Advancing Autonomous Green Stormwater Infrastructure

by

Brooke Mason

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in The University of Michigan
2023

Doctoral Committee:

Associate Professor Branko Kerkez, Chair
Professor Cyndee Gruden, University of New Hampshire
Professor Nancy Love
Professor Jerome Lynch, Duke University
Associate Professor Ramanarayan Vasudevan

Brooke Mason
bemason@umich.edu
ORCID iD: 0000-0001-5868-7026

DEDICATION

To my wife for her unwavering love and support.
To my brother who showed me the path forward.
To my parents who made so many sacrifices.

ACKNOWLEDGMENTS

First, I must thank my advisor, Prof. Branko Kerkez, for his encouragement, support, and guidance throughout my PhD journey. The PhD process is challenging and there were many days where I was not sure I could do it. His unwavering belief in me helped me forge ahead and become the confident researcher I am today. I would like to thank my dissertation committee, Dean Cyndee Gruden, Prof. Nancy Love, Dean Jerry Lynch, and Prof. Ram Vasudevan for their support and guidance throughout my academic career. A special thank you to Dean Gruden, who has been a mentor of mine since she advised me as an MS student at the University of Toledo.

I also want to thank all the amazing collaborators I worked with over the years. The best day of any week was when I had a meeting or field visit with one of our collaborators. These collaborators include: Ric Lawson and Andrea Paine from the Huron River Watershed Council; Erma Leaphart, Elayne Elliott, and Cyndi Ross from the Detroit Sierra Club; and Harry Sheehan, Catherine Wytychak, and Grayson Wilcox from Washtenaw County. I would also like to thank the site owners in Detroit for allowing us to install green infrastructure sensors at their homes, churches, and schools. These installations were only possible with the help of Angela Hojnacki, Ian Thompson, and Kevin Kaya, so thank you. I also have to thank our Project Manager, Kate Kusiak Galvin. We are so lucky to have her on our team– she keeps us organized, on schedule, well stocked, and not to mention, caffeinated and fed!

I have worked alongside amazing peers in both the Digital Water Lab and in the broader department. To my labmates Brandon Wong, Matt Bartos, Abhi Mullapudi, Sara Troutman, Ernesto Martinez, Meagan Tobias, Jack Schmidt, Stef Escobar, Travis Dantzer, Ariel Roy, Gina Kittleson, and Bryon Banman, thanks for always being there to vent about the trials and tribulations of graduate school, for making me laugh, and for forcing me to take breaks to have fun. I also worked with so many

amazing students on the Graduate Student Advisory Council, the DEI Collaborative, and the DEI Committee. You all made my graduate school experience so much more impactful and meaningful.

I have to thank my friends Serena Falkenberg, Lorena Ganser, Jess Buckley, and Andrea Paine (A.K.A. Sandy Cheeks). If I needed a distraction or a break, I could always count on them to find us a fun adventure. Their love and laughter is always the perfect re-energizer. I also have to thank Mel Hudson Nowak, a friend and mentor I met years ago at Bowling Green State University. When we met, she saw something in me that I could not yet see in myself. Over time, her steadfast belief in me has always helped me push through hard times. I love you all and am so thankful to call you friends.

My wife Julia and our pup Cammy have been on this journey with me almost from the beginning. They have been right by my side for the ups and downs, never wavering in their love and support. Julia never complained when I would go MIA for days to study for an exam or write a paper, and then for a few more days as I slept and recuperated. She would surprise me with little notes and treats, reminding me she loved me and I could do this. I am so grateful to have them by my side and I am so excited for our journey together once our third wheel (my PhD) is finally in the rearview mirror! My love for you two is unbounded.

I would not be here today without the love and support of my family. From as young as I can remember my mom told my brother and I that we were going to college so we could have opportunities she never had. We both took what she said very seriously– we went to college, and then didn't stop until we had PhDs. I am truly grateful for all the sacrifices my parents have made for me over the years. Those sacrifices gave me incredible opportunities that I would not have had otherwise. Without them, I would not be here today. My little brother, Lance, has always been my best friend and biggest cheerleader. His love and support has carried me forward on hard days. As the older sister, I generally chartered the path for us but it took him

showing me that a PhD was possible for me to even consider it an option for me. To the rest of my family, those with us and today and those who left this world too soon, thank you for the unconditional love and for helping shape me into the person I am today. And my new family, the Kings and Friedrichs, you have all been so welcoming and supportive during the craziness of this PhD journey. I am so thankful for and love you all very much.

Finally, I would like to acknowledge my funding sources, the US National Science Foundation (Award Numbers: 1737432 and 1750744) and the Great Lakes Protection Fund (Award Number: 1035).

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

ABSTRACT

Flooding causes more damage and fatalities than any other natural disaster. Simultaneously, storms carry large quantities of pollutants into receiving waters. These challenges are compounded by urbanization and a changing climate. Traditional infrastructure solutions, such as larger storage basins and pipes, are cost prohibitive. Green infrastructure (GI) has been proposed as a nature-based alternative to new construction. GI includes assets such as rain gardens and bioswales, which are designed to capture runoff, treat pollutants, and infiltrate water into underlying soils. However, the scalability of these distributed solutions has yet to be vetted and measured at scale. Another promising solution, autonomous stormwater systems, leverage recent advances in wireless sensing, communications, and controls. Infrastructure assets are retrofitted with wireless sensors and controllable valves, enabling them to adapt to changing weather, flows, and pollutant loads. These controllable assets are coordinated at the system-scale to achieve flooding and pollutant objectives across watersheds. While these novel solutions have gained traction for traditional stormwater infrastructure, they have only just begun to be investigated for GI. Before autonomous technologies can be adopted for GI, several fundamental knowledge gaps must be closed. Broadly, these include a lack of understanding around how GI can be effectively and optimally measured at scale, as well as how pollutant transformations should be modeled to support real-time control. This dissertation addresses these knowledge gaps and presents foundational work towards enabling autonomous GI. The second chapter introduces an end-to-end data toolchain, underpinned by a wireless GI sensor network for continuously measuring real-time water levels. The toolchain automatically isolates storms in the sensor data to parameterize a dynamical system model of GI drawdown dynamics. The model outputs are then used to investigate the explanatory features of drawdown dynamics. We show how invest-

ments in monitoring networks support a more targeted and data-driven approach to GI design, placement, and maintenance. Investing in monitoring networks requires knowing where to place sensors and how many are needed. The third chapter introduces a sensor placement methodology for urban drainage networks using publicly available datasets and Gaussian Processes. The methodology is flexible enough to work for any stormwater sensor or spatial parameter of interest, has guarantees of optimality, and is computationally efficient. We show that the methodology maximizes information gained from the sensor network while minimizing its size. The fourth chapter addresses the lack of simulation tools necessary for modeling complex pollutant transformations affected by real-time control in urban drainage networks. A new water quality package, *StormReactor*, is introduced. *StormReactor* provides an open-source Python programming interface for simulating complex pollutant generation, treatment, and real-time control processes. Two case studies are presented to illustrate the fidelity of *StormReactor* and the potential of using real-time control for ecological benefits. Expanding upon these case studies, the fifth chapter evaluates the impact of real-time control on pollutant removal in GI. *StormReactor* is used to simulate real-time control of a real-world inspired GI to capture phosphorus. We show that real-time control not only provides a "digital" alternative to existing, passive GI upgrades, like soil amendments, but it also provides long-term flexibility. This flexibility enables stormwater managers to dynamically balance trade-offs in existing GI designs and aids in the larger goal of system-level control. The dissertation concludes with a broader discussion on how the discoveries made can support the development of a new generation of autonomous GI.

CHAPTER 1

Introduction

Flooding causes more damage and fatalities than any other natural disaster [1]. Simultaneously, storms carry large quantities of pollutants into receiving waters [2–4]. In fact, runoff pollution is acknowledged as one of our greatest environmental challenges [5]. These problems are compounded by urbanization and a changing climate [6]. More than half of the world's population lives in urban areas [7], and this number continues to rise. With urbanization comes an increase in impervious area, which results in fewer surfaces to infiltrate stormwater [6, 8, 9]. At the same time, storms have become more frequent and intense [9, 10]. As our cities become boxed in by development, more frequent flooding and larger quantities of pollutants degrade urban waterways and downstream ecosystems [11].

Presently, the main solution to combat flooding and pollution is to build larger infrastructure, such as pipes and storage basins [12]. Bigger pipes can be built to move stormwater out of urban areas faster, while bigger basins provide storage for excess stormwater. The additional capacity provided by larger infrastructure is vital but there are several drawbacks. First, this solution is cost prohibitive for most urban areas [13]. Second, these solutions often have adverse environmental impacts. The increased volume and velocity discharged into natural waterways can cause erosion and damage aquatic ecosystems [11, 14–16]. Finally, these solutions are passive, meaning they lack the ability to adapt to storms and changing urban landscapes [13, 15]. As such, more adaptive solutions have been sought to manage stormwater and protect our communities.

Adaptive stormwater solutions can be enabled by taking advantage of the autonomy that is presently being embedded in self-driving vehicles, smart homes, and

related "smart" systems [13, 17]. Leveraging the recent advancements and cost reductions in wireless sensing, computation, and communication [17], infrastructure can be transformed from passive to active with the addition of distributed sensors and controllable valves [13]. These technologies enable a stormwater asset to respond in real-time to changing weather, flows, and pollutant loads [18, 19]. Autonomous stormwater systems may one day coordinate tens to hundreds of individual stormwater assets to route water and treat pollutants in real-time to meet watershed goals [20–22]. While these technological solutions are gaining traction for traditional stormwater infrastructure (e.g., pipes, basins), they have yet to be investigated for green infrastructure (GI).

GI is a nature-based stormwater management alternative that filters and absorbs stormwater where it falls [23]. One popular design in the rapidly growing toolbox of GI are rain gardens, which are the focus of this dissertation. Rain gardens enable pervious management in impervious urban areas [24]. Examples of rain gardens include bioretention cells and bioswales, which capture and reduce runoff by allowing it to evapotranspire or exfiltrate into surrounding soil [23]. Benefits of rain gardens include reducing runoff, promoting infiltration and evapotranspiration, recharging groundwater, protecting stream channels, reducing peak flows, and treating pollutants [25, 26]. The general design features of a rain garden are the ponding area, soil media layer, native plants, and an optional gravel layer and underdrain (Figure 1.1). The gravel layer and underdrain are generally recommended for areas with slow draining native soils (e.g., $\leq$ 1.27 cm/hr [27]).

GI is often presented as a more natural and cost-effective alternative to upgrading traditional stormwater infrastructure [28] but some studies have called attention to its limitations [29–31]. While effective at capturing small rainfall events, GI have had varying results capturing large or consecutive storms and treating pollutants [25, 32, 31, 33]. These performance limitations are a consequence of GI being underpinned by complex biological and physical processes [34]. GI performance can be hindered

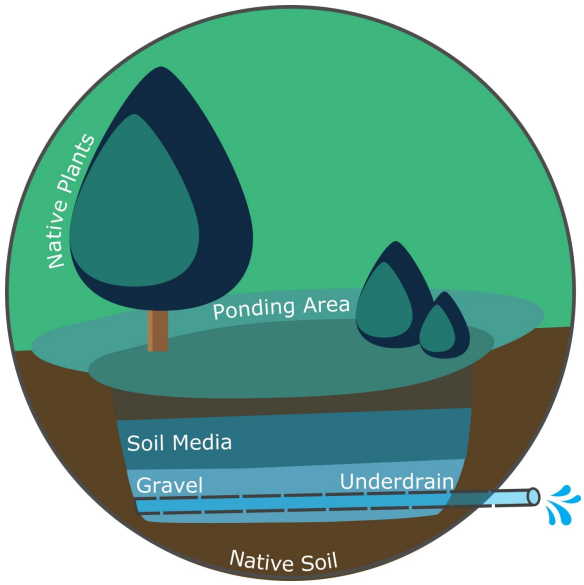

Figure 1.1: The general design features of a rain garden.

further if it is not frequently monitored and maintained [31], however, monitoring GI at the city-scale is costly and time-intensive [25]. Furthermore, the distributed nature of GI has made it difficult to evaluate how the impacts of GI scale; for example, if GI effectively mediates downstream water quantity and quality [35]. These limitations highlight the need for new ways to monitor and boost the performance of GI at both the site- and system-scale. Autonomous technologies, such as those added to traditional stormwater infrastructure, may enable GI to scale and to perform better than ever imagined (Figure 1.2).

The goal of this dissertation is to strike a middle ground between the distributed benefits of GI and the system-level benefits that can be achieved with autonomous stormwater systems. Specifically, the objective is to evaluate the fundamental underpinnings of autonomous GI, and to close the following major knowledge gaps that prevent this vision from becoming reality.

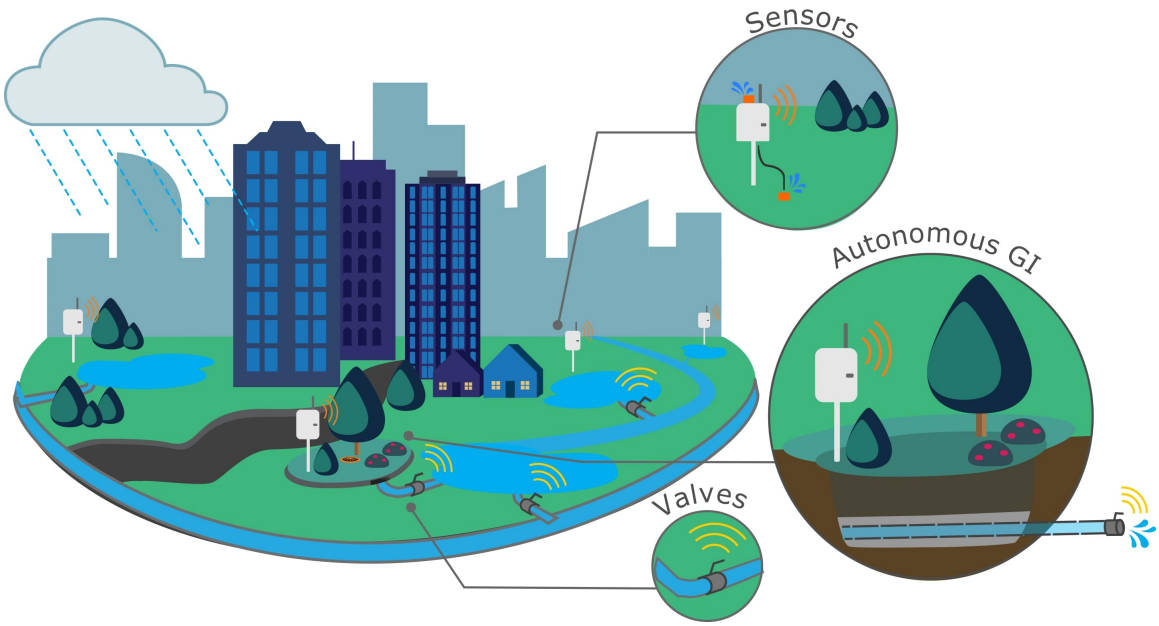

Figure 1.2: Visualization of autonomous stormwater systems with autonomous GI.

- We do not understand how to utilize recent technological advances in sensing and computation for city-scale GI monitoring, preventing us from knowing if individual GI are performing as designed, as well as if and how GI impacts scale. New sensing devices and automated data analytic toolchains are needed to process the high-resolution datasets obtained from such networks.

- We do not understand the value of information in urban drainage networks, nor where we should install sensors in our urban drainage networks, including GI. Without a comprehensive sensor placement methodology, communities can only intuit where sensors are needed and assume they are obtaining the necessary information for managing their urban drainage networks.

- We do not understand how to model water quality-based real-time control in

stormwater infrastructure, including GI, which prevents us from knowing if or how it will work. Existing stormwater models cannot represent water quality processes at the fidelity required to evaluate real-time control at either the site- or system-scale.

- We do not understand how to build an autonomous GI site, nor how to control such a site, preventing us from knowing if real-time control can improve GI pollutant treatment performance. While real-time control has been shown to improve pollutant removal in traditional stormwater infrastructure, extrapolating these benefits to GI is difficult because pollutant removal is underpinned by much more complex biological and physical processes.

This dissertation will close these knowledge gaps by introducing new advances in wireless sensing, data analytics, and control for green stormwater infrastructure. In addition to improving our fundamental understanding of autonomous green infrastructure systems, this work will provide practical, data-driven tools water managers can use to guide GI development from design to maintenance. To that end, the specific contributions of this dissertation include:

- **Chapter 2:** An end-to-end data toolchain, underpinned by a novel wireless sensor network for continuously measuring real-time water levels in GI. The toolchain isolates storms in the sensor data to parameterize a dynamical system model of GI drawdown dynamics. The model outputs are then used to investigate the explanatory features of drawdown dynamics. The methodology is evaluated on a GI sensor network in Detroit (Michigan, US), showing that depth to groundwater, imperviousness, longitude, and drainage area to surface area (DA/SA) ratio are the most important features explaining GI drawdown dynamics in Detroit.

- **Chapter 3:** An iterative sensor placement methodology for urban drainage

networks using sensor data, publicly available GIS datasets, and Gaussian Processes (GP). The proposed methodology can be used for any urban drainage network sensor or spatial parameter of interest, has guarantees of optimality, and is easy and efficient to use. The proposed methodology is evaluated on the Detroit GI sensor network, showing that the size of existing network can be reduced without compromising the predictive capabilities of the network.

- **Chapter 4:** *StormReactor*, a new water quality package providing an open-source Python programming interface for simulating complex pollutant generation, treatment, and real-time control processes. The package's ability to model complex pollutant transformations and real-time control actions is evaluated using two case studies. The first case study shows that *StormReactor* improves pollutant process representation by including pollutant generation processes, while the second case study shows that a real-time controlled asset can achieve the same pollutant improvements as an uncontrolled asset in a quarter of the spatial footprint.

- **Chapter 5:** A study that evaluates the impact of real-time control on pollutant removal in GI. Based on a real-world GI retrofitted for real-time control in Toledo (Ohio, US), *StormReactor* is used to simulate real-time control of the GI's underdrain to improve phosphorus removal. The autonomous GI is compared to an uncontrolled GI and a GI passively upgraded with soil amendments. Results show the autonomous GI matches the pollutant treatment performance of the uncontrolled GI in half the spatial footprint. The autonomous GI also matches the performance of the passive upgrade, suggesting real-time control may provide a "digital" alternative to existing, passive upgrades.

1.1 Background

1.1.1 Green Infrastructure Performance

GI have shown varied performance in terms of reducing runoff and peak flows (Table 1.1). Generally, rain gardens effectively capture small storms but struggle with large and/or consecutive storms [25, 32, 31, 33, 36]. Winston et al. [37] found that three rain gardens reduced overall runoff by 36 to 59%, despite slow draining native soils. During small rainfall events (i.e, 1-year design rainfall intensities), the rain gardens reduced peak flows by 24 to 96%. Whereas the gardens released outflows during large events (i.e., 5.5 to 13.8 mm).

In terms of pollutants, removal is highly dependent on the complex biological and physical interactions between the pollutant and the GI plant/soil systems [34]. Greater success has been observed with capturing particular pollutants (e.g., total suspended solids), through sedimentation and filtration by the soil media [24]. Dissolved pollutants (e.g., dissolved phosphorus, dissolved nitrogen), however, are even more challenging for rain gardens to treat [24, 38]. Rain gardens primarily remove dissolved pollutants through adsorption, but they can also leach (i.e., release) pollutants previously stored in the soil media and vegetation [24, 39]. Table 1.1 provides a summary of GI performance reported in several literature review studies.

1.1.2 GI Design Standards

Many communities rely on established stormwater management manuals, which detail how to select, design, construct, and maintain stormwater infrastructure, including GI. A manual's goal is to set forth best management practices which will elicit a certain level of performance, such as mitigating peak flow or infiltrating a certain fraction of runoff [42]. Regional and local manuals set design requirements (e.g., site selection, GI selection/sizing, soil media composition, underdrain sizing, plant selection) as well

Literature Citation	No. Studies per Review	Runoff	Peak Flow	TSS	TP	TN
[25]	17	48 to 97%	N/A	47 to 99%	−3 to 99%	32 to 99%
[23]	4	N/A	N/A	−170 to 96%	−240 to 87%	40 to 59%
[40]	12	N/A	N/A	54 to 99%		
[41]	14	0 to 100%	0 to 99%	−170 to 100%	−240 to 100%	−3 to 99%

Table 1.1: Review studies of rain garden performance reporting reductions in runoff, peak flow, total suspended solids (TSS), total phosphorus (TP), and total nitrogen (TN).

as performance metrics [24]. These design requirements and performance metrics exist for a variety of reasons, for example to ensure public safety and limit liability by eliminating trip hazards, adding barriers around water features, and to control mosquitoes, but most fundamentally, to ensure that stormwater is being managed consistently across various sites. As an example, two common metrics for rain gardens and bioretention cells include the maximum allowable ponding time, generally 12–48 hours [27, 43, 44], and infiltration rate, typically 2.5–5 cm/hr [24, 27, 44].

While infiltration rates can vary substantially even within the same GI, drawdown rates are representative of the entire system [45, 46]. Drawdown rate is the speed at which water is evapotranspired or exfiltrated into the surrounding native soil [47, 37]. The drawdown rate of GI is a function of the design features, building and maintenance practices, and the surrounding and underlying physiographic features [47, 48]. Design features include size, soil type, and vegetation. During site construction, how the sites are excavated and graded can cause significant soil compaction which ultimately impacts GI drawdown rates [49]. Physiographic features include the native soils, topography, land use type, depth to groundwater, and sunlight [50, 48]. While GI design can be optimized, in most cases the surrounding physiographic features cannot be changed. These features may have a strong effect on GI drawdown. For example, a shallow groundwater table ($<$ 2–3 m) may result in more saturated media,

which forms a smaller hydraulic gradient, impeding infiltration into the GI and exfiltration out of the GI into the native soil [51, 52]. This suggests that the drawdown rate of GI is governed by complex interactions between design features and surrounding physiographic features. Few large-scale data sets exist to verify these patterns at scale, however.

1.1.3 Measuring GI Performance

Monitoring is needed to confirm whether a GI is meeting desired management goals. Additionally, monitoring can be used to determine whether local stormwater manuals are setting appropriate design standards and performance metrics. Due to the sheer number of sites and the cost of measuring quantitative metrics such as drawdown rate, cities often rely on visual inspection or modeling to assess performance [25]. If GI monitoring is carried out, it is generally limited to certain time periods and conditions [25, 48, 53].

Drawdown rate has been traditionally measured via drawdown testing. A GI is filled with water (either synthetically or via rainfall) until ponding occurs, then the drain depth and time are recorded to calculate the drawdown rate [47, 54]. These measurements are typically conducted manually with the help of a watch and gauge plate. Drawdown testing is generally only done pre- and post-installation [37], but occasionally assets are tested as they age to track how they change over time [55, 56]. Unfortunately, the laboriousness of drawdown testing results in most communities having sparse datasets of in-situ GI drawdown. Furthermore, drawdown is inherently non-linear [37], meaning that drawdown rate may change over the course of a storm and in response to ambient conditions. To gain a complete picture of GI behavior, more data are needed than what can be obtained from a single drawdown test taken during a single storm event.

Recent technological advances in Internet of Things (IoT) technologies have opened

up new possibilities for low-cost, high resolution stormwater sensing [20, 57]. Real-time sensing has been successfully deployed to monitor depths and flows in stormwater [19] and sewer networks [58, 59]. Recently, some studies have used sensors, such as pressure transducers connected to data loggers, to monitor GI [60, 56, 37, 54]. While these studies provided high resolution measurements, they required frequent field maintenance (e.g., downloading the data onsite, replacing batteries), making this approach impractical for obtaining large-scale, and/or long-term data. Aside from these studies, the uptake of these technologies for GI management has been limited. According to a national survey of officials in water utilities and agencies, assumed high construction and maintenance costs associated with smart GI are the two main barriers to adoption [61]. As such, there is a need to vet low-cost, low-maintenance IoT solutions for GI at scale.

1.1.4 Optimal Sensor Placement

Even with access to reliable GI sensors, knowing where to place these devices becomes a challenge. Communities that want to measure GI performance at scale, or stormwater infrastructure more broadly, at scale are faced with the challenge of determining how many sensors are needed and where they should be placed in the urban drainage network. Several studies have investigated optimization-based methods for sensor placement in urban drainage networks [62–69]. These studies formulated and then solved an objective function using computational models and optimization algorithms. For example, Fattoruso et al. [62] explored flood gauge placement for calibrating an EPA Storm Water Management Model (SWMM). The study searched for the model parameters that minimized the error between the predicted and measured values in the real urban drainage network. The sensors were selected using an enumerative search algorithm. This approach has several limitations, however. First, this methodology works for model calibration, but it is not transferable to other

objectives, like event detection or understanding drawdown dynamics. Second, the approach relies on a calibrated model, which many communities do not have. Finally, the selected algorithm does not necessarily guarantee optimality.

Studies have also investigated optimal sensor placement for detecting pollutants [63, 66, 67] and flooding [64, 68, 69]. These studies developed their own objective functions based on domain knowledge. The required data to evaluate their objective functions were obtained through model simulations. The majority of these studies used SWMM [63, 62, 66, 68], while the remaining studies used either a graph-based model [64], a Bayesian network model [69], or a derived model of riverine contaminant transport [67]. Using the model outputs, the objective functions were then solved using optimization methods, including Bayesian decision networks [66], agglomerative clustering [68], and genetic algorithms [63, 64]. While these approaches resulted in viable sensor placements, they have the same limitations as Fattoruso et al. [62].

Yazdi [65] proposed a more flexible objective function based on entropy theory to place water quality sensors in urban drainage networks. The objective was to maximize the joint entropy, or the information content obtained by the sensor network. To solve the objective function, SWMM and an evolutionary algorithm was used. While more flexible, two of the limitations described above still remain— it uses a calibrated model and it does not guarantee optimality.

Ideally, a sensor placement methodology would have the following properties. First, it would use an objective function that is flexible enough to work for any sensor or spatial parameter of interest to urban drainage networks. Second, it would have some guarantees of optimality. Finally, it would be easy and efficient to use, meaning it is computationally efficient and does not require a calibrated physics-based model.

1.1.5 Modeling GI Performance

Data provide us with invaluable metrics to measure performance, but do not enable models or scenario planning tools that are needed by community managers to make investments in new infrastructure. GI are generally modeled using an existing stormwater model, which can be broadly grouped into two categories: water quantity models and water quality models. Most stormwater models primarily focus on coupled hydrologic-hydraulic processes with limited capabilities for modeling water quality (e.g., MIKE URBAN+, DR3M, STORM, MUSIC, SWMM) [70, 71]. However, some stormwater models do focus on high resolution water quality processes. These finite element models (e.g., HYDRUS-CWMI, FITOVERT) simulate complex pollutant transformations within individual sites [72–74]. Unfortunately, scaling from site- to watershed-scale becomes very difficult due to the input data requirements and the difficulty of parameterization. The chasm between these two types of stormwater models forces a trade-off between either comprehensively modeling water quality at the site scale, or less comprehensively modeling watershed-scale processes.

To avoid this tradeoff, researchers have modified existing stormwater models, like SWMM, to expand their pollutant modeling capabilities. SWMM, widely used in the US stormwater community, is an open-source urban stormwater model [75]. SWMM's water quality model provides users the ability to introduce pollutants and pollutant treatment, while also routing and calculating mass balance for each pollutant [76]. *SWMM-TSS* modified SWMM to simulate total suspended solids (TSS) transport, accumulation, and erosion in sewers and retention tanks [77]. As its name implies, this modification is only for TSS. Baek et al. [78] modified SWMM's water quality module for low impact development (LID) to include straining, decay, and decomposition of pollutants. However, this modification does not work for stormwater storage assets or links. Talbot et al. ([79]) modified PCSWMM, a licensed version of SWMM, to simulate sediment loading due to soil erosion. This modification is not open source and

thus not open for exploration or expansion by the community. All of these packages are very useful for specific modeling tasks; however, they do not offer general water quality modeling solutions.

Although these packages provide additional functionality, SWMM has many remaining pollutant modeling limitations that must be addressed. The water quality module is limited by the range of treatment measures that can be modeled [80], specifically, limited nutrient treatment capabilities inside storage nodes (e.g., basins, wetlands) [81, 82]. SWMM cannot simulate pollutant treatment inside links (e.g., conduits, channels) or pollutant generation processes (e.g., resuspension, erosion) inside any stormwater asset. Pollutant treatment cannot be turned on or off based on site conditions or other parameters, requiring treatment to run for the entire simulation. All of these constraints limit a user's ability to model complex pollutant transformations, necessitating a more generalizable and scalable approach.

Aside from water quality limitations, many stormwater quality models have limited or no ability to simulate real-time control. Real-time control is made possible through the installation of sensors and controllers [83, 84]. To realize the goal of autonomous stormwater systems, we must be able to model real-time control strategies [85, 86]. One open-source and popular real-time control package is PySWMM, a Python wrapper for the SWMM computational engine. PySWMM queries stormwater states directly from SWMM, which is used to apply control actions by setting the control parameters for valves, gates, and pumps in real-time [87]. However, PySWMM presently only enables real-time control decisions to be made based on water quantity parameters (e.g., flow, head, depth, volume). Therefore, there is a need for a comprehensive package that can both simulate water quality processes and real-time control.

1.1.6 Boosting GI Performance

1.1.6.1 Passive Modifications

Both simulation and field measurement studies have reported inconsistent results in terms of GI's ability to capture flows and pollutants [25, 33]. To boost GI performance, passive design modifications have been explored in the literature [88–90, 24, 39]. For GI with an underdrain, modifying the underdrain and outlet configuration can boost hydrologic and pollutant performance. For example, to increase retention time, the size of the outlet can be reduced and a larger diameter outlet placed above ground [88]. The smaller outlet slows the flows from the underdrain, providing more time for the rain garden to treat pollutants and exfiltrate/evapotranspirate flows, while the outlet above ground ensures the site does not flood. Aside from reducing the outlet size, the outlet configuration can also be redesigned. For example, adding an upturned elbow to the underdrain increases the the internal water storage layer [89, 90]. The upturned elbow provides more time for the system to treat nitrogen and exfiltrate water into the surrounding soil instead of flowing through the underdrain into the sewer system. While these solutions improve performance for well-spaced storms, they may hinder performance for consecutive storms. If these systems do not drain quickly, they may not have adequate storage space for the next storm. To that end, more flexible solutions are needed that can improve performance for both well-spaced and consecutive storms.

In terms of pollutants, treatment can also be improved by adding soil amendments (organic or inorganic) to GI [24, 39]. For example, it is known that the phosphorus sorption capacity can be improved by amending the iron oxide and aluminum oxide contents of the soil inside a bioretention cell [91]. Soil amendments are mixed into the soil media (5-30% of the total soil volume). Their price is highly dependent on the type of amendment, ranging from free water treatment plant residuals (a by-product of the water treatment plant) to \$16,000 USD/m^3 for iron filings. While

soil amendments are excellent at removing specific pollutants, they deplete over time, provide limited hydrologic benefits, and have limited or no impact on pollutants for which they were not amended [91]. As such, discovering more flexible ways to manage the limitations of GI is important for meeting runoff and pollutant goals.

1.1.6.2 Real-Time Control

While passive modifications may improve GI performance, their static nature mean GI cannot adapt to changing conditions in real-time. Adding a controllable valve to the end of a GI's underdrain transforms it from a passive to active system, enabling the GI to switch between storing and releasing water [92]. The opening and closing of this valve allows for many of the trade-offs in existing bioretention designs to be dynamically balanced. Intuitively, this suggests that controlling water levels using a valve would improve exfiltration and pollutant treatment due to extended residence times. However, real-time control creates a dynamic environment for soils, plants, and microbes that is unexplored and may have unintended consequences [92]. A laboratory column study by Persaud et al. [92] found that real-time control can improve upon standard bioretention designs, but further optimization is required to balance water quality benefits against storage needs for impending storms. Aside from this column study, the benefits and practicality of real-time control have yet to be quantified for GI.

1.2 Summary

When combined, the prior sections illustrate that a set of first steps need to be taken to close key knowledge gaps, including (1) how to measure and analyze city-scale GI performance; (2) where to place urban drainage network sensors; (3) how to model stormwater quality-based real-time control; and (4) how to build and control

autonomous GI. The following chapters chart a journey to close these knowledge gaps, while highlighting new frontiers for the field of autonomous GI.

CHAPTER 2

Measuring City-Scale Green Infrastructure Drawdown Dynamics Using Internet-Connected Sensors in Detroit

Abstract

The impact of green infrastructure (GI) on the urban drainage landscape remains largely unmeasured at high temporal and spatial scales. To that end, a data toolchain is introduced, underpinned by a novel wireless sensor network for continuously measuring real-time water levels in GI. The internet-connected sensors enable the collection of high-resolution data across large regions. A case study in Detroit (MI, US) is presented, where the water levels of 14 GI sites were measured in-situ from June to September 2021. The large dataset is analyzed using an automated storm segmentation methodology, which automatically extracts and analyzes individual storms from measurement time series. Storms are used to parameterize a dynamical system model of GI drawdown dynamics. The model is completely described by the decay constant α, which is directly proportional to the drawdown rate. The parameter is analyzed across storms to compare GI dynamics between sites and to determine the major design and physiographic features that drive drawdown dynamics. A correlation analysis using Spearman's rank correlation coefficient reveals that depth to groundwater, imperviousness, longitude, and drainage area to surface area ratio are the most important features explaining GI drawdown dynamics in Detroit. A discussion is provided to contextualize these finding and explore the implications of data-driven strategies for GI design and placement.

2.1 Introduction

Urban areas around the world are struggling to manage stormwater runoff and flooding– a challenge compounded by rapid urbanization and climate change [93, 6]. Gray infrastructure, which consists of gutters, drains, and pipes, is the traditional method for collecting and conveying stormwater away from urban areas. Recently, green infrastructure (GI) has become a popular alternative, used either as a standalone stormwater management practice or in concert with traditional gray infrastructure [48, 35]. GI attempts to mimic the natural water cycle by using plants, soil, and landscape design to capture and filter local runoff [25, 48]. One of the most common GI practices is bioretention cells, or rain gardens, which are depressed vegetated areas that capture and reduce runoff by allowing it to evapotranspire or exfiltrate into surrounding soil [24].

Communities worldwide are investing in GI for managing stormwater at increasing scales. For example, China plans to spend over US$ 1.5 trillion on GI in 657 cities by 2030 [94]. In the midwestern US, the city of Detroit, Michigan invested US$ 15 million in GI between 2013–2017 and will invest US$ 50 million by 2029 [95]. These investments assume adding more GI assets will positively impact stormwater outcomes, however, sufficient data to support this claim has yet to be produced [96, 25, 97, 48].

Real-time monitoring of stormwater infrastructure at high temporal and spatial resolutions is now possible with Internet of Things (IoT) technologies [57, 20]. Real-time sensing has been successfully deployed to monitor depths and flows in stormwater [19] and sewer networks [58, 59]. Recently, some studies have used sensors, such as pressure transducers connected to data loggers, to monitor GI [60, 56, 37, 54]. While these studies provided high resolution measurements, they required frequent field maintenance (e.g., downloading the data onsite, replacing batteries), making this approach impractical for obtaining large-scale, and/or long-term data. Therefore,

there is still a need for GI IoT solutions.

To that end, we introduce an end-to-end data toolchain based on new wireless sensors for estimating real-time drawdown in GI, the speed at which stormwater is evapotranspired and exfiltrated into the native soil [37, 25]. These wireless sensors are low-cost, easy to install, and can be deployed at scale to create large, long-term, high-resolution datasets of urban drainage conditions. When combined with an analytics toolchain, our approach can be used to automatically learn GI dynamics from data on a storm-by-storm basis. To study the value of a city-wide dataset, we present a case study of these GI sensors deployed in Detroit. This novel dataset is used to characterize the drawdown dynamics of GI over multiple storms. The core contribution of this paper is a new sensor and data analysis methodology, along with experimental results that show which factors are the strongest predictors of drawdown dynamics for the studied GI network.

2.2 Materials and Methods

2.2.1 Green Infrastructure Wireless Sensors

A wireless sensor was designed to continuously measure drawdown in GI (Figure 2.1). Specifically, the device measures water level fluctuations in real-time. At the time of writing, the sensor costs approximately US$ 1,000 to build and US$ 25 annually for telecommunication and data storage services. The form factor of the sensor is similar to a water well, consisting of a 1.5 m long, slotted PVC pipe with one end holding the sensor and the other holding the remaining hardware components. The sensor uses the vetted Open Storm hardware and cloud services stack detailed in Bartos et al. [19]. The hardware layer relies on an ultra-low power ARM Cortex-M3 microcontroller (Cypress PSoC). The microcontroller manages the sensing and data transmission logic of the embedded system. The sensor measures water levels to a

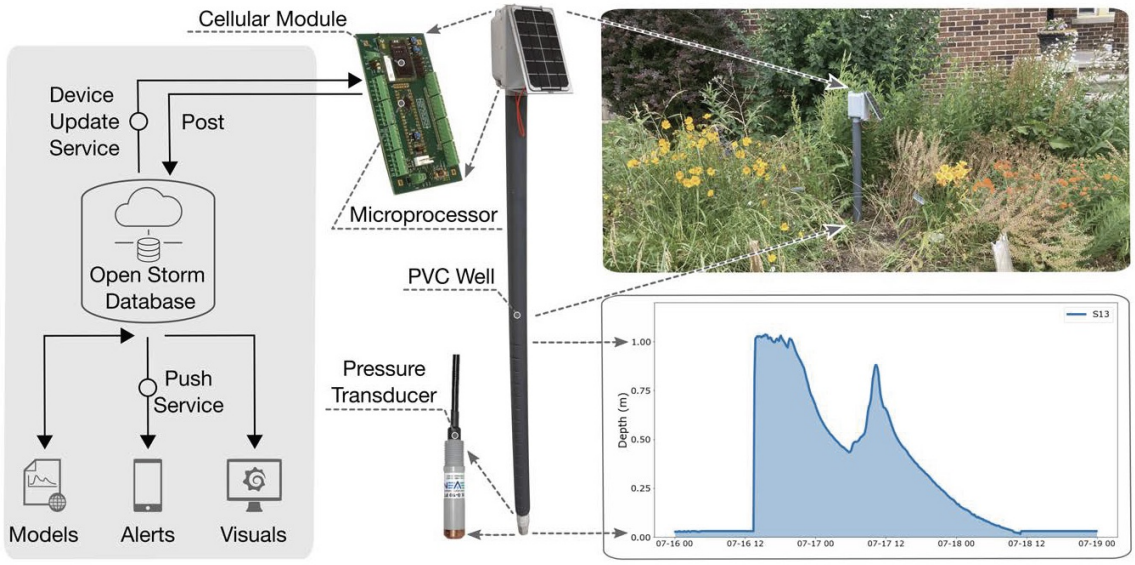

Figure 2.1: A GI sensor installed in a rain garden (top right). The sensor's hardware layer (center) includes the PVC well, microcontroller, cellular modem, and pressure transducer. The cloud services layer (left) includes the database backend, along with applications for controlling sensor behavior and visualizing data (bottom right).

reported accuracy of ± 0.762 cm using a pressure transducer (Stevens SDX 93720-110), which converts a barometric reading to a 4–20 milliampere (mA) output. The sensor is equalized for atmospheric pressure changes and was calibrated in the laboratory using a standard water column. The device is connected to the internet with a 4G LTE CAT-4 cellular modem (Nimbelink NL-SW-LTE). The cellular modem enables bi-directional communication between the sensor and a remote cloud-hosted web server. The device is powered using a 3.7 V lithium-ion battery (Tenergy) that is recharged by a solar panel (Adafruit 500). Power consumption measurements were used to confirm that when the device is on, power consumption is in the milli-amperage range and when the device is in sleep mode, it is in the micro-amperage range. With these power consumption numbers the sensor can stay in the field for up to 10 years

without needing a battery replacement.

The main reason for field maintenance occurs if a sensor malfunctions. The first type of sensor malfunction is sensor drift, which is defined as a small temporal variation in the sensor output under unchanging conditions. Sensor drift can be detected in this case when the sensor's "zero" reading changes over time. The other type of sensor malfunction occurs if a sensor provides a zero reading during periods of rainfall. There are several possible explanations for this malfunction. First, since the sensor operates by converting current to depth, there could be an issue with the analog circuitry resulting in inaccurate current measurements. Second, the sensor could be physically damaged during node assembly or deployment. Third, the sensor provides a venting tube for equalizing atmospheric pressure changes. Although a cap is added to the tube to keep moisture out, if the cap is faulty, condensation can enter the tube and cause inaccurate readings. Finally, the PVC well may clog with sediment. To rectify any of the above sensor malfunctions, the sensor is swapped for a new one, which only takes a few minutes of field work.

The sensor measurements were validated in the field using a gauge plate and digital, time-lapse photography by an outside consultant [98]. During rain events, photos were taken of the ponded water and gauge plate measurement every ten minutes (Figure A.1). There was an average alignment of 11 mm between the camera-recorded and sensor-recorded depth measurements (Figure A.2).

Installation of the sensor takes less than 30 minutes by one person and requires digging a 1 m deep hole using a simple, off-the-shelf, handheld post hole digger. The sensor is placed in the hole and backfilled with soil. Real-time data begins streaming to a web dashboard as soon as the unit is deployed. The sensor is deployed such that an water level of 0 m indicates dry conditions, while a measurement above 1 m indicates water is ponding on the surface.

The sensor takes measurements every ten minutes and reports data to the server once every hour. Measurements are transmitted over the cellular network via a secure

connection to a cloud-hosted server. Data and metadata are stored in an InfluxDB database [99]. Measurements are then made available for visualization and sharing with partners through Grafana [100], a dashboarding software used to plot measured water level over time. Both InfluxDB and Grafana instances are hosted on an Amazon Web Services (AWS) Elastic Cloud Computing (EC2) instance [101]. The system is entirely open source and the complete codebase, hardware schematics, and how-to guides have been made available as part of this paper on `github.com/kLabUM/GI_Sensor_Node`.

2.2.2 Automatically Learning GI Dynamics from Data

To enable comparisons between sites without losing temporal information due to averaging, we synthesize and parameterize a drawdown model automatically from data. We assume that water levels inside GI can be approximated as a first-order linear dynamical system, which evolves according to the differential equation:

$$\frac{dh}{dt} = \alpha h \; ; \; \alpha < 0 \tag{2.1}$$

where h represents the water level in GI and α is the decay constant– a measure of how fast the water level inside a GI recedes following a storm. In this formulation, this decay constant is directly proportional to drawdown rate and provides a single parameter that can be compared between sites. A relatively larger magnitude α corresponds to a faster rate of drawdown, while a smaller magnitude α corresponds to more slowly changing water levels. More relevant to cross comparisons between sites, however, is that α embeds both temporal and magnitude information in one parameter. In other words, two sites could have similar bulk performance metrics, such as average volume capture over 24 hours, but exhibit vastly different drawdown curves. As such, studying the decay constant α allows us to compare sites while taking advantage of the temporal granularity of our sensor data.

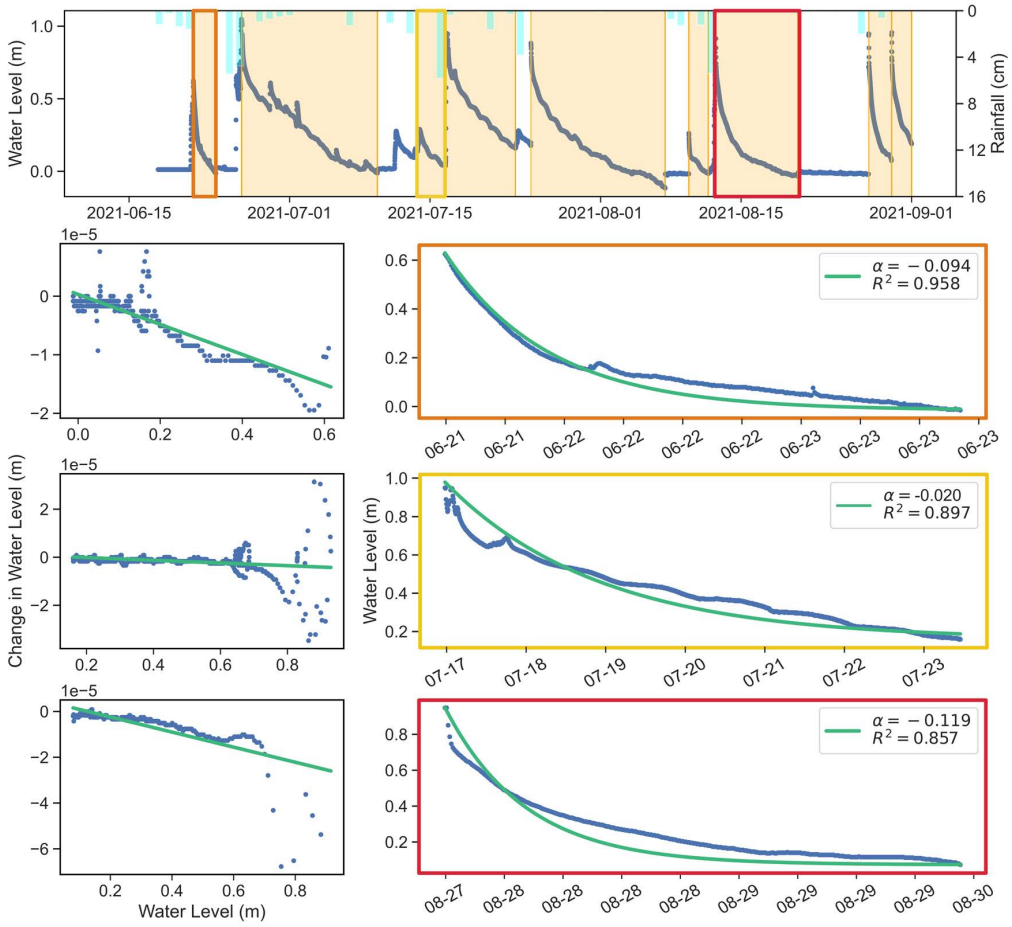

Figure 2.2: (Top row) Time series water level measurement from a GI overlaid with nearby publicly-available precipitation data. The orange boxes indicate distinct storm events automatically detected by a peak finding algorithm. The decay constant α is fit for three distinct storms in the same GI. (rows 2–4, left) To find α, we fit a line for the relationship between water level (x-axis) and the change in water level (y-axis). (rows 2–4, right) The found α's are then plotted against the actual water levels experienced from the three distinct storms. The R^2 value for each fit is also provided.

Linear regression is used to fit the drawdown model to the water level sensor data of each storm. To fit the data to Eqn. 2.1, we find the fit that best captures the relationship between the water level and its first derivative $[h(t), \frac{dh}{dt}]$ (Figure 2.2, left col.). The slope of this line is the decay constant, α. This method selects the most dominant rate of decay in the data. The fit of the model is evaluated using two metrics: the coefficient of determination (R^2) and root mean squared error (RMSE). To illustrate the methodology, the fit of the drawdown model to the sensor data for three distinct storms is shown in Figure 2.2.

Since we calculate α for every storm, drawdown dynamics of each site can be compared on a storm-by-storm basis, or the set of α's can be combined into a single value for a given site. A single value of α can be thought of as a regression in $[h(t), \frac{dh}{dt}]$ feature space across all storms. This allows us to model the expected water level drawdown curve for a future storm. The resulting model could be used to inform estimates on how long a GI would take to drain given an initial water level of $h(0)$ m, for example. A parameterized decay model can also be used to simulate the GI's behavior as part of a broader hydrologic simulator (e.g., US EPA SWMM [75]).

2.2.2.1 Implementation

An automated process is developed to identify individual storms in the sensor data. This methodology requires water level time data, in this case provided by our sensors. Storm events are automatically identified by marking local minima and maxima using the `find_peaks()` function of Python Scipy Signal library [102]. To find the maxima we pass the water level time series to the function, which returns a list of indices corresponding to peaks (local maxima). To find the minima, we pass the negative of the water level time series, which then returns a list of indices for local minima. We use two of the function's optional parameters to refine which points qualify as "peaks": prominence (p) and distance. Prominence is a measure of how high a local maxima stands out in comparison to its neighboring local minima. The prominence parameter

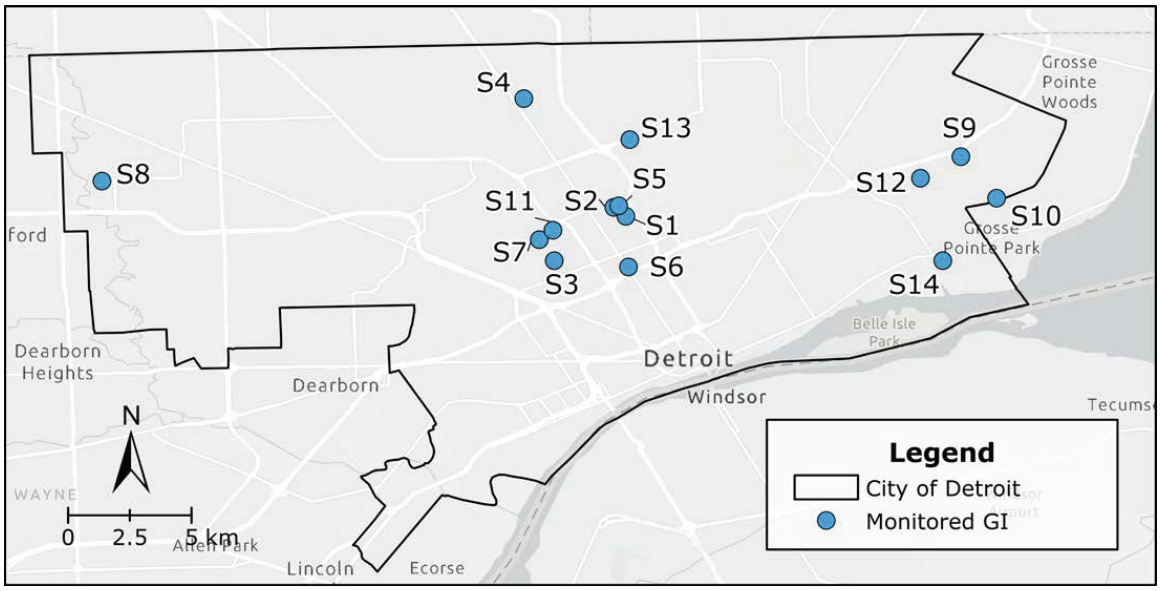

Figure 2.3: Map of the 14 GI sites selected for sensors in Detroit.

was adjusted for each site such that the selected peaks corresponded reasonably well to local rainfall measurements and captured a meaningful segment of water level drawdown for each storm. We set the distance parameter to 3 hours, meaning adjacent local minima/maxima must be at least 3 hours apart to be selected. An example of the resultant automated storm segmentation is provided in Figure 2.2, top row. While rainfall data are not required for the method, they can nonetheless be used as a secondary check, by visually lining up storms detected in the water levels with those measured by nearby rain gages.

Once the storms were isolated, the drawdown model is fit to the data using the `poly_fit()` function of Python's Numpy library [103]. The function uses least squares to fit a polynomial to the provided data. We pass $[h(t), \frac{dh}{dt}]$ to the function with the degree set to one. The function returns the α that minimizes the squared error. Figure 2.2 (rows 2–4) show these fits along with the resultant drawdown model for

three storms measured at the same site. Taken out of the differential form, the drawdown model follows $x = Ce^{\alpha t} + b$, where C and b are scaling and offset parameters that are adjusted to fit the magnitude of the storm. The coefficient of determination (R^2) and root mean squared error (RMSE) are calculated for each fit using Python's Scikit-learn library [104].

2.2.3 Case Study

We selected Detroit, Michigan, US for the GI monitoring network (latitude 42°19'53", longitude −83°2'44"). Detroit has a unique opportunity for extensive GI installations because approximately 103 km^2 (28%) of the city is classified as vacant land [105]. The city is located at the outlet of three major watersheds (i.e., Rouge River, Clinton River, Lake St. Clair) where flows eventually discharge into either Lake St. Clair or the Detroit River. Due to Detroit's location in the floodplain, most of its soil is poorly drained clay and silt [106]. Detroit also has a shallow groundwater table. Teimoori et al. [107] found that the modeled depth to groundwater in Detroit ranged from approximately 1–3 meters below the ground surface. Detroit's climate follows a four-season pattern, with average temperatures ranging from −7.11°C to 28.7°C. Detroit averages 87 cm and 137 days of precipitation per year [108]. Precipitation is dispersed relatively evenly throughout the year as rain and snow, but heavier amounts occur in spring and winter [106].

Detroit has a combined sewer system for managing stormwater and wastewater which flows into the second largest wastewater plant in the world [106]. During extreme rainfall events in 2021, the sewer conveyance and wastewater plant's treatment capacity was exceeded on multiple occasions, resulting in billions of gallons of raw sewage being directly discharged into Detroit waterways [109]. In addition, residential basements were flooded with sewage-laden runoff [109]. The need to mitigate flooding and sewer overflows has driven the City of Detroit and organizations like the Detroit

Sierra Club to prioritize GI installations [95].

In partnership with the Detroit Sierra Club, a non-profit organization, 14 GI sites were selected for deployment in summer 2021 across 155 km^2 of Detroit to monitor GI performance (Figure 2.3). Since 2015, the Detroit Sierra Club has been working with community partners and Detroit residents to build GI, primarily small residential rain gardens. GI were selected that varied in terms of age, size, and surrounding land use type. Twelve sites were rain gardens designed and built by Detroit Sierra Club and their partners, and two were engineered and commercially built bioretention cells. The design and site data for the GI were provided by Detroit Sierra Club (Figure A.4). Moving forward, each site is identified by an alpha numeric code (e.g., S1 for site 1).

2.2.4 Correlation Analysis

Once the decay constants were extracted from the Detroit sensor network, a correlation analysis was conducted to determine which design and physiographic features explain GI drawdown, as quantified by the decay constant α. Design features included the GI's location, surface area, drainage area, storage volume, soil media depth, age, and drainage area to surface area ratio (DA/SA ratio). The DA/SA ratio was calculated by dividing the drainage area by the surface area. The physiographic features for each GI were extracted from public GIS datasets of percent imperviousness, land use type, elevation, slope, native soil type (i.e., hydrologic soil group), and depth to groundwater. Appendix B provides detailed steps on how the GIS datasets were downloaded, processed, and the features were extracted for each GI.

The datasets investigated included both non-normal continuous (e.g., surface area, elevation) and ordinal (e.g., land use type, hydrologic soil group) variables. To handle both types of variables, Spearman's rank correlation coefficient was selected for the correlation analysis [110]. Spearman's rank correlation coefficient is a nonparametric measure of the strength and direction of the monotonic relationship between two

ranked variables [111].

Spearman's rank correlation coefficients were computed using the `corr()` function of Python's Pandas library [112]. A dataframe of the mean decay constants, physiographic features, and design features for the GI monitoring network was passed to the function. The function requires a correlation method, which was set to `'spearman'`. Readers are directed to a Zenodo web portal to freely obtain the data and code referenced in this paper [113].

2.3 Results

2.3.1 Sensor Network Performance

Deployment of the GI monitoring network began mid-June 2021 and 14 operational sensors were deployed by early July 2021 (installation dates provided in Table A.1). The measurement period consists of data collected between June 15, 2021, and September 1, 2021. During the measurement period, there were only two instances of prolonged data loss— S8 and S12 had a two-hour and 24-hour data gap, respectively. These losses did not impact the measurement of storm response at either site. Sensor drift was not an issue, with an average drift of < 2.5 cm. There was one maintenance trip on August 11th to swap S12's sensor because it indicated the GI was empty during periods of rain (Table A.1).

2.3.2 GI Drawdown Analysis

The measurement period coincided with Detroit's 7th wettest summer on record, which included several historic rain events: 15.2 cm of rain on June 25th, 5.6 cm on July 16th, and 6.9 cm on August 12th [114]. During the measurement period, a total of 122 storms were identified across the network (orange boxes in Figure 2.4 (left)). Of the 122 storms, 15 storms were excluded as outliers from the analysis due to poor

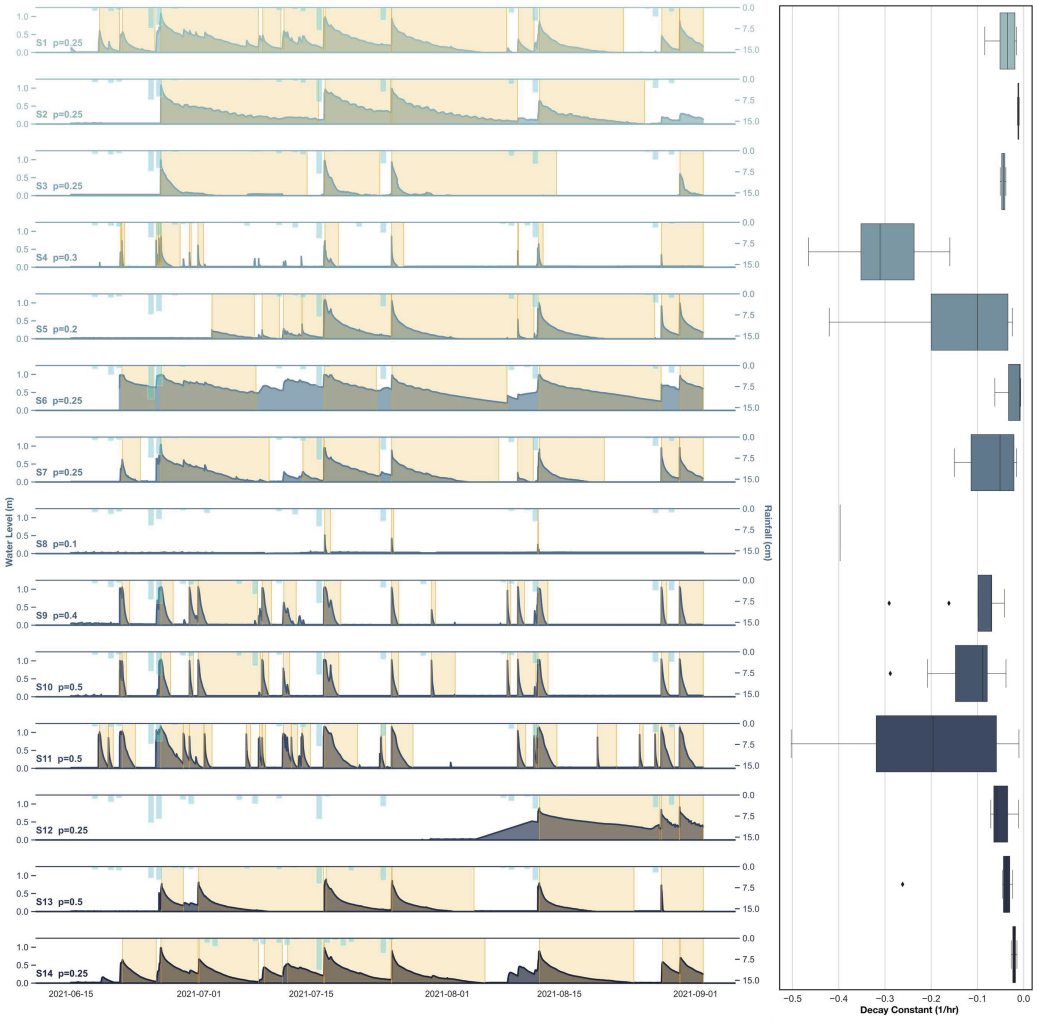

Figure 2.4: (left) Water level (m) measured across all sites on the left y-axis with rainfall (cm) on the right y-axis. Storm events are highlighted by the orange boxes. Prominence (p), the minimum increase in water level needed for a storm event to be considered distinct, is labeled for each site. (right) A boxplot showing the variance in each GI's decay constants measured for all highlighted storms.

fit of the drawdown model (negative R^2). A mean of 7.4 storm events were analyzed for each site with the number of distinct storm events varying widely per site: 21 for S11 versus 1 for S8. The variation in the number of storms captured by site is due to both the installation date (Table A.1) and the spatial variation in rainfall [115].

Site	No. Storms Analyzed	α (mean)	RMSE (mean)	R^2 (mean)
S1	11/11	−0.040	5.159	0.834
S2	3/3	−0.011	6.306	0.875
S3	4/4	−0.044	4.776	0.885
S4	9/12	−0.305	9.109	0.524
S5	9/9	−0.146	4.611	0.727
S6	5/6	−0.024	3.420	0.916
S7	9/9	−0.069	6.088	0.802
S8	1/3	−0.397	2.998	0.922
S9	9/12	−0.102	15.964	0.606
S10	11/12	−0.119	13.744	0.697
S11	21/24	−0.200	12.209	0.738
S12	3/3	−0.047	4.531	0.806
S13	6/6	−0.072	3.777	0.921
S14	7/8	−0.021	6.630	0.637

Table 2.1: The results from fitting the drawdown model to the storms captured by the GI monitoring network. We report the mean decay constant α for each GI and how well the decay constant α fit the sensor data as measured by RMSE and R^2.

The mean fit of the drawdown model to the sensor data was $R^2 = 0.746 \pm 0.111$ and RMSE = 8.579 ± 4.168. The fitted decay constant α varied by storm and by GI (Figure 2.4 (right)). Across all storms and sites, the mean decay constant α and standard deviation was -0.119 ± 0.124 hr^{-1}. The average decay constant per site varied by two orders of magnitude, from -0.011 hr^{-1} (S2) to -0.397 hr^{-1} (S8). The number of storms identified versus analyzed, as well as the mean decay constant α, RMSE, and R^2 for each GI is provided in Table 2.1.

The decay constant α corresponds with the GI's drainage dynamics. During the measurement period, most GI completely drained between storm events (S4, S8–S11), providing full storage for the next storm event (Figure 2.4 (left)). S2, S6, and S12 always had some water present in their soil media, limiting the amount of storage for each subsequent storm. During the measurement period, most sites experienced ponding (water level $>$ 1 m). However, ponding did not exceed 12 hours for most sites (11 of 14 sites). S6, S11, and S9 experienced extended periods of ponding during the June 25th storm for 22, 29, and 21 hours, respectively. Sites S6 and S11 also experienced extended ponding for approximately 24 hours during the July 16th storm, and S11 ponded for about 16 hours during the August 12th storm.

2.3.3 Correlation Analysis

Spearman's rank correlation coefficients between the GI design features and the decay constants ranged from 0.01 (site age) to 0.34 (DA/SA ratio) (Figure 2.5). The decay constants were most correlated with the DA/SA ratio (0.34) and drainage area (0.23). Drainage area and DA/SA ratio were highly correlated with each other (0.92); therefore, we focus analysis on the DA/SA ratio. The sites with the largest DA/SA ratios had the smallest magnitude decay constants (i.e., drained the slowest). Soil media depth, storage volume, surface area, and age had limited impact on the decay constants (0.16, -0.09, 0.06, and 0.01, respectively).

The correlation coefficients between the physiographic features and the decay constants ranged from -0.02 (slope) to -0.64 (groundwater depth) (Figure 2.5). The decay constants were most correlated with groundwater depth (-0.64), latitude (-0.56), imperviousness (0.43), and longitude (0.37). The closer groundwater was to the surface, the slower the site drained (i.e., the smaller the decay constant's magnitude). Groundwater is also highly correlated with latitude (0.98), which explains the correlation between latitude and the decay constants. Longitude, however, is not correlated

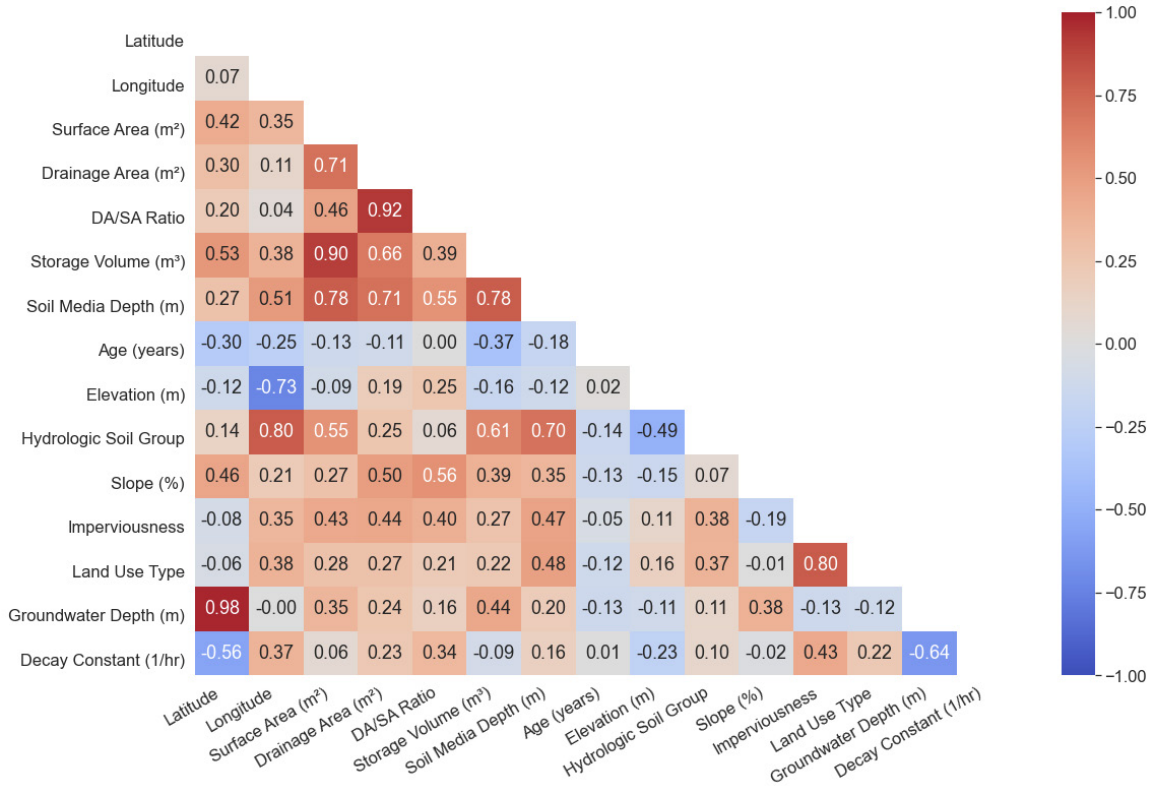

Figure 2.5: Spearman's rank order correlation coefficients for the decay constants, design features, and physiographic features.

with groundwater but still has a positive correlation with the decay constants. The decay constants' magnitude decreases for sites further away from the western border towards central Detroit, where the smallest magnitude decay constants are, increasing again towards the eastern border. In terms of imperviousness, the greater the imperviousness, the smaller the decay constant's magnitude. This was not always the case, however. For example, S1 and S12 are 53 and 52% impervious and their mean α's are -0.040 and -0.047 hr^{-1}, respectively, while S9 is 92% imperviousness with a mean α of -0.102 hr^{-1}. The remaining physiographic features are either highly

correlated with the explanatory variables discussed above (elevation and longitude: −0.73; land use type and imperviousness: 0.80) or are minimally correlated with the decay constants (hydrologic soil group: 0.10; slope: −0.02).

The relationship between the decay constant and its most correlated design feature, DA/SA ratio, and physiographic feature, groundwater depth, was explored further. We show groundwater depth versus DA/SA ratio for estimated decay constants in Figure 2.6a. Given that decay constants were retrieved for individual sites and individual storms, the figure reflects averaged surface fit across all the observations. The shape of Figure 2.6a is bounded by the observations made by the sensor network and was not extrapolated beyond those bounds. The colored contours indicate the expected decay constant based on the combination of groundwater depth and DA/SA ratio. The red contours indicate slower drawdown while the blue/grey contours indicate faster drawdown. To frame the interpretation of the figure, the corresponding drawdown rates are also color coded in (Figure 2.6b).

In our study, decay constants with magnitudes ≥ -0.20 hr^{-1} result in the drainage of one meter of water in under 24 hours (Figure 2.6b). Figure 2.6a shows there are various combinations of groundwater depth and DA/SA ratio that achieve this performance metric. On one end of the spectrum, groundwater can be as shallow as 7.5 m if it has a small DA/SA ratio of 1–2. On the other end of the spectrum, groundwater must be at least 10 m deep with a DA/SA ratio no larger than 8. Furthermore, if the groundwater table is < 7.5 m, a slower drawdown rate is observed regardless of the DA/SA ratio (bottom edge of Figure 2.6a). Similarly, when the DA/SA ratio is > 8, the drawdown rate is slow regardless of the groundwater depth (right edge Figure 2.6a).

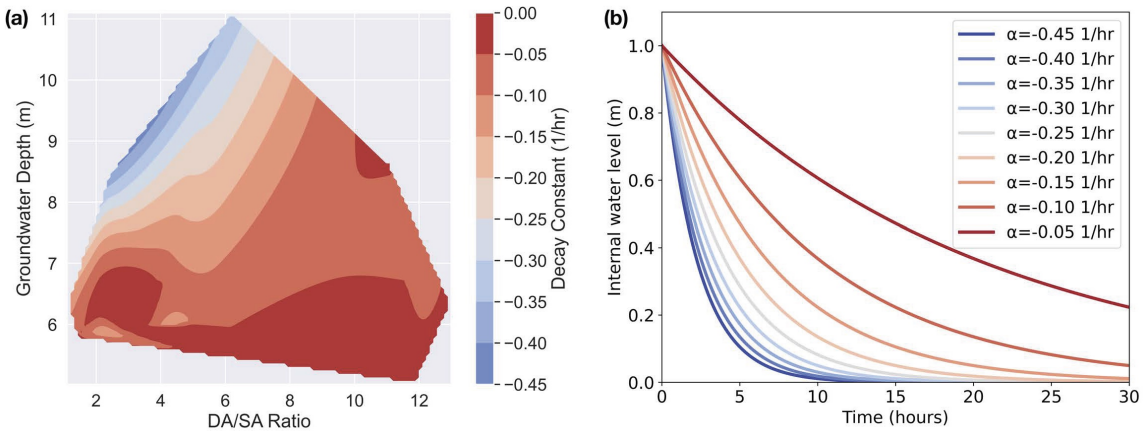

Figure 2.6: (a) A surface fit of the calculated decay constants (hr^{-1}) based on groundwater depth (m) (y-axis) and DA/SA ratio (x-axis). (b) The drawdown model curves for the range of decay constants found in (a). Blue indicates faster drawdown rates while red indicates slower rates.

2.4 Discussion

2.4.1 GI Drawdown Dynamics

The data toolchain introduced in this paper provides an automated way to analyze high resolution hydrologic data, such as water levels in GI. This is enabled by the storm segmentation methodology, which automatically extracts and analyzes data from individual storms. As sensor networks scale, manual data analysis will become infeasible, demanding that we discover means by which to automatically extract relevant data for analysis or training of machine learning algorithms becomes infeasible. As demonstrated here, the approach automatically identified storm events and subsequently analyzed them to train models for the decay constants. The application of a peak-find algorithm to extract events from other types of data (flows, rainfall, soil moisture, etc) should be explored in future studies.

The water levels from the 14 sensors indicate the GI are generally performing as designed, despite record rainfall. The GI met and exceeded the requirement specified by Detroit's GI design manual that ponding time should not exceed 24 hours [44]. Below the ground surface, the performance varied by site and storm. To completely drain 1 m of water in 24 hours a GI must have a decay constant ≥ -0.2 hr^{-1} (Figure 2.6b). Only 2 of the 14 gardens had an average decay constant above this threshold. Therefore, most sites have restricted storage capacity when they experience consecutive storms.

Fitting a drawdown model for each storm and each site resulted in variability across decay constant estimates. Statistical uncertainty is inherent in a study of this scale, and may manifest across measurements, deployment consistency, and model assumptions. Some variability in the decay constants was likely due in part to the spatial and temporal variation in rainfall [115]. The decay constants may also have been impacted by changes in GI conditions such as the swelling and shrinking of the soil media following wet and dry periods, and the creation of preferential flow paths after extended dry periods [38].

Naturally, a highly granular and continuous sensor dataset can be expected to reveal dynamics and nonlinearities that are not apparent in single measurements or short-term experimental campaigns. We contend that the use of the decay constant poses a first step in the analysis of this large dataset and provides an initial balance by enabling a metric for cross-site comparisons without compressing large amounts of sensor data into an over simplistic summary that ignores dynamics entirely. Future studies could explore the nuanced variabilities dynamics more explicitly.

Cross-site comparisons of water level dynamics revealed patterns driven by site design and physiographic features. It is difficult to directly attribute the variation seen between sites to the variations in these features due to the complexity of the physical processes that govern GI drainage dynamics. The correlation analysis found broadly, however, that GI with DA/SA ratios smaller than 8 have faster drawdown

rates. Therefore, when designing GI, the size of the garden in relation to the size of the drainage area is critically important. These results align with Davis [116], which found that a large cell media volume to drainage area ratio and drainage configurations were the two most dominant factors that improved GI performance.

Across the broader landscape, GI drawdown dynamics were highly correlated with two physiographic features: groundwater depth and longitude. Faster drawdown rates were correlated with a deeper groundwater table and locations on the outskirts of Detroit. This illustrates the importance of evaluating groundwater levels when planning urban GI installations, especially since many urban areas have shallow groundwater tables [117], including Detroit [107]. The correlation with longitude may be explained by prolonged soil compaction from development in central Detroit [118].

Some physiographic features had low correlation with the decay constants. Detroit is relatively flat, which may explain the low correlation with elevation and slope. The low correlation between the decay constants and the hydrologic soil group of the surrounding soil is more difficult to posit. Our physiographic input data were limited to public datasets, whose accuracy is driven by factors outside of the control of this study. The low spatial resolution of publicly available raster datasets may oversimplify the physiographic features at a GI site. In the future, site surveys may provide better data for analyzing these physiographic features interaction with the decay constants.

Our results have several implications for the future of stormwater management. Considering the broader urban drainage landscape and the potential impact of physiographic features on GI drawdown rates, measurements should become a core component of how managers choose to invest in GI. For example, measuring the drawdown rate, groundwater depth, and/or soil compaction at a site before installation could reduce the risk of installing GI in locations that will have impeded drainage regardless of how well they are engineered. Beyond single sites, an investment into an entire measurement network may help support a more targeted and data-driven approach to GI design, placement, and maintenance. The application of this methodology could

result in empirical design guidance, such as an empirical "heatmap", as shown in Figure 2.6a. Such illustrations could serve as a field-validated guide for managers who want to push the performance of their infrastructure without focusing all of their limited resources into one particular design or locale. Naturally, this would require the collection and analysis of more data, but the increasing reliability of technology and automation afforded by some of the tools in this paper may reduce the barrier to adoption.

One potential limitation of this work is the duration of our study period. Over longer periods of time we would expect to see fluctuations in the decay constants due to seasonal conditions (e.g., the rate of evapotranspiration falling during colder months [119]) and due to longer-term trends (e.g., deterioration of the GI's drainage capacity due to clogging [31]). In future work, how the decay constants vary over time should be investigated to determine these seasonal and long-term changes. The reliability of the sensors should enable long-term data collection with reduced measurement overhead.

2.4.2 Beyond Site-Level Drawdown Dynamics

This study used the high temporal and spatial resolution dataset produced by a sensor network to provide a first order analysis of the variability in GI drawdown dynamics, but the sensor network could also be used for a variety of other purposes. Large GI sensor networks have potential for use in long-term GI monitoring. These data can used to develop a deeper understanding of how GI installations fit into the larger urban drainage network, but this may also require the application of expanded tools for data analysis. Given the accessibility to and availability of modern Machine Learning libraries, the data collected by these networks could be used to inform predictive tools and interactive design guides. The sensor data can also be used to iterate on site design or inform maintenance schedules. Measurements showing

when drainage slows over time could indicate that the GI soil media is clogged and should be replaced. A science-based method to validate such scenarios should be investigated. These data may also be used for community education and engagement by communicating to residents and community groups how and where GI may be expected to work well.

2.5 Conclusion

This study introduces a wireless, real-time sensor for measuring GI drawdown. Networked together across Detroit, these sensors provide high temporal and spatial resolution data for analyzing city-scale urban drainage conditions. To isolate individual storms in this large dataset, we designed an automated storm segmentation methodology based on peak finding. To our knowledge, this study is the first to monitor GI at this scale and combine it with a data-driven workflow to reveal explanatory features of drawdown dynamics. In Detroit, the groundwater table, imperviousness, longitude, and DA/SA ratio are the most important features impacting drawdown rates. To confirm this finding for other regions, high resolution and long-term GI monitoring is necessary.

CHAPTER 3

Sensor Placement for Urban Drainage Networks using Gaussian Processes

Abstract

Low-cost sensors have enabled monitoring of urban drainage networks at unprecedented temporal and spatial scales. The next frontier is determining how many sensors are needed and where to place them. This study presents a comprehensive and iterative methodology for placing sensors in urban drainage networks using sensor data, publicly available GIS datasets, and Gaussian Processes (GP). The proposed methodology can be used for any urban drainage network sensor or spatial parameter of interest, has guarantees of optimality, and is easy and efficient to use. The proposed methodology is evaluated on a real-world green infrastructure (GI) sensor network in Detroit (Michigan, US). Using sensor data and GIS datasets, a GP model is trained to estimate city-scale GI drawdown rates. The model's uncertainty is then input into a mutual information-based greedy algorithm to find the optimal sensor locations in Detroit. The study's results can be used for future GI development and GI sensor placement in Detroit. Beyond Detroit, this study provides a roadmap for other communities who want to find optimal sensor placements for their urban drainage networks. The proposed methodology will improve our models and management of our urban drainage networks.

3.1 Introduction

Over the past five years, real-time infrastructure monitoring has become a standard practice as Internet of Things (IoT) technologies have matured, enabling low-cost, high-resolution sensing [57, 20, 120]. Internet-connected sensors are now measuring a wide variety of stormwater parameters, such as water levels [18], soil moisture [92], and pollutants [121]. With these advancements, the question of *where* to place sensors becomes critically important. Several studies have investigated sensor placement for urban drainage networks [62–69], however, they do not offer a methodology that is flexible enough to be applied to any sensor and/or spatial parameter of interest. Therefore, this study seeks to answer the following question: *where should sensors be placed in urban drainage networks to characterize the parameter of interest using the minimum number of sites?*

To that end, we propose a sensor placement methodology for urban drainage networks using Gaussian Processes (GPs). The methodology consists of training a GP regression model using a sensor network and publicly available GIS datasets. The GP model's uncertainty is then used along with a mutual information-based greedy algorithm to optimize sensor placement. To our knowledge, this is the first from-the-ground-up study that seeks to fuse sensors with regional datasets and machine learning to optimally place sensors in an urban drainage network.

3.1.1 Motivating example

The city of Detroit (Michigan, US) struggles to manage its stormwater for a variety of reasons. Located at the outlet of three major watersheds, Detroit receives significant volumes of stormwater. The stormwater is hard to infiltrate because the city has poorly drained clay and silt soils [106], and has a shallow groundwater table [107]. These factors are compounded by increasing precipitation, aging infrastructure, and rising surface water levels [122]. In response, Detroit has been installing green

infrastructure (GI) [123, 122], a nature-based stormwater practice [23].

To investigate Detroit's GI, wireless sensors were deployed in 14 of 144 rain gardens and bioretention cells in summer 2021 (Figure 3.1b) [124]. The sensor's form factor is that of a water well, consisting of a 1.5 m long slotted PVC pipe with the sensing device (pressure transducer) at one end and additional hardware components (microcontroller, cellular modem, lithium-ion battery, solar panel) at the other end (Figure 3.1a). The sensor measures real-time water level fluctuations inside GI with the goal of better understanding how these sites capture and drain water.

The water level measurements can be used to compute the GI's drawdown rate. Drawdown rate is the speed at which water is captured by evapotranspiration and exfiltration into native soils [37]. As a proxy for GI performance, drawdown rate is a means to understand the role of GI in the broader drainage landscape. A drawdown model was parameterized for each site using the water level dataset [124]. The model approximates water level as a first-order linear dynamical system, which evolves according to the differential equation:

$$\frac{dh}{dt} = \alpha h \ , \ \alpha < 0 \tag{3.1}$$

where h represents the water level in the GI and the decay constant α is a measure of how fast the water level recedes. The decay constant is directly proportional to drawdown rate: the larger α's magnitude, the faster the water recedes. To compare drawdown across Detroit, α was calculated for every storm event captured by the GI sensor network during the measurement period (June 15 – September 1, 2021) [124].

These decay constants were then used to investigate which design and physiographic features improve GI drawdown. To do this, a Spearman's rank correlation analysis was conducted. The features investigated included publicly available physiographic GIS datasets and GI design data. From this correlation analysis, five features were correlated with the decay constant but not collinear: depth to groundwater (-

0.64), imperviousness (0.43), longitude (0.37), drainage area to surface area (DA/SA) ratio (0.34), and soil media depth (0.16) [124].

To gain a deeper understanding of how GI fits into Detroit's larger urban drainage network, we can use these data and results to analyze city-scale drawdown patterns. The GI sensors, however, may not be optimally placed to investigate these patterns. Therefore, we ask the following question: *how many sensors are needed and where should they be placed to characterize drawdown, as quantified by α, across Detroit?*

3.2 Methods

We propose a methodology for sensor placement in urban drainage networks that is flexible, has guarantees of optimality, and is easy and efficient to use. In Section 3.2.1, we introduce GP regression for creating city-scale predictive models. Aside from the predictions, GPs also output the uncertainty associated with those predictions. In Section 3.2.2, we explain how this uncertainty along with a mutual information-based greedy algorithm can be used to find optimal sensor placements. The methodology is designed to be iterative, which is described in Section 3.2.3. Finally, in Section 3.2.4, we describe how we will evaluate the proposed sensor placement methodology using the GI sensor network introduced in Section 3.1.1.

3.2.1 Gaussian Process Regression Model

GP regression is a nonparametric and probabilistic method for modeling nonlinear relationships [125], like those inherent in urban drainage networks [126, 127]. A GP regression model assumes the standard regression form [125]:

$$\mathbf{y} = f(\mathbf{x}) + \epsilon \tag{3.2}$$

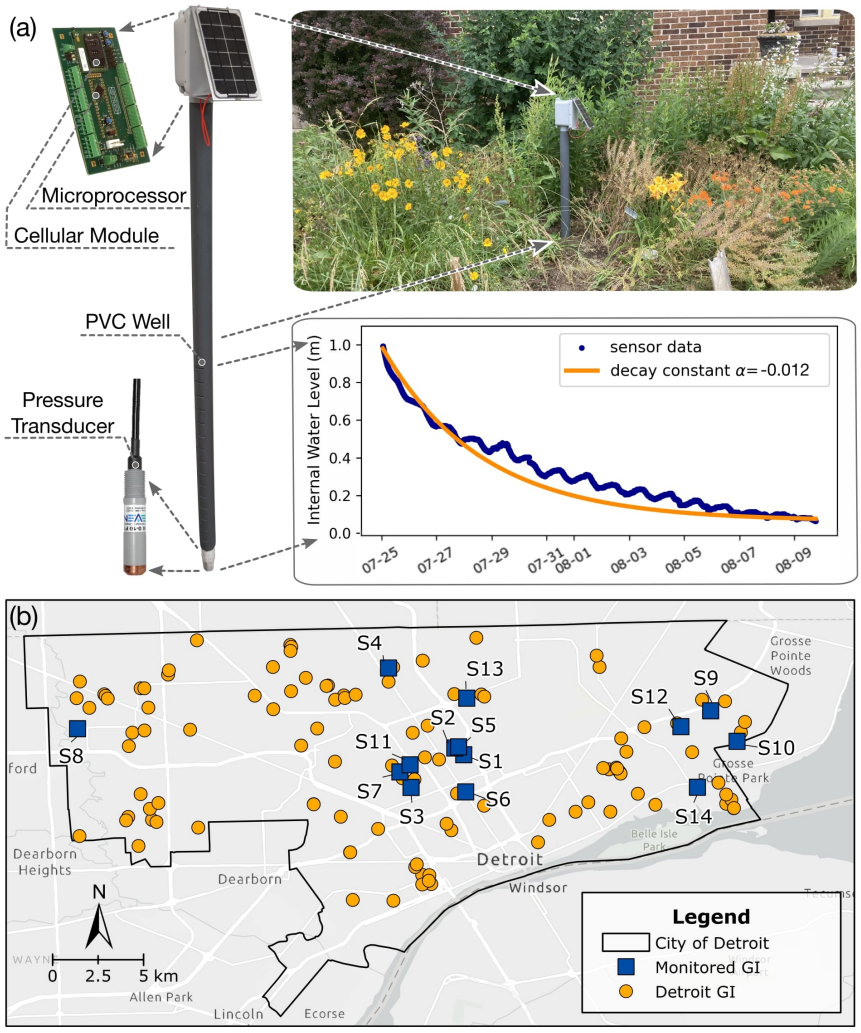

Figure 3.1: (a) The sensor and its components (left) installed in a Detroit rain garden (top right) (modified from Mason et al. [124]). The bottom right plot shows the sensor's data output, water level, and the drawdown model (decay constant $\alpha = -0.012$) for one storm. (b) Map of the sensor locations (blue squares) selected from all of the rain gardens and bioretention cells (orange circles) in Detroit.

where the measured parameter ($\mathbf{y}$) is a function of the true parameter ($f(\mathbf{x})$) plus Gaussian noise ($\epsilon \sim \mathcal{N}(\mu_n, \sigma_n)$).

Formally, a GP is a collection of Gaussian random variables, where any finite subset are also jointly Gaussian [125]:

$$P(f(\mathbf{x}_{n+1})|f(\mathbf{x}_1)f(\mathbf{x}_2)...f(\mathbf{x}_N)) \sim \mathcal{N}(\mu, \mathbf{K}). \tag{3.3}$$

In Eq. 3.3, N is the number of random variables, μ is their mean, and $\mathbf{K}$ is their covariance. For a sensor network, this means the joint distribution over a sensor-measured parameter at a finite number of sensor locations is also Gaussian. GPs use the jointly Gaussian property to estimate a measurement at an unmeasured location ($\mathbf{x}_{N+1}$) using the prior ($\mu_{1:N}$, $\mathbf{K}_{1:N}$). Given a set of observed measurements ($\mathbf{D}_N$: $\{\mathbf{x}_{1:N}, \mathbf{y}_{1:N}\}$), the mean function predicts the most probable value of y at $\mathbf{x}_{N+1}$, and the the covariance function, also known as the kernel, estimates the uncertainty of that prediction. In this way, a GP is completely described by it's mean and covariance functions (Eq. 3.4).

$$f(\mathbf{x}|\mathbf{D}_N) \sim GP(\mu(\mathbf{x}|\mathbf{D}_N), \mathbf{K}(\mathbf{x}|\mathbf{D}_N)). \tag{3.4}$$

For this study, we selected the squared exponential (SE) kernel (Eq. 3.5). The squared exponential kernel is commonly used in GP literature to model non-linear relationships [128], like those inherent in urban drainage networks [126, 127].

$$\mathcal{K}_{SE}(x, x') = \sigma^2 exp(-\frac{1}{2l^2}(x - x')^2) + \delta_{xx'}\sigma_n^2 \tag{3.5}$$

The kernel is controlled by its hyperparameters $\theta : \{\sigma^2, l, \sigma_n^2\}$. The lengthscale ($l$) determines the smoothness of the function, the output variance (σ^2) determines the average distance the function is away from its mean, and the noise variance (σ_n^2) is only applied when $x = x'$. The hyperparameters that best fit the observed data are

found by maximizing the log marginal likelihood (Eq. 3.6) [125], where $\mathbf{K}_{N\times N}$ is the kernel matrix and $\mathbf{I}_{N\times N}$ is the identity matrix.

$$\underset{\theta}{\operatorname{argmax}} \log p(\mathbf{y}|\mathbf{X},\theta)$$
$$where \quad \log p(\mathbf{y}|\mathbf{X},\theta) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{N}{2}\log 2\pi, \tag{3.6}$$
$$\mathbf{K}_y = \mathbf{K}_{N\times N} + \sigma_n^2\mathbf{I}_{N\times N}$$

3.2.2 Sensor Placement

The goal is to place sensors which are most informative with respect to the entire measurement space $\mathcal{V}$. A common sensor placement methodology is to add sensors where uncertainty, or entropy (H), is highest [65, 129, 130]. Eq. 3.7 quantifies this uncertainty, called conditional differential entropy, for the unobserved locations $\mathcal{V}\backslash\mathcal{A}$, referred to as $\bar{\mathcal{A}}$ for simplicity of notation, after placing sensors at locations $\mathcal{A}$.

$$H(\mathbf{X}_{\bar{\mathcal{A}}}|\mathbf{X}_{\mathcal{A}}) = -\int p(\mathbf{x}_{\bar{\mathcal{A}}},\mathbf{x}_{\mathcal{A}})\log p(\mathbf{x}_{\bar{\mathcal{A}}}|\mathbf{x}_{\mathcal{A}})d\mathbf{x}_{\bar{\mathcal{A}}}d\mathbf{x}_{\mathcal{A}} \tag{3.7}$$

The conditional differential entropy of a Gaussian random variable $\mathbf{X}_{\bar{\mathcal{A}}}$, conditioned on a set of variables $\mathbf{X}_{\mathcal{A}}$, is a monotonic function of its variance:

$$\begin{aligned} H(\mathbf{X}_{\bar{\mathcal{A}}}|\mathbf{X}_{\mathcal{A}}) &= \frac{1}{2}\log(2\pi e\sigma^2_{\mathbf{X}_{\bar{\mathcal{A}}}|\mathbf{X}_{\mathcal{A}}}) \\ &= \frac{1}{2}\log(2\pi e(\sigma^2_{\mathbf{X}_{\bar{\mathcal{A}}}} - \Sigma_{\mathbf{X}_{\bar{\mathcal{A}}}\mathbf{X}_{\mathcal{A}}}\Sigma^{-1}_{\mathbf{X}_{\bar{\mathcal{A}}}\mathbf{X}_{\bar{\mathcal{A}}}}\Sigma^T_{\mathbf{X}_{\bar{\mathcal{A}}}\mathbf{X}_{\mathcal{A}}})) \end{aligned} \tag{3.8}$$

where $\Sigma_{\mathbf{X}_{\bar{\mathcal{A}}}\mathbf{X}_{\mathcal{A}}}$ is a covariance vector. Entropy, however, has a significant drawback, which is that the selected sensors are those most uncertain about each other's measurements. Generally, this means the sensors are placed far away from each other, which tends to be along the borders of the area of interest [131]. This phenomenon results in "wasted" sensing space [132].

An alternative to entropy is mutual information [133]. Mutual information expands the concept of entropy to a set of random variables, measuring their mutual dependence. Specifically, it quantifies the amount of information obtained about one random variable by observing the others [134].

We use mutual information as our optimization criterion for sensor placement. We search for the subset of sensor locations that most significantly reduces the uncertainty about the estimates in the rest of the space. Specifically, we want to find the set of k sensors that give good predictions at all unmeasured locations, $\bar{\mathcal{A}}$, after placing sensors at locations $\mathcal{A}$. We do this by maximizing mutual information ($I(\mathbf{X}_{\mathcal{A}}; \mathbf{X}_{\bar{\mathcal{A}}})$)[132]:

$$\begin{aligned} \mathcal{A}^* &= \text{argmax}_{\mathcal{A} \subseteq \mathcal{V}:|\mathcal{A}|=\text{k}} I(\mathbf{X}_{\mathcal{A}}; \mathbf{X}_{\bar{\mathcal{A}}}) \\ &= \text{argmax}_{\mathcal{A} \subseteq \mathcal{V}:|\mathcal{A}|=\text{k}} H(\mathbf{X}_{\bar{\mathcal{A}}}) - H(\mathbf{X}_{\bar{\mathcal{A}}}|\mathbf{X}_{\mathcal{A}}) \end{aligned} \tag{3.9}$$

Solving this maximization problem directly, however, is NP hard. Fortunately, this problem can be solved near-optimally using a greedy algorithm (Figure 3.2 [132]). In each iteration, the algorithm selects the sensor that provides the maximum increase in mutual information (δ_y) and adds it to the set $\mathcal{A}$. The algorithm sequentially adds sensors until $|\mathcal{A}| = k$ or the marginal gain becomes negative. The greedy algorithm gives a $(1-1/e)$ approximation of the optimal sensor placement [132]. Unfortunately, this greedy algorithm is computationally intense for large n where $n = |\mathcal{V}|$. A lazy implementation of the greedy algorithm reduces the computational complexity from $O(kn^4)$ to $O(kn^3)$ [132]. It is this greedy lazy algorithm (Figure C.1) that we use to solve Eq. 3.9.

3.2.3 Iterative Process

The sensor placement methodology we propose is an iterative process. An initial network of sensors is deployed. The data collected is used to train a GP model. The model and the mutual information-based greedy algorithm are used to find the

Algorithm Approximation algorithm for maximizing mutual information.

Input : covariance matrix $\mathbf{K}$, number of sensors to select k, set of location to choose from $\mathcal{V}$
Output : selected sensors $\mathcal{A} \subset \mathcal{V}$
begin
  $\mathcal{A} \leftarrow \emptyset$;
  for $j = 1$ **to** k **do**
    for $y \in \bar{\mathcal{A}}$ **do** $\delta_y \leftarrow \frac{\sigma^2_{y|\mathcal{A}}}{\sigma^2_{y|\bar{\mathcal{A}}}}$;
      $y^* \leftarrow \text{argmax}_{y \in \bar{\mathcal{A}}} \delta_y$;
    $\mathcal{A} \leftarrow \mathcal{A} \cup y^*$;
end

Figure 3.2: An approximation algorithm for maximizing mutual information (modified from [132]). The inputs are the covariance matrix $\mathbf{K}$, the number sensors to select k, and the set of locations to choose from $\mathcal{V}$. The set of selected sensors $\mathcal{A}$ is the output. For each possible sensor location y, the algorithm calculates the marginal mutual information gain δ_y, the conditional variance $\sigma^2_{y|\mathcal{A}} = \sigma^2_y - \Sigma_{y\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\Sigma^T_{y\mathcal{A}}$, and the conditional variance $\sigma^2_{y|\bar{\mathcal{A}}} = \sigma^2_y - \Sigma_{y\bar{\mathcal{A}}}\Sigma^{-1}_{\bar{\mathcal{A}}\bar{\mathcal{A}}}\Sigma^T_{y\bar{\mathcal{A}}}$. Note $\Sigma_{y\mathcal{A}}$ is a covariance vector with one entry for each location in $\mathcal{A}$.

optimal sensor locations. These locations are then used to augment the network, which includes removing redundant sensors from the original network and adding sensors at the newly identified locations. Once data is obtained from the augmented network, the GP model is updated and the process is repeated.

3.2.4 Evaluation

3.2.4.1 GP model

A GP regression model was trained to estimate drawdown dynamics, as quantified by α, across Detroit. The GP's input features were identified by a Spearman's rank correlation analysis as described in Section 3.1.1. The input features included depth to groundwater, imperviousness, longitude, DA/SA ratio, and soil media depth. Since

soil media depth and DA/SA ratio are design features, they had to be estimated. To do this, we used the average soil media depth (0.3 m) and average DA/SA ratio (4.0) from the GI sensor network. We denote these input features as $\mathbf{X}_S$, a 107×5 matrix. The GP's output $\mathbf{y}$, is the decay constant α where $|y| = 107$.

The GP was implemented using GPy, a Guassian Process framework for Python [135]. The 'tnc' solver was used to optimize the GP's hyperparameters. The GP model provides the option to add optimization restarts to help ensure an optimal solution is found. To do this, GPy randomly restarts the model optimization and sets the model to the best seen solution. The `optimize_restarts` parameter was set to ten. The goodness of fit was analyzed using the log marginal likelihood, which was computed using GPy.

Once the GP model was trained, it was used to estimate drawdown dynamics across Detroit. We denote these new input features for Detroit as $\mathbf{X}_D$, a 14701×5 matrix. $\mathbf{X}_S$ and $\mathbf{X}_D$ are summarized in Table C.1 and Table C.2.

We also investigated the correlation between each sensor and the rest of the prediction space. To do this, the GP_{og} model's covariance matrix was converted to a correlation matrix. Then the column associated with each sensor was displayed on a map to visualize the correlations.

3.2.4.2 Sensor placement

We evaluated sensor placement for two cases. We first determined which sensors were redundant and should be removed. We then investigated where future sensors should be installed considering all possible GI in Detroit. To find the optimal locations, we used MATLAB's (ver. R2022b) Submodular Function Optimization (SFO) toolbox [136]. The algorithm, `sfo_greedy_lazy()`, required two inputs: `sigma`, the covariance matrix; and `V_sigma = 1:size(sigma,1)`, the ground set of possible sensor locations. The algorithm's output includes `A`, the set of selected locations, and `scores`, the mutual information gained by each location.

To determine the redundant sensors in the initial network, the covariance matrix ($\mathbf{K}_{14\times14}$) was computed by passing the GI sensor network input features ($\mathbf{X}_S$) into the squared exponential kernel. The covariance matrix and the ground set ($|\mathcal{V}_S| = 14$) were then input into the greedy lazy algorithm. The algorithm-selected sensors ($|\mathcal{A}_S| = k$) were used to train a new GP (GP_{opt_k}). The log marginal likelihood of GP_{opt_k} was compared to the original GP (GP_{og}).

GP_{opt_k} was also compared to GPs created from randomly-selected sensors. To do this, all of the possible sensor combinations were computed: $\binom{14}{1}$, $\binom{14}{2}$,..., $\binom{14}{k}$. From each of these sets, 250 combinations were randomly chosen and used as training sets for 250 new GPs. Root mean squared error (RMSE) was used to compare GP_{opt_k} and the randomly-selected GP models (GP_{rand_k}). To do this, each model was tested on the remaining sensors that were not used for training. RMSE was calculated using Python's Scikit-learn library [104].

To determine where to place future sensors, we combined the initial GI sensor network dataset with two new GI datasets. The Detroit Sierra Club provided data for their remaining unmeasured rain gardens and bioretention cells in Detroit. Additional rain gardens and bioretention cells were obtained online [137]. The input features $\mathbf{X}_D$ were extrapolated from publicly available GIS datasets following the methodology outlined in Mason et al. [124]. The input features $\mathbf{X}_D$ were passed to the squared exponential kernel to obtain the covariance matrix. The covariance matrix ($\mathbf{K}_{144\times144}$) and the ground set $|\mathcal{V}_D| = 144$ were then passed to the greedy lazy algorithm. The algorithm's optimal sensor locations ($\mathcal{A}_D$) are presented but cannot be evaluated because we do not yet have sensor-measured decay constants for most of these locations.

Readers are directed to a Zenodo web portal to freely obtain the data and code reference in this paper [138].

3.3 Results and Discussion

3.3.1 Gaussian Process Regression Model

The negative log likelihood of the GP_{og} model was -88.58. The model's optimal hyperparameters are provided in Table 3.1. The smaller a feature's lengthscale, the more sensitive the model is to that feature. The GP_{og} model was most sensitive to groundwater depth and DA/SA ratio, and least sensitive to imperviousness.

Hyperparameter	GP_{og}	GP_{opt}
$l_{groundwater\ depth}$	11.76	12.30
$l_{DA/SA\ ratio}$	14.74	273.63
$l_{longitude}$	777.01	402.79
$l_{soil\ media\ depth}$	3120	960.14
$l_{imperviousness}$	42,249	721,765
σ^2	0.032	0.041
σ_N^2	0.009	0.005

Table 3.1: The optimal hyperparameters for the GP_{og} and the GP_{opt} models.

Detroit's mean predicted decay constant was -0.160 ± 0.083 hr^{-1}. Figure 3.3 (left) maps the predicted mean decay constants across Detroit. The map shows slower drainage in Central Detroit, along the eastern border, and a small portion of the western border. Faster drainage occurs along the northern border and southwestern tip. The large-scale trend of poor predicted drainage near the river and better drainage towards the periphery of the city generally follows the groundwater depths across Detroit (Figure B.3).

The mean variance across Detroit was -0.013 ± 0.002 hr^{-2}. Figure 3.3 (right) maps the variance associated with the mean decay constant predictions. The map shows the areas of highest uncertainty are along the northern border and in central Detroit. This result is expected since the GP is most sensitive to groundwater level

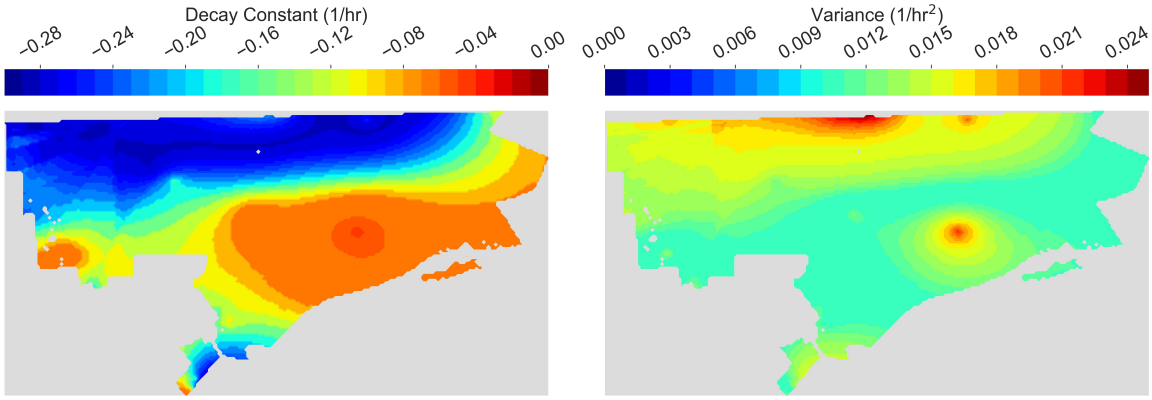

Figure 3.3: Maps of the predicted mean decay constant α across Detroit (left) and the uncertainty associated with those predictions (right) from the GP_{og} model.

and the shallowest and deepest groundwater levels are located in central Detroit and along the northern border, respectively (Figure B.3). To reduce the uncertainty of of our GP model, more sensor data is needed for these regions.

Overall, the GP methodology worked well for modeling GI drawdown because spatial phenomena, like drawdown, are well described by GPs [132]. The methodology was also easy and efficient to use. The GP was computationally fast to train (9.9 sec) and to estimate drawdown across Detroit (0.1 sec) using a 2018 MacBook Pro (Processor: 2.2 Ghz 6-Core Intel Core i7; Memory: 16 GB 2400 MHz DDR4). The input features were easily obtained online. These publicly available GIS datasets are the most realistic option for building large-scale predictive models since surveying an entire city is impractical. However, some site surveying may be necessary to check these datasets because of their inherent errors and/or uncertainties.

The mapping tool introduced can aid engineers and city planners in scientifically determining where future GI development should be prioritized. The predicted drawdown map indicates where in the city faster drainage can be achieved and where it is expected to be dampened by subpar site conditions. Comparing the locations

of existing GI (Figure 3.1) with the map of predicted drawdown (Figure 3.3 (left)), GI are generally absent from the fastest drawdown areas. Therefore, Detroit should consider focusing future GI installations along the northern border and southwestern tip to maximize drawdown performance. This methodology offers a novel method for data-driven decision making and strategic placement of GI development.

While we focused on GI drawdown, GPs are transferable to other urban drainage network sensors or parameters of interest. For example, if the existing datasets were supplemented with rain data, a GP model could be developed to estimate volume reduction. Aside from estimating a single output, GPs can also estimate multiple outputs. If we have several outputs we would like to estimate, we would add an additional covariance function (kernel) that specifies the covariance between the outputs. Combining this output kernel with our input kernel, we would then use this new input-output kernel to estimates our outputs. For example, we could create a predictive model of both drawdown and GI stormwater treatment if a water quality sensor was added to the GI sensor.

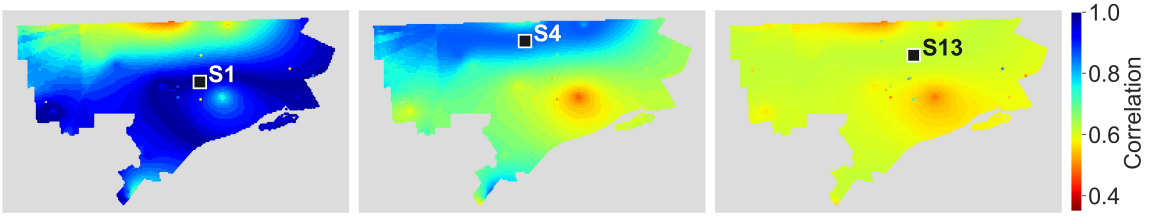

Figure 3.4: The correlation between S1 (left), S4 (center), and S13 (right) with the other sensors and the entire prediction space.

3.3.2 Sensor Placement

Figure 3.4 shows the correlation between three sensors (S1, S4, S13) and the rest of the prediction space (other sensors provided in Figure C.2). The correlation map for

each sensor is unique. S1 and S13 are both correlated with most of Detroit except the northern border. S13, however, is less correlated overall than S1 and has an uncorrelated region in central Detroit. S4 is also uncorrelated with central Detroit but highly correlated with the northern border. These results show that sensors are not necessarily most correlated with their nearest neighbors, underscoring the challenge of sensor placement, and why a sensor placement strategy is critical.

The lazy greedy algorithm found the optimal number of sensors needed to predict drawdown across Detroit was $k = 6$. In order of selection, the algorithm-selected sensors ($\mathcal{A}_S$) were S14, S12, S10, S9, S4, and S6. Four of these sensors (S9, S10, S12, S14) are clustered together on the eastern border, one sensor is located near the northern border (S4), and the final sensor is located in central Detroit (S6). Figure 3.5 (row 1) shows the mutual information gained by each sensor. Adding additional sensors after S6 resulted in a negative marginal gain.

The negative log likelihood of the GP_{opt_6} model was -46.24, a slightly poorer fit than GP_{opt}. The hyperparameters of GP_{opt_6} are provided in Table 3.1. Similar to GP_{og}, GP_{opt_6} was most sensitive to groundwater depth and DA/SA ratio and least sensitive to imperviousness.

Figure 3.5 (row 2) shows the new maps of predicted mean decay constants across Detroit. The general trend in GP_{og} is also found in GP_{opt_6} and a similar range of decay constants were predicted by both models. The minimum and maximum decay constants were -0.280,-0.045 and -0.293,-0.021 hr^{-1} for GP_{og} and GP_{opt_6}, respectively. The mean decay constant and standard deviation was -0.160 ± 0.083 and -0.132 ± 0.082 hr^{-1} for GP_{og} and GP_{opt_6}, respectively. The GP_{opt_6} model replicated the ranges and patterns of GP_{og} using fewer sensors. Therefore, the eight unselected sensors (S1, S2, S3, S5, S7, S8, S11, and S13) can be removed from the network or moved to other Detroit GI. These results suggest that sensor networks can be built smaller without sacrificing its predictive power.

The uncertainty associated with the mean decay constant predictions are shown

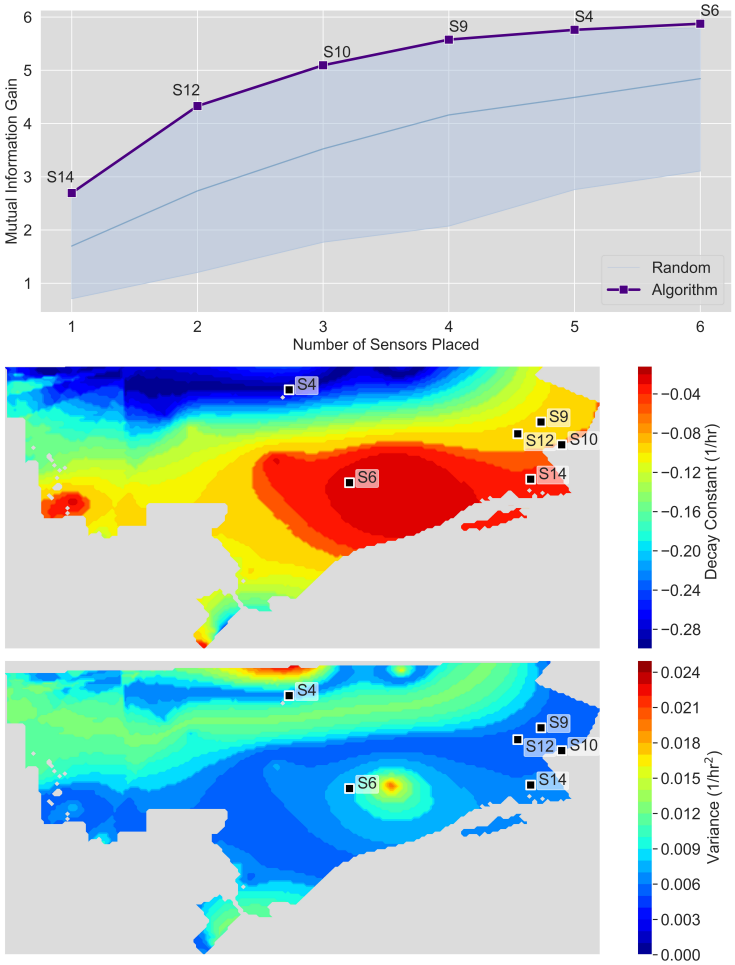

Figure 3.5: As k increases from $k = 1$ to $k = 6$, the mutual information increases for both GP_{opt_k} and GP_{rand_k} (top row). The algorithm-selected sensors, however, outperform the randomly-selected sensors for every value of k. Maps of the predicted decay constants (middle) and the uncertainty associated with those predictions (bottom) from GP_{opt_6}. The algorithm-selected sensors,$\mathcal{A}_S$, are plotted on top of the predicted decay constants.

in Figure 3.5 (row 3). The bottom half of Detroit as well as a thin band in northern Detroit were the areas of highest uncertainty (colored blue). Those regions in the GP_{og}'s uncertainty map are teal/green, indicating higher levels of uncertainty (Figure 3.3 (right)). While the the northern border and central Detroit had the highest uncertainty (colored yellow to red) in both models, GP_{opt_6} was more certain in these regions. Overall, GP_{opt_6} was more accurate than GP_{og}, with a mean variance and standard deviation of 0.008 ± 0.003 hr^{-2} and 0.013 ± 0.002, respectively. These results illustrate that uncertainty can be reduced if the near-optimal sensor configuration is used to create the predictive model.

Figure 3.5 (row 1) compares the mutual information gained from $\mathcal{A}_S$ to those randomly-selected ($\mathcal{A}_{rand_k}$). For $k = 1$ to $k = 6$, the range of mutual information gained (i.e., minimum, mean, maximum) is plotted for the randomly-selected locations. $\mathcal{A}_S$ outperformed $\mathcal{A}_{rand_k}$ for every value of k. Similarly, Table 3.2 provides the RMSE from the test set for GP_{opt_k} and GP_{rand_k}. While GP_{opt_k}'s and GP_{rand_k}'s mean RMSE were similar for all values of k, GP_{opt_k}'s RMSE was always lower than GP_{rand_k}'s maximum RMSE. These results demonstrate that GP_{opt_k} outperformed GP_{rand_k} for all values of k, and the proposed methodology selects more optimal sensor locations than a random methodology.

Sensors (k)	GP_{opt_k} **RMSE**	GP_{rand_k} **RMSE**		
		min	**average**	**max**
1	0.176	0.145	0.170	0.180
2	0.179	0.108	0.166	0.185
3	0.165	0.107	0.160	0.192
4	0.187	0.090	0.152	0.220
5	0.142	0.093	0.144	0.204
6	0.142	0.085	0.138	0.201

Table 3.2: For each value of k, the RMSE from the test set for GP_{opt_k} and the RMSE range (i.e., minimum, average, and maximum) for GP_{rand_k}.

Beyond the original network, the algorithm selected 60 of the 144 GI sites ($\mathcal{A}_D$) for future sensor deployments (Figure 3.6 (left)). The locations are grouped in sets of ten and color-coded by relative mutual information gain (Figure 3.6 (right)). The locations that result in the greatest mutual information gain (dark blue), should be installed first while those that result in minimal gain (red), should be installed last. Significant mutual information gains can be obtained by the first 40 locations and then the gains only marginally increase. The algorithm selected seven sensors from the original GI sensor network (S12, S8, S3, S9, S13, S4, S7). Interestingly, only S12, S9, and S4 were selected for both $\mathcal{A}_S$ and $\mathcal{A}_D$.

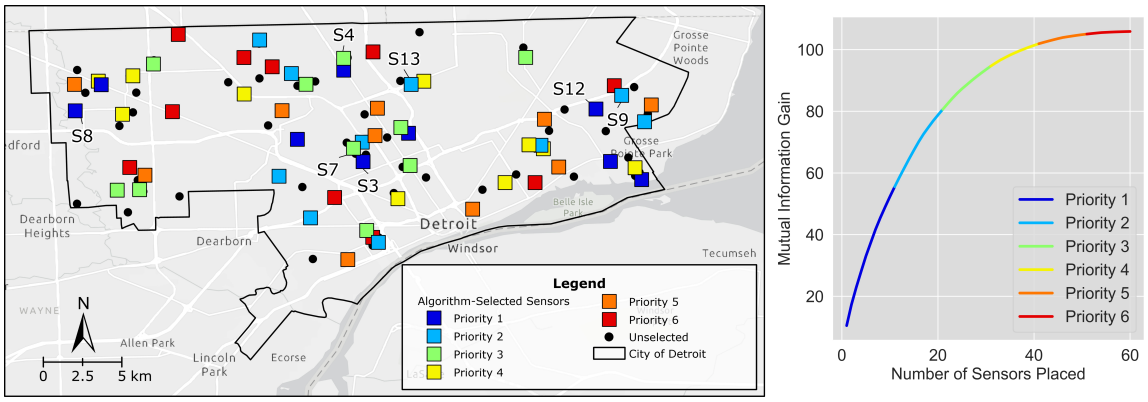

Figure 3.6: The 60 locations selected by the algorithm, color-coded by priority (left). The mutual information gain as k increases from $k = 1$ to $k = 60$, also color-coded by priority. The first 40 locations result in significant increases in mutual information and then the gains marginally increase.

To investigate why the algorithm selected different sensors from the original network for $\mathcal{A}_S$ and $\mathcal{A}_D$, we compared the normalized histograms of the input dataset, X_D, with the subset for the algorithm-selected locations, $X_{\mathcal{A}_D}$ (Figure 3.7). The histograms show the distributions of $X_{\mathcal{A}_D}$ align with the distributions of X_D. The algorithm, therefore, selects locations so that these two distributions align. In this

way, communities who do not have access to GPs could use the histogram method as a "rule of thumb" to determine where to place sensors.

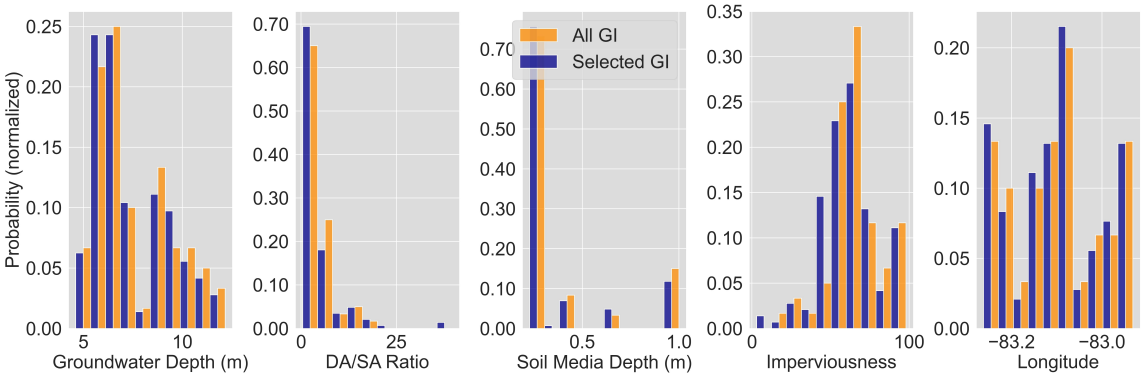

Figure 3.7: Normalized histogram of the GP features for all Detroit locations (blue) and those of the algorithm-selected locations (orange).

$\mathcal{A}_D$ could not be evaluated in the same way as $\mathcal{A}_S$ because we did not have measured decay constants for most of these sites. Future work will evaluate these sites after the next set of sensors are deployed. New sensors will result in new decay constants which can then be used to retrain the GP model, improving it's predictive abilities. In this way, the proposed methodology is iterative: install sensors, train the GP model, find optimal sensor placements, and repeat.

Using mutual information, the proposed sensor placement methodology is flexible for any urban drainage network sensor or spatial parameter. The method selects the sensors that result in the most mutual information gain. In this way, sensor networks can be made smaller without losing predictive abilities as shown by GP_{opt_6}. As an iterative process, the model's accuracy can be continually improved as sensor placement is refined. The algorithm used to select the sensors has an optimally guarantee $(1-1/e)$. In addition, the methodology did not require a calibrated physics-based model and was computationally efficient. It required 0.45 sec to find $\mathcal{A}_S$ and

0.51 sec to find $\mathcal{A}_D$ using a a 2018 MacBook Pro (Processor: 2.2 Ghz 6-Core Intel Core i7; Memory: 16 GB 2400 MHz DDR4).

3.4 Conclusions

This study presents a comprehensive sensor placement methodology for urban drainage networks. The methodology is flexible enough to work for any urban drainage network sensor or spatial parameter of interest, has guarantees of optimality, and is easy and efficient to use. To demonstrate the methodology, we used sensor data from a GI sensor network to create a predictive GP model of drawdown rates across Detroit. The GP model's uncertainty was then used to identify optimal sensor locations in Detroit. While these results are critical for future GI development and sensor deployments in Detroit, more importantly, they provide a roadmap for other communities to follow. Optimal sensor placements will improve our models and management of our urban drainage networks.

CHAPTER 4

StormReactor: An Open-Source Python Package for the Integrated Modeling of Urban Water Quality and Water Balance

Abstract

Retrofitting watersheds with sensing and control technologies promises to enable autonomous water systems, which control themselves in real-time to improve water quality. To realize this vision, there is a need to improve the degree of fidelity in the underlying representation of pollutant processes. This paper presents an open-source Python package, *StormReactor*, which integrates the Stormwater Management Model's water balance engine with a new water quality module. *StormReactor* includes a variety of predefined pollutant generation and treatment processes, while allowing users to implement additional processes on their own. To demonstrate the range of possible water quality methodologies that can be modeled, we simulated suspended solids and nitrates in a real and anonymized stormwater network. To illustrate *StormReactor*'s real-time control capabilities, a control strategy was implemented to maximize denitrification. Case study results indicate a controlled asset can achieve the same pollutant improvements as an uncontrolled asset in a quarter of the spatial footprint.

4.1 Introduction

A reliable and cost-effective method for treating stormwater pollutants is real-time control [139–141]. Retrofitting stormwater assets with sensing and control technologies enables watersheds to adapt in real-time to individual storms or pollutant loads [92, 142]. These smart stormwater assets can be coordinated at the watershed-scale to maximize pollutant treatment [83, 21, 20]. In essence, this supports the analogy of transforming our natural or urbanized watersheds into distributed treatment plants by combining knowledge from stormwater systems and process control [143]. To realize this vision, we must first be able to model both pollutant transformations and the impact of real-time control actions on water quality at the watershed scale [80, 85, 20].This can be achieved with integrated environmental modeling.

Integrated environmental modeling dynamically links distinctly separate models during run-time to better understand the environmental system's response to human and natural stressors [144, 145]. Recently, integrated environmental modeling has been used to combine climate and streamflow data with a water budget model and a dynamic groundwater model [146], simulate the hydrological effects of land use changes on karst systems [147], link precipitation forecasts with real-time hydrological and hydraulic modeling for urban flood forecasting [148], couple hydrodynamic and closed nutrient cycle ecological models to predict dissolved oxygen (DO) in surface waters [149], and create a catchment-scale water quality modeling and monitoring framework [150].

Integrated environmental modeling of stormwater requires the coupling of water quantity and quality models. This necessitates simulating a number of underlying processes, including precipitation, runoff, climatic variables, land use, flow and pollutant routing, and pollutant transformations [151–153]. While a number of existing models are able to represent these individual components effectively at a granular scale, an all-in-one modeling package is still lack-

ing. Given the complexity of stormwater, specifically its nonlinear dynamics [154, 85], most existing models understandably seem to draw a line between flow and quality [70, 71]. There has been a stated need to integrate these two types of environmental models [143, 150, 155]. To that end, the specific contributions of this paper are:

- *StormReactor*, a new water quality package implemented as an extension of the popular US Environmental Protection Agency's (EPA) Stormwater Management Model (SWMM), which provides an open-source Python programming interface for simulating complex pollutant generation, treatment, and real-time control processes.

- An evaluation of the package's ability to model complex pollutant transformations and real-time control actions using two case studies.

These contributions provide researchers and practitioners more flexibility in simulating water quality processes and pollutant-based real-time control at site and watershed scales.

4.2 New Package for Modeling Stormwater Quality

A watershed-scale pollutant transformation model is comprised of the water quantity and water quality representations of the stormwater network (Figure 4.1). These representations provide insight into which sub-components are already well addressed by existing models, and which others should be expanded or developed. The water quantity representation focuses on the conveyance of water through the network of links (e.g., channels, conduits) and nodes (e.g., detention basins, retention basins, wetlands). The hydrologic and hydraulic processes, which underpin the water quantity sub-component, are well established in stormwater models [70, 71]. The water quality representation includes the pollutant generation and treatment processes that occur

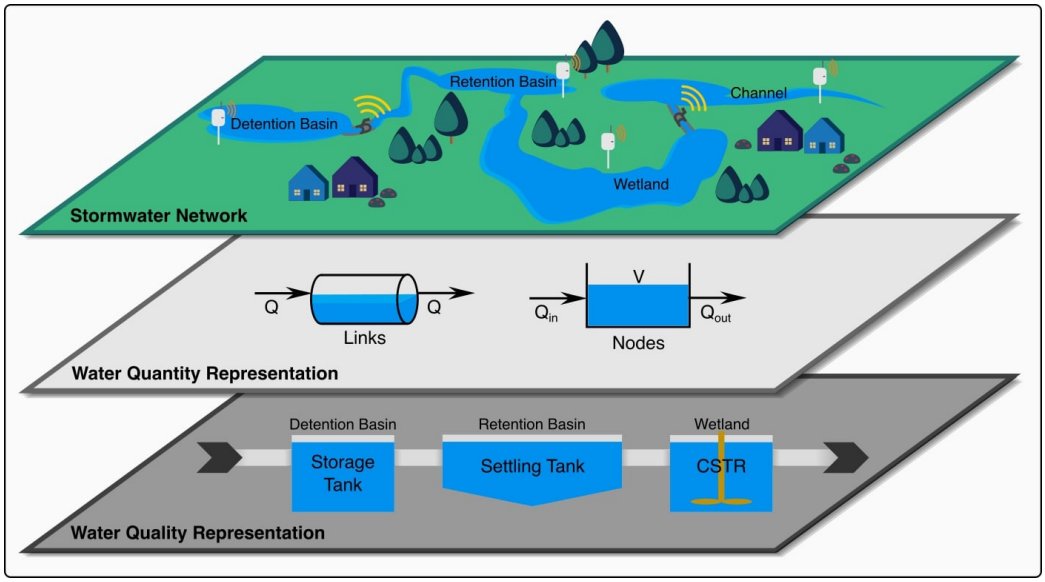

Figure 4.1: A watershed-scale pollutant transformation model is comprised of the water quantity and quality representations of the stormwater network. The water quantity representation, often modeled by SWMM, focuses on the conveyance of water through the network of links (e.g., channels, conduits) and nodes (e.g., detention basins, retention basins, wetlands). The water quality representation, often modeled using water treatment plant process literature, focuses on the water treatment processes that occur in stormwater assets.

in stormwater assets (e.g., wetland as a continuously tank reactor (CSTR), retention basin as a settling tank). Often, this sub-component is significantly simplified (e.g., first order decay models) instead of drawing from water treatment process literature [143], leaving room for expansion.

Guided by the state of these sub-components in current stormwater models, we developed *StormReactor*, a new water quality Python package, coupled with SWMM. The choice to build a module for SWMM was based on a number of factors. First, SWMM has a verified hydraulic solver, which is critically important for accurately

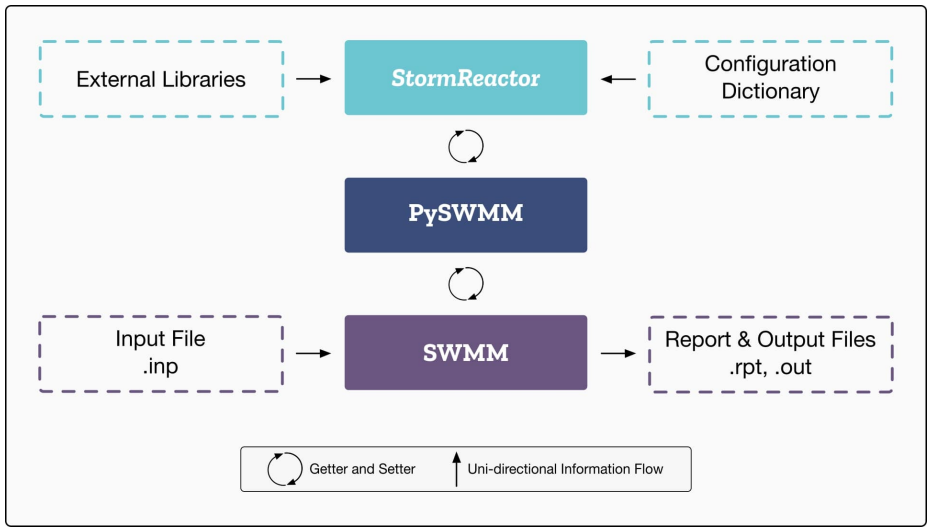

Figure 4.2: *StormReactor* follows an object-oriented programming paradigm. This modular approach allows for modifications and reuse by users. *StormReactor* uses a configuration dictionary and can work with external Python libraries. *StormReactor* interacts with PySWMM which interacts with SWMM all via *getters* and *setters*. SWMM requires an input file and then when a simulation is complete, it creates the report and output files.

modeling flow and pollutant routing [75]. In addition, building upon SWMM's popularity engages a large user base ensuring it is accessible to more people. Finally, SWMM is open source, which enables modification of its code and the use of popular Python wrappers, such as PySWMM.

Section 4.2.1 and Section 4.2.2 detail the development and structure of *StormReactor*. *StormReactor* was created by (1) modifying the SWMM and PySWMM source code to allow water quality states to be modified and (2) building an additional Python library to interface water quality modeling with these popular tools.

4.2.1 SWMM and PySWMM

To address the limitations of SWMM's water quality module, we modified SWMM's C source code[1] by introducing *getters* and *setters* to allow for real-time access of the model states during simulation (Table 4.1). A *getter* enables a user to access a variable while a *setter* enables a user to change the value of a variable. We then modified PySWMM's Python source code[2] to gain access to SWMM water quality states and to provide the convenience of modeling in a popular scripting language. While PySWMM already allowed for the interaction with SWMM's quantity states (e.g., flows, depths), it needed to be expanded to support interaction with water quality states (Table 4.1). Now a user can interact with a pollutant's concentration in any node or link during any routing time step. In this way, SWMM is used to transport pollutants using its reliable hydraulic and routing engine, PySWMM is used to support Python interaction with SWMM's C engine, and *StormReactor* adds supplementary support for water quality modeling (Figure 4.2).

4.2.2 StormReactor

StormReactor enables users to model water quality, while fully leveraging the well validated SWMM functionality for flow and routing. *StormReactor* provides a high-level programming interface that removes the user from the complex interactions between SWMM, PySWMM, and *StormReactor*, and only requires a few Python command statements to model pollutant transformations. Users have the ability to select a water treatment method in any stormwater asset and specify the routing time steps across which to carry out simulations. To promote uptake by an existing community of modelers, a user can select any of the already existing SWMM treatment functions outlined in the *SWMM Reference Manual Volume III: Water Quality* (Table 4.2) [76].

[1] github.com/OpenWaterAnalytics/Stormwater-Management-Model
[2] github.com/OpenWaterAnalytics/pyswmm

Table 4.1: The *getters* and *setters* added to both SWMM and PySWMM.

Variable	Type	Description
`NODEQUAL`	*getter*	current pollutant concentration in a node
`NODECIN`	*getter*	inflow concentration in a node
`NODEREACTORC`	*getter*	updated concentration after the mass balance of flows and pollutants in a node
`NODEHRT`	*getter*	hydraulic residence time (hours) in a node
`LINKQUAL`	*getter*	current pollutant concentration in a link
`TOALLOAD`	*getter*	total quality mass loading in a link
`LINKREACTORC`	*getter*	updated concentration after the mass balance of flows and pollutants in a link
`Node.extQual`	*setter*	current pollutant concentration in a link
`Link.extQual`	*setter*	current pollutant concentration in a node

Users can also select from a library of our new water quality methods, including reactor models and stream processes, such as erosion (Table 4.2). More importantly, users can implement their own custom pollutant models using a Python interface (Section 4.2.2.3). These custom pollutant models can be built upon states of the various water quantity and quality parameters in SWMM (e.g., flow, depth, volume, concentration) as well as interact with other Python packages (e.g., SciPy). Readers are directed to Zenodo[3] for *StormReactor*'s source code and documentation.

4.2.2.1 User Experience

StormReactor can be installed using pip[4]. To use *StormReactor*, first import both *StormReactor* and PySWMM (Figure 4.3). Next, define a configuration dictionary stating at which nodes and links water quality will be modeled, as well as the desired pollutants, water quality methods, and the parameters required for each method.

[3]DOI: 10.5281/zenodo.4913493

[4]pypi.org/project/stormreactor

Then, create an instance of the water quality class by calling `WaterQuality()` which takes two arguments: `config`, the configuration dictionary; and `sim`, a PySWMM simulation object, which encapsulates all the SWMM simulation functionality (e.g. start/stop simulation, get/set attributes). Finally, call the class instance method `updateWQState()` to run the desired water quality
method.

Once initialized, *StormReactor* executes the simulation loop. First, *StormReactor* queries the necessary water quantity and quality parameters (e.g., water depth, pollutant concentration) for specific stormwater assets at the current routing time step. Next, it uses the queried parameters to compute and set the new pollutant concentration using a predefined or custom water quality method. If a water quality computation requires a time parameter, the length of the routing time step is used. If real-time control is being modeled, selected water quality and/or quantity data are used to calculate the control decisions. SWMM then enacts the real-time control decisions and routes the pollutant(s) and flows through the network. This process can be repeated at any or every routing time step. The simulation loop terminates after the number of desired routing time steps or the SWMM model is complete.

4.2.2.2 Architecture

StormReactor's architecture follows an object-oriented programming paradigm. This matches already popular Python conventions and maximizes potential for user customization. *StormReactor* begins by defining a class: `WaterQuality()`. The class has an `__init__` method which takes three parameters: `self`, an instance of the class; `sim`, the PySWMM simulation object; and `config`, the configuration dictionary. When an instance of the class is created, it automatically calls the `__init__` method, which does the following: (1) initializes the asset flag; (2) calls the PySWMM method `sim.start_time` to get the start time of the simulation; (3) initializes the variable `last_timestep` to aid in calculating the length of the routing time step; (4)

initializes the ordinary differential equation (ODE) solver for the CSTR water quality method; and (5) defines the callable names of the water quality instance methods. The `WaterQuality()` class also defines two important methods: `updateWQState()` and `updateWQState_CSTR()`, which update the pollutant concentrations during a SWMM simulation for non-CSTR and CSTR methods, respectively. The class also has a collection of Python instance methods which specify the various treatment and generation processes that can be performed on a pollutant (Table 4.2).

Time steps are handled by *StormReactor* by relying on SWMM. Many of the treatment methods do not require a time parameter (e.g., event mean concentration, constant removal, k-C* method). *StormReactor* handles these methods just as they would be handled in native SWMM. These methods grab the current pollutant concentration and then calculates and sets the new concentration at the end of the current routing time step. For the methods that do require a time parameter (e.g., N-th order reaction kinetics, erosion, gravity settling), *StormReactor* computes the routing time step length (`dt`) using the same method as SWMM. To calculate `dt`, *StormReactor* calls the PySWMM function `sim.current_time` to get the current simulation time, subtracts the previous routing time step saved in the variable
`last_timestep`, and then converts it to seconds. In this way, *StormReactor* is dependent on SWMM to get `dt`. Once `dt` is calculated and the current concentration is queried, the new concentration is computed and set at the end of the current routing time step. This new concentration then becomes the concentration at the beginning of the next routing time step. Routing time steps are usually on the order of seconds, whereas water quality processes may take much longer. Therefore, users must also parameterize water quality coefficients on the order of seconds.

4.2.2.3 Implementing Custom Pollutant Models

To implement a new custom pollutant model, users can either (1) add their new class instance method to *StormReactor*'s code base or (2) build their model directly in their Python script using the appropriate *getters* and *setters* (Table 4.1). We recommend the first option if code is to be more seamlessly shared with others. To add a new method to the code base a user must:

1. Define the new method using the following convention: `_NewMethod(self, ID, pollutantID, parameters, flag)`. Non-public Python instance methods should always start with an underscore. The new method requires five parameters: `self`, an instance of the class; `ID`, the node or link name in SWMM; `pollutantID`, the pollutant index in SWMM; `parameters`, the water quality method parameters; and `flag`, used to determine if the method is for a link or node.

2. Provide a text description of the method including the water quality method parameters and their required units. Be sure to note if the method is for links, nodes, or both.

3. Write the pollutant transformation code for the new method.

   (a) Define any variables that may be needed for the pollutant transformation calculations.

   (b) Query SWMM variables that are necessary for the computation (e.g., pollutant concentration, water depth, current simulation time) using PySWMM *getters*.

   (c) Compute the pollutant transformation concentration.

   (d) Set the new pollutant concentration using PySWMM *setters*.

```python
# import packages
from StormReactor import WaterQuality
from pyswmm import Simulation

# build water quality configuration dictionary
config = { 'detention_basin': { 'pollutant': 0, 'method': 'GravitySettling', 'parameters': {'k': 0.0005,'C_s': 21.0}, \
    'retention_basin': { 'pollutant': 0, 'method': 'GravitySettling', 'parameters': {'k': 0.0005,'C_s': 21.0} }, \
    'wetland': { 'pollutant': 1, 'method': 'CSTR', 'parameters': {'k': -0.000089,'n': 1.0,'Co': 0.0} }, \
    'channel': { 'pollutant': 0, 'method': 'Erosion', 'parameters': {'w': 10.0,'So': 0.037,'Ss': 1.6, 'd50': 0.04} }, \
               { 'pollutant': 0, 'method': 'GravitySettling', 'parameters': {'k': 0.0005, 'C_s': 21.0} } }

# initialize water quality
with Simulation ('example.inp') as sim:
    WQ = WaterQuality(sim,config)

    for step in sim:
        # update each routing time step
        WQ.updateWQState()
```

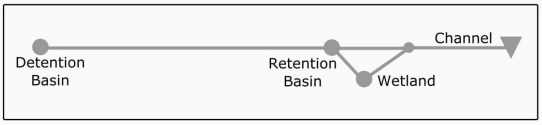

Figure 4.3: A Python code snippet that illustrates how some of the TSS and nitrate methods from the two case studies were implemented using *StormReactor*. The package is imported and the configuration dictionary is defined. The configuration dictionary includes the node/link IDs from the SWMM input file, the pollutant indices based on the order in which they are defined in the SWMM input file, the pollutant transformation methods selected, and the required pollutant transformation parameters. The methods are initialized by calling `waterQuality(sim,config)` and the pollutant transformations are computed by calling `updateWQState()` each routing time step.

4. Define the callable name in the `__init__` method.

5. Write unit tests for the new method and add them to test_links.py and/or test_nodes.py in the tests folder.

Once the new method is added to *StormReactor*'s code base, the user can then use it following the steps outlined in Section 4.2.2.1.

4.3 Water Quality Case Studies

The study area is a 7.8 km^2 urban, separated stormwater network (Figure 4.1) located in Michigan, which suffers from erosion problems due to high flashy flows. In this network, stormwater first flows through a detention basin into a long channel. A detention basin has its outlet at the bottom of the basin so between storms it is usually dry. The long channel then flows into a retention basin. A retention basin has its outlet at a higher point so it tends to retain a permanent pool of water. If the height of the water in the retention basin is less than a specified threshold, water flows directly into a constructed treatment wetland. Otherwise, water bypasses the wetland and overflows into another channel. Water leaving the wetland flows into the same channel as the overflow from the retention basin. The end of this channel is considered the outfall of the stormwater network.

For the two case studies, we isolated the network described above from a calibrated SWMM model of the larger, regional stormwater network. Since we removed the upstream assets from the model, we added inflows to simulate the real system response. The network was forced with a 5-year, 12-hour storm, which corresponds with design guidelines in the study region [22]. Readers are directed to Zenodo[5] for the SWMM input files and simulation code.

We provide these case studies to illustrate the following capabilities of *StormReactor*: (1) *StormReactor* can model SWMM's pollutant treatment equations as if we used SWMM's water quality module directly; (2) *StormReactor* can model new water quality processes (e.g., channel erosion, CSTRs in series); and (3) *StormReactor* enables water quality-based real-time control actions. The first case study uses TSS to illustrate the first two capabilities (Section 4.3.1) and the second case study uses nitrate to demonstrate the third capability (Section 4.3.2).

[5]DOI:10.5281/zenodo.4913515; DOI: 10.5281/zenodo.4913501

4.3.1 TSS Case Study

TSS (often measured as concentration in mg/L) is a commonly monitored pollutant because it negatively impacts water quality. These impacts include increasing turbidity, inhibiting plant growth, reducing species diversity, as well as providing transportation for nutrients and heavy metals
[158–160]. To mitigate these negative impacts, researchers and practitioners must be able to model deposition, erosion, and transport processes. Section 4.3.1.1 details how *StormReactor* was used to model these TSS processes and Section 4.3.1.2 provides the simulation results and discussion.

4.3.1.1 TSS Methods

Gravity settling was assumed to occur in the wetland, basins, and channels. We selected the gravity settling equation from the *SWMM Reference Manual Volume III: Water Quality* to illustrate how *StormReactor* allows users to model and match existing SWMM treatment equations [76]. The gravity settling equation is defined as:

$$C = C^* + (C - C^*)\, exp\,(-k\, \Delta t/d) \tag{4.1}$$

The values for the steady state concentration (C^*=21 mg/L) and the settling velocity (k=0.0005 m/s) were selected based on prior monitoring campaigns in the region. At each routing time step (Δt), depth (d) was queried from SWMM and the current concentration (C) was computed.

Along with gravity settling, erosion was also assumed to occur in both channels. Many equations exist for modeling erosion and sediment transport, many of which can be implemented in our library. For illustration purposes, we selected the Engelund-Hansen sediment transport formula [157].

The formula of Engelund and Hansen formula [157] can be expressed as:

$$f \cdot \phi = 0.10^{5/2} \tag{4.2}$$

where

$$f = (2 \cdot g \cdot d \cdot S_o)/v^2 \tag{4.3}$$

$$\theta = (d \cdot S_o)/[(S_s - 1)d_{50}] \tag{4.4}$$

$$q_t = \phi \left[(S_s - 1)\, g \cdot d_{50}^3\right]^{1/2} \tag{4.5}$$

where f is a friction factor, ϕ is a dimensionless sediment transport function, θ is a dimensionless shear parameter, g is gravitational acceleration, d is hydraulic depth, S_o is channel slope, v is mean channel velocity, S_s is specific gravity of sediment, d_{50} is mean particle diameter, and q_t is total bed-material sediment discharge by weight per unit width [161, 162]. The values for mean particle diameter (d_{50} = 0.04 mm), sediment specific gravity (S_s = 1.6), and channel slope (0.037-1.8 m/m) were selected based on site data. At each routing time step, the required parameter values were queried from SWMM, the sediment discharge concentration was computed, and the new TSS concentration was set in SWMM.

Root mean square error was used to validate both settling and erosion in the nodes and links. For gravity settling in the nodes, root mean square error was calculated for the cumulative TSS load from the *StormReactor* simulation and a native SWMM simulation (Figure 4.4). The root mean squared error was zero for all three nodes. Since treatment in SWMM links is a new feature of *StormReactor*, gravity settling and erosion in the channels had to be validated differently. The load leaving the channel was compared to the load entering the outfall. The root mean squared error was 6.19E-13.

TSS concentrations measured directly downstream of our outfall average 21 mg/L

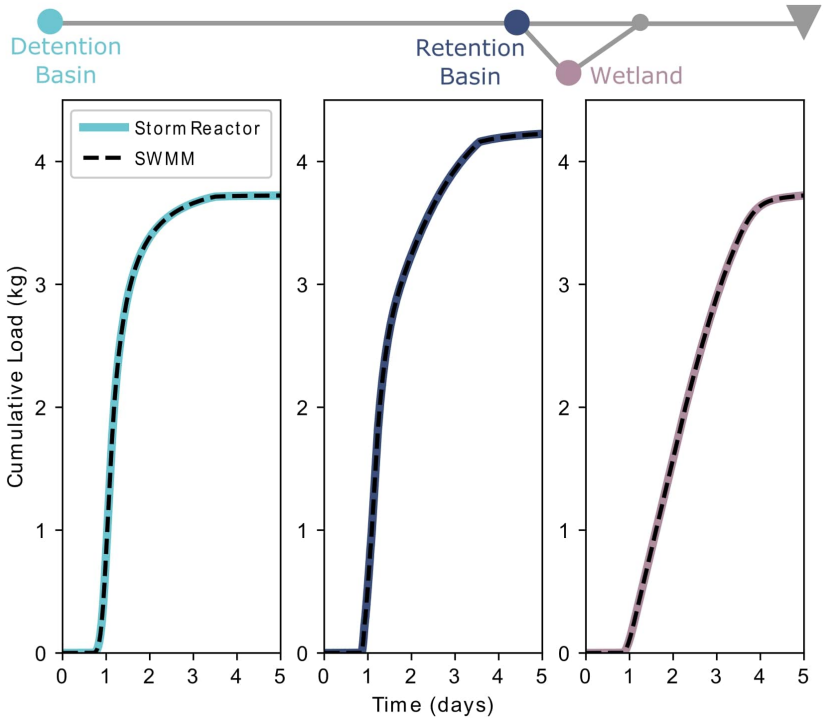

Figure 4.4: Cumulative TSS load comparing gravity settling using SWMM's traditional water quality module and *StormReactor*'s water quality module for the detention basin, retention basin, and wetland. Root mean squared error was zero for each asset.

during steady state conditions and 175 mg/L during storm conditions. For our simulation, TSS was assumed to follow an event mean concentration (EMC) wash-off model [75]. Since this network is dominated by channel erosion and not subcatchment wash-off, the steady state EMC was used in the wash-off model. The additional TSS needed to match storm event concentrations was provided by the erosion model.

4.3.1.2 TSS Results and Discussion

Results show that this system is dominated by erosion processes with only small reductions due to gravity settling (Figure 4.5). The detention basin's TSS concentration averaged 13 mg/L due to the small EMC used in the wash-off model. The retention basin saw higher concentrations throughout the simulation, with an average TSS concentration of 121 mg/L. This was a result of significant erosion occurring in the channel that connects the two basins. The wetland's TSS concentration was lower than in the retention basin, but still averaged 100 mg/L during the simulation. The reduction was due to settling in the wetland. The outfall's average TSS concentration was 107 mg/L. The increase in concentration at the outfall was again due to channel erosion occurring between the wetland and the outfall.

StormReactor improved TSS process representation by including channel erosion. Prior to *StormReactor*, users could not model pollutant generation processes unless they modified the parameters in the SWMM build-up and wash-off equations. In our case study, this would have not reflected reality because it would have resulted in high TSS concentrations in the detention basin. Since most of the TSS added to this system comes from downstream channel erosion, high TSS concentrations should only be found in the downstream assets. *StormReactor* now provides the ability to model pollutant generation processes in the assets in which they occur.

The TSS simulation took 42.35 seconds on a 2018 MacBook Pro (Processor: 2.2 Ghz 6-Core Intel Core i7; Memory: 16 GB 2400 MHz DDR4) as compared to 6.75 seconds without the TSS model. As we scale to larger networks, future work must

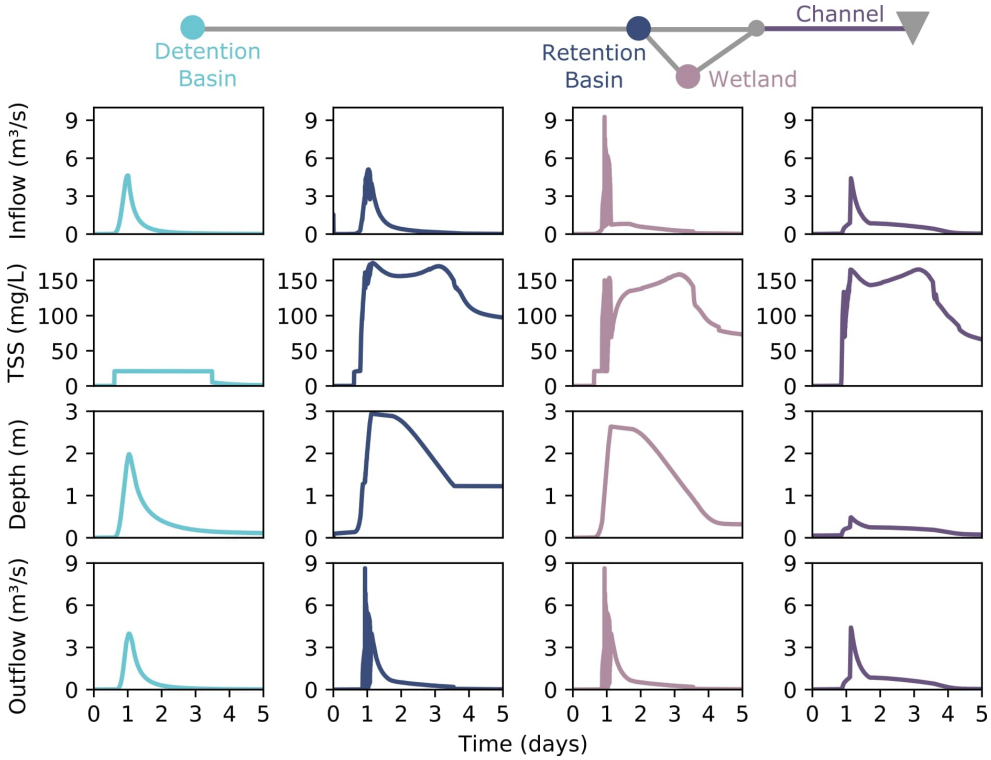

Figure 4.5: Simulation results for the various assets in the stormwater network including inflow rate (top panel), TSS concentration (second panel), storage depth (third panel), and outflow rate (bottom panel).

evaluate the computational efficiency of *StormReactor*.

4.3.2 Nitrate Case Study

Excess nitrogen can cause water quality impairments, such as eutrophication, harmful algal blooms, and fish kills [163, 164]. In order to mitigate these negative impacts, researchers and practitioners must be able to model the nitrogen cycle. This is presently not possible in models like SWMM, because the multiphase, multicomponent reac-

tions which are affected by the aerobic/anoxic conditions in the network cannot be simulated [81, 82].

Section 4.3.2.1 details how *StormReactor* was used to model nitrate. Section 4.3.2.2 explains the addition of real-time control, which will control the stormwater network in response to water quality states. To our knowledge, this case study is the first to model nitrate treatment through real-time control at the scale of an entire stormwater network.

4.3.2.1 Nitrate Methods

Modeling nitrogen interactions in stormwater is difficult because nitrogen exists in various forms (e.g., nitrate, nitrite, particulate nitrogen, ammonia, ammonium, dissolved organic nitrogen, nitrogen gas) and undergoes numerous transformations (e.g., denitrification, nitrification, ammonification, fixation, and dissimilatory reduction) [81]. In stormwater basins and wetlands, nitrogen is typically removed through three main mechanisms: assimilation, sedimentation, and denitrification. However, the primary mechanism is denitrification [3]. High denitrification rates are a result of high nitrate concentrations, low DO concentrations, and readily available sources of carbon (e.g., decaying plants and grass) [156, 165].

For this case study, we focused only on nitrogen in the form of nitrate and therefore, denitrification as the primary removal mechanism. We selected nitrate because site data and other studies indicate runoff is dominated by this form of nitrogen [156]. Denitrification was assumed to occur only in the wetland because wetlands tend to have large quantities of biomass and thus higher denitrification capacity than other storage nodes [166, 167]. Since this case study assumed high nitrate concentrations and readily available sources of carbon, DO became the limiting factor for denitrification, necessitating us to model DO concentrations as well.

The wetland DO model was implemented using the CSTR method in *StormReactor*. Based on findings by Kadlec [168], we assumed the wetland functioned as

three CSTRs in series. We selected CSTRs to illustrate how *StormReactor* enables wastewater treatment process models. Often CSTRs are modeled assuming steady state conditions, where the influent concentration, inflow rate, and outflow rate are constant, and therefore, the concentration in the control volume is also constant. Steady state condition allows for a closed form solution to the CSTR equation. However, in a wetland, influent concentration and flows are dynamic and therefore, the CSTR should be assumed to be unsteady. We solved the unsteady CSTR with an ODE solver to show how *StormReactor* integrates with other computational Python packages. We selected the SciPy ODE numerical solver using the explicit runge-kutta method[6] [102]. The CSTR equation is defined as:

$$\frac{dC}{dt}\, V = Q_{in}C_{in} - Q_{out}C - kCV \tag{4.6}$$

Based on data collected in this network, the influent DO concentration (C_{in}) to the wetland was assumed to be 9.6 mg/L. The reaction rate constant (k_{DO}) was assumed to be 0.2/hr [169]. At each routing time step, the dynamic parameters were queried from SWMM (Q_{in}, Q_{out}, V) and the ODE solver computed the current concentration (C). Since the DO concentration was only relevant to triggering denitrification in the wetland, DO was tracked only in Python and therefore, the new DO concentration did not need to be set in SWMM (i.e., DO was not added as a pollutant in the SWMM input file).

Nitrate treatment was triggered when the DO concentration dropped below 1 mg/L, signally anoxic conditions. Nitrate treatment in the wetland was also modeled in *StormReactor* using three CSTRs in series [156]. The nitrate concentration in the real stormwater network averages less than 1 mg/L during steady state and storm conditions. Although this low level may exceed recommended water quality criteria [170], assuming a larger concentration will result in higher rates of denitrification for

[6] docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.ode.html

simulation purposes. Therefore, for our simulation, nitrate was added to the system using SWMM's wash-off model assuming an EMC of 10 mg/L, which aligns with 13% of stream sites monitored by Mueller and Spahr [171]. The nitrate reaction rate constant (k_{NO}) was assumed to be 1.5/day [169]. At each routing time step, the dynamic parameters were queried from SWMM (Q_{in}, Q_{out}, C_{in}, V), the ODE solver computed the current concentration (C), and that concentration was then set in SWMM. To validate the CSTRs in series model, *StormReactor*'s steady state concentration at the end of the simulation was compared with the steady state analytical solution. The wetland's nitrate concentration from *StormReactor* converged to the computed steady state analytical solution (5.7% error).

4.3.2.2 Nitrate Real-Time Control Strategy

A water quality-based controller was constructed to maximize denitrification without flooding the wetland (Figure 4.6). The controller held water in the wetland until the nitrate was treated or flooding was imminent. It also held water in the upstream detention basin until the downstream wetland had sufficient storage capacity to handle more inflow. When the controller opened a valve, it regulated the size of the opening (0-100%) to release water at a rate proportional to the asset's water level by solving the submerged orifice equation [75]. It was assumed that the network had the necessary water quantity and quality sensors and the outlets of the detention basin and the wetland had controllable valves. To reflect real world implementation, control decisions were constrained to every 15 minutes. The controlled scenario was compared against a baseline, uncontrolled scenario to determine the effectiveness of the controller.

OLD CAPTION: The controller's objective was to maximize denitrification without flooding the wetland. The controller computed the valve's percent opening for the detention basin ($valve_{DB}$) and wetland ($valve_W$). Water was released proportionally by solving the submerged orifice equation ($Q_{max} = CA/\sqrt{2gd}$) for C, the discharge

Algorithm Algorithm to maximize denitrification without flooding the wetland.

Compute wetland's DO and nitrate concentrations: $\frac{dC}{dt} = \frac{Qin_t \cdot Cin_t - Qout_t \cdot Cout_t}{V_t} - k_t \cdot C_t$

for *i in controllable valves* **do**

- **if** $DO^t > 1mg/L$ **then**
  - **if** $d_W \leq 3m$ **then**
    - $valve_W = 0 \quad valve_{DB} = f \cdot Q_{max}/A_{DB}\sqrt{2 \cdot g \cdot d_{DB}}$
  - **else**
    - $valve_W = f \cdot Q_{max}/A_W\sqrt{2 \cdot g \cdot d_W} \quad valve_{DB} = 0$
- **else if** $DO \leq 1mg/L$ **then**
  - **if** $C_t \leq 5mg/L$ **then**
    - $valve_i = f \cdot Q_{max}/A_i\sqrt{2 \cdot g \cdot d_i})$
  - **else**
    - **if** $d_W \leq 3m$ **then**
      - $valve_i = 0$
    - **else**
      - $valve_W = f \cdot Q_{max}/A_W\sqrt{2 \cdot g \cdot d_W}) \quad valve_{DB} = 0$

end

Figure 4.6: Algorithm to maximize denitrification without flooding the wetland.

coefficient, where Q_{max} was the maximum flow rate desired ($Q_{max} = 2m^3/s$), A was the completely open orifice area, g was acceleration due to gravity, and d was water depth. Q_{max} was the flow rate threshold at which downstream sediments were assumed to re-suspend [143].The computed value for C was multiplied by a scaling factor f (f=1.75 in this study).

4.3.2.3 Nitrate Results and Discussion

The controller met the control objective of maximizing denitrification (Figure 4.7). The controlled scenario saw a 95% nitrate load reduction at the outfall as compared to the uncontrolled scenario. The load reduction was a result of keeping the valves closed when either the wetland was oxic or the wetland's nitrate concentration was too high. The controller used both the wetland and the upstream basin for storage until the conditions were appropriate to release flows. To put this load reduction into context, SWMM was used to determine how large the studied wetland would need to be to obtain the same load reduction without real-time control. After incrementally increasing the area of the wetland and rerunning the SWMM simulation several times, it was determined that the wetland would need to be four times as large to obtain the same load reduction.

The controller also ensured that flooding did not occur in any of the assets (Figure 4.7). The water depths in the detention basin and wetland were kept below their flooding thresholds. These two assets did not flood because the detention basin had significant storage capacity, and the controller opened the wetland valve whenever it was close to its maximum capacity. In both scenarios, the retention basin depth resulted in some flows bypassing the wetland. Unfortunately, this is because of how the retention basin/wetland system was designed. If a control valve was installed or the bypass height was increased on the retention basin, these bypass flows could have been reduced.

StormReactor provided the ability to implement a water quality-based controller

in SWMM. Prior to this package, users trying to meet water quality goals with controllers could only access water quantity states. Now, users can access water quality states and build a pollutant concentration-based controller with only a few lines of Python code.

The real-time controlled nitrate simulation took 86.84 seconds on a 2018 MacBook Pro (Processor: 2.2 Ghz 6-Core Intel Core i7, Memory: 16 GB 2400 MHz DDR4). The nitrate simulation without real-time control took 86.22 seconds, as compared to the simulation without water quality or real-time control which took 12.30 seconds. The increased computational time was a result of the longer simulation (twelve days instead of five) and the ODE solver. Therefore, to increase computational efficiency in the future, a discrete form update could be used instead of an ODE solver.

4.4 Discussion

As shown in the case studies, *StormReactor* improved water quality process representation at both the site and watershed scale. Rather than implementing an all-in-one quality-quantity model, we coupled the popular water quantity features of SWMM with *StormReactor*'s water quality model. To illustrate the fidelity of *StormReactor*, we showed how a variety of pollutant transformations (e.g., erosion, settling, CSTR) matched expectations from established models and methods. Therefore, *StormReactor* was shown to be an effective tool for modeling water quality.

To the best of our knowledge, our modular framework supports many of the features seen in advanced hydraulic and water quality packages. For advanced users, *StormReactor*'s integration with Python will support numerical solvers and packages, higher order reaction kinetics, wastewater process models (e.g., ASM-1), and combined sewer networks. In its present implementation, *StormReactor* poses a few constraints which users need to be aware of before choosing to use it in their stormwater studies. It does not presently support LID (i.e., green infrastructure) water quality

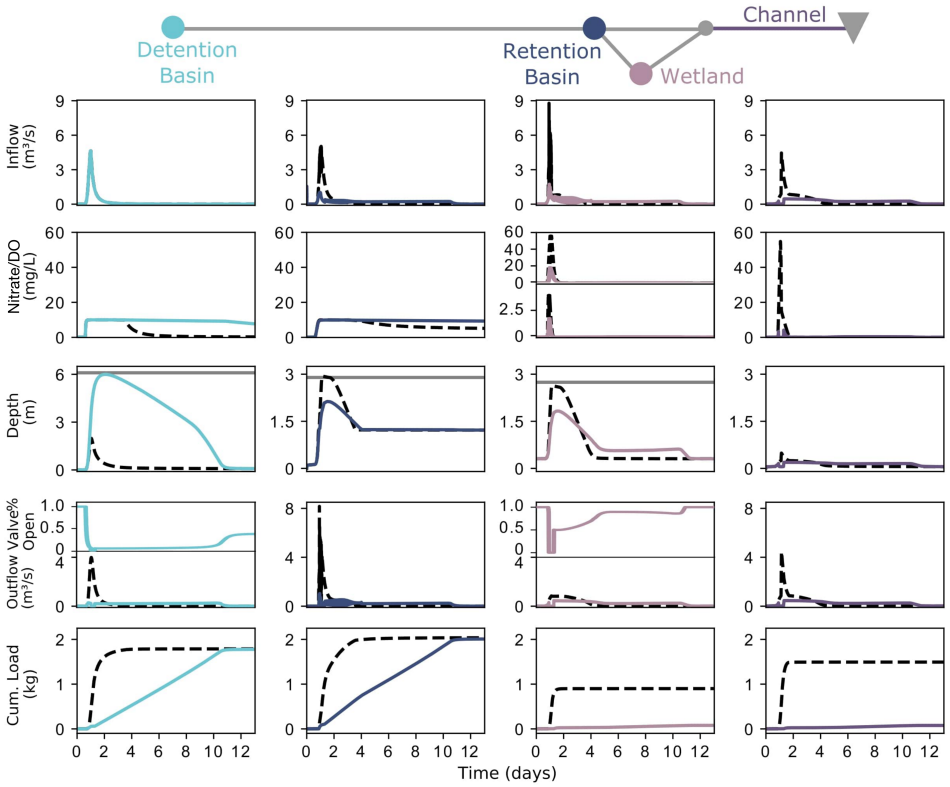

Figure 4.7: Comparison of the uncontrolled (dotted lines) and controlled (solid lines) scenarios for the various assets in the stormwater network including inflow rate (top panel), nitrate and DO concentration (second panel), storage depth (third panel), valve position and outflow rate (fourth panel), and cumulative nitrate load (bottom panel). In the depth panel, the gray solid lines depict the flooding thresholds for the detention basin and the wetland and the bypass threshold for the retention basin. No flooding occurred but some flows did bypass the wetland in both the uncontrolled and controlled scenarios.

processes because SWMM handles LID water quality outside of its link and node data structures. In addition, *StormReactor* does not support high spatial resolution water quality processes (e.g., advection, diffusion). Both LID access and high spatial resolution models can be added and are proposed as future work. Aside from these limitations, *StormReactor* provides a general water quality modeling solution that is flexible and expandable.

The nitrate case study points to the potential of using real-time control or "smart" stormwater systems for ecological benefits. Watershed water quality goals can be achieved by tuning real-time control. The ability to model complex water quality interactions enables the development and testing of real-time control algorithms that use pollutant concentration, load, and sensor data. We can now utilize formal control theory (e.g., PID, MPC, genetic algorithms) to explore emergent behavior, stability, and optimal control strategies at both the site and watershed scale. We can then use this information to optimize asset treatment performance, pushing our watersheds to behave like distributed water treatment plants, and ultimately improve watershed water quality.

4.5 Conclusions

StormReactor improves the fidelity of modeling pollutant transformations and pollutant-based real-time control; moving us a step closer to realizing the goal of controlling entire watersheds as real-time distributed treatment plants. Additional fidelity could be gained by adding LID access and high spatial resolution models to *StormReactor*. The flexibility of *StormReactor* gives researchers and practitioners immense freedom in modeling water quality. We hope that this package will become a community-driven resource. We see opportunities for the research community to collaborate on the development of *StormReactor* by contributing their own pollutant generation and treatment methods. As we scale to larger networks, future work must

evaluate the computational efficiency of *StormReactor*. In addition, significant future research stands to be enabled through the use of holistic frameworks, such as those posed in this paper. In particular, future studies have the potential to evaluate how to control entire watersheds in response to ecological objectives.

Table 4.2: Overview of the current water quality methods that can be selected from *StormReactor* including a method explanation and the asset type (node, link, or both) it can be used for.

Water Quality Method	Asset Type	Method Explanation
Event Mean Concentration	Both	Treatment results in a constant concentration
Constant Removal	Both	Treatment results in a constant percent removal
Co-Removal	Both	Removal of some pollutant is proportional to the removal of some other pollutant
Concentration-Dependent Removal	Both	When higher pollutant removal efficiencies occur with higher influent concentrations
Nth Order Reaction Kinetics	Both	When treatment of pollutant X exhibits nth order reaction kinetics where the instantaneous reaction rate is kC^n
k-C* Model	Node	The first-order model with background concentration made popular by Kadlec and Knight [156] for long-term treatment performance of wetlands.
Gravity Settling	Both	During a quiescent period of time within a storage volume, a fraction of suspended particles will settle out
CSTR	Node	CSTR is a common model for a chemical reactor. The behavior of this CSTR is modeled assuming it is not in steady state because outflow, inflow, volume, and concentration are constantly changing.
Erosion	Link	Engelund and Hansen [157] developed a procedure for sediment transport in streams.

CHAPTER 5

Improvement of Phosphorus Removal in Bioretention Cells Using Real-Time Control

Abstract

Retrofitting urban watersheds with wireless sensing and control technologies will enable the next generation of autonomous water systems. While many studies have highlighted the benefits of real-time controlled grey infrastructure, few have evaluated real-time controlled green infrastructure. Motivated by a controlled bioretention site, where phosphorus is a major runoff pollutant, we present an analysis of phosphorus removal over a range of influent concentrations and storm conditions for three scenarios: a passive, uncontrolled bioretention cell (baseline), a real-time controlled cell (autonomous upgrade), and a cell with soil amendments (passive upgrade). Results suggest the autonomous upgrade matched the pollutant treatment performance of the baseline scenario, only using half the spatial footprint. The autonomous upgrade matched the performance the passive upgrade; suggesting real-time control may provide a "digital" alternative to existing, passive upgrades. These findings may help stormwater managers, who are often constrained by site or cost constraints, meet their water quality goals.

5.1 Introduction

An emerging generation of autonomous stormwater solutions promises to shrink the size of infrastructure needed to manage runoff pollution and changing weather. Retrofitting stormwater infrastructure with wireless sensors, gates, valves, and pumps

will reduce flooding and improve pollutant treatment [143, 20]. This will be achieved by dynamically adjusting water levels to take advantage of excess storage and enhanced treatment conditions. At the core of this vision is system-level control, where tens to hundreds of individual stormwater assets will coordinate in real-time to route water and promote the uptake of pollutants across the scale of entire watersheds [20, 21]. In lieu of new construction, this will repurpose existing infrastructure by dynamically adapting it on a storm-by-storm basis [143].

Many studies have highlighted the benefits of autonomous stormwater infrastructure for gray infrastructure, including basins, ponds, and underground infrastructure [172], but few have explicitly evaluated real-time control of green infrastructure [92, 173]. Real-time control of larger and non-biologically active sites, such as detention basins, have shown significant benefits—a valve at the outlet can be used to extend retention time and drastically improve the capture of sediment-bound pollutants [172, 174]. Extrapolating these benefits to green infrastructure, and more specifically bioretention systems, is difficult because pollutant removal is underpinned by much more complex biological and physical processes [34]. Real-time control of a bioretention cell can be achieved cost effectively by adding an actuated valve and a water level sensor (Figure 5.1b). At the time of writing, a fully automated and internet-connected control system could be constructed for $1,500 USD using open-source solutions. Readers are directed to Bartos et al. [19] and www.open-storm.org for details and best practices on implementation.

The purpose of this chapter is to begin exploring these processes for phosphorus, a runoff pollutant associated with harmful algal blooms [175], using *StormReactor*, a new water quality modeling toolchain [173]. Specifically, this chapter compares phosphorus removal in a passive, uncontrolled bioretention cell (i.e., baseline) scenario to two upgraded scenarios: one with real-time control (i.e., autonomous upgrade) and another with water treatment plant residuals (i.e., passive upgrade). The approach ingests laboratory-measured data into a model of bioretention-based phosphorus re-

moval under real-time control. Treatment performance is evaluated across a broad range of influent concentrations and storm conditions.

5.2 Methods

5.2.1 Hydrologic and Control Model

A bioretention cell at the Toledo Zoo in Toledo, Ohio, US was retrofitted for real-time control (Figure 5.1). To estimate site performance, a hydrological model was built and calibrated using the hydraulic conductivity rate of the existing site (i.e., 5.1 cm/hr). This site is considered oversized by most US design guidelines, which stipulate that bioretention cells should generally capture the runoff of no more than 8,094 m^2 (2-acres), and that the surface area of a cell should be 5-10% of this contributing impervious area [27]. Therefore, the hydrological model assumed a contributing impervious area of 4,047 m^2 (1-acre) and cell surface area of 405 m^2 (0.1-acre).

The hydrologic model was built using the U.S. EPA's Stormwater Management Model (SWMM), a physically-based, discrete-time, storm-runoff simulation model [75]. While recent updates to SWMM feature a tool for modeling green infrastructure, the tool does not include the ability to dynamically control the flow through the underdrain. To circumvent this issue, the bioretention hydrologic model was built following the methodology vetted by Lucas [88], which represents the individual components of the cell (i.e., ponding area, soil media, gravel storage, underdrain, and surrounding soil) using SWMM junctions, conduits, orifices, outlets, weirs, and outfalls (Figure 5.2).

A level controller was designed as an initial step in exploring bioretention control. The controller released water from the underdrain at a rate proportional to the cell's ponding height (i.e., the deeper the ponding height, the larger the valve opening). Specifically, the controller sets the underdrain's valve to a position between

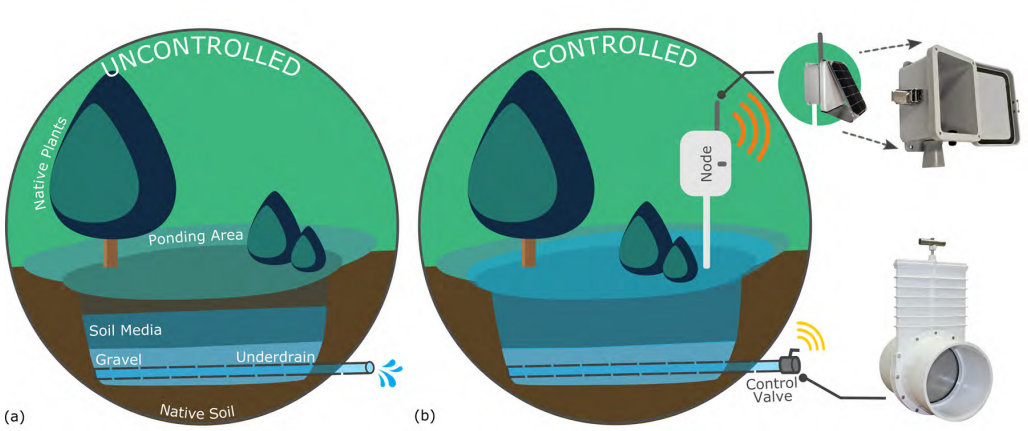

Figure 5.1: The cross-section of (a) a passive, uncontrolled bioretention cell (i.e., baseline) and (b) a real-time controlled bioretention cell (i.e., autonomous upgrade) with images of the sensor node and control valve installed on a cell in Toledo, Ohio, US.

closed (0%) and open (100%) based on the following formula: $valve\ position = 10\% \times ponding\ height$. Control decisions were made once every 15 minutes, as implemented in the field. The controller promotes improved hydrologic conditions and, by extension, improved pollutant removal. Readers can access the simulation documentation from the GitHub online repository (`github.com/bemason/RTC_GreenInfrastructure`).

5.2.2 Phosphorus Model

Although SWMM has the ability to model nutrient removal, it is limited to percent removal and first order dynamics [76], which cannot effectively represent the complex nutrient interactions triggered by real time control [143]. Therefore, the phosphorus model was added as a new custom pollutant model to the open-source Python package *StormReactor* [173]. *StormReactor* provides a high-level programming interface

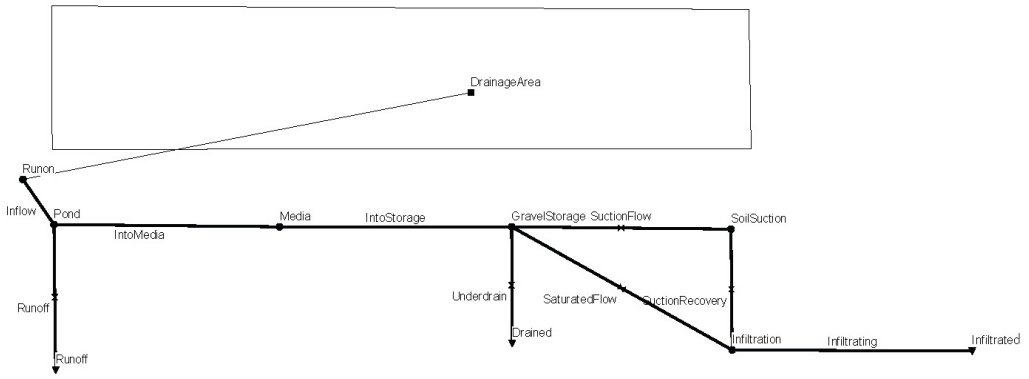

Figure 5.2: A bioretention cell hydrologic model was built following the methodology vetted by Lucas [88], which represents the individual components of the cell (i.e., ponding area, soil media, gravel storage, underdrain, and surrounding native soil) using SWMM junctions, conduits, orifices, outlets, weirs, and outfalls. An orifice is used to simulate a controllable valve on the underdrain.

for users to model pollutant transformations while leveraging the flow and routing functionality of SWMM. *StormReactor* provides the ability to model complex pollutant transformations (e.g., higher order reaction kinetics, wastewater process models, and differential equations). Readers can access the source code and documentation from the GitHub online repository (github.com/kLabUM/StormReactor).

Li and Davis's [176] phosphorus model was implemented, which represents a bioretention cell as a plug flow reactor and one-dimensional adsorption column (Figure 5.3). The model allows for advective flow in and out of a horizontal differential element of the soil media and considers filtration, adsorption, and leaching reactions. The mass balances for the reactor are shown for the particulate and dissolved phosphorus concentrations (CPP, CDP) in Equations 1 and 2, respectively (Figure 5.3).

The water quality model parameters were experimentally derived and calibrated by Li and Davis [176]. In the baseline and autonomous scenarios, these parameters

assume the cell's soil media consists of 74% bioretention soil media, 22% additional sand, and 3% mulch, on an air-dry mass basis. In the passive upgrade scenario, 5% of the bioretention soil media was replaced with water treatment plant residuals [177].

5.2.3 Control Performance Evaluation

As discussed, bioretention cells can be upgraded in a variety of ways to boost performance. An analysis was performed to compare phosphorus removal in a passive, uncontrolled bioretention cell (baseline) scenario to two upgraded scenarios: one with real-time control (autonomous upgrade) and another with water treatment plant residual soil amendments (passive upgrade). To capture the wide range of conditions a bioretention cell may experience, the scenarios were evaluated under a variety of influent concentrations and design storms. Five total phosphorus influent concentrations (0.2, 0.6, 1.0, 1.4, and 1.8 mg/L) were selected based on literature values for commercial, residential, and agricultural settings [178]. Since dissolved and particulate phosphorus removal are independent of each other, equal amounts of each were used in the evaluation (i.e., for 0.2 mg/L of total phosphorus, 0.1 mg/L of both dissolved and particulate phosphorus were used). Three 6-hour Soil Conservation Service (SCS) Type II design storms were evaluated as conventional in the green infrastructure community: 12.7 mm (0.5 in), 25.4 mm (1.0 in), and 50.8 mm (2.0 in) [179].0

In addition, a simulation using rain data from the wet summer of 2015 in Toledo, OH, US was used to evaluate performance under real, dynamic weather conditions for the same three scenarios described above. The weather data selected includes several consecutive storms, dry periods, as well as large and small storms. The total phosphorus influent concentration was assumed to be 0.38 mg/L, the average concentration for all US land use types according to the US National Stormwater Quality Database [180].

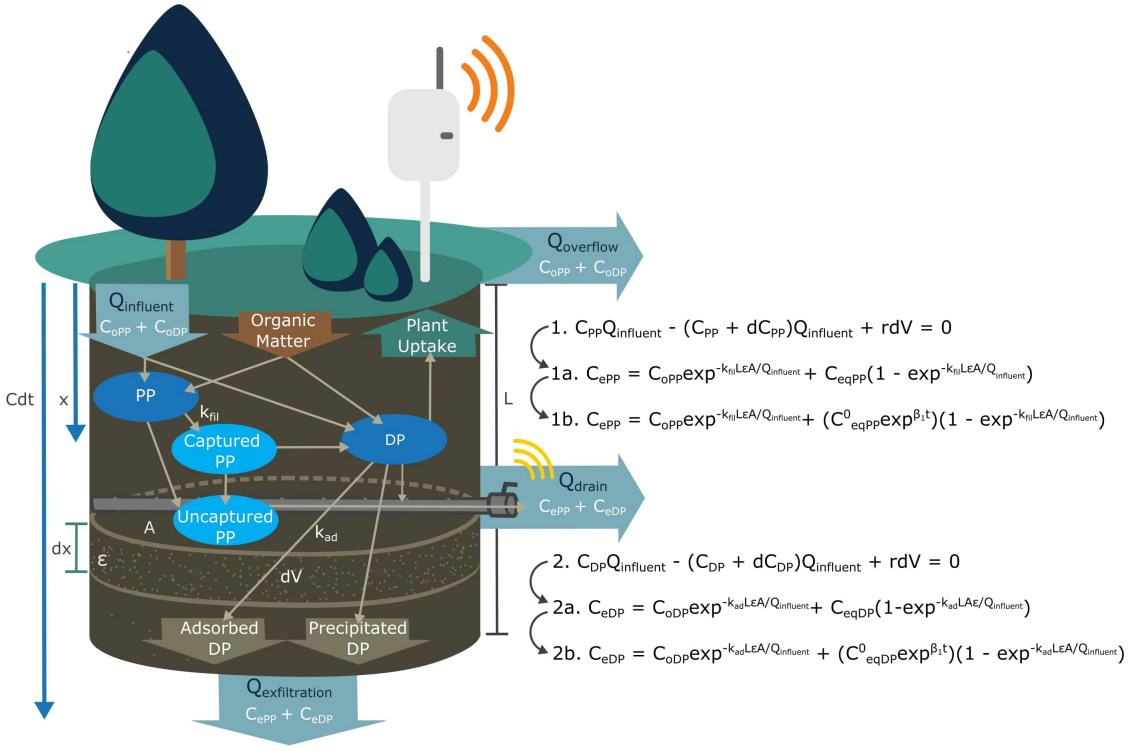

Figure 5.3: On the left, a controllable bioretention cell modeled as a plug flow reactor showing the dissolved (DP) and particulate phosphorus (PP) transformations in the soil. The model represents the flow rate (Q) through a differential element (dx). The pollutant's concentration (C) changes as it moves through the length (L) of the reactor. The area (A), porosity (ϵ), and Q determine infiltration. On the right, mass balance equations. To create Eq. 1a and 2a, the variables were separated and integrated over L and the change in concentration (Co to Ce); and the adsorption and filtration rate constants (k_{ad}, k_{fil}) were substituted in for the reaction rates (r). To create Eq. 1b and 2b, $C^0_{eq} \exp^{\beta_1 t}$ was substituted in for the equilibrium concentration (Ceq) to account for the variability in C_{eq} over the lifetime of the soil; where C^0_{eq} is the initial C_{eq} for a storm event, β_1 is a constant describing the rate at which Ceq approaches Co, and t is the cumulative time elapsed from the start of a storm event.

5.3 Results and Discussion

5.3.1 Comparative Analysis

The results of the comparative analysis are shown in Figure 5.5. Illustrated are various permutations of design storms (x-axis) and influent concentration (y-axis) across multiple scenarios: baseline (a), autonomous upgrade (b), and passive upgrade (c). For each simulation, the total phosphorus load (flow times concentration) was computed for the flows entering the bioretention cell, exiting through the underdrain, exfiltrating into the surrounding soil, and overflowing when the ponding height was exceeded. A mass balance of these loads then determined the final value (i.e., color) used in the figure. The figure colors indicate if the overall mass balance resulted in phosphorus capture (blue) or release (red). The black line indicates the transition from phosphorus capture to release.

The baseline scenario released phosphorus during the low influent simulations and removed phosphorus during the high influent simulations (Figure 5.5a). The equilibrium concentration essentially defines the point separating removal from release. When the influent concentration is larger than the equilibrium concentration, removal occurs, otherwise, desorption of the pollutant occurs, resulting in a net increase in the pollutant concentration [176].

The autonomous upgrade captured phosphorus during most simulations (Figure 5.5b). During the smaller storms, the controlled underdrain remained closed for most of the simulation, resulting in little to no phosphorus being released. The transition from phosphorus capture to release occurred during the larger storms with lower influent concentrations. Akin to the baseline, this was due to the system trying to reach the equilibrium concentration.

The autonomous upgrade outperformed the baseline scenario (Figure 5.5a,b), aligning with the results of the real-time controlled bioretention column study by Persaud et al. [92]. Since the autonomous upgrade released at least two times less

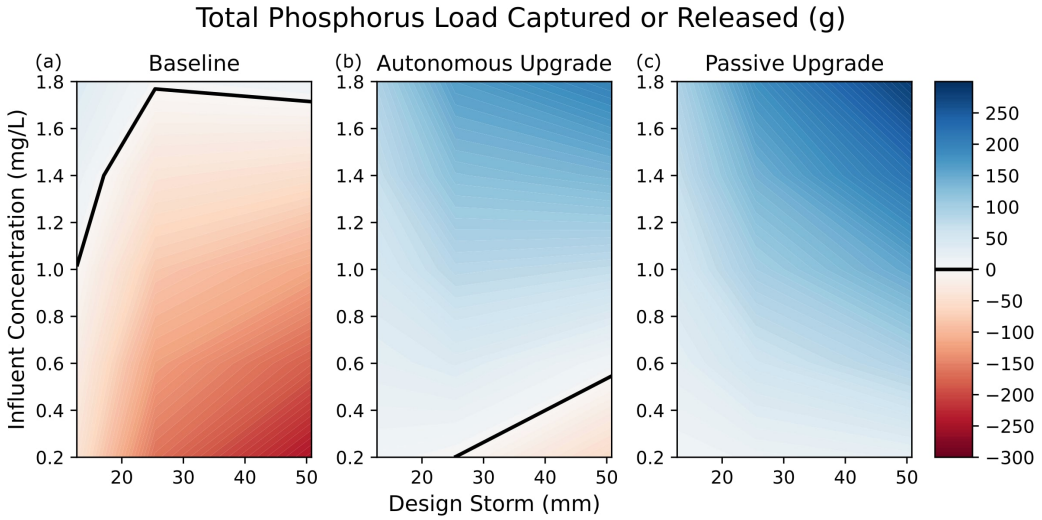

Figure 5.4: Total phosphorus (g) either captured or released by a bioretention cell over various storm sizes (x-axis) and influent concentrations (y-axis) for the baseline (a), autonomous upgrade (b), and passive upgrade (c) scenarios. The black lines denote a shift from capturing (blue) to releasing (red) phosphorus.

phosphorus load than the baseline during all design storms, the autonomous upgrade could match the pollutant treatment performance of the baseline in half the spatial footprint. This result aligns with other research that has shown that real-time controlled stormwater infrastructure can be built smaller without compromising performance [143, 173].

The passive upgrade resulted in phosphorus capture for all design storms and influent concentrations (Figure 5.5c). By design, soil amendments have high reaction rate constants and a low equilibrium concentration. Both factors worked together to ensure the cell's phosphorus concentration remained low. Therefore, even though water left the unregulated underdrain, the load released was relatively small.

Although the modeled soil amendments were successful at treating phosphorus, they have several drawbacks. Their efficacy will inevitably deplete over time, requir-

ing the installation of new amendments. Soil amendments are targeted pollutant solutions; they have limited or no impact on other pollutants [91]. Similarly, they do not provide hydrologic benefits, and may even reduce hydraulic conductivity [181]. Although the water treatment residual amendments are a free by-product from the water treatment plant, there is a cost to excavating and reinstalling the amended soil. For the modeled site at the time of writing, the installation of the passive upgrade would cost an estimated $4,800 USD [182]. There would be additional costs for monitoring and maintenance but calculating these costs are outside of the scope of this chapter.

Real-time control pushed the autonomous upgrade to perform similarly to the passive upgrade (Figure 5.5b,c). The performance was essentially equivalent during the smallest design storm, but the autonomous upgrade released up to seven times more phosphorus load during the larger two storms. By analogy, real-time control enables a bioretention cell to perform as if it has been "digitally" upgraded to achieve benefits of passive soil amendments. By controlling the flow through the underdrain the system is forced to mimic the pollutant treatment of the passive upgrade. The removal mechanism, however, is different for the autonomous upgrade. Removal is primarily through volume reduction rather than adsorption and filtration. By exfiltrating water and phosphorus, the volume of water leaving the site is reduced, thus also diminishing potential adverse impacts on downstream waterways.

The autonomous upgrade can be used to address a variety of pollutants (e.g., phosphorus, metals, solids) through volume reduction, while the passive upgrade only targets one pollutant through adsorption and filtration. Therefore, the autonomous upgrade provides long-term management flexibility by enabling the cell to be "reprogrammed" to tailor retention times whenever a new pollutant needs to be treated, or when knowledge of site dynamics changes. This flexibility is even more pronounced when considering system-level control. Stormwater managers can coordinate a network of autonomously upgraded sites, allowing them to decide where and how pol-

lutants are treated to meet system-level water quality goals [22]. In addition, the autonomous upgrade is cost-effective ($1,500 USD including parts and installation at the time of writing). That being said, since real-time control is not readily offered as a commercial solution, it is still difficult to project this cost into a future commercial market. Like the passive upgrade, there may be additional costs for monitoring and maintaining the site that are outside of the scope of this chapter.

5.3.2 Dynamic Storm Analysis

In the previous section, phosphorus removal was evaluated using design storms to better understand performance across the wide range of conditions a bioretention cell may experience. A dynamic storm simulation was carried out to evaluate performance using local, measured storm data. The three scenarios received the same cumulative influent load (5.9 g) (Figure 5.4a,d,g). The differences, however, occur when comparing the cumulative released and captured loads. The cumulative load released from the underdrain (no overflow/runoff occurred) was 7.7 g and 7.1 g for the baseline and passive upgrade scenarios, respectively (Figure 5.4h). These results suggest the baseline and passive upgrades leached phosphorus previously captured in the soil (Figure 5.4i). Leaching is attributed to the wetting/drying cycle in the bioretention soil media. When dried soil is wetted, the phosphorus concentration initially increases as previously captured phosphorus is washed and eluted from the soil media [176]. The higher phosphorus concentration combined with the larger volumes of water leaving their underdrains (Figure 5.4b) resulted in net export of phosphorus (Figure 5.4e). The autonomous upgrade also exhibited higher phosphorus concentrations at the start of each storm (due to the wet-dry cycle) but only a fraction of water was released from the underdrain and the rest was exfiltrated (Figure 5.4b,c). This is why the autonomous upgrade captured 5.4 g and released 0.6 g, about twelve times less phosphorus load than the uncontrolled cells (Figure 5.4h,i).

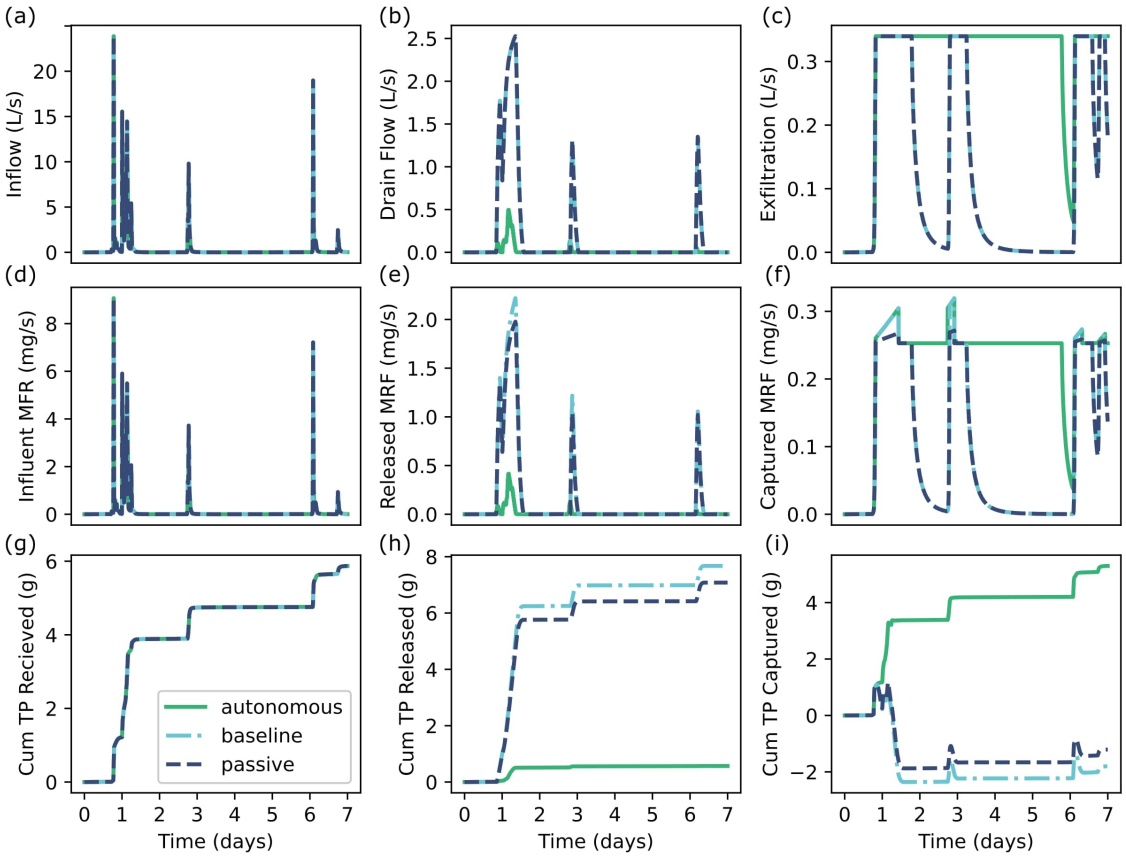

Figure 5.5: Dynamic storm simulation results showing the inflow (a), drain flow (b), and exfiltration (c) rates; influent (d), released (f), and captured (f) mass flow rates (MFR); and cumulative total phosphorus (TP) load in grams received (g), released (h), and captured (i) for the baseline (light blue), autonomous upgrade (green), and passive upgrade (dark blue) scenarios.

5.4 Conclusions

This chapter explored the early potential of real-time controlled bioretention cells. Using a controlled bioretention cell as motivation, phosphorus removal was simulated for a variety of influent concentrations and storm conditions. Three scenarios were evaluated including the baseline, autonomous upgrade, and passive upgrade. Both the autonomous and passive upgrades improved pollutant removal. Future work should evaluate the benefits of a combined autonomous-passive upgrade and more mathematically complex control algorithms.

The autonomous upgrade was shown to release at least two times less phosphorus load than the baseline during all design storm simulations. Therefore, the autonomous upgrade matched the pollutant treatment performance of the baseline in half the spatial footprint for the system studied here. These findings need to be generalized but may stand to benefit stormwater managers who often cannot design retention systems to the recommended size due to site or cost constraints.

Water quality goals (e.g. phosphorus removal) can be achieved by adding real-time control as illustrated in both the design storm and dynamic storm analyses. Not only does real-time control potentially provide a "digital" alternative to existing, passive upgrades, like soil amendments, but it also provides long-term management flexibility. This flexibility enables stormwater managers to dynamically balance trade-offs in existing bioretention designs and aids in the larger goal of system-level control. A real-world experiment is necessary to validate these findings in-situ.

CHAPTER 6

Conclusion

6.1 Summary of Contributions

The objective of this dissertation was to advance autonomous green stormwater infrastructure. To that end, this dissertation tackled both theoretical and technological knowledge gaps, which ultimately led to a number of fundamental contributions, including, but not limited to:

- **Chapter 2:** We introduced a low-cost, low-maintenance sensor for real-time, high-resolution GI monitoring. When coupled with an automated data toolchain, we showed how investments in monitoring networks support a more targeted and data-driven approach to GI design, placement, and maintenance.
- **Chapter 3:** We introduced a comprehensive sensor placement methodology for urban drainage networks that is flexible enough to work for any sensor or spatial parameter of interest, has guarantees of optimality, and is easy and efficient to use. We showed the mutual information criterion maximizes information gained while minimizing the size of the sensor network.
- **Chapter 4:** We introduced *StormReactor*, a Python package that improves the fidelity of modeling pollutant transformations and pollutant-based real-time control. We showed that the improved fidelity enables the development and testing of real-time control algorithms that use pollutant concentration, load, and/or sensor data.

- **Chapter 5:** Using *StormReactor*, we discovered that real-time control not only provides a "digital" alternative to existing, passive GI upgrades, like soil amendments, but it also provides long-term management flexibility. This flexibility enables stormwater managers to dynamically balance trade-offs in existing bioretention designs and aids in the larger goal of system-level control.

6.2 Future Research Directions

While the contributions of this dissertation move us closer to autonomous green stormwater infrastructure systems, work remains to continue to accelerate the adoption of this vision. The following sections present some promising future research directions towards this end, with more details and other directions highlighted in the respective chapters.

6.2.1 Analyzing Long-term GI Performance

The reliability of the GI sensor introduced in Chapter 2 should enable long-term data collection with reduced measurement overhead. Long-term GI performance has not yet been investigated at the temporal and spatial scales enabled by the GI sensor. Future research should examine how the decay constants vary over time to determine seasonal and long-term changes. In addition, these long-term datasets may be able to be used to inform maintenance schedules. Slowed drainage may indicate the GI soil media is clogged and should be replaced. A science-based method to validate such scenarios should be investigated.

6.2.2 Empirical GI Design and Placement Guidelines

Future work should investigate how the GI measurements from Chapter 2 can be used to inform future GI design and placement. The methodology presented could

be used to create empirical design guides, such as an empirical “heatmap” shown in Figure 2.6a. The empirical "heatmap" could be used to inform future GI design and placement in Detroit. For example, the design (i.e., DA/SA ratio) and placement (i.e., depth to groundwater) of a future GI asset could be selected based on the desired drawdown rate. A new sensing modality should be explored and deployed to validate the approach. The results could then be used to iterate on site design.

The mapping tool introduced in Chapter 3 provides another approach for strategic placement of GI assets. Engineers and city planners could use the map to scientifically determine where future GI development should be prioritized. For example, the predicted drawdown map showed faster drainage occurs along the northern border of Detroit. New GI could be built and monitored in this region. These sites can then be used to not only validate the predictive map’s output, but also to validate the predictive map’s ability as a planning tool.

6.2.3 Iterative Sensor Placement

Future work should validate the algorithm-selected sensor locations in the broader Detroit GI landscape. To do this, sensors should be installed in at least the "Priority 1" locations (Figure 3.6). Once data has been collected for a sufficient period of time, the data should be analyzed using the automated data toolchain introduced in Chapter 2. The new decay constants can then be used to retrain the GP model. This work should show that the new GP model is more accurate than the previous model, illustrating the iterative nature of the sensor placement.

6.2.4 Improving StormReactor

Future work should address the limitations of *StormReactor*. Presently, *StormReactor* does not support LID water quality processes, LID real-time control, or high spatial resolution water quality processes. To add access to a GI’s water quality

processes, SWMM's LID module (`lid.c`) must be modified. Currently, SWMM reads the percent removal for each pollutant and LID from the input file using `readRemovalsData()`. The percent removal is then used by `lid_addDrainRunon()` to update the LID's pollutant concentration. The pollutant loads are then updated with `lid_addDrainLoads()`. These functions would need to be modified so that instead of using the percent removal provided in the input file, they would use the user provided concentrations computed using *StormReactor*. In addition, the ability to control the LID's underdrain during the simulation must be added. We investigated changing the underdrain's coefficient in the input file to simulate opening and closing the underdrain, however, this method is not computationally stable. Therefore, a different method must be developed.

Adding the ability to model high spatial resolution water quality processes, like advection or diffusion, in *StormReactor* will prove more challenging. SWMM models nodes and links as completely mixed stirred tank reactors; assuming the concentration is constant throughout the entirety of a node or link. This assumption is contrary to high spatial resolution water quality processes. One approach to model advection may be to discretize links or nodes into smaller sections. Each section would remain completely mixed, but advection could be estimated by tracking the pollutant as it moves from one section to the next. To do this, a Python function could be added to read the link and node information from a SWMM input file and then update the input file with discretized links and nodes before running the simulation.

6.2.5 Next Steps for Autonomous GI

StormReactor now enables the development and testing of real-time control algorithms that use pollutant concentration, load, and sensor data. Future work should explore the use of formal control theory to explore emergent behavior, stability, and optimal control strategies at both the site and watershed scale [183]. We can then use

this information to optimize asset treatment performance, pushing our watersheds to behave like distributed water treatment plants, and ultimately improve watershed water quality.

Chapter 5 illustrated that real-time control provides long-term management flexibility by enabling the GI to be "reprogrammed". Future work should investigate how to reprogram GI to address different pollutants. Not only will individual pollutants need to be targeted, but several pollutants may need to be targeted at once, requiring the tailoring of retention times to competing objectives. To handle these competing objectives, more complex control algorithms will likely be needed. Beyond the site-scale, future work must investigate how to coordinate a network of autonomously upgraded sites to meet system-level water quality goals.

Future work should also evaluate the benefits of a combined autonomous-passive upgrade. Adding soil amendments would improve the passive treatment of pollutant treatment while real-time control would provide the ability to control outflows and residence time. Together, these upgrades should push the performance boundary of GI even further. Our simulated study showed that the autonomous upgrade matched the pollutant treatment performance of the baseline in half the spatial footprint for the system studied. This result may enable resource-scarce water managers to design GI smaller without compromising performance. These findings should be validated with a real-world experiment.

APPENDIX A

GI Sensor Network

The GI sensor was validated by an outside consultant [98] using a gage plate and camera (Figure A.1, Figure A.2). For more details, please refer to Dierks [98].

Figure A.1: An example of sensor and camera-measured depths in one GI asset from the validation study completed by an outside consultant.

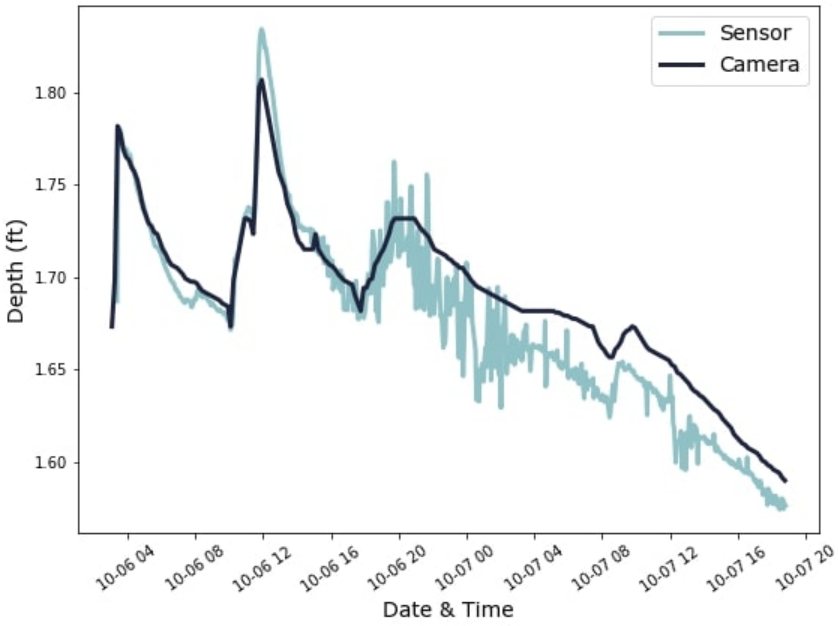

Figure A.2: The sensor validation setup using a gage plate and camera

Figure A.4 provides a list of the 14 monitored GI sites with their design features and physiographic features. The site type refers to if the site is a rain garden (RG) or a bioretention cell (BRC). The land use type refers to if the site is classified as a developed low intensity (DLI), developed medium intensity (DMI), or developed high intensity (DHI). The installation and maintenance dates for these sites are provided in Table A.1. To analyze rainfall at the GI sites, data was obtained from eight rain gauges in Detroit (Figure A.3). The closest gauge to each site was selected.

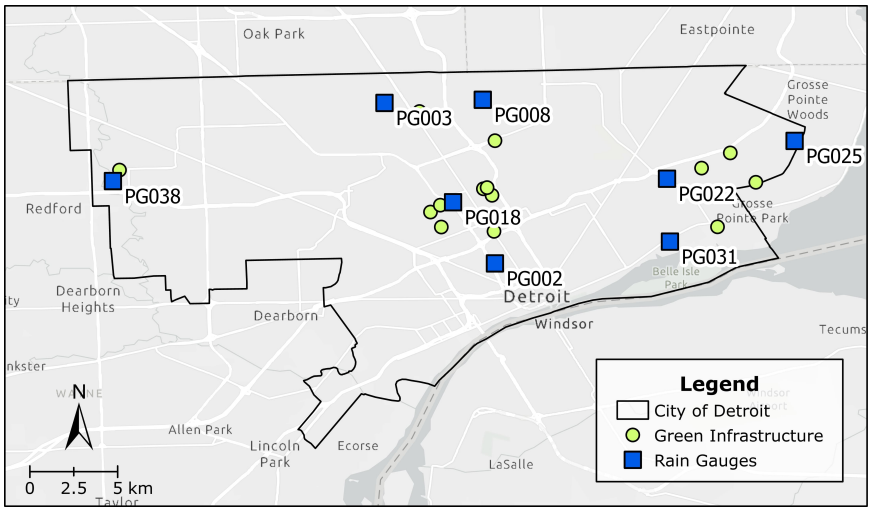

Figure A.3: The eight Detroit rain gauges used to analyze rainfall.

Site ID	Deployment	Maintenance
S1	6/11/21	N/A
S2	6/18/21	N/A
S3	6/11/21	N/A
S4	6/11/21	N/A
S5	7/02/21	N/A
S6	6/11/21	N/A
S7	6/11/21	N/A
S8	7/02/21	N/A
S9	6/18/21	N/A
S10	6/18/21	N/A
S11	6/11/21	N/A
S12	6/18/21	8/11/21
S13	6/11/21	N/A
S14	6/18/21	N/A

Table A.1: Records of the field visits (deployment and maintenance) for the 14 GI devices.

Site	Site Type	Surface Area (sq. m)	Drainage Area (sq. m)	Drainage Area to Surface Area Ratio	Storage Volume (cu. m)	GI Depth (m)	Install Year	Percent Impervious	Land Use Type	Latitude	Longitude	Elevation (m)	Slope (%)	Hydrologic Soil Group	Depth to Groundwater (m)	Decay Constant (1/hr)
S1	RG	15.8	60.4	3.8	3.5	0.3	2020	53%	DMI	42.38413	-83.07061	191.76	1.85	C	5.99	-0.040
S2	RG	18.6	241.5	13.0	2.2	0.3	2017	71%	DMI	42.387459	-83.075043	192.31	1.12	A	6.35	-0.011
S3	RG	16.4	23.2	1.4	1.0	0.3	2017	59%	DMI	42.36798	-83.09664	192.62	0.19	C	5.77	-0.044
S4	RG	27.9	173.7	6.2	6.6	0.3	2018	60%	DMI	42.42695	-83.10778	194.77	2.02	C	11.1	-0.305
S5	RG	11.1	23.2	2.1	2.5	0.3	2018	52%	DMI	42.38802	-83.07315	192.21	1.68	C	6.36	-0.146
S6	BRC	260.9	3106.7	11.9	14.1	1.0	2017	98%	DHI	42.36578	-83.06965	192.68	0.94	D	5.02	-0.024
S7	RG	14.9	40.5	2.7	3.3	0.3	2020	69%	DMI	42.37563	-83.10217	193.98	0.11	C	5.88	-0.069
S8	RG	20.4	49.1	2.4	4.1	0.3	2015	46%	DLI	42.39698	-83.26136	192.1	0.60	C	8.46	-0.397
S9	RG	39.0	163.5	4.2	8.9	1.0	2018	87%	DHI	42.40581	-82.94867	182.28	2.04	D	6.79	0.102
S10	RG	21.4	24.5	1.1	4.7	0.3	2019	61%	DMI	42.39074	-82.93565	178.21	0.42	D	6.37	0.119
S11	RG	9.3	53.0	5.7	0.9	0.3	2017	54%	DMI	42.37914	-83.09715	194.13	1.07	C	6.03	-0.200
S12	BRC	138.9	1740.1	12.5	17.3	1.0	2019	61%	DLI	42.39801	-82.96338	182.88	3.51	D	6.76	-0.047
S13	RG	33.4	371.6	11.1	8.0	1.00	2019	98%	DHI	42.41205	-83.06913	192.4	0.64	D	8.63	-0.072
S14	RG	13.9	40.1	2.9	1.6	0.3	2016	68%	DMI	42.36806	-82.95522	175.3	0.69	D	5.95	-0.021

Figure A.4: List of the 14 monitored GI and their design and physiographic features.

APPENDIX B

GIS

B.1 GIS Data Pre-Processing

The following steps were taken to download and pre-process the GIS datasets used in the correlation analysis using ArcGIS Pro. Table B.1 provides the details on the GIS datasets including the year, source, type, and resolution.

1. Added a CSV file with the spatial coordinates and decay constants of each GI location.
2. The GI data was displayed using the Display XY Data tool. Set the X field as longitude and the Y field as Latitude.
3. Reprojected the GI layer using the Project tool to GCS_WGS_1984.
4. Downloaded the City of Detroit Boundary JSON file from `https://data.detroitmi.gov/datasets/detroitmi::city-of-detroit-boundary/about`.
5. Converted Detroit boundary to shapefile and deleted center cutout of Hamtramck and Highland Park using the Edit Vertices tool.
6. Downloaded 3m (1/9th arc second) elevation data for Wayne County, Michigan, US from `https://earthexplorer.usgs.gov` [184]
7. Combined the eleven elevation images into one using the Mosaic to New Raster tool.

8. Added the USA SSURGO – Soil Hydrologic Group [185], USA NLCD Land Cover [186], and USA NLCD Impervious Surface Time Series [186] raster datasets from the ArcGIS Virtual Portal.

9. Reprojected the four raster datasets using the Project Raster tool to GCS_WGS_1984.

10. Clipped the four raster layers to Detroit boundary shapefile the Extract by Mask tool

11. Made all four raster files have the same resolution (30 m) using the Resample tool.

12. Used the elevation layer to create a new slope layer using the Surface Parameters tool [187]. Selected the quadratic option (which is default and recommended option for most data and applications), the default calculated neighborhood distance, the z unit was set to meter, and the output slope measurement was set for percent rise.

13. The land cover layer contains categorical data, so we needed to change these to numerical values for the correlation analysis using the Reclassify tool. The "High Developed Intensity" value was set to 3, "Medium Developed Intensity" to 2, "Low Developed Intensity" to 1, and all other categories to 1 since they are most closely related to "Low Developed Intensity". "Open Water" was set to NODATA since we cannot install GI there.

14. Obtained the well data for Michigan from the State of Michigan's Water Well Viewer [188], Wellogic System [189], and the US Geologic Survey's Groundwater Watch [190]. Combined the three well datasets into one Excel file. Plotted histogram of static water level to check for outliers, kurtosis, and skewness to show it's a normal distribution. The CSV file was then added to ArcGIS

Pro. The data is displayed using the Display XY Data tool. Set the X field as longitude and the Y field as Latitude.

15. Reprojected the wells layer using the Project tool to GCS_WGS_1984.

16. Interpolated groundwater levels using the Empirical Bayesian Kriging tool. The output cell size was set to the same size as the other raster datasets (30 m). Data transformation type was none and the semivariogram type was Power. Additional model parameters were 50 for the maximum # of points in each local model; 1 for the local model area overlap factor; and 1,000 for the number of simulated semivariograms. The search neighborhood parameters used a Smooth Circular search neighborhood with a smoothing factor of 0.85 and the default calculated radius (21,399 m). Used GA Layer to Rasters to convert the geostatistical layer to a raster file for both the prediction and the prediction standard error. Masked it to the Detroit boundary shapefile and selected GCS_WGS_1984 as the projection.

17. Used the Extract Multi Values to Points tool to extract the values from each raster layer (elevation, slope, HSG, groundwater, imperviousness, land use type) at the GI locations.

18. Converted this data to an Excel file using the Table to Excel tool and loaded it into Python for the correlation analysis.

B.2 Groundwater Interpolation

Detroit has a shallow groundwater system, with the groundwater table being one to three meters below the surface in some regions [107]. For this reason, it is critical to include groundwater in the analysis. To the best of our knowledge, the only

Dataset	Year	Source	Type	Resolution
City of Detroit Boundary	2021	City of Detroit	Vector	N/A
National Elevation Dataset 1/9 Arc Second [184]	2017	USGS	Raster	3 m
USA SSURGO Soil Hydrologic Group [185]	2021	Esri	Raster	30 m
USA NLCD Land Cover [186]	2019	Esri	Raster	30 m
USA NLCD Impervious Surface [186]	2019	Esri	Raster	30 m
Well Records [188]	2021	State of Michigan	CSV	N/A
Well Records [189]	2021	State of Michigan	CSV	N/A
Well Records [190]	2021	US Geologic Survey	CSV	N/A

Table B.1: Details on the GIS datasets.

available groundwater data for the State of Michigan are a collection of water well records which provide the static water level (ft), or depth to the groundwater, for each well. Three sets of well records were found: (1) Water Well Viewer by the State of Michigan's Department of Environmental Quality [188]; (2) Wellogic System by the State of Michigan's Department of Environment, Great Lakes and Energy (previously the Department of Environmental Quality) [189]; and (3) Groundwater Watch by the US Geologic Survey [190]. It is important to note that although the derived data in these files represents the best readily available data, they do not represent a complete database of all wells or well records in existence. The well records include three CSV files with each well's ID, location (latitude, longitude), static water level, and other data that is irrelevant for this analysis. A limitation of these records is that there is a single static water level reading for each well. And since groundwater fluctuates, the variation in groundwater is missing, creating some uncertainty.

We want to use the groundwater data to see if there is a correlation between GI drawdown rates and the depth to groundwater. To do this, we need an estimate of

Total Wells	1670
Mean (ft)	34.6
St. Dev. (ft)	22.7
Max (ft)	120.0
Min (ft)	1.0
Kurtosis	0.93
Skewness	0.96

Table B.2: Groundwater well record summary statistics.

groundwater depth for each GI, which requires the well records need to be manipulated into a usable form. The three water well records are combined into a single Excel file. A histogram of the static water level is plotted to check for outliers (Figure B.1). In addition, summary statistics are computed (Table B.2). Since the values for kurtosis and skewness are between $\pm$ 2, this is considered acceptable to prove the data follows a normal univariate distribution [191]. Therefore, the full dataset is used (no outliers removed), and no data transformations are used. The dataset is added to ArcGIS Pro using the Excel to Table tool, displayed using the Display XY Data tool, and then reprojected to "GCS WGS 1984". The next step is to interpolate the static water level across Detroit.

Kriging is an accepted method of estimating groundwater at sites where the water level data are available but where there may be insufficient additional data necessary for groundwater flow modeling [192]. Traditional kriging methods estimate a variogram that is considered the true variogram of the observed data without explicitly considering uncertainty. Recently, a new form of kriging, Empirical Bayesian Kriging (EBK), has been shown to perform better than other types of kriging methods [193]. It has been successfully used to evaluate inter-annual water-table evolution in Mexico [194] and to quantify uncertainty in groundwater modeling [195]. The fundamental advantage of EBK over classical kriging methods is that it creates a spectrum of var-

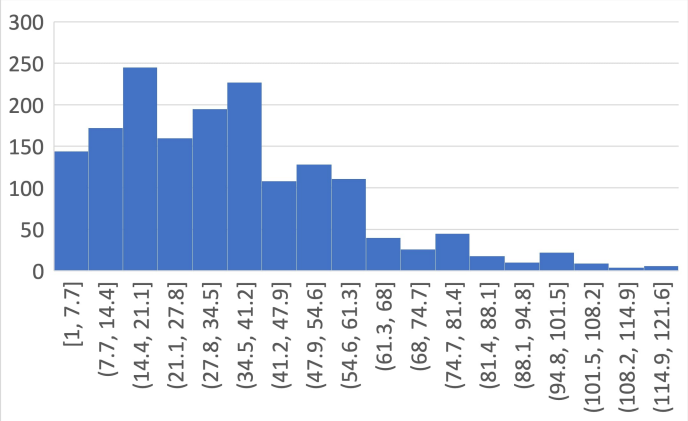

Figure B.1: Histogram of the static water level (ft) in the Detroit region water wells.

iograms which account for the uncertainty introduced by estimating a variogram in the first place [194]. Therefore, EBK is used to interpolate groundwater following the methodology of Li et al. [194].

In ArcGIS Pro, the EBK tool is selected with the groundwater point data as the input feature. The Z value field was set to groundwater depth. The output cell size is set to the same size as the other raster datasets (30 m). The data transformation type is set to "None" because the data is normally distributed. There are three semivariogram options when the data transformation is "None": power, linear, and thin plate spline. Power is selected because it is relatively fast, flexible, and balances performance and accuracy. The search neighborhood parameters used a "Smooth Circular" search neighborhood with a smoothing factor of 0.85 and the default calculated radius (21,399 m) [194]. ArcGIS Pro's leave-one-out cross-validation is used to find the remaining model parameters: maximum number of points in each local model, the local model area overlap factor, and the number of simulated semivariograms. Following the methodology of Li et al. [194], we calculate the mean error (ME), root mean square error (RMSE), average standard error (ASE), mean standardized error

Check if the following hold:	**The prediction variability is:**
ASE $\approx$ RMSE and RMSE$\approx$ 1	Correctly assessed
ASE $>$ RMSE and RMSSE $<$ 1	Overestimated
ASE $<$ RMSE and RMSSE $>$ 1	Underestimated

Table B.3: Groundwater well record summary statistics.

Inside 90% Interval	91.1
Inside 95% Interval	94.4
Mean	0.230
RMSE	16.7
MSE	0.00979
RMSSE	0.998
ASE	16.4

Table B.4: ArcGIS Pro cross-validation report for the optimal parameter values.

(MSE), and root mean square standardized error (RMSSE) for each subset of parameters. The different errors from cross-validation ware analyzed with the rules in Table B.3 to assess the variability of predictions and evaluate the performance (under or over-estimation) of the EBK model.

Cross-validation results found the optimal parameters to be: 50 for the maximum number of points in each local model; 1 for the local model area overlap factor; and 1,000 for the number of simulated semivariograms. The interpolation errors suggest the EBK model performs correctly and does not under or overestimate groundwater (Table B.4). As a secondary check, Li et al. [194] reported an RMSE $\approx$ ASE $\approx$ 13.16-14.43, and our values are close (16.4-16.7). Figure B.2 shows the predicted versus true groundwater depths.

The EBK model with the optimal parameters is used to interpolate groundwater across Detroit. The GA Layer to Rasters tool is used to convert the geostatistical layer to a raster file for both the prediction and the prediction standard error. The

output is set to be masked to the Detroit boundary shapefile and projected to "GCS WGS 1984". The interpolated groundwater along with the groundwater wells are shown in Figure B.3. The interpolated groundwater map aligns with the literature. Teimoori et al. [107] found that the depth to groundwater is deepest in the northwest and gradually decreases as you move southeast. The standard error of prediction plot shows the errors are higher in areas where well data does not exist (Figure B.4).

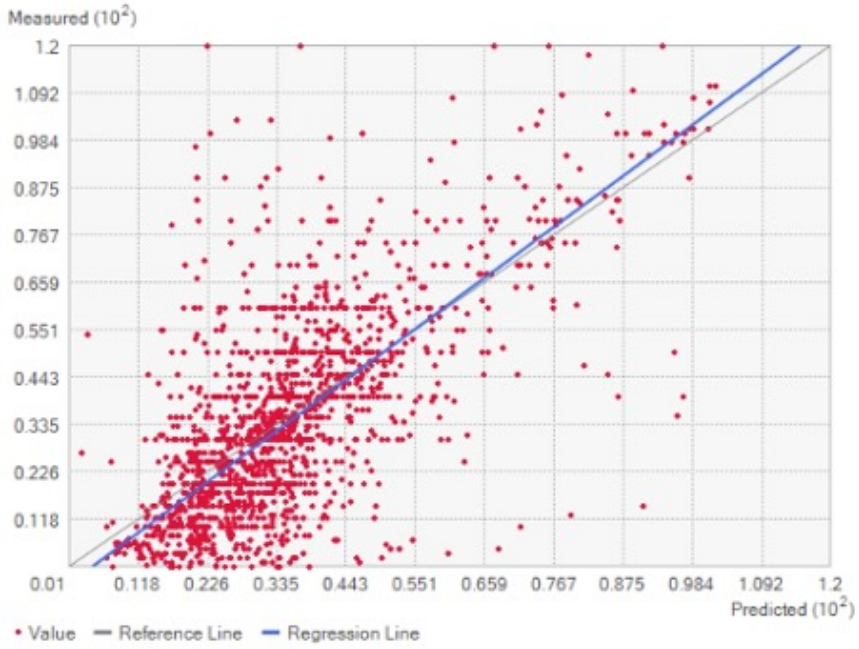

Figure B.2: The predicted versus true groundwater depth from the EBK model.

The final steps are to prepare the data for the correlation analysis. The Extract Multi Values to Points tool is used to extract the values from the groundwater raster layer at the GI locations and then it is converted to an Excel file using the Table to Excel tool. Then the groundwater data is also converted into an Excel file using the Table to Excel tool. Finally, these files are loaded into Python.

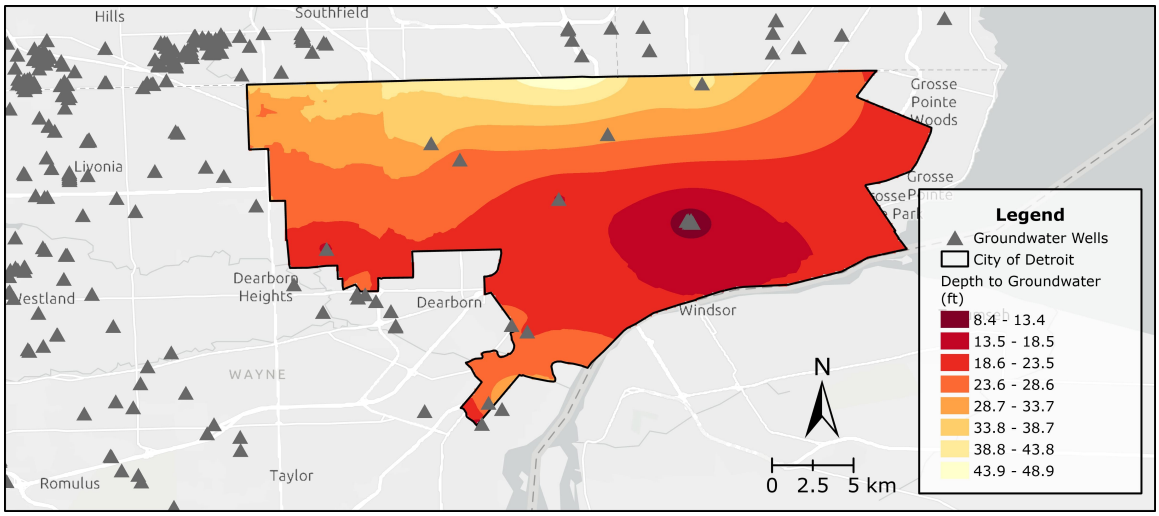

Figure B.3: Detroit's groundwater wells and interpolated groundwater depth (ft).

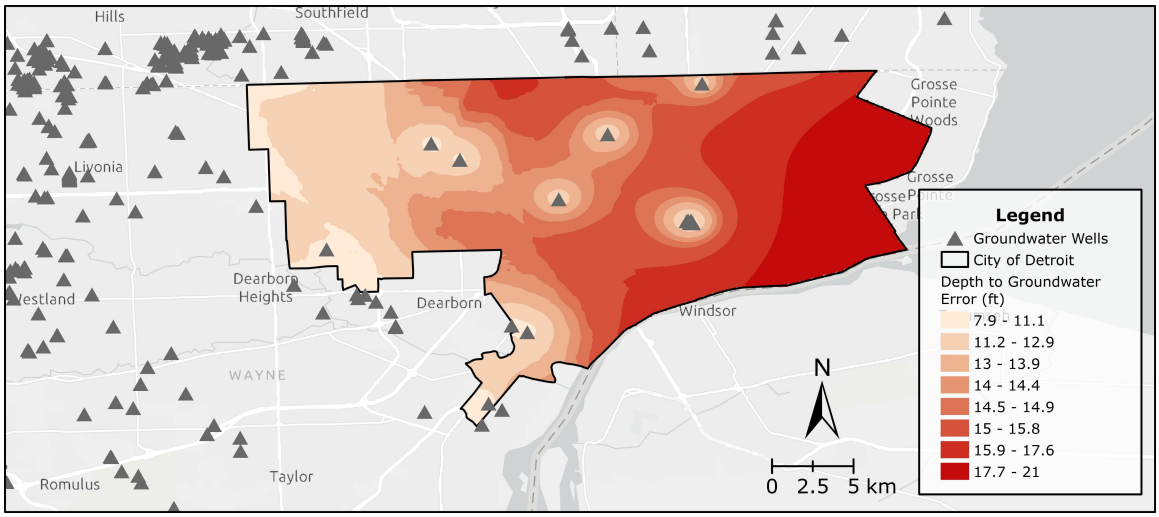

Figure B.4: The standard error of prediction of the interpolated groundwater depth.

APPENDIX C

Gaussian Processes and Sensor Placement

Algorithm Approximation algorithm for maximizing mutual information using lazy evaluation.

Input : covariance matrix $\mathbf{K}$, number of sensors to select k, set of location to choose from $\mathcal{V}$
Output : selected sensors $\mathcal{A} \subset \mathcal{V}$
begin
  $\mathcal{A} \leftarrow \emptyset$;
  foreach $y \in \mathcal{V} \backslash \mathcal{A}$ **do** $\delta_y \leftarrow +\infty$;
  for $j = 1$ *to* k **do**
    foreach $y \in \mathcal{V} \backslash \mathcal{A}$ **do** $current_y \leftarrow$ **false**;
    while true do
      $y^* \leftarrow \text{argmax}_{y \in \mathcal{V} \backslash \mathcal{A}} \delta_y$;
      if $current_{y^*}$ **then break**;
      $\delta_{y^*} \leftarrow H(y \mid \mathcal{A}) - H(y \mid \mathcal{V} \backslash \mathcal{A})$;
      $current_{y^*} \leftarrow$ **true**
    $\mathcal{A} \leftarrow \mathcal{A} \cup y^*$;
end

Figure C.1: Approximation algorithm for maximizing mutual information using lazy evaluation.

Input Feature	**Minimum**	**Mean ± Std Dev**	**Maximum**
groundwater depth	5.02	6.66±1.46	11.10
DA/SA ratio	1.10	5.00±3.43	13.00
longitude	−83.26	−83.05±0.07	−82.94
soil media depth	0.30	0.45±0.29	1.00
imperviousness	39.00	64.23±16.93	98.00
decay constant α	−0.503	−0.119±0.124	−0.006

Table C.1: The range of the input features for training GP_{og}.

Input Feature	**Minimum**	**Mean ± Std Dev**	**Maximum**
groundwater depth	2.44	7.63±2.01	14.63
DA/SA ratio	N/A	4.00	N/A
longitude	−83.29	−83.10±0.09	−82.91
soil media depth	N/A	0.30	N/A
imperviousness	0.00	63.92±23.06	100.00

Table C.2: The range of the input features for predicting decay constant α across Detroit.

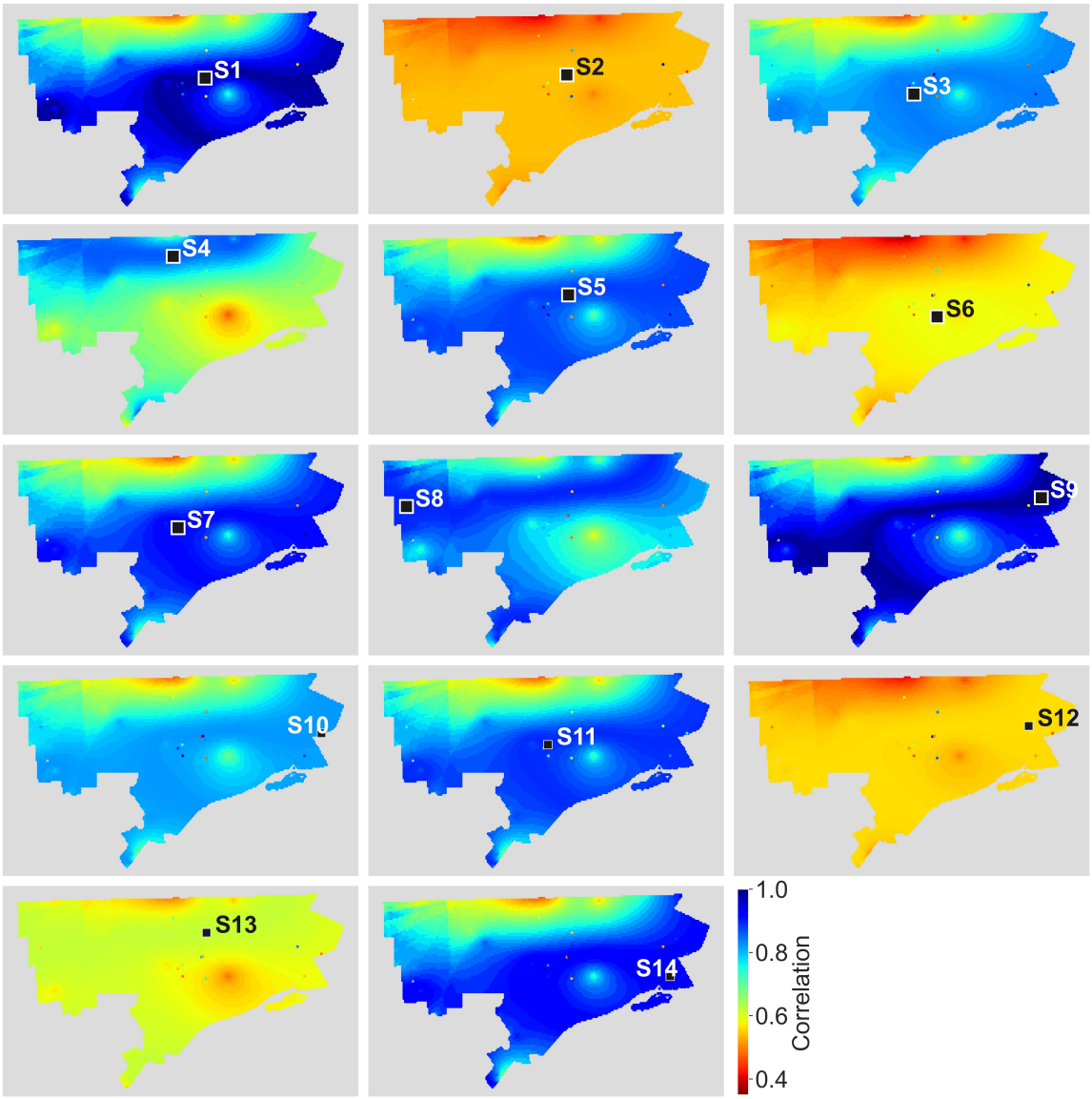

Figure C.2: The correlation between each sensor and the entire prediction space.

BIBLIOGRAPHY

[1] Shannon Doocy, Amy Daniels, Sarah Murray, and Thomas D. Kirsch. The Human Impact of Floods: a Historical Review of Events 1980-2009 and Systematic Literature Review. *PLoS Currents*, 2013. Publisher: Public Library of Science.

[2] L. Lundy, J.B. B. Ellis, and D.M. M. Revitt. Risk prioritisation of stormwater pollutant sources. *Water Research*, 46(20):6589–6600, 2012. Publisher: Elsevier Ltd.

[3] Yun-Ya Yang and Mary G. Lusk. Nutrients in Urban Stormwater Runoff: Current State of the Science and Potential Mitigation Options. *Current Pollution Reports*, 4(2):112–127, June 2018.

[4] Laurel Christian, Thomas Epps, Ghada Diab, and Jon Hathaway. Pollutant Concentration Patterns of In-Stream Urban Stormwater Runoff. *Water*, 12(9):2534, September 2020.

[5] Lawrence A Baker. Introduction to nonpoint source pollution in the United States and prospects for wetland use. *Ecological Engineering*, 1(1):1–26, March 1992.

[6] Tingju Zhu, Jay R Lund, Marion W Jenkins, Guilherme F Marques, and Randall S Ritzema. Climate change, urbanization, and optimal long-term floodplain protection. *Water Resour. Res*, 43, 2007.

[7] United Nations, Department of Economic and Social Affairs, Population Division. World Population Prospects 2022: Methodology of the United Nations population estimates and projections. Technical Report UN DESA/POP/2022/TR/NO. 4, United Nations, 2022.

[8] David M Theobald, Scott J Goetz, John B Norman, and Patrick Jantz. Watersheds at Risk to Increased Impervious Surface Cover in the Conterminous United States. *Journal of Hydrologic Engineering*, 14(4), 2009.

[9] Lea Rosenberger, Jorge Leandro, Stephan Pauleit, and Sabrina Erlwein. Sustainable stormwater management under the impact of climate change and urban densification. *Journal of Hydrology*, 596:126137, May 2021.

[10] Narae Kang, Soojun Kim, Yonsoo Kim, Huiseong Noh, Seung Hong, and Hung Kim. Urban Drainage System Improvement for Climate Change Adaptation. *Water*, 8(7):268, June 2016.

[11] Lizhu Wang, John Lyons, Paul Kanehl, and Roger Bannerman. Impacts of Urbanization on Stream Habitat and Fish Across Multiple Spatial Scales. *Environmental Management*, 28(2):255–266, August 2001.

[12] Eric A. Rosenberg, Patrick W. Keys, Derek B. Booth, David Hartley, Jeff Burkey, Anne C. Steinemann, and Dennis P. Lettenmaier. Precipitation extremes and the impacts of climate change on stormwater infrastructure in Washington State. *Climatic Change*, 102(1-2):319–349, September 2010.

[13] Branko Kerkez, Cyndee Gruden, Matthew Lewis, Luis Montestruque, Marcus Quigley, Brandon P. Wong, Alex Bedig, Ruben Kertesz, Tim Braun, Owen Cadwalader, Aaron Poresky, and Carrie Pak. Smarter Stormwater Systems. *Environ. Sci. Technol.*, 50(14):7267–7273, July 2016.

[14] C. Zevenbergen, W. Veerbeek, B. Gersonius, and S. Van Herk. Challenges in urban flood management: travelling across spatial and temporal scales: Challenges in urban flood management. *Journal of Flood Risk Management*, 1(2):81–88, July 2008.

[15] Qianqian Zhou. A Review of Sustainable Urban Drainage Systems Considering the Climate Change and Urbanization Impacts. *Water*, 6(4):976–992, April 2014.

[16] Thomas F. Waters. *Sediment in streams: sources, biological effects, and control*. Number 7 in American Fisheries Society monograph. American Fisheries Society, Bethesda, Md, 1995.

[17] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, September 2013.

[18] Abhiram Mullapudi, Matthew Bartos, Brandon P. Wong, and Branko Kerkez. Shaping streamflow using a real-time stormwater control network. *Sensors*, 18(2259), 2018.

[19] Matthew Bartos, Brandon P. Wong, and Branko Kerkez. Open storm: a complete framework for sensing and control of urban watersheds. *Water Research & Technology*, 4(346), 2018.

[20] Emily Zechman Berglund, Jacob G. Monroe, Ishtiak Ahmed, Mojtaba Noghabaei, Jinung Do, Jorge E. Pesantez, Mohammad Ali Khaksar Fasaee, Eleni Bardaka, Kevin Han, Giorgio T. Proestos, and James Levis. Smart Infrastructure: A Vision for the Role of the Civil Engineering Profession in Smart Cities. *Journal of Infrastructure Systems*, 26(2), jun 2020.

[21] Sven Eggimann, Lena Mutzner, Omar Wani, Mariane Yvonne Schneider, Dorothee Spuhler, Matthew Moy De Vitry, Philipp Beutler, and Max Maurer. The Potential of Knowing More: A Review of Data-Driven Urban Water Management. *Environmental Science and Technology*, 51(5):2538–2553, 2017.

[22] Brandon P. Wong and Branko Kerkez. Real-Time Control of Urban Headwater Catchments Through Linear Feedback: Performance, Analysis, and Site Selection. *Water Resources Research*, 54(10):7309–7330, oct 2018.

[23] Michael E. Dietz. Low impact development practices: A review of current research and recommendations for future directions. *Water, Air, and Soil Pollution*, 186:351–363, 2007.

[24] William F. Hunt, Allen P. Davis, and Robert G. Traver. Meeting Hydrologic and Water Quality Goals through Targeted Bioretention Design. *Journal of Environmental Engineering*, 138(6):698–707, 2012.

[25] Laurent M. Ahiablame, Bernard A. Engel, and Indrajeet Chaubey. Effectiveness of Low Impact Development Practices: Literature Review and Suggestions for

Future Research. *Water, Air, & Soil Pollution*, 223(7):4253–4273, September 2012.

[26] Allen P. Davis. Field Performance of Bioretention: Hydrology Impacts. *Journal of Hydrologic Engineering*, 13(2):90–95, 2008.

[27] John Mathews. Post-Construction Stormwater Management Practices. In *Rainwater and Land Development: Ohio's Standards for Stormwater Management, Land Development and Urban Stream Protection*, pages 69–84. Ohio Department of Natural Resources, Division of Soil and Water Conservation, Columbus, Ohio, 3 edition, 2021.

[28] Trisha L. Moore, John S. Gulliver, Latham Stack, and Michael H. Simpson. Stormwater management and climate change: vulnerability and capacity for adaptation in urban and suburban contexts. *Climatic Change*, 138(3-4):491–504, October 2016.

[29] E. Gregory McPherson. Accounting for benefits and costs of urban greenspace. *Landscape and Urban Planning*, 22(1):41–51, September 1992.

[30] Jari Lyytimäki and Maija Sipilä. Hopping on one leg – The challenge of ecosystem disservices for urban green management. *Urban Forestry & Urban Greening*, 8(4):309–315, January 2009.

[31] Vinicius J Taguchi, Peter T Weiss, John S Gulliver, Mira R Klein, Raymond M Hozalski, Lawrence A Baker, Jacques C Finlay, Bonnie L Keeler, and John L Nieber. It Is Not Easy Being Green : Recognizing Unintended Consequences of Green Stormwater Infrastructure. *Water*, 12(522), 2019.

[32] Basanta Kumar Biswal, Kuppusamy Vijayaraghavan, Max Gerrit Adam, Daryl Lee Tsen-Tieng, Allen P. Davis, and Rajasekhar Balasubramanian. Biological nitrogen removal from stormwater in bioretention cells: a critical review. *Critical Reviews in Biotechnology*, 0(0):1–23, September 2021.

[33] Yaoze Liu, Bernard A. Engel, Dennis C. Flanagan, Margaret W. Gitau, Sara K. McMillan, and Indrajeet Chaubey. A review on effectiveness of best management practices in improving hydrology and water quality: Needs and opportunities. *Science of the Total Environment*, 601-602:580–593, 2017.

[34] Allen P Davis, Mohammad Shokouhian, Himanshu Sharma, and Christie Minami. Water Quality Improvement through Bioretention Media: Nitrogen and Phosphorus Removal. *Water Environment Research*, 78(3):284–293, 2006.

[35] Heather E. Golden and Nahal Hoghooghi. Green infrastructure and its catchment-scale effects: an emerging science. *Wiley Interdisciplinary Reviews: Water*, 5(1):e1254–e1254, January 2018.

[36] Allen P. Davis, Robert G. Traver, William F. Hunt, Ryan Lee, Robert A. Brown, and Jennifer M. Olszewski. Hydrologic Performance of Bioretention Storm-Water Control Measures. *Journal of Hydrologic Engineering*, 17(5):604–614, May 2012.

[37] Ryan J. Winston, Jay D. Dorsey, and William F. Hunt. Quantifying volume reduction and peak flow mitigation for three bioretention cells in clay soils in northeast Ohio. *Science of The Total Environment*, 553:83–95, 2016.

[38] B.E. E Hatt, T.D. Fletcher, and A. Deletic. Hydraulic and pollutant removal performance of stormwater filters under variable wetting and drying regimes. *Water Science & Technology*, 56(12):11–19, 2007.

[39] F Qiu, S Zhao, D Zhao, J Wang, and K Fu. Enhanced Nutrients Removal in Bioretention Systems Modified with Water Treatment Residual and Internal Water Storage Zone. *Environ. Sci.: Water Res. Technol*, 5:993–1003, 2019.

[40] Allen P. Davis, William F. Hunt, Robert G. Traver, and Michael Clar. Bioretention Technology: Overview of Current Practice and Future Needs. *Journal of Environmental Engineering*, 135(3):109–117, March 2009.

[41] Jia Liu, David Sample, Cameron Bell, and Yuntao Guan. Review and Research Needs of Bioretention Used for the Treatment of Urban Stormwater. *Water*, 6(4):1069–1099, April 2014.

[42] Allison H. Roy, Seth J. Wenger, Tim D. Fletcher, Christopher J. Walsh, Anthony R. Ladson, William D. Shuster, Hale W. Thurston, and Rebekah R. Brown. Impediments and Solutions to Sustainable, Watershed-Scale Urban Stormwater Management: Lessons from Australia and the United States. *Environmental Management*, 42(2):344–359, August 2008.

[43] New York City Environmental Protection. New York City Stormwater Manual. In *Chapter 19.1: Industrial, Commercial, Construction, and Post-Construction Stormwater Sources*. City of New York, New York City, New York, 2022.

[44] Water and Sewerage Department. Stormwater Management Design Manual, October 2022.

[45] F. Ahmed, R. Nestingen, J.L. Nieber, J.S. Gulliver, and R.M. Hozalski. A Modified Philip-Dunne Infiltrometer for Measuring the Field-Saturated Hydraulic Conductivity of Surface Soil. *Vadose Zone Journal*, 13(10):vzj2014.01.0012, October 2014.

[46] Brooke C. Asleson, Rebecca S. Nestingen, John S. Gulliver, Raymond M. Hozalski, and John L. Nieber. Performance Assessment of Rain Gardens. *JAWRA Journal of the American Water Resources Association*, 45(4):1019–1031, August 2009.

[47] Ali Ebrahimian, Kristin Sample-Lord, Bridget Wadzuk, and Robert Traver. Temporal and spatial variation of infiltration in urban green infrastructure. *Hydrological Processes*, 34(4):1016–1034, 2020.

[48] Kyle Eckart, Zach McPhee, and Tirupati Bolisetti. Performance and implementation of low impact development – A review. *Science of The Total Environment*, 607-608:413–432, December 2017.

[49] Robert A. Brown and William F. Hunt. Impacts of Construction Activity on Bioretention Performance. *Journal of Hydrologic Engineering*, 15(6):386–394, June 2010.

[50] U.S Environmental Protection Agency. Reducing stormwater costs through low impact development (LID) strategies and practices. Technical Report EPA 841-F-07-006, Nonpoint Source Control Branch, USEPA, Washington, D.C., 2007.

[51] N. Jackisch and M. Weiler. The hydrologic outcome of a Low Impact Development (LID) site including superposition with streamflow peaks. *Urban Water Journal*, 14(2):143–159, 2017.

[52] Kun Zhang and Ting Fong May Chui. Evaluating hydrologic performance of bioretention cells in shallow groundwater. *Hydrological Processes*, 31:4122–4135, 2017.

[53] V. Grace Mitchell. Applying Integrated Urban Water Management Concepts: A Review of Australian Experience. *Environmental Management*, 37(5):589–605, May 2006.

[54] Zachary Zukowski, Clay H. Emerson, Andrea L. Welker, and Brendon Achey. Evaluation of Field Hydraulic Conductivity Data: Comparing Spot Infiltrometer Test Data to Continuous Recession Data. In *Sustainable Materials and Resource Conservation*, Chicago, Illinois, 2016. American Society of Civil Engineers.

[55] Björn Kluge, Arvid Markert, Michael Facklam, Harald Sommer, Mathias Kaiser, Matthias Pallasch, and Gerd Wessolek. Metal accumulation and hydraulic performance of bioretention systems after long-term operation. *J Soils Sediments*, 18:431–441, 2018.

[56] Conor Lewellyn, Cara E. Lyons, Robert G. Traver, and Bridget M. Wadzuk. Evaluation of Seasonal and Large Storm Runoff Volume Capture of an Infiltration Green Infrastructure System. *Journal of Hydrologic Engineering*, 21(1):04015047, January 2016.

[57] Frank Blumensaat, João P Leitao, Christoph Ort, Jörg Rieckermann, Andreas Scheidegger, Peter A Vanrolleghem, and Kris Villez. How Urban Water Management Prepares for Emerging Opportunities and Threats: Digital Transformation, Ubiquitous Sensing, New Data Sources, and Beyond-a Horizon Scan. *Environ. Sci. Technol.*, 2019.

[58] G Cembrano, J Quevedo, M Salamero, V Puig, J Figueras, and J Martı. Optimal control of urban drainage systems. A case study. *Control Engineering Practice*, 12(1):1–9, January 2004.

[59] Congcong Sun, Vicenç Puig, and Gabriela Cembrano. Real-Time Control of Urban Water Cycle under Cyber-Physical Systems Framework. *Water*, 12(406):1–17, 2020.

[60] Hamidreza Kazemi, Thomas D. Rockaway, Joshua Rivard, and Sam Abdollahian. Assessment of Surface Infiltration Performance and Maintenance of Two Permeable Pavement Systems in Louisville, Kentucky. *Journal of Sustainable Water in the Built Environment*, 3(4):04017009, November 2017.

[61] Ting Meng and David Hsu. Stated preferences for smart green infrastructure in stormwater management. *Landscape and Urban Planning*, 187(June 2018):1–10, July 2019.

[62] Grazia Fattoruso, Annalisa Agresta, Guido Guarnieri, Bruno Lanza, Antonio Buonanno, Mario Molinara, Claudio Marrocco, Saverio De Vito, Francesco Tortorella, and Girolamo Di Francia. Optimal sensors placement for flood forecasting modelling. *Procedia Engineering*, 119:927–936, 2015.

[63] Bijit Kumar Banik, Leonardo Alfonso, Cristiana Di Cristo, Angelo Leopardi, and Arthur Mynett. Evaluation of different formulations to optimally locate sensors in sewer systems. *Journal of Water Resources Planning and Management*, 143(7), July 2017.

[64] R.I. Ogie, N. Shukla, F. Sedlar, and T. Holderness. Optimal placement of water-level sensors to facilitate data-driven management of hydrological infrastructure assets in coastal mega-cities of developing nations. *Sustainable Cities and Society*, 35:385–395, November 2017.

[65] J. Yazdi. Water quality monitoring network design for urban drainage systems, an entropy method. *Urban Water Journal*, 15(3):227–233, March 2018.

[66] Mariacrocetta Sambito, Cristiana Di Cristo, Gabriele Freni, and Angelo Leopardi. Optimal water quality sensor positioning in urban drainage systems for illicit intrusion identification. *Journal of Hydroinformatics*, 22(1):46–60, January 2020.

[67] Matt Bartos and Branko Kerkez. Observability-based sensor placement improves contaminant tracing in river networks. *Hydrology*, September 2020.

[68] Jiada Li. Exploring the potential of utilizing unsupervised machine learning for urban drainage sensor placement under future rainfall uncertainty. *Journal of Environmental Management*, 296:113191, October 2021.

[69] Hamed Farahmand, Xueming Liu, Shangjia Dong, Ali Mostafavi, and Jianxi Gao. A network observability framework for sensor placement in flood control networks to improve flood situational awareness and risk management. *Reliability Engineering & System Safety*, 221, May 2022.

[70] Christopher C. Obropta and Josef S. Kardos. Review of Urban Stormwater Quality Models: Deterministic, Stochastic, and Hybrid Approaches. *JAWRA Journal of the American Water Resources Association*, 43(6):1508–1523, dec 2007.

[71] Peter M. Bach, Wolfgang Rauch, Peter S. Mikkelsen, David T. McCarthy, and Ana Deletic. A critical review of integrated urban water modelling - Urban drainage and beyond. *Environmental Modelling & Software*, 54:88–107, apr 2014.

[72] A Rizzo, G Langergraber, A Galvão, F Boano, R Revelli, and L Ridolfi. Modelling the response of laboratory horizontal flow constructed wetlands to unsteady organic loads with HYDRUS-CWM1. *Ecological Engineering*, 68:209–213, 2014.

[73] T G Pálfy and G Langergraber. The verification of the Constructed Wetland Model No. 1 implementation in HYDRUS using column experiment data. *Ecological Engineering*, 68:105–115, 2014.

[74] D. Giraldi, M. de Michieli Vitturi, and R. Iannelli. FITOVERT: A dynamic numerical model of subsurface vertical flow constructed wetlands. *Environmental Modelling & Software*, 25(5):633–640, may 2010.

[75] Lewis A. Rossman. *Storm Water Management Model User's Manual Version 5 . 1.* U.S. EPA, Cincinnati, 2015.

[76] Lewis A. Rossman and Wayne C. Huber. *Storm Water Management Model Reference Manual Volume III-Water Quality.* United States Environmental Protection Agency, Cincinnati, 2016.

[77] Congcong Sun, Bernat Joseph-Duran, Thibaud Maruejouls, Gabriela Cembrano, Jordi Meseguer, Vicenç Puig, and Xavier Litrico. Real-time control-

oriented quality modelling in combined urban drainage networks. *IFAC-PapersOnLine*, 50(1):3941–3946, 2017. 20th IFAC World Congress.

[78] Sang-Soo Baek, Mayzonee Ligaray, Jongcheol Pyo, Jong-Pyo Park, Joo-Hyon Kang, Yakov Pachepsky, Jong Ahn Chun, and Kyung Hwa Cho. A novel water quality module of the SWMM model for assessing Low Impact Development (LID) in urban watersheds. *Journal of Hydrology*, 586(124886), jul 2020.

[79] M T Talbot, O Mcguire, C Olivier, and R Fleming. Parameterization and Application of Agricultural Best Management Practices in a Rural Ontario Watershed Using PCSWMM. *Journal of Water Management Modeling C400*, 2016.

[80] Tony H F Wong, Tim D. Fletcher, Hugh P. Duncan, and Graham A. Jenkins. Modelling urban stormwater treatment-A unified approach. *Ecological Engineering*, 27(1):58–70, 2006.

[81] Brendan Troitsky, David Z. Zhu, Mark Loewen, Bert van Duin, and Khizar Mahmood. Nutrient processes and modeling in urban stormwater ponds and constructed wetlands. *Canadian Water Resources Journal*, 44(3):230–247, 2019.

[82] Mehran Niazi, Chris Nietch, ; Mahdi Maghrebi, A M Asce, Nicole Jackson, Brittany R Bennett, Michael Tryby, Arash Massoudieh, and M Asce. Storm water management model: Performance review and gap analysis. *J. Sustainable Water Built Environ.*, 3, 2017.

[83] Branko Kerkez, Cyndee Gruden, Matthew Lewis, Luis Montestruque, Marcus Quigley, Brandon P. Wong, Alex Bedig, Ruben Kertesz, Tim Braun, Owen Cadwalader, Aaron Poresky, and Carrie Pak. Smarter Stormwater Systems. *Environ. Sci. Technol.*, 50(14):7267–7273, jul 2016.

[84] Schu, Manfred Tze, Alberto Campisano, Hubert Colas, Wolfgang Schilling, and Peter A. Vanrolleghem. Real time control of urban wastewater systems - Where do we stand today? *Journal of Hydrology*, 299(3-4):335–348, 2004.

[85] L. García, J. Barreiro-Gomez, E. Escobar, D. Téllez, N. Quijano, and C. Ocampo-Martinez. Modeling and real-time control of urban drainage systems: A review. *Advances in Water Resources*, 85:120–132, nov 2015.

[86] P A Vanrolleghem, L Benedetti, and J Meirlaen. Modelling and real-time control of the integrated urban wastewater system. *Environmental Modelling & Software*, 20:425–442, 2005.

[87] Bryant E Mcdonnell, Katherine Ratliff, Michael E Tryby, Jennifer Jia, Xin Wu, and Abhiram Mullapudi. PySWMM: The Python Interface to Stormwater Management Model (SWMM). *The Journal of Open Source Software*, 5(52):2292, 2020.

[88] William C. Lucas. Design of Integrated Bioinfiltration-Detention Urban Retrofits with Design Storm and Continuous Simulation Methods. *Journal of Hydrologic Engineering*, 15(6):486–498, 2010.

[89] R. A. Brown and W. F. Hunt. Underdrain Configuration to Enhance Bioretention Exfiltration to Reduce Pollutant Loads. *Journal of Environmental Engineering*, 137(11):1082–1091, November 2011.

[90] William Lucas and Margaret Greenway. Hydraulic Response and Nitrogen Retention in Bioretention Mesocosms. *Water Environment Research*, 83(8), 2011.

[91] R. Andrew Tirpak, ARM Nabiul Afrooz, Ryan J. Winston, Renan Valenca, Ken Schiff, and Sanjay K. Mohanty. Conventional and amended bioretention soil media for targeted pollutant treatment: A critical review to guide the state of the practice. *Water Research*, 189(2021), February 2021.

[92] P P Persaud, A A Akin, Branko Kerkez, D T Mccarthy, and J M Hathaway. Real time control schemes for improving water quality from bioretention cells. *Blue Green Systems*, 1(1):55–71, 2019.

[93] Barney Cohen. Urbanization in developing countries: Current trends, future projections, and key challenges for sustainability. *Technology in Society*, 28(1):63–80, January 2006.

[94] H. Jia, Z. Wang, X. Zhen, M. Clar, and S.L. Yu. China's sponge city construction: A discussion on technical approaches. *Frontiers of Environmental Science and Engineering*, 11(4), 2017.

[95] Water and Sewage Department. Green Stormwater Infrastructure Projects. Technical report, City of Detroit, Michigan, 2022.

[96] Jane Clary, Marcus Quigley, Aaron Poresky, Andrew Earles, Eric Strecker, Marc Leisenring, and Jonathan Jones. Integration of Low-Impact Development into the International Stormwater BMP Database. *Journal of Irrigation and Drainage Engineering*, 137(3):190–198, March 2011.

[97] T. D. Fletcher, H. Andrieu, and P. Hamel. Understanding, management and modelling of urban hydrology and its consequences for receiving waters: A state of the art. *Advances in Water Resources*, 51:261–279, 2013.

[98] Scott Dierks. Developing a Stormwater Control Measure Sizing Credit for the Infiltration Improvements Attributable to Plants. In *Proceedings of the Water Environment Federation*. Water Environment Federation, 2019.

[99] InfluxDB: Scalable datastore for metrics, events, and real-time analytics, 2022.

[100] Grafana – the open platform for analytics and monitoring, 2022.

[101] Elastic Compute Cloud (EC2): Cloud Server & Hosting, 2022.

[102] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J van der Walt, Matthew Brett, Joshua Wilson, K Jarrod Millman, Nikolay Mayorov, Andrew R J Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E A Quintero, Charles R Harris, Anne M Archibald, Antônio H Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, pages 1–12, feb 2020.

[103] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark

Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array Programming with NumPy. *Nature*, 585(February):357–362, 2020.

[104] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[105] Sara Meerow and Joshua P. Newell. Spatial planning for multifunctional green infrastructure: Growing resilience in Detroit. *Landscape and Urban Planning*, 159:62–75, March 2017.

[106] Andrea R. McFarland, Larissa Larsen, Kumelachew Yeshitela, Agizew Nigussie Engida, and Nancy G. Love. Guide for using green infrastructure in urban environments for stormwater management. *Environmental Science: Water Research & Technology*, 5(4):643–659, 2019.

[107] Sadaf Teimoori, Brendan F. O'Leary, and Carol J. Miller. Modeling Shallow Urban Groundwater at Regional and Local Scales: A Case Study in Detroit, MI. *Water*, 13(11):1515, May 2021.

[108] National Weather Service. NOWData - NOAA Online Weather Data. Technical report, National Oceanic and Atmospheric Administration, 2022.

[109] Emma Stein. 150 million gallons of raw sewage discharged into Michigan waters after storms. *Detroit Free Press*, August 2021.

[110] Ronald N. Forthofer, Eun Sul Lee, and Mike Hernandez. Descriptive Methods. In *Biostatistics*, pages 21–69. Elsevier, 2007.

[111] Jerrold H. Zar. Spearman Rank Correlation: Overview. In N. Balakrishnan, Theodore Colton, Brian Everitt, Walter Piegorsch, Fabrizio Ruggeri, and Jozef L. Teugels, editors, *Wiley StatsRef: Statistics Reference Online*. Wiley, 1 edition, September 2014.

[112] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[113] Brooke E. Mason, Jacquelyn Schmidt, and Branko Kerkez. Real-Time Sensor Network of Detroit Green Infrastructure: Datasets and Code, 2023.

[114] Mark Hicks. Summer 2021 among southeast Michigan's top 20 warmest, wettest, weather service says. *The Detroit News*, September 2021.

[115] Elena Cristiano, Marie-Claire ten Veldhuis, and Nick van de Giesen. Spatial and temporal variability of rainfall and their effects on hydrological response in urban areas – a review. *Hydrology and Earth System Sciences*, 21(7):3859–3878, July 2017.

[116] Allen P. Davis. Field Performance of Bioretention: Water Quality. *Environmental Engineering Science*, 24(8):1048–1064, October 2007.

[117] C. Small and R.J. Nicholls. A global analysis of human settlement in coastal zones. *Journal of Coastal Research*, 19(3):584–599, 2003.

[118] Jeffrey L. Howard, Katharine M. Orlicki, and Sarah M. LeTarte. Evaluation of some proximal sensing methods for mapping soils in urbanized terrain, Detroit, Michigan, USA. *CATENA*, 143:145–158, August 2016.

[119] Sylvie Spraakman, Jean-Luc Martel, and Jennifer Drake. How much water can bioretention retain, and where does it go? preprint, Preprints, September 2021.

[120] Adam Rose, Dan Wei, Juan Machado, and Kyle Spencer. Benefit–cost analysis of low-cost flood inundation sensors. *Natural Hazards Review*, 24(1), February 2023.

[121] Jianyin Huang, Christopher W. K. Chow, Zhining Shi, Rolando Fabris, Amanda Mussared, Gary Hallas, Paul Monis, Bo Jin, and Christopher P. Saint. Stormwater monitoring using on-line UV-Vis spectroscopy. *Environmental Science and Pollution Research*, October 2021.

[122] Jamie Steis Thorsby, Carol J. Miller, and Lara Treemore-Spears. The role of green stormwater infrastructure in flood mitigation (Detroit, MI USA) – case study. *Urban Water Journal*, 17(9):838–846, October 2020.

[123] Water and Sewage Department. Green stormwater infrastructure projects. Technical report, City of Detroit, Michigan, 2022.

[124] Brooke E. Mason, Jacquelyn Schmidt, and Branko Kerkez. Measuring city-scale green infrastructure drawdown dynamics using internet-connected sensors in detroit. *arXiv*, 2023.

[125] Carl Edward Rasmussen. *Gaussian Processes in Machine Learning*, volume 3176, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[126] Kyle Eckart, Zach McPhee, and Tirupati Bolisetti. Multiobjective optimization of low impact development stormwater controls. *Journal of Hydrology*, 562:564–576, July 2018.

[127] Daniel Ochoa, Gerardo Riano-Briceno, Nicanor Quijano, and Carlos Ocampo-Martinez. Control of urban drainage systems: Optimal flow control and deep learning in action. In *2019 American Control Conference (ACC)*, pages 4826–4831, Philadelphia, PA, USA, July 2019. IEEE.

[128] David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University Of Cambridge, November 2014.

[129] M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14(2):165–170, January 1987.

[130] Noel A. C. Cressie. *Statistics for spatial data*. John Wiley & Sons, Inc, Hoboken, NJ, revised edition edition, 2015.

[131] Naren Ramakrishnan and Chris Bailey-Kellogg. *Gaussian Process Models in Spatial Data Mining*, pages 639–644. Springer International Publishing, 2017.

[132] Andreas Krause. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 2008.

[133] W.F. Caselton and J.V. Zidek. Optimal monitoring network designs. *Statistics & Probability Letters*, 2(4):223–227, August 1984.

[134] Tyrone E. Duncan. On the calculation of mutual information. *SIAM Journal on Applied Mathematics*, 19(1):215–220, July 1970.

[135] GPy. Gpy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`, since 2012.

[136] Andreas Krause. Submodular function optimization. MATLAB Central File Exchange, January 2023.

[137] Water and Sewerage Department. Green stormwater infrastructure locations opendata. Technical report, City of Detroit, Michigan, December 2022.

[138] Brooke Mason and Branko Kerkez. Gaussian Process Model and Sensor Placement for Detroit Green Infrastructure: Datasets and Code, February 2023.

[139] Congcong Sun, Jan Lorenz Svensen, Morten Borup, Vicenç Puig, Gabriela Cembrano, and Luca Vezzaro. An MPC-Enabled SWMM Implementation of the Astlingen RTC Benchmarking Network. *Water*, 12(1034):1–13, 2020.

[140] Giuseppina Garofalo, Andrea Giordano, Patrizia Piro, Giandomenico Spezzano, and Andrea Vinci. A distributed real-time approach for mitigating CSO and flooding in urban drainage systems. *Journal of Network and Computer Applications*, 78(January 2016):30–42, January 2017.

[141] Shadab Shishegar, Sophie Duchesne, and Geneviève Pelletier. An Integrated Optimization and Rule-based Approach for Predictive Real Time Control of Urban Stormwater Management Systems. *Journal of Hydrology*, 577:124000, 2019.

[142] Pingping Zhang, Yanpeng Cai, and Jianlong Wang. A simulation-based real-time control system for reducing urban runoff pollution through a stormwater storage tank. *Journal of Cleaner Production*, 183:641–652, may 2018.

[143] Abhiram Mullapudi, Brandon P. Wong, and Branko Kerkez. Emerging investigators series: building a theory for smart stormwater systems. *Water Research & Technology*, 3(1):66–77, 2017.

[144] Gerard F. Laniak, Gabriel Olchin, Jonathan Goodall, Alexey Voinov, Mary Hill, Pierre Glynn, Gene Whelan, Gary Geller, Nigel Quinn, Michiel Blind, Scott Peckham, Sim Reaney, Noha Gaber, Robert Kennedy, and Andrew Hughes. Integrated environmental modeling: A vision and roadmap for the future. *Environmental Modelling & Software*, 39:3–23, 2013.

[145] J. Sutherland, I. H. Townend, Q. K. Harpham, and G. R. Pearce. From integration to fusion: The challenges ahead. *Geological Society Special Publication*, 408:35–54, 2017.

[146] Christopher K. Shuler and Katrina E. Mariner. Collaborative groundwater modeling: Open-source, cloud-based, applied science at a small-island water utility scale. *Environmental Modelling & Software*, 127, 5 2020.

[147] Daniel Bittner, Ayla Rychlik, Tobias Klöffel, Anna Leuteritz, Markus Disse, and Gabriele Chiogna. A gis-based model for simulating the hydrological effects of land use changes on karst systems – the integration of the lukars model into freewat. *Environmental Modelling & Software*, 127, 5 2020.

[148] Conrad E. Brendel, Randel L. Dymond, and Marcus F. Aguilar. Integration of quantitative precipitation forecasts with real-time hydrology and hydraulics modeling towards probabilistic forecasting of urban flooding. *Environmental Modelling & Software*, 134, 12 2020.

[149] Vivian V. Camacho Suarez, Robert J. Brederveld, Marieke Fennema, Antonio Moreno-Rodenas, Jeroen Langeveld, Hans Korving, Alma N.A. Schellart, and James D. Shucksmith. Evaluation of a coupled hydrodynamic-closed ecological cycle approach for modelling dissolved oxygen in surface waters. *Environmental Modelling & Software*, 119:242–257, 9 2019.

[150] Qian Wang, Rui Zou, Alvi Khalid, and Tao Yang. Uncertainty-based parameter estimation for urban pollutant buildup and washoff simulation using a multiple pattern inverse modeling approach. *Mathematics and Computers in Simulation*, jul 2019.

[151] Ana B. Deletic and C. T. Maksimovic. Evaluation of Water Quality Factors in Storm Runoff from Paved Areas. *Journal of Environmental Engineering*, 124(9):869–879, sep 1998.

[152] Prasanna Egodawatta, Evan Thomas, and Ashantha Goonetilleke. Mathematical interpretation of pollutant wash-off from urban road surfaces using simulated rainfall. *Water Research*, 41(13):3025–3031, 2007.

[153] David T. McCarthy, V. G. Mitchell, A. Deletic, and C. Diaper. Escherichia coli in urban stormwater: Explaining their variability. *Water Science and Technology*, 56(11):27–34, dec 2007.

[154] D E Overton and M E Meadows. *Stormwater Modeling*. Elsevier Science, 2013.

[155] Camilla Tuomela, Nora Sillanpää, and Harri Koivusalo. Assessment of stormwater pollutant loads and source area contributions with storm water management model (SWMM). *Journal of Environmental Management*, 233:719–727, 2018.

[156] Robert H. Kadlec and Scott D. Wallace. *Treatment Wetlands*. CRC Press, 2 edition, 2009.

[157] Frank Engelund and Eggert Hansen. A Monograph on Sediment Transport in Alluvial Streams. Technical Report 2, Technical University of Denmark, Copenhagen, 1967.

[158] Yaser Shammaa and David Z Zhu. Techniques for Controlling Total Suspended Solids in Stormwater Runoff. *Canadian Water Resources Journal*, 26(3):359–375, 2001.

[159] Keith E. Schilling, Sea Won Kim, and Christopher S. Jones. Use of water quality surrogates to estimate total phosphorus concentrations in Iowa rivers. *Journal of Hydrology: Regional Studies*, 12:111–121, aug 2017.

[160] A. Dong, G. Chesters, and G. V. Simsiman. Metal composition of soil, sediments, and urban dust and dirt samples from the Menomonee River Watershed, Wisconsin, U.S.A. *Water, Air, and Soil Pollution*, 22(3):257–275, apr 1984.

[161] USDA. Transmission of Sediment by Water. In *National Engineering Handbook*, chapter 4, pages 1–39. United States Department of Agriculture, 1983.

[162] Baosheng Wu, M Asce, Albert Molinas, and Pierre Y Julien. Bed-Material Load Computations for Nonuniform Sediments. *J. Hydrol. Eng.*, 130(10), 2004.

[163] Daniel J. Conley, Hans W. Paerl, Robert W. Howarth, Donald F. Boesch, Sybil P. Seitzinger, Karl E. Havens, Christiane Lancelot, and Gene E. Likens. Controlling Eutrophication: Nitrogen and Phosphorus. *Science*, 323, 2009.

[164] Robert Howarth and Hans W. Paerl. Coastal marine eutrophication: Control of both nitrogen and phosphorus is necessary. *Proceedings of the National Academy of Sciences of the United States of America*, 105(49), dec 2008.

[165] Shane E. Perryman, Gavin N. Rees, Christopher J. Walsh, and Michael R. Grace. Urban Stormwater Runoff Drives Denitrifying Community Composition Through Changes in Sediment Texture and Carbon Content. *Microbial Ecology*, 61(4):932–940, may 2011.

[166] John R. White and K.R. Reddy. Biogeochemical Dynamics I: Nitrogen Cycling in Wetlands. In Edward Maltby and Tom Barker, editors, *The Wetlands Handbook*, chapter 9, pages 213–227. Blackwell Publishing Ltd, 2009.

[167] Lian Scholes, D. Michael Revitt, and J. Bryan Ellis. A systematic approach for the comparative assessment of stormwater pollutant removal potentials. *Journal of Environmental Management*, 88(3):467–478, aug 2008.

[168] Robert H. Kadlec. Nitrate dynamics in event-driven wetlands. *Ecological Engineering*, 36(4):503–516, 2010.

[169] K. R. Reddy and W. H. Patrick. Nitrogen Transformations and Loss in Flooded Soils and Sediments. *Critical Reviews in Environmental Control*, 13(4):273–309, 1984.

[170] EPA. Summary Table for the Nutrient Criteria Documents. Technical report, United States Environmental Protection Agency, 2002.

[171] David K Mueller and Norman E Spahr. Water-quality, streamflow, and ancillary data for nutrients in streams and rivers across the nation, 1992-2001. Technical report, United States Geological Survey, 2005.

[172] E. Gaborit, D. Muschalla, B. Vallet, P. A. Vanrolleghem, and F. Anctil. Improving the performance of stormwater detention basins by real-time control using rainfall forecasts. *Urban Water Journal*, 10(4):230–246, 2013.

[173] Brooke E. Mason, Abhiram Mullapudi, and Branko Kerkez. StormReactor: An open-source Python package for the integrated modeling of urban water quality and water balance. *Environmental Modelling & Software*, 145:105175, November 2021.

[174] Jason Faber Carpenter, Bertrand Vallet, Geneviève Pelletier, Paul Lessard, and Peter A Vanrolleghem. Pollutant removal efficiency of a retrofitted stormwater detention pond. *Water Quality Research Journal of Canada*, 49(2):124–134, 2014.

[175] Anna M Michalak, Eric J Anderson, Dmitry Beletsky, Steven Boland, Nathan S Bosch, Thomas B Bridgeman, Justin D Chaffin, Kyunghwa Cho, Rem Confesor, Irem Daloglu, Joseph V Depinto, Mary Anne Evans, Gary L Fahnenstiel, Lingli He, Jeff C Ho, Liza Jenkins, Thomas H Johengen, Kevin C Kuo, Elizabeth Laporte, Xiaojian Liu, Michael R McWilliams, Michael R Moore, Derek J Posselt, R Peter Richards, Donald Scavia, Allison L Steiner, Ed Verhamme, David M Wright, and Melissa A Zagorski. Record-setting algal bloom in Lake Erie caused by agricultural and meteorological trends consistent with expected future conditions. *Proceedings of the National Academy of Sciences of the United States of America*, 110(16):6448–52, April 2013.

[176] Jiake Li and Allen P. Davis. A unified look at phosphorus treatment using bioretention. *Water Research*, 90:141–155, 2016.

[177] Sean W. O'Neill, Allen P. Davis, Sean W O'Neill, and Allen P. Davis. Water Treatment Residual as a Bioretention Amendment for Phosphorus. II: Long-Term Column Studies. *Journal of Environmental Engineering*, 138(3):328–336, 2012.

[178] Yaoze Liu, Laurent M. Ahiablame, Vincent F. Bralts, and Bernard A. Engel. Enhancing a rainfall-runoff model to assess the impacts of BMPs and LID practices on storm runoff. *Journal of Environmental Management*, 147:12–23, 2015.

[179] Lewis A. Rossman. Modeling Low Impact Development Alternatives with SWMM. *Journal of Water Management Modelling*, 6062:167–182, 2010.

[180] R Pitt, A Maestre, and J Clary. The National Stormwater Quality Database (NSQD), Version 4.02 Background. Technical report, The Water Research Foundation, 2018.

[181] Michael R. Ament, Stephanie E. Hurley, Mark Voorhees, Eric Perkins, Yongping Yuan, Joshua W. Faulkner, and Eric D. Roy. Balancing Hydraulic Control and Phosphorus Removal in Bioretention Media Amended with Drinking Water Treatment Residuals. *ACS ES&T Water*, 1(3):688–697, March 2021.

[182] Minnesota Pollution Control Agency. Cost-benefit Considerations for Bioretention. Technical report, Minnesota Pollution Control Agency, 2021.

[183] Brooke E. Mason, Abhiram Mullapudi, Cyndee Gruden, and Branko Kerkez. Improvement of phosphorus removal in bioretention cells using real-time control. *Urban Water Journal*, pages 1–7, August 2022.

[184] U.S. Geological Survey. 1/9th Arc-second Digital Elevation Models (DEMs) - USGS National Map 3DEP Downloadable Data Collection: U.S. Geological Survey. Technical report, U.S. Geological Survey, 2017.

[185] Soil Survey Staff. Web Soil Survey. Technical report, Natural Resources Conservation Service, United States Department of Agriculture, 2021.

[186] Jon Dewitz and U.S. Geological Survey. National Land Cover Database (NLCD) 2019 Products, 2021.

[187] Tomislav Hengl and Hannes I. Reuter, editors. *Geomorphometry: concepts, software, applications*, volume 33 of *Developments in Soil Science*. Elsevier, 2008.

[188] and Energy Department of Environment, Great Lakes. Water Well Viewer. Technical report, State of Michigan, 2021.

[189] Department of Environmental Quality. Wellogic system. Technical report, State of Michigan, 2021.

[190] U.S. Geological Survey. Michigan Real-Time Groundwater Level Network. Groundwater watch, U.S. Department of the Interior, 2021.

[191] Darren George and Paul Mallery. *SPSS for Windows step by step: a simple guide and reference 18.0 update*. Allyn & Bacon/Pearson, Boston, MA, 11th ed edition, 2011.

[192] Rachel M. Uetrecht, Andrew C. Elmore, Joe D. Guggenberger, and Zane D. Helwig. Practical considerations for kriging groundwater surfaces. *Remediation Journal*, 29(4):83–91, September 2019.

[193] Konstantin Krivoruchko and Alexander Gribov. Evaluation of empirical bayesian kriging. *Spatial Statistics*, 32:100368, August 2019.

[194] Yanmei Li, J. Horacio Hernandez, Manuel Aviles, Peter S. K. Knappett, John R. Giardino, Raúl Miranda, María Jesús Puy, Francisco Padilla, and Jorge Morales. Empirical Bayesian Kriging method to evaluate inter-annual water-table evolution in the Cuenca Alta del Río Laja aquifer, Guanajuato, México. *Journal of Hydrology*, 582:124517, March 2020.

[195] Hannah M. Cooper, Caiyun Zhang, and Donna Selch. Incorporating uncertainty of groundwater modeling in sea-level rise assessment: a case study in South Florida. *Climatic Change*, 129(1-2):281–294, March 2015.