BiLSTM memory vectors (H_t, C_t) during rollouts, is detailed in **Algorithm 3.1**.

Algorithm 3.1 Surrogate-Accelerated SVERL-P Calculation

Require: Trained Agent π, Surrogate Env E, Target State s_{target}, Horizon H
Require: Background Dataset D_{bg} (Historical samples)
Initialize characteristic values dictionary $V \leftarrow \{\}$
Generate all feature coalitions C_{all} ▷ Power set of state features
for each coalition $c \in C_{all}$ **do**
    $R_{cumulative} \leftarrow 0$
    for $k \leftarrow 1$ to K samples **do** ▷ Expectation over background distribution
        E.restore_state(s_{target}) ▷ Synchronize physics and LSTM memory
        $\hat{s} \leftarrow$ Mask(s_{target}, c, Sample(D_{bg})) ▷ Marginalize missing features
        $r_{episode} \leftarrow 0$
        for $h \leftarrow 1$ to H **do** ▷ Lookahead Rollout (Lookahead=6hr)
            $a_h \leftarrow \pi(\hat{s})$ ▷ Policy execution on hybrid state
            $s_{next}, r, \cdots \leftarrow E.\text{step}(a_h)$ ▷ Surrogate state transition
            $r_{episode} \leftarrow r_{episode} + r$
            $\hat{s} \leftarrow$ Mask(s_{next}, c, Sample(D_{bg}))
        end for
        $R_{cumulative} \leftarrow R_{cumulative} + r_{episode}$
    end for
    $V[c] \leftarrow R_{cumulative}/K$
end for
return ComputeShapleyValues(V) ▷ Game-theoretic attribution

3.2.4 Experimental Protocol

Stress Testing with Synthetic Climate Data

To ensure the agent is robust to climate change scenarios beyond historical observations, we generated a custom stress test dataset using the Huff Rainfall Distribution method.

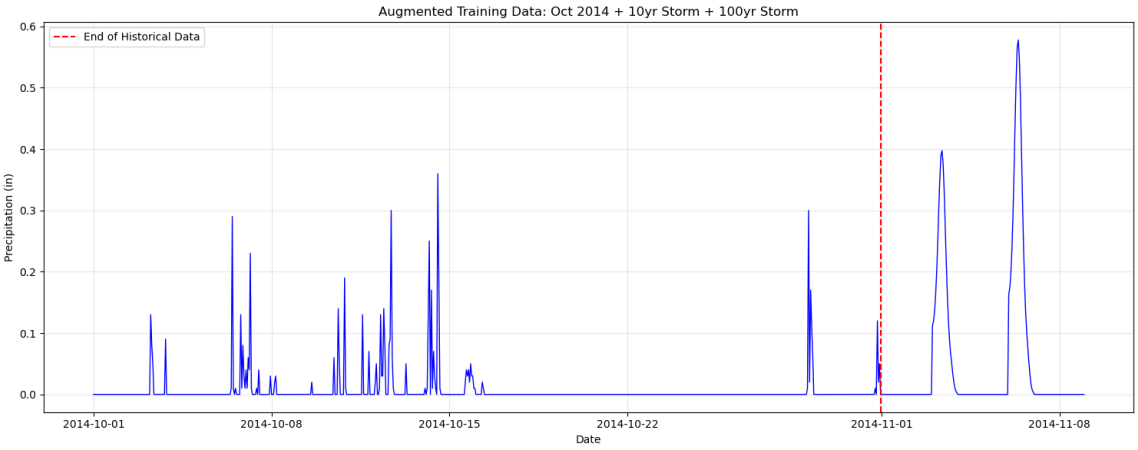


Figure 3.3: Design of the stress test dataset. Synthetic 10-year and 100-year design storms are stitched after historical data, separated by 48-hour dry periods. This design protocol is critical as it forces the agent to not only manage the storm peak but also actively drain the storage ponds during the cooldown recovery period to restore capacity for subsequent events.

As shown in **Figure 3.3**, synthetic 10-year and 100-year design storms were stitched into the timeline, separated by 48-hour dry periods. This design protocol is critical as it forces the agent to not only manage the storm peak but also actively drain the storage ponds during the cooldown recovery period to restore capacity for subsequent events.

Diagnostic Pipeline: Snapshot Selection

Calculating Shapley values is computationally intensive. Analyzing every timestep of a multi-day simulation would generate significant data redundancy. Therefore, we implemented a two-stage diagnostic pipeline defined in **Algorithm 3.2**. First, the

system scans the simulation to detect active states—moments where the agent makes a significant control decision or the system water levels exceed safety thresholds. Only these critical snapshots, including the necessary BiLSTM hidden states, are stored for the detailed SVERL analysis.

Algorithm 3.2 Diagnostic State Collection Workflow

Require: Trained Agent π, Environment E, Safety Threshold ϵ

Initialize storage $S_{interesting} \leftarrow \emptyset$
$s \leftarrow E.\text{reset}()$
while simulation not done **do**
    $a \leftarrow \pi(s)$
    $s', r, \cdots \leftarrow E.\text{step}(a)$
    $ValveOpen \leftarrow (\sum a > 0)$
    $HighWater \leftarrow (\max(s_{depths}) > \epsilon)$
    if $ValveOpen$ **or** $HighWater$ **then**
        $Snapshot \leftarrow E.\text{get_state}()$ ▷ Capture LSTM hidden states
        $S_{interesting}.\text{append}(Snapshot)$
    end if
    $s \leftarrow s'$
end while
Store $S_{interesting}$ to disk for SVERL-P analysis

Validation Proxy: The Logic Agent

To provide a benchmark for interpretability, we utilize a hard-coded Logic Agent as a ground-truth proxy. Since the decision rules of this agent are known, it serves as the calibration standard for SVERL-P. This proxy operates on a Safety Buffer Heuristic, where the decision logic is based on the ratio of current depth (d_t) to the maximum safe operating depth (d_{max}).

We define the decision rule as:

$$a_t = \begin{cases} 100\% \text{ (Open)} & \text{if } \frac{d_t}{d_{max}} > \gamma \\ 0\% \text{ (Closed)} & \text{otherwise} \end{cases} \tag{3.2}$$

where the threshold $\gamma = 0.7$ ensures the agent maintains a 30% safety buffer. If SVERL-P is accurate, it must demonstrate that Pond Depth is the positive driver of control actions, while identifying Rainfall as a negative stressor that reduces the potential reward.

3.3 Results and Discussion

The application of the Surrogate-Accelerated SVERL-P framework enables a multi-stage behavioral audit of autonomous stormwater control policies. The results are presented in three distinct analytical stages: (1) **Framework Validation**, where the interpretability tool is verified against a known ground-truth; (2) **Failure Diagnosis**, where SVERL-P is utilized to identify the root cause of policy collapse during training; and (3) **Structural Analysis**, where we evaluate the limits of reward shaping in the presence of watershed-scale topological conflicts.

3.3.1 Framework Validation: Ground-Truth Calibration via Logic Proxy

Prior to auditing the opaque DRL neural networks, we verify that SVERL-P accurately reports physical causality. As defined in **Section 3.2.4**, the Logic Agent operates on a fixed 0.7 safety buffer threshold. We subjected this agent to a synthetic 100-year storm event to evaluate the resulting attribution values.

As illustrated in **Figure 3.4**, the SVERL-P engine successfully decoupled the control trigger from the environmental stressor:

- **Trigger Identification:** The engine assigned high positive Shapley values to Pond Depth. This confirms that the agent's performance is primarily driven by its awareness of the hydraulic head relative to the safety buffer, aligning with the agent's internal heuristic.

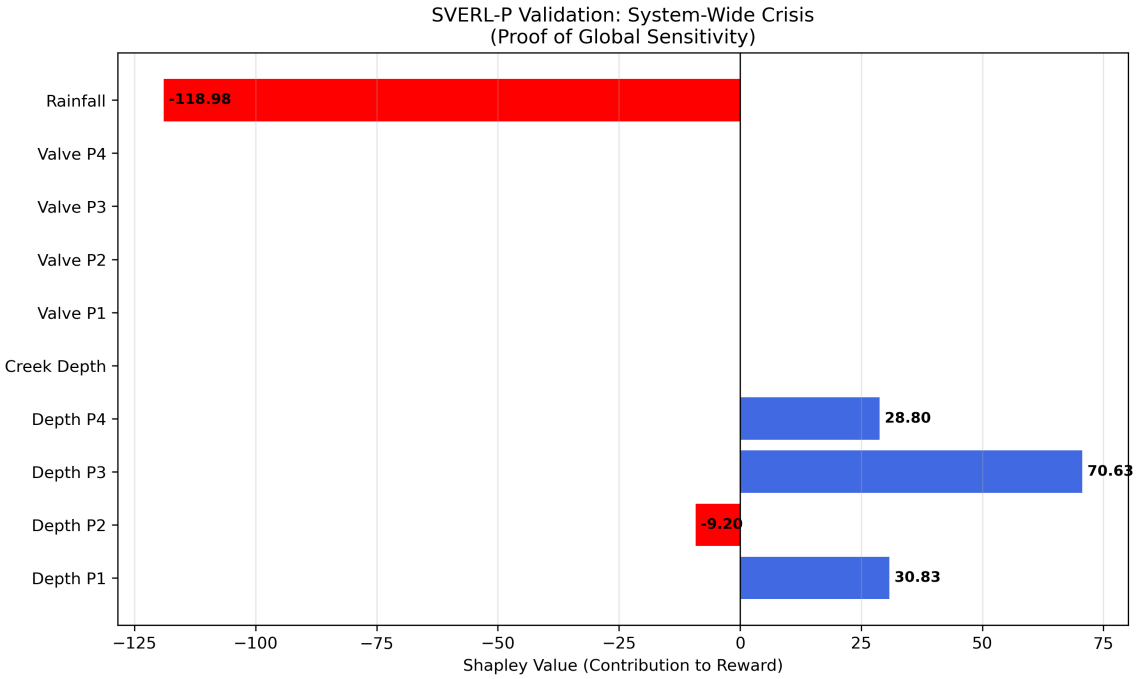


Figure 3.4: SVERL-P validation results for the Logic Agent proxy. The analysis correctly attributes the positive reward contribution to Pond Depth (the trigger for opening valves) while identifying Rainfall as a negative environmental stressor. Irrelevant sensors, such as Creek Depth, are assigned zero value, confirming the engine's ability to filter stochastic noise.

- **Stressor Attribution:** Rainfall received a negative value. This distinction is critical for hydrological interpretability; SVERL-P correctly identifies that rainfall does not help the agent but rather creates an unavoidable hydraulic load that decreases the maximum achievable reward.

- **Noise Rejection:** Sensors excluded from the Logic Agent's hard-coded rules (e.g., Creek Depth) were assigned Shapley values of zero. This confirms the engine's precision in reporting only causal drivers without hallucinating correlations.

3.3.2 Failure Diagnosis: Identifying Policy Collapse in Long-Duration Training

A significant risk in deploying autonomous systems for water infrastructure is the phenomenon of policy collapse—where an agent discovers that a static strategy is statistically safer than attempting complex, multi-objective control. We utilized SVERL-P to audit the agent's strategy evolution over a 20-hour training period on the Stress Test dataset.

Table 3.1 details the degradation of the control strategy. During early training (4–12 hours), the agent demonstrates dynamic behavior, modulating valves to throttle flow and protect the downstream channel. However, as training progresses to 20 hours, the agent converges to a static always-open strategy to minimize the variance of the high erosion penalty ($w_{erosion} = 15,000$).

Table 3.1: Longitudinal audit of agent behavior during extreme design storms. The convergence toward an Average Valve setting of 2.00 (Fully Open) at 16–20 hours indicates a transition from reactive control to static policy collapse.

Training Time	Max Pond Depth	Flood Risk	Max Shear (τ)	Eco-Status	Avg Valve (0-2)	Verdict
4 Hours	13.54 ft	Critical	0.099	Violation	1.17	Reactive
8 Hours	13.55 ft	Critical	0.100	Violation	1.91	Aggressive
12 Hours	**13.33 ft**	**Critical**	**0.098**	**Violation**	**1.74**	**Strategic Modulation**
16 Hours	13.33 ft	Critical	0.098	Violation	2.00	Policy Collapse
20 Hours	13.33 ft	Critical	0.098	Violation	2.00	Policy Collapse

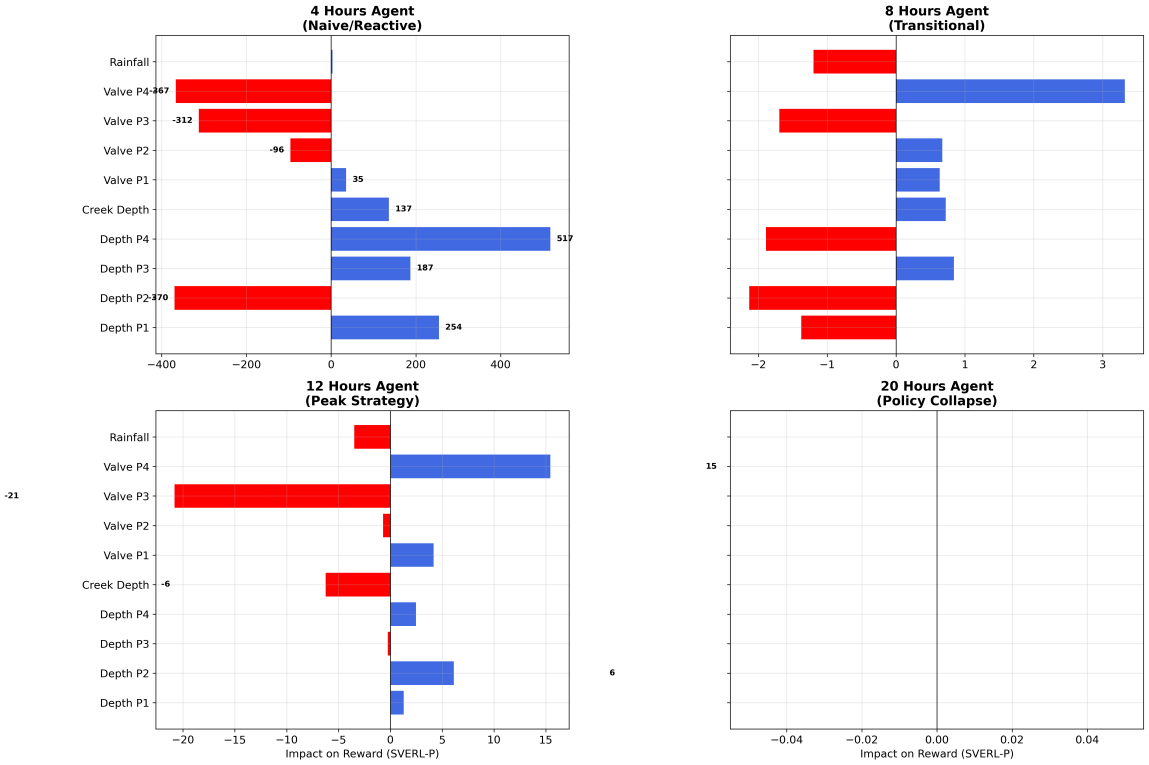


Figure 3.5: Visualizing Policy Collapse. The 12-hour agent (A) actively throttles valves during the storm peak to protect the creek. The 20-hour agent (B) adopts a static policy, locking valves fully open to prioritize flood safety, resulting in zero Shapley attribution across all sensors.

Using SVERL-P as a liveness probe, we diagnosed the mathematical cause of this failure (**Figure 3.5**). For the collapsed 20-hour agent, the attribution values for all sensor inputs dropped to near zero. Because the agent's policy became deterministic and independent of state ($\pi(S) = \pi(S_{masked})$), masking the sensors resulted in identical reward trajectories. This zero-value attribution serves as a formal diagnostic of a "dead" agent that has ceased to utilize environmental information for decision-making.

3.3.3 Structural Analysis: Topological Conflicts at Conner Creek

The failure of the balanced agent to reach stability highlights a fundamental topological conflict within the Conner Creek network. By utilizing SVERL-P as a forensic tool, we identified that Valve P2 serves as a physical bottleneck. During 100-year storm events, the upstream storage requirements and downstream erosion limits become mutually exclusive. SVERL-P attributions revealed that any action taken to mitigate pond overtopping (Objective A) directly resulted in a violation of the downstream shear stress threshold (Objective B).

This result demonstrates that the failure was not a limitation of the DRL algorithm, but a capacity failure of the existing infrastructure. SVERL-P provides the mathematical evidence necessary to distinguish between a learning failure (which would justify more training) and a systemic failure (which justifies infrastructure retrofits such as pond expansion).

3.3.4 Practical Implications

Sensor Optimization and Information Redundancy

A significant finding of the SVERL-P analysis was the near-zero attribution assigned to the Creek Depth logger. While hydrologically intuitive to monitor the downstream

creek, the agent learned to infer the creek's state purely through valve positions and rainfall data. This suggests that the downstream logger is mathematically redundant for this specific control policy, offering a financially actionable insight for utility managers seeking to minimize maintenance costs without degrading control performance.

Forensic Audit: Distinguishing Malpractice from Force Majeure

SVERL-P provides a quantitative framework for liability assessment in autonomous systems. By decomposing the failure modes, we can distinguish between:

- **Force Majeure:** Cases where the Rainfall attribution is extremely negative while Valve attributions are positive, indicating the storm exceeded the system's physical capacity despite optimal control.
- **Control Malpractice:** Cases where Valve attributions are negative (as observed with Valve P2), indicating the agent made suboptimal decisions that actively contributed to the failure.

The Requirement of the Surrogate for Real-Time Explainability

This depth of analysis was enabled solely by the BiLSTM surrogate. Generating the SVERL-P heatmaps required approximately 10,000 counterfactual simulations. On a standard physics-based solver, this analysis would require 14 hours; our surrogate-accelerated engine completed the task in under 5 minutes. This confirms that high-fidelity surrogate modeling is a structural prerequisite for deploying trustworthy, auditable AI in real-world water systems.

3.4 Conclusion

This study presented Surrogate-Accelerated SVERL-P, a framework designed to bridge the gap between black box DRL agents and the rigorous interpretability

requirements of civil engineering. By leveraging a BiLSTM surrogate to achieve a 33× speedup over traditional physics-based solvers, we enabled a game-theoretic analysis of hydrological control policies that was previously computationally prohibitive.

The results demonstrate that our proposed framework serves as a powerful diagnostic instrument throughout the lifecycle of an AI agent. Specifically, our work provides three primary contributions to the field of smart stormwater management:

1. **Causal Validation of Control Logic:** By utilizing a hard-coded Logic Agent proxy, we demonstrated that the SVERL-P engine correctly distinguishes between physical triggers (Pond Depth) and environmental stressors (Rainfall), ensuring that attributions align with hydrological ground truths.

2. **Diagnostic Detection of Policy Collapse:** We established that SVERL-P functions as a liveness probe for DRL training. By identifying moments where sensor attribution drops to zero, our proposed methodology can detect when an agent has ceased to learn from its environment, allowing for the early termination of inefficient training runs.

3. **Forensic Auditing of Structural Limits:** Crucially, our proposed framework functions as a forensic tool to distinguish between algorithmic learning failures and physical capacity failures. In the case of the Conner Creek watershed, the analysis pinpointed a structural topological conflict at Valve P2, where flood prevention and erosion control objectives were found to be mutually exclusive under extreme design storms. This provides engineers with the mathematical evidence needed to justify infrastructure retrofits rather than further reward-tuning.

Ultimately, this methodology provides the necessary transparency to audit DRL failures, optimize sensor networks by identifying redundant loggers, and transition from theoretical simulation to deployment in sensitive urban watersheds.

3.4.1 Limitations and Future Work

While our proposed framework successfully diagnosed the causes of control failure, a fully stable, multi-objective control agent was not achieved for the 100-year design storm. This limitation is attributed to the identified topological conflict within the current watershed infrastructure. However, this outcome highlights the value of the explainability framework; it allowed us to identify why the control problem was physically unsolvable with the existing pond configurations.

Future work should focus on:

- **Constrained Optimization:** Implementing Constrained Markov Decision Processes (CMDPs) to enforce safety limits as hard constraints rather than relying on soft reward weights.
- **Real-Time Monitoring:** Integrating SVERL-P into active control dashboards to provide human operators with real-time justifications for autonomous valve movements.
- **Infrastructure Retrofit Justification:** Utilizing the forensic insights from this study to simulate how downstream storage expansion might resolve the identified topological conflicts.

By transforming DRL from a black box into a trusted, auditable engineering tool, our proposed methodology enables the responsible integration of AI into safety-critical water infrastructure.

Data and Software Availability

The code and datasets for the explainability framework used in this study are available at: https://github.com/isifletcher/Performance-Based-Explainability-for-Deep-Reinforcement-Learning-in-Stormwater-Control

Chapter 4

Eco-Mimicry in Real-Time: Leveraging Deep Reinforcement Learning to Replicate Natural Flow Regimes

Disclosure

Author Contributions and Roles: Isidora Fletcher was the primary investigator and lead author for this study. She was responsible for the conceptual design of the framework, the development of the modeling procedure, the implementation of the RL architecture, and the drafting of the manuscript.
Anahita Khojandi and Jon Hathaway supervised the project. They provided guidance on formulation, domain expertise, and critical review.

Abstract

Urbanization fundamentally alters watershed hydrology, replacing permeable landscapes with impervious surfaces that amplify peak flows and degrade downstream channel stability. While traditional stormwater infrastructure effectively mitigates flood risk, static detention systems lack the dynamic capacity to restore the natural flow variability required for ecological health. This study presents an Eco-Mimicry control framework that utilizes Deep Reinforcement Learning (DRL) to transform passive urban infrastructure into adaptive systems capable of emulating pre-development hydrology. We developed a physics-based Natural Proxy model of the Conner Creek watershed to generate real-time reference hydrographs representing a forested landscape. The control problem was formulated as a reference tracking task, where an Advantage Actor-Critic (A2C) agent modulated outlet valves to minimize the deviation between the engineered urban outflow and the natural target signal. Experimental results quantify the hydrologic divergence of the current system and identify critical physical and computational boundaries. Specifically, we demonstrate that while DRL agents can optimize release timing, the efficacy of eco-mimicry appears bounded by the hydraulic capacity gap between urban detention volume

and natural soil retention. This research validates intelligent control as a scalable software retrofit while establishing requisite physical bounds for its deployment.

4.1 Introduction

Urbanization is recognized globally as a critical driver of aquatic ecosystem degradation (Miltner et al., 2004). The conversion of natural landscapes to cities, marked by increased impervious surfaces and the installation of highly efficient drainage systems, leads to fundamental changes in the flow regimes of downstream rivers and streams (Walsh et al., 2005; Levin et al., 2020). These hydrologic alterations often result in the phenomenon known as the Urban Stream Syndrome, a condition characterized by high flow regime flashiness, increased severity of extreme flow events, and channel degradation (Wu et al., 2015; Walsh et al., 2005; Hawley and Bledsoe, 2011).

The negative consequences of hydrologic alteration are profound for aquatic life. Altered flow conditions increase streambed scouring and sedimentation, which negatively affects fish and macroinvertebrate population assemblages, often favoring non-native species more tolerant of high sediment loads (Coleman II et al., 2011). Furthermore, the increased flashiness of flow regimes tends to favor fish species with generalized feeding strategies over those with specialized life histories or limited mobility (Coleman II et al., 2011). In urban areas, stormwater runoff is a critical source of degradation, and management approaches that focus solely on traditional flood and pollution mitigation are frequently insufficient to protect the ecological structure and function of streams (Walsh et al., 2016).

Conversely, maintaining a natural flow regime is essential for stream health and resilience. Small- to medium-sized stream ecosystems are strongly influenced by the flow regime as a primary driver of their ecological structure (Walsh et al., 2016; Poff et al., 1997). The underlying hypothesis for ecologically successful restoration requires catchment-scale stormwater management to replicate pre-development hydrologic processes (Walsh et al., 2016). The concept of the natural flow regime identifies five

critical components—magnitude, frequency, duration, timing, and rate of change—all of which are vital to stream biota (Poff et al., 1997). By attempting to mimic natural flow, stream ecosystems can be maintained in a state that is both sustainable and resilient (Meyer, 1997).

Recognizing the need to link hydrological management to ecological outcomes, numerous studies have sought to integrate flow-ecology research with watershed planning (Wu et al., 2015). Frameworks such as the Ecological Limits of Hydrologic Alteration (ELOHA) (Poff et al., 2010; Stein et al., 2017) and the Functional Flows Approach (FFA) (Irving et al., 2022; Mazor et al., 2018) have been developed to prioritize areas for flow management. However, a significant challenge remains: translating these theoretical ecological flow targets into dynamic, operational control strategies for existing, active urban stormwater infrastructure (Akin, 2021).

Real-time control (RTC), particularly through Deep Reinforcement Learning (DRL), has emerged as an adaptive method for managing urban drainage systems (UDS) (Tian et al., 2022b, 2024a; Wang et al., 2020, 2022; Tian et al., 2022a). DRL agents learn optimal control policies by maximizing a quantitative reward function. Yet, previous DRL applications have primarily focused on traditional hydraulic goals, such as mitigating combined sewer overflows (CSOs) or capping flows at arbitrary static thresholds (Tian et al., 2022b; Saliba et al., 2020; Bowes et al., 2022). Few studies have attempted to force an urban system to dynamically mimic the complex, non-linear hydrograph of a natural watershed in real-time.

This chapter addresses this gap by proposing and evaluating a novel Eco-Mimicry control framework applied to the Conner Creek watershed. Rather than relying on static safety thresholds, we generate a Natural Proxy model—a modified version of the system parameters representing pre-development conditions (e.g., high infiltration, low imperviousness). The flow output from this natural proxy serves as a dynamic target baseline. By integrating the deviation between the urban flow and the natural proxy flow into the RL reward function, the agent learns to manipulate the detention pond valves to reshape the urban hydrograph.

This chapter presents the comprehensive experimental evaluation of the Eco-Mimicry framework. The study yielded critical insights into three distinct areas: (1) the limitations of data-driven surrogate models in replicating complex, variable-aperture hydraulic control; (2) the physical capacity constraints of urban infrastructure that bound the solution space for any control logic; and (3) the computational trade-offs required to train intelligent agents in hydrologic environments. While the framework successfully established a decoupled architecture to resolve hardware conflicts, the results indicate that achieving perfect eco-mimicry is likely constrained by both the stochastic nature of surrogate training and the hydraulic bottlenecks of the existing gray infrastructure.

4.2 Methodology

This section details the computational and experimental framework used to evaluate the feasibility of Eco-Mimicry. We describe the development of the Natural Proxy for target generation, the formulation of the decision-making process, and the surrogate modeling techniques used to accelerate agent training. The methodology is designed to isolate the agent's performance from computational bottlenecks, allowing for a focused assessment of hydraulic control capabilities.

4.2.1 Overview of the Eco-Mimicry Control Framework

This study proposes a shift from traditional flood-minimization control to an Eco-Mimicry reference tracking approach. The objective is to utilize DRL to actuate urban detention infrastructure such that the downstream hydrograph replicates the behavior of a pre-development (natural) watershed. The framework integrates three core components: (1) a Natural Proxy Model to generate the target signal, (2) a BiLSTM Surrogate Model to accelerate training, and (3) an Actor-Critic agent to perform real-time control.

4.2.2 Developing the Natural Proxy Model

To generate the baseline for ecological mimicry, we modified the calibrated SWMM model of Conner Creek to approximate pre-development conditions. This Natural Proxy relies on parameter adjustments to simulate a forested watershed response:

- **Imperviousness:** Reduced to 5% across all subcatchments to simulate natural ground cover.
- **Roughness (Manning's n):** Increased to 0.40 for pervious areas (representing dense woods) and 0.24 for channel routing.
- **Infiltration:** Green-Ampt parameters were adjusted to reflect uncompacted soil, increasing hydraulic conductivity.
- **Topology:** Detention ponds were functionally removed by converting storage nodes to junctions. Engineered orifices were replaced with open trapezoidal channels, allowing water to flow unrestricted by artificial impoundments.

To enable a direct comparison for the reward function, flow rates must be monitored at physically equivalent locations in both models. **Figure 4.1** illustrates the spatial layout of the urban watershed and highlights the four detention pond locations. In the Urban Model, Q_{urban} is measured at the conduit immediately downstream of the detention facility; this ensures that both controlled valve discharge and uncontrolled weir overflow are captured. In the Natural Proxy, $Q_{natural}$ is measured at the corresponding open channel segment that replaced the pond.

4.2.3 High-Resolution Hydrologic Data Generation

To ensure the RL agent is robust against diverse storm dynamics, we developed a stochastic rainfall generator capable of synthesizing high-resolution hyetographs from coarse historical data or design standards (NRCS Type II, Huff Quartiles).

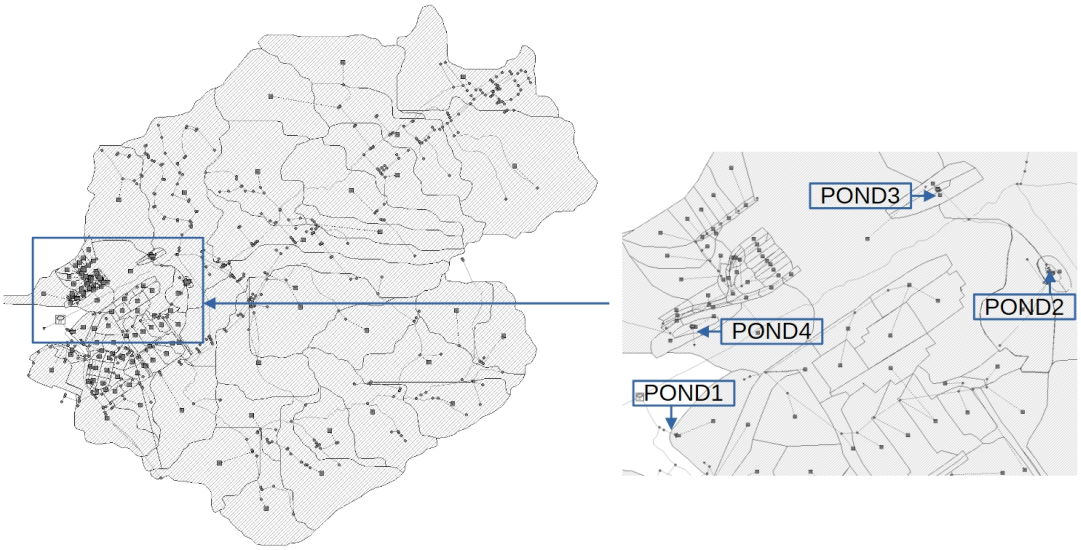


Figure 4.1: Conner Creek SWMM model structure. **Left:** The full urban watershed network. **Right:** Zoomed-in view of the four detention ponds (Ponds 1–4). For the Natural Proxy, these storage units are converted to junctions, and the outlet links are replaced with natural trapezoidal channels to generate the reference flow target $Q_{natural}$.

The generator utilizes Cubic Spline Interpolation to convert discrete cumulative rainfall depth-duration-frequency (DDF) points into continuous cumulative mass curves. This allows for the temporal downscaling of aggregate data into arbitrary time steps (e.g., 15 minutes). The process ensures that the first derivative (rainfall intensity) remains smooth and continuous, avoiding unrealistic abrupt changes common in linear interpolation. The procedure is detailed in **Algorithm 4.1**.

Algorithm 4.1 Stochastic Rainfall Generation via Cubic Spline Interpolation

Require: Discrete Data (T_{known}, Y_{known}), Total Depth D_{tot}, Duration H, Step Δt

Initialize Spline $S(\cdot) \leftarrow \text{FitCubicSpline}(T_{known}, Y_{known})$
$T_{target} \leftarrow [0, \Delta t, 2\Delta t, \ldots, H]$
$D_{cum} \leftarrow [0]$
for each $t \in T_{target}$ **do**
    $y_{raw} \leftarrow S(t)$
    $y_{mono} \leftarrow \max(y_{raw}, \text{Last}(D_{cum})/D_{tot})$ ▷ Enforce monotonicity
    Append $(y_{mono} \times D_{tot})$ to D_{cum}
end for
$R_{series} \leftarrow \text{Diff}(D_{cum})$ ▷ Calculate incremental depth
$R_{final} \leftarrow \text{Pad}(R_{series}, \text{DryDays} = 4)$
return R_{final}

4.2.4 Surrogate Modeling for Computational Efficiency

Direct interaction with the SWMM hydraulic engine is computationally prohibitive for the millions of time steps required for DRL training. To address this, we train a Bidirectional Long Short-Term Memory (BiLSTM) deep neural network to serve as a differentiable surrogate of the urban watershed.

- **Input:** A sequence of past states (rainfall, depths, valve settings).
- **Output:** The predicted hydraulic state (d_{t+1}) for all ponds and critical conduits.
- **Training Data:** The surrogate is trained on synthetic and historical events generated by the stochastic hydrology engine using 15-minute time steps.

This surrogate allows the RL agent to explore the action space orders of magnitude faster than real-time simulation, with the final validation performed on the full SWMM physics engine.

4.2.5 Markov Decision Process (MDP) Formulation

The control problem is formulated as a tuple (S, A, R, γ).

State Space (S_t)

The state vector consists of 13 features:

$$S_t = \{\mathbf{d}_{ponds}, \mathbf{p}_{valves}, i_{rain}, \mathbf{Q}_{natural}\} \tag{4.1}$$

Crucially, this includes $\mathbf{Q}_{natural}$, the target flow rates derived from the Natural Proxy, explicitly informing the agent of the tracking objective.

Action Space (A_t)

The action space is Multi-Discrete, allowing each of the four valves to be set to one of 11 positions $(0.0, 0.1, ..., 1.0)$. This granularity enables the fine flow modulation required for hydrograph tracking.

Eco-Mimicry Reward Function (R_t)

The objective is to minimize tracking error subject to safety constraints:

$$R_t = -w_1 \sum_{i=1}^{4} \left| Q^{(i)}_{urban}(t) - Q^{(i)}_{natural}(t) \right| - w_2 P_{flood} - w_3 P_{stability} \tag{4.2}$$

where P_{flood} penalizes overflows and $P_{stability}$ penalizes valve oscillation.

4.2.6 Reinforcement Learning Framework

We utilize the Advantage Actor-Critic (A2C) algorithm. The framework utilizes two neural networks: an actor (π_θ) that outputs the probability distribution of valve settings, and a critic (V_ϕ) that estimates the value of the current state. The system architecture, illustrating the flow of the reference signal $Q_{natural}$ into both the reward function and the agent's state, is visualized in **Figure 4.2**.

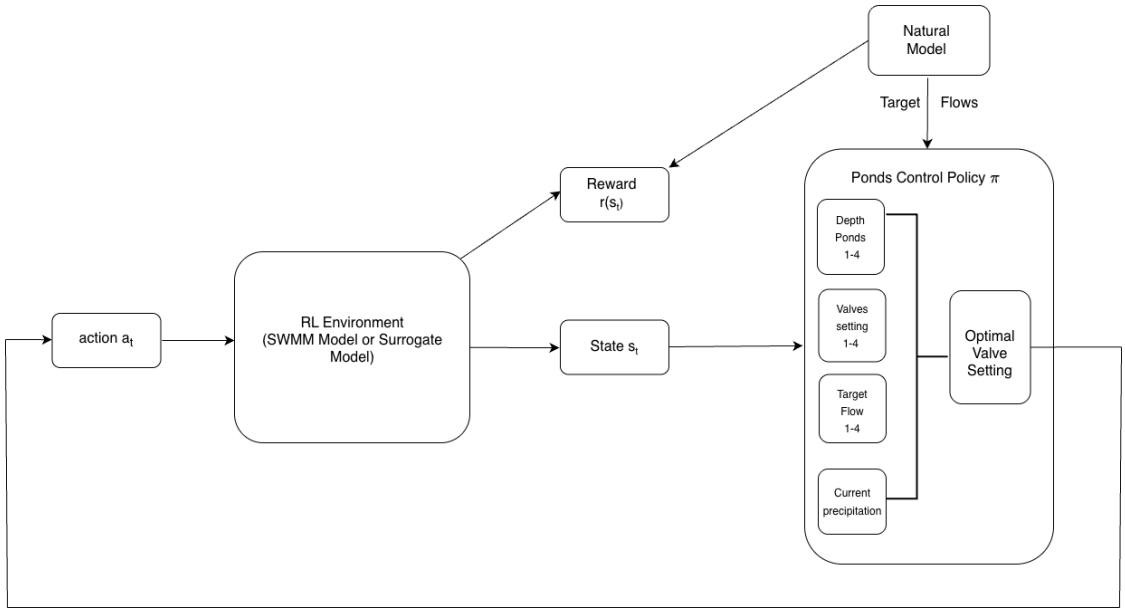


Figure 4.2: Schematic of the Eco-Mimicry Reinforcement Learning Architecture. The Natural Proxy Model generates the target reference flow (Q_{nat}), which is fed into both the Reward calculation (for error minimization) and the Ponds Control Policy π (as a state input). This allows the agent to actively track the pre-development hydrograph.

The training procedure is detailed in **Algorithm 4.2**.

4.3 Results and Discussion

This section presents the validation of the surrogate model and the subsequent performance of the control agent under storm conditions. We first analyze the hydrologic baseline to establish the tracking difficulty, followed by an assessment of

the surrogate's accuracy in capturing variable-valve hydraulics. Finally, we evaluate the agent's ability to minimize tracking error and discuss the physical constraints identified during the experiment.

4.3.1 Baseline Hydrologic Characterization

Prior to deploying the Reinforcement Learning agent, the hydrologic divergence between the current urban system and the pre-development landscape was characterized to establish the control objective. **Figure 4.3** presents the comparative hydrographs for December 2015 at the four detention pond locations.

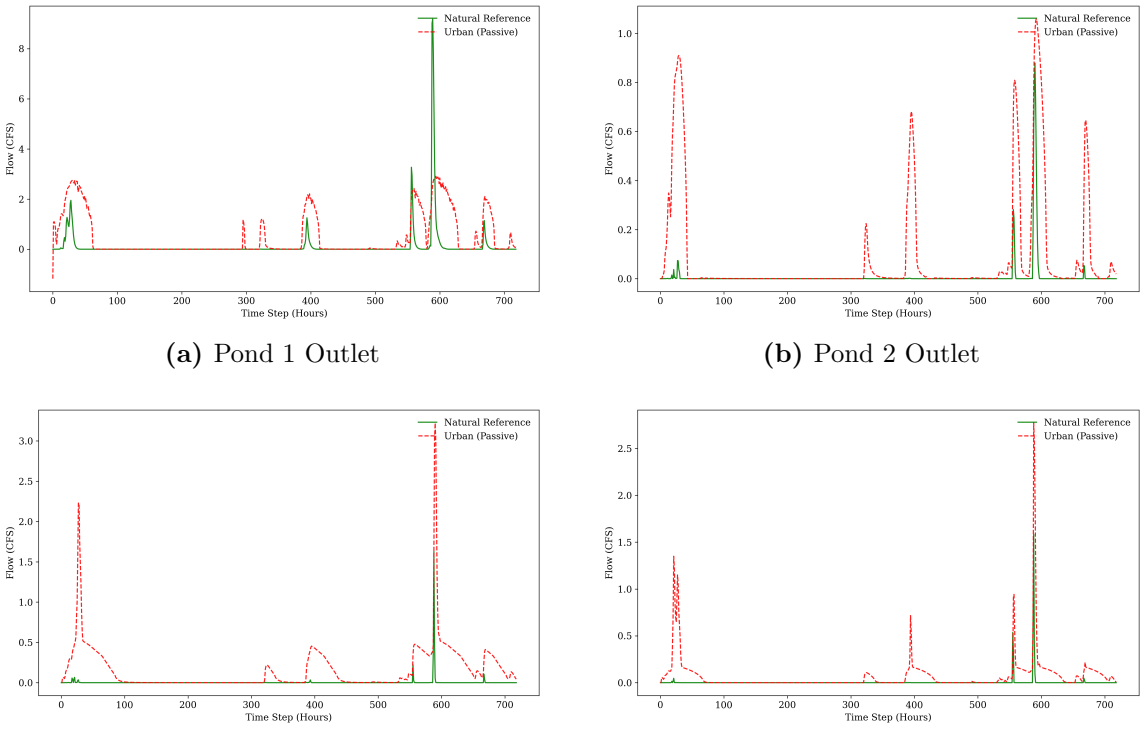
The results reveal a stark contrast in hydrologic response:

- **Urban Flashiness (Red Dashed Line):** The passive urban system (valves 100% open) exhibits rapid, high-magnitude peak flows immediately following rainfall events. For example, at Pond 1, urban peaks reach approximately 9 CFS, and at Pond 3, they exceed 2 CFS.
- **Natural Attenuation (Green Solid Line):** The Natural Proxy model demonstrates extreme attenuation. In Ponds 2, 3, and 4, the natural discharge remains near zero for the majority of the simulation, indicating that the pre-development landscape would have infiltrated nearly all rainfall from these specific winter storm events.

This comparison defines the difficulty of the control task: to mimic the Green line, the RL agent must learn to keep valves predominantly closed to utilize the full storage capacity of the ponds, releasing water only at a trickle to match the slow interflow of the natural watershed.

4.3.2 Surrogate Model Performance and Limitations

To accelerate the training of the RL agent, a Bi-Directional LSTM (BiLSTM) was developed to approximate the physics of the SWMM engine. Unlike previous



(a) Pond 1 Outlet

(b) Pond 2 Outlet

(c) Pond 3 Outlet

(d) Pond 4 Outlet

Figure 4.3: Baseline hydrologic comparison for December 2015. The **Red Dashed Line** indicates the engineered urban system with fully open valves, showing sharp peaks. The green solid line indicates the Natural Proxy target ($Q_{natural}$). The significant gap between the two demonstrates the loss of natural retention and the necessity for active control.

studies which often focus on binary control elements (pumps on/off), this study modeled multi-discrete variable orifice openings (action space of 11 discrete levels: $0.0, 0.1, \ldots, 1.0$). This significantly increased the complexity of the mapping function the neural network had to learn, as flow through an orifice is non-linear with respect to both depth and the discrete opening ratio.

The surrogate model performance was evaluated using the Nash-Sutcliffe Efficiency (NSE) given by:

$$NSE = 1 - \frac{\sum(Q_{nat}^t - Q_{rl}^t)^2}{\sum(Q_{nat}^t - \bar{Q}_{nat})^2} \tag{4.3}$$

The results of the surrogate model are presented in Table 4.1.

Table 4.1: Performance metrics of the BiLSTM Surrogate Model.

Variable	NSE Range	Interpretation
System Outflow (Q_{out})	$0.95 - 0.99$	High Fidelity (Flow Routing)
Pond Depth (d)	0.70 to -46.0	Instability in Storage Prediction

While flow routing achieved high accuracy ($NSE > 0.95$), the prediction of pond depth exhibited significant instability. This failure is attributed to the zero-inflated data sparsity inherent in the training set. The historical data is dominated by dry periods where pond depth is zero. Consequently, the BiLSTM struggled to learn the precise onset of accumulation, often predicting zero depth even as inflow began, leading to cascading volume balance errors. This result highlights a critical limitation of purely data-driven surrogates in handling the complex state transitions of multi-level valve control.

4.3.3 RL Agent Training and Convergence

The Advantage Actor-Critic (A2C) agent was initialized and trained on the direct SWMM environment for a preliminary session. **Figure 4.4** illustrates the total reward trajectory during this phase.

Algorithm 4.2 Eco-Mimicry Reference Tracking (A2C)

Require: Actor π_θ, Critic V_ϕ, Environment $\mathscr{E}$, Natural Data $\mathbf{Q}_{nat}$

for episode $m \leftarrow 1$ to M **do**
  $\mathscr{E}$.reset()
  $s_0 \leftarrow \text{GetState}(\mathscr{E}, \mathbf{Q}_{nat}[0])$
  while $t < T_{max}$ **and not** Done **do**
    $a_t \sim \pi_\theta(s_t)$ ▷ Sample valve settings
    $v_t \leftarrow V_\phi(s_t)$ ▷ Estimate state value
    $Q_{urb}, \mathbf{d}_{ponds} \leftarrow \mathscr{E}.\text{sim}(a_t)$ ▷ Run physics engine
    $Q_{ref} \leftarrow \mathbf{Q}_{nat}[t]$ ▷ Lookup target flow
    $r_t \leftarrow -w_1|Q_{urb} - Q_{ref}| - w_2 P_{flood}(\mathbf{d}_{ponds})$
    Store (s_t, a_t, r_t, v_t)
    $s_{t+1} \leftarrow \text{GetState}(\mathscr{E}, \mathbf{Q}_{nat}[t+1])$
    $t \leftarrow t + 1$
  end while
  $G_t \leftarrow \sum \gamma^k r_{t+k}$ ▷ Calculate Returns
  $A_t \leftarrow G_t - v_t$ ▷ Calculate Advantage
  Update $\theta \leftarrow \nabla_\theta \log \pi_\theta(a_t|s_t) A_t$
  Update $\phi \leftarrow \nabla_\phi (G_t - V_\phi(s_t))^2$
end for

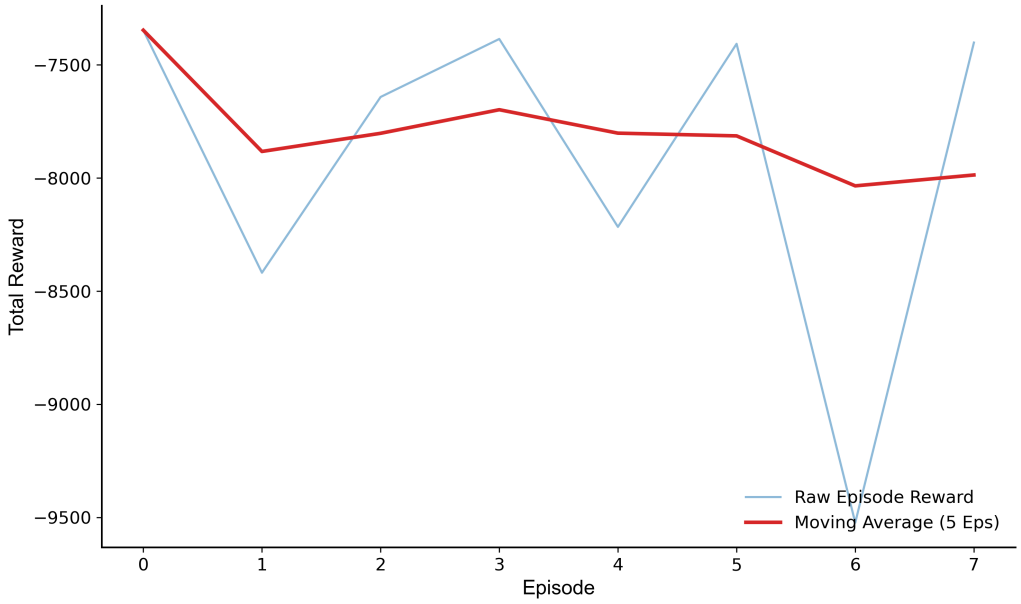


Figure 4.4: Early training trajectory of the Eco-Mimicry Agent. The blue line represents the raw episodic reward, and the red line represents the moving average. The volatility indicates the agent is in the exploration phase, learning the causal relationship between valve actions, storage depth, and tracking error.

As expected, the agent exhibited high volatility as it explored the Multi-Discrete action space. The sharp drops in reward correspond to the discovery of high-penalty states—likely keeping valves closed too long during a peak, triggering the heavy P_{flood} penalty. This negative feedback was essential for the Critic network to learn the safety boundaries of the system.

4.3.4 Hydraulic Bottlenecks and the Flatline Phenomenon

To evaluate the limits of the control framework, we compared the behavior of the system under three conditions during a 10-year design storm: (1) the **Natural Target** (pre-development baseline), (2) a **Random Agent** (representing an untrained or exploratory policy), and (3) a **Static Baseline** (valves fixed at 50% open).

Figure 4.5 presents a critical divergence between the desired and realized flows.

- **The Natural Target (Black Dashed):** Exhibits an unconstrained peak discharge of approximately 50 cfs, reflecting a wide, natural stream channel.
- **The Urban Response (Red/Green):** Both the Random Agent and the Static 50% Baseline are constrained to a peak of approximately $1 - 2$ cfs.

This result demonstrates a Hydraulic Bottleneck. The downstream urban conduit (stream) has a fixed diameter that limits conveyance. The inability of either the random agent or the static baseline to approach the natural flow peak suggests that this is not primarily a failure of learning, but a failure of capacity. Even if the RL agent were to act perfectly (opening valves to 100%), it appears physically impossible to match the peak magnitude of the natural watershed. This finding indicates that Eco-Mimicry in high-intensity storms is not solely a control problem, but an infrastructure sizing problem. Control logic cannot override the laws of fluid mechanics; it can only optimize within the feasible hydraulic envelope.

4.4 Conclusion

This study successfully established a computational framework for Eco-Mimicry but identified significant physical and methodological boundaries. By developing a Natural Proxy model and a decoupled training architecture, we resolved the primary computational barriers to high-resolution stormwater control. However, the experimental results demonstrate a fundamental dichotomy between algorithmic intelligence and physical capacity.

First, the evaluation of the surrogate model revealed the limitations of model-free RL for variable-aperture hydraulics. The stochastic noise introduced by continuous valve settings makes purely data-driven prediction difficult, suggesting that future research might benefit from leverage high-performance computing (HPC) to train agents directly on physics-based engines.

Second, and most critically, the experiment identified a Hydraulic Capacity Gap. Comparison against random and static baselines demonstrated that the system was physically unable to match the natural hydrograph during peak events due to the conveyance limits of the existing pipe network. This suggests that while AI can identify the maximum safe operating capacity of a system, it cannot overcome physical infrastructure constraints.

Ultimately, this research concludes that smart control is a necessary but insufficient condition for full ecological restoration. Effective Eco-Mimicry likely requires a hybrid approach, where intelligent algorithms optimize the timing of release, but are coupled with Green Infrastructure retrofits to expand the physical retention capacity of the watershed.

Data and Software Availability

The code for the Eco-Mimicry framework and the Natural Proxy model generation is available at: https://github.com/isifletcher/Eco-Mimicry-in-Real-Time-Leveraging-Deep-Reinforcement-Learning-to-Replicate-Natural-Flow-Regimes

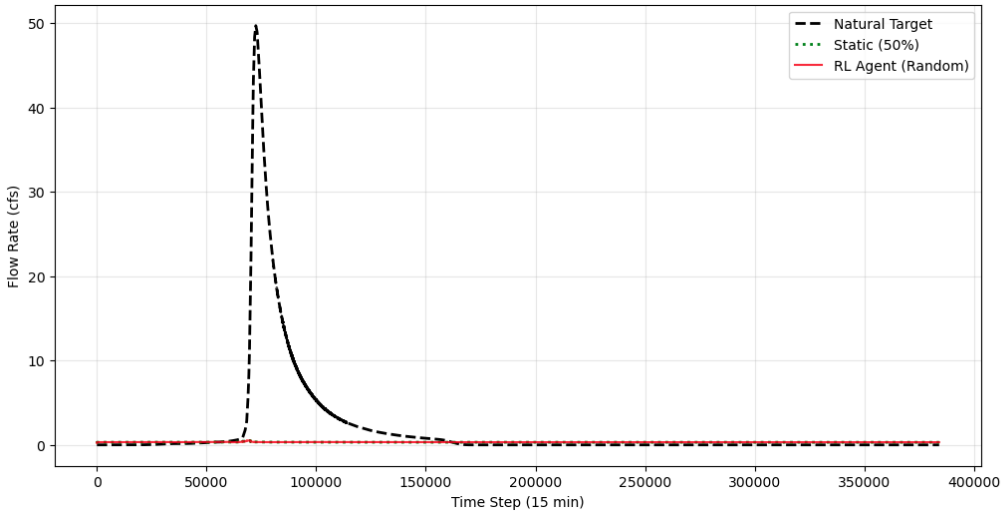


Figure 4.5: Evaluation of the Eco-Mimicry Agent under 10-year storm conditions. The massive gap between the Natural Target (peak $\approx$ 50 cfs) and the urban scenarios (Random Agent and 50% Static Baseline, both peaking $\approx$ 2 cfs) illustrates the hydraulic capacity limit of the infrastructure. Regardless of the control strategy, the existing pipe network cannot convey the natural flow magnitude.

Conclusions

5.1 Summary of Contributions

This dissertation has addressed fundamental barriers to the adoption of Deep Reinforcement Learning (DRL) in critical urban water infrastructure. By integrating high-fidelity deep learning emulators with game-theoretic interpretability frameworks, this research establishes a pathway for transitioning DRL from theoretical simulation to practical, trustworthy deployment.

The specific contributions of this work are summarized below:

5.1.1 Surrogate Modeling for Accelerated Training

As presented in Chapter 2, the computational cost of physics-based hydraulic solvers (SWMM) renders them impractical for the iterative training required by DRL algorithms. This work developed and validated a Bidirectional Long Short-Term Memory (BiLSTM) surrogate model that emulates the hydraulic dynamics of the Conner Creek watershed with high fidelity (NSE > 0.96).

- The surrogate model achieved a 33-fold acceleration in simulation speed compared to the native SWMM engine.
- This acceleration enabled an Advantage Actor-Critic (A2C) agent to complete over 1,000 training episodes and converge to a competent policy within a 5-hour

computational budget, whereas an agent trained directly on SWMM failed to learn due to data starvation.

5.1.2 Performance-Based Explainability and Trust

Chapter 3 addressed the "Black Box" problem, a primary impediment to the regulatory approval of AI in civil engineering. By adapting Shapley Values to the reinforcement learning domain (Surrogate-Accelerated SVERL), this research moved beyond explaining policy (what the agent did) to explaining performance (how the input contributed to physical goals).

- The framework successfully quantified the "Value of Information," identifying that downstream creek sensors were mathematically redundant for flood control in the studied topology.
- It served as a forensic tool, diagnosing "Policy Collapse" by detecting when the agent ceased to utilize sensor inputs, allowing for the early termination of failed training runs.
- It identified structural limitations in the watershed (Topological Conflicts), proving that certain multi-objective failures were due to physical infrastructure constraints rather than algorithmic deficiencies.

5.1.3 Eco-Mimicry and Ecological Restoration

Chapter 4 introduced a paradigm shift from purely hydraulic control (flood prevention) to ecological control. By proposing the Eco-Mimicry framework, this research established a methodology for using existing detention ponds to actively restore natural flow regimes and evaluated its limits.

- We defined a procedure for creating a Natural Proxy baseline using parameter adjustment (e.g., infiltration and roughness), providing a scalable method for generating ecological targets.

- We identified the limits of data-driven surrogates for variable-aperture control, demonstrating that epistemic errors at low depths can destabilize fine-grained flow modulation.

- Most critically, we identified the Hydraulic Capacity Gap. Experimental results showed that neither intelligent agents nor static baselines could match the natural peak flow during high-intensity storms. This finding quantitatively demonstrates that software retrofits must be coupled with physical capacity expansion (Green Infrastructure) to achieve true ecological restoration.

5.2 Future Research Directions

While this dissertation provides a robust framework for training and explaining autonomous agents, several avenues remain for future investigation to fully realize smart water systems.

5.2.1 Integration of Quantitative Precipitation Forecasts (QPF)

The current Eco-Mimicry agent operates primarily on reactive data (current depths and rainfall). However, our results indicate that the current storage capacity is insufficient to match natural infiltration during storm peaks. Future work will integrate long-horizon weather forecasts into the state space. This will allow the agent to pre-release water (drawdown) hours before a storm begins, effectively creating artificial storage capacity to better approximate the retention capabilities of a pre-development watershed.

5.2.2 From Soft Rewards to Hard Constraints

The current A2C framework utilizes soft reward penalties to discourage unsafe actions (e.g., downstream erosion or flooding). However, in safety-critical infrastructure, violations of physical limits cannot always be tolerated. Future investigations

should apply Constrained Markov Decision Processes (CMDPs) or Lagrangian relaxation techniques to enforce hard safety constraints, ensuring that the agent is mathematically forbidden from violating critical flood thresholds while optimizing for ecological mimicry.

5.2.3 Sim-to-Real Transfer and Physical Deployment

Validating the surrogate-trained policies on physical hardware remains the ultimate goal. Future research must address the Sim-to-Real gap, specifically analyzing how agents handle sensor noise, actuator delays, and hydraulic discrepancies that were not captured in the simulation environment. A pilot deployment using low-cost IoT controllers on the Conner Creek valves would provide essential validation data.

Bibliography

Aderemi, A. et al. (2025). Beyond the black box: A systematic review of explainable ai for transparent and trustworthy water quality monitoring. *IEEE Sensors Reviews*. In Press. 38

Akin, A. A. (2021). *Impact and Application of Real-Time Control on Stormwater Systems*. PhD thesis, University of Tennessee. PhD diss. 1, 7, 9, 14, 15, 37, 60

ASCE Task Committee (2023). Adoption of artificial intelligence in drinking water operations: A survey of progress in the united states. *Journal of Water Resources Planning and Management*. 38

Baumann, H., Ravn, N. H., and Schaum, A. (2022a). Efficient hydrodynamic modelling of urban stormwater systems for real-time applications. *Modelling*, 3(4):464–480. 8

Baumann, H., Ravn, N. H., and Schaum, A. (2022b). Efficient hydrodynamic modelling of urban stormwater systems for real-time applications. *Modelling*, 3(4):464–480. 37

Beechey, D., Smith, T. M. S., and Şimşek, (2023). Explaining reinforcement learning with shapley values. 39, 45

Behzadian, K., Kapelan, Z., Savic, D., and Ardeshir, A. (2009). Stochastic sampling design using a multi-objective genetic algorithm and adaptive neural networks. *Environmental Modelling & Software*, 24(4):530–541. 8

Bowes, B., Tavakoli, A., Wang, C., Heydarian, A., Behl, M., Beling, P., and Goodall, J. (2020). Flood mitigation in coastal urban catchments using real-time stormwater infrastructure control and reinforcement learning. *Journal of Hydroinformatics*, 23. 9, 37, 39

Bowes, B. D., Wang, C., Ercan, M. B., Culver, T. B., Beling, P. A., and Goodall, J. L. (2022). Reinforcement learning-based real-time control of coastal urban stormwater systems to mitigate flooding and improve water quality. *Environmental Science: Water Research & Technology*, 8:2065–2086. 1, 7, 37, 60

Coleman II, J. C., Miller, M. C., and Mink, F. L. (2011). Hydrologic disturbance reduces biological integrity in urban streams. *Environmental Monitoring and Assessment*, 172:663–687. 59

Fan, M., Zhang, L., Liu, S., and Lu, D. (2023). Investigation of hydrometeorological influences on reservoir releases using explainable machine learning methods. *Frontiers in Water*, 5:1112970. 38

Fletcher, I., Hathaway, J., and Khojandi, A. (2026). Accelerating decision-making for smart stormwater management through swmm surrogate modeling for reinforcement learning. *ResearchGate Preprint*. Preprint. 38, 39, 42, 43

Fuente, N. D. L. and Guerra, D. A. V. (2024). A comparative study of deep reinforcement learning models: Dqn vs ppo vs a2c. 9, 20, 38

Ge, Y., Lu, W., and Pan, Z. (2023). Groundwater contamination source identification based on Sobol sequences–based sparrow search algorithm with a BiLSTM surrogate model. *Environmental Science and Pollution Research*, 30(18):53191–53203. 8

Ghasemlounia, R., Gharehbaghi, A., Ahmadi, F., and Saadatnejadgharahassanlou, H. (2021). Developing a novel framework for forecasting groundwater level fluctuations

using bi-directional long short-term memory (BiLSTM) deep neural network. *Computers and Electronics in Agriculture*, 191:106568. 8

Girihagama, L., Naveed Khaliq, M., Lamontagne, P., Perdikaris, J., Roy, R., Sushama, L., and Elshorbagy, A. (2022). Streamflow modelling and forecasting for canadian watersheds using LSTM networks with attention mechanism. *Neural Computing and Applications*, 34(22):19995–20015. 8

Green, W. H. and Ampt, G. A. (1911). Studies on soil phyics. *The Journal of Agricultural Science*, 4(1):1–24. 15

Hawley, R. J. and Bledsoe, B. P. (2011). How do flow peaks and durations change in suburbanizing semi-arid watersheds? a southern california case study. *Journal of Hydrology*, 405(1-2):69–82. 59

He, Z., Tian, W., Wang, J., Yan, H., Xin, K., and Tao, T. (2025). Deep reinforcement learning control as an innovative approach for urban drainage systems: review and prospects. *Water Research*, 284:123954. 8, 9, 37, 38

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780. 16

Huang, S., Xia, J., Wang, Y., Lei, J., and Wang, G. (2024). Water quality prediction based on sparse dataset using enhanced machine learning. *Environ. Sci. Ecotechnol.*, 20(100402):100402. 18

Huff, J., Watts, J., Khojandi, A., and Hathaway, J. (2023). Deep temporal neural networks for water level predictions of watershed systems. In *2023 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE. 8

Husic, A., Hammond, J., Price, A. N., and Roundy, J. K. (2025). Interrogating process deficiencies in large-scale hydrologic models with interpretable machine learning. *Hydrology and Earth System Sciences*, 29:4457–4472. 38

Irving, K., Taniguchi-Quan, K. T., Aprahamian, A., Rivers, C., Sharp, G., Mazor, R. D., Theroux, S., Holt, A., Peek, R., and Stein, E. D. (2022). Application of flow-ecology analysis to inform prioritization for stream restoration and management actions. *Frontiers in Environmental Science*, Volume 9 - 2021. 60

Keating, E. H., Doherty, J., Vrugt, J. A., and Kang, Q. (2010). Optimization and uncertainty assessment of strongly nonlinear groundwater models with high parameter dimensionality. *Water Resources Research*, 46(10). 8

Kerkez, B., Gruden, C., Lewis, M., Montestruque, L., Quigley, M., Wong, B., Bedig, A., Kertesz, R., Braun, T., Cadwalader, O., Poresky, A., and Pak, C. (2016). Smarter stormwater systems. *Environmental Science & Technology*, 50(14):7267–7273. PMID: 27227574. 1, 7, 37, 38

Langtangen, H. P. (2004). *Python Scripting for Computational Science*. Springer, Berlin, Heidelberg, 1st edition. 16

Le, X.-H., Ho, H.-V., Lee, G., and Jung, S.-W. (2019). Application of long short-term memory (LSTM) neural network for flood forecasting. *Water*, 11(7):1387. 8

Levin, P. S., Howe, E. R., and Robertson, J. C. (2020). Impacts of stormwater on coastal ecosystems: the need to match the scales of management objectives and solutions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1814):20190460. 59

Majeske, N., Zhang, X., Sabaj, M., Gong, L., Zhu, C., and Azad, A. (2022). Inductive predictions of hydrologic events using a long short-term memory network and the soil and water assessment tool. *Environmental Modelling Software*, 152:105400. 8

Mazor, R. D., Stein, E. D., Ode, P. R., and Schiff, K. (2018). Tools for managing hydrologic alteration on a regional scale: Setting targets to protect stream health. *Freshwater Biology*, 63(8):786–803. 60

McDonnell, B. E., Ratliff, K., Tryby, M. E., Wu, J. J. X., and Mullapudi, A. (2020). PySWMM: The python interface to stormwater management model (SWMM). *Journal of Open Source Software*, 5(52):2292. 11, 15

Meyer, J. L. (1997). Stream health: incorporating the human dimension to advance stream ecology. *Journal of the North American Benthological Society*, 16(2):439–447. 60

Miltner, R. J., White, D., and Yoder, C. (2004). The biotic integrity of streams in urban and suburbanizing landscapes. *Landscape and Urban Planning*, 69(1):87–100. 59

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. 43

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533. 37

Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D., and Veith, T. L. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, 50(3):885–900. 15, 37

Mullapudi, A. and Kerkez, B. (2018). Autonomous control of urban storm water networks using reinforcement learning. In Loggia, G. L., Freni, G., Puleo, V., and Marchis, M. D., editors, *HIC 2018. 13th International Conference on Hydroinformatics*, volume 3 of *EPiC Series in Engineering*, pages 1465–1469. EasyChair. 7, 8, 9

Mullapudi, A., Lewis, M. J., Gruden, C. L., and Kerkez, B. (2020). Deep reinforcement learning for the real time control of stormwater systems. *Advances in Water Resources*, 140:103600. 1, 7, 8, 9, 37, 38

Poff, N. L., Allan, J. D., Bain, M. B., Karr, J. R., Prestegaard, K. L., Richter, B. D., Sparks, R. E., and Stromberg, J. C. (1997). The natural flow regime. *BioScience*, 47(11):769–784. 59, 60

Poff, N. L., Richter, B. D., Arthington, A. H., Bunn, S. E., Naiman, R. J., Kendy, E., Acreman, M., Apse, C., Bledsoe, B. P., Freeman, M. C., et al. (2010). The ecological limits of hydrologic alteration (eloha): a new framework for developing regional environmental flow standards. *Freshwater Biology*, 55(1):147–170. 60

Razavi, S., Tolson, B. A., and Burn, D. H. (2012). Review of surrogate modeling in water resources. *Water Resources Research*, 48(7). 8, 39

Rossman, L. A. (2015). Storm Water Management Model User's Manual Version 5.1. Technical report, U.S. Environmental Protection Agency, Office of Research and Development, Washington, DC, USA. 15, 41

Sadler, J. M., Goodall, J. L., Behl, M., Bowes, B. D., and Morsy, M. M. (2020). Exploring real-time control of stormwater systems for mitigating flood risk due to sea level rise. *Journal of Hydrology*, 583:124571. 1, 37

Saliba, S. M., Bowes, B. D., Adams, S., Beling, P. A., and Goodall, J. L. (2020). Deep reinforcement learning with uncertain data for real-time stormwater system control and flood mitigation. *Water*, 12(11):3222. 2, 7, 8, 9, 37, 38, 60

Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2019). The performance of LSTM and BiLSTM in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3285–3292. 8, 16

Skaggs, R. W., Youssef, M. A., and Chescheir, G. M. (2012). DRAINMOD: Model use, calibration, and validation. *Transactions of the ASABE*, 55(4):1509–1522. 15, 37

Stein, E. D., Sengupta, A., Mazor, R. D., McCune, K., Konrad, C. P., and Bledsoe, B. P. (2017). Application of regional flow–ecology relationships to inform watershed management decisions: Application of the eloha framework in the san diego river watershed, california, usa. *Ecohydrology*, 10(7):e1869. 60

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. The MIT Press, second edition. x, 8, 18, 37

Tian, W., Liao, Z., Zhang, Z., Wu, H., and Xin, K. (2022a). Flooding and overflow mitigation using deep reinforcement learning based on Koopman operator of urban drainage systems. *Water Resources Research*, 58(7):e2021WR030939. 2, 7, 8, 9, 37, 39, 60

Tian, W., Liao, Z., Zhi, G., Zhang, Z., and Wang, X. (2022b). Combined sewer overflow and flooding mitigation through a reliable real-time control based on multi-reinforcement learning and model predictive control. *Water Resources Research*, 58(7):e2021WR030703. 9, 20, 60

Tian, W., Zhang, Z., Xin, K., Liao, Z., and Li, F. (2024a). Improving the interpretability of deep reinforcement learning in urban drainage system operation. *Water Research*, 249:120912. 2, 8, 9, 60

Tian, W., Zhang, Z., Xin, K., Liao, Z., and Li, F. (2024b). Improving the interpretability of deep reinforcement learning in urban drainage system operation. *Water Research*, 249:120912. 38

Vermont Department of Environmental Conservation (n.d.). Appendix o-4: Shear stress. Assessment protocol appendix, Vermont Agency of Natural Resources. Available at https://dec.vermont.gov/sites/dec/files/wsm/rivers/doc

s/assessment-protocol-appendices/0-Appendix-0-04-Shear-Stress.pdf. 14

Walsh, C. J., Booth, D. B., Burns, M. J., Fletcher, T. D., Hale, R. L., Hoang, L. N., Livingston, G., Rippy, M. A., Roy, A. H., Scoggins, M., and Wallace, A. (2016). Principles for urban stormwater management to protect stream ecosystems. *Freshwater Science*, 35(1):398–411. 59

Walsh, C. J., Roy, A. H., Feminella, J. W., Cottingham, P. D., Groffman, P. M., and Morgan, R. P. (2005). The urban stream syndrome: current knowledge and the search for a cure. *Journal of the North American Benthological Society*, 24(3):706–723. 59

Wang, C., Bowes, B., Tavakoli, A., Adams, S., Goodall, J., and Beling, P. (2020). Smart stormwater control systems: A reinforcement learning approach. In *Proceedings of the 2020 International Conference on Artificial Intelligence*. CSREA Press. 2, 37, 60

Wang, C., Bowes, B. D., Beling, P. A., and Goodall, J. L. (2021). Reinforcement learning for flooding mitigation in complex stormwater systems during large storms. In *IEEE EUROCON 2021 - 19th International Conference on Smart Technologies*. IEEE. 7, 9

Wang, Y., Zhang, C., Yu, T., and Ma, M. (2022). Recursive least squares advantage actor-critic algorithms. 9, 20, 60

Wong, B. P. and Kerkez, B. (2018). Real-time control of urban headwater catchments through linear feedback: Performance, analysis, and site selection. *Water Resources Research*, 54(10):7309–7330. 7

Wood-Ponce, R., Diab, G., Liu, Z., Blanchette, R., Hathaway, J., and Khojandi, A. (2024). Developing data-driven learning models to predict urban stormwater runoff volume. *Urban Water Journal*, 21(5):549–564. 8

Wu, H., Bolte, J. P., Hulse, D., and Johnson, B. R. (2015). A scenario-based approach to integrating flow-ecology research with watershed development planning. *Landscape and Urban Planning*, 144:74–89. 59, 60

Zhao, C., Liu, C., Li, W., Tang, Y., Yang, F., Xu, Y., Quan, L., and Hu, C. (2023). Simulation of urban flood process based on a hybrid LSTM-SWMM model. *Water Resources Management*, 37(13):5171–5187. 8

Vita

Originally from Chile, Isidora Fletcher moved to the United States to pursue higher education. She attended the University of Tennessee, Knoxville, where she received a Bachelor of Science degree in Physics with a minor in Astronomy. It was during her undergraduate studies that she developed a passion for programming and computational problem-solving. This interest led her to continue her education at the University of Tennessee, where she joined the Bredesen Center for Interdisciplinary Research and Graduate Education to pursue a Doctor of Philosophy degree in Data Science and Engineering. Her research interests focus on the intersection of artificial intelligence, computational modeling, and optimal decision-making, with a specific emphasis on applying these techniques to solve complex challenges in environmental science and sustainable water management.