Reinforcement Learning for Flooding Mitigation in Complex Stormwater Systems during Large Storms

Cheng Wang
Engineering Systems and Environment
University of Virginia
Charlottesville, USA
cw8xk@virginia.edu

Benjamin D. Bowes
Engineering Systems and Environment
University of Virginia
Charlottesville, USA
bdb3m@virginia.edu

Peter A. Beling
Engineering Systems and Environment
University of Virginia
Charlottesville, USA
pb3a@virginia.edu

Jonathan L. Goodall
Engineering Systems and Environment
University of Virginia
Charlottesville, USA
goodall@virginia.edu

IEEE EUROCON 2021 - 19th International Conference on Smart Technologies | 978-1-6654-3299-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/EUROCON52738.2021.9535587

***Abstract*—Compared with capital improvement projects, real-time control of stormwater systems may be a more effective and efficient approach to address the increasing risk of flooding in urban areas. One way to automate the design process of control policies is through reinforcement learning (RL). Recently, RL methods have been applied to small stormwater systems and have demonstrated better performance over passive systems and simple rule-based strategies. However, it remains unclear how effective RL methods are for larger and more complex systems. Current RL-based control policies also suffer from poor convergence and stability, which may be due to large updates made by the underlying RL algorithm. In this study, we use the Proximal Policy Optimization (PPO) algorithm and develop control policies for a medium-sized stormwater system that can significantly mitigate flooding during large storm events. Our approach demonstrates good convergence behavior and stability, and achieves robust out-of-sample performance.**

***Keywords*—Stormwater Systems, Reinforcement Learning, Real-time Control, Flood Mitigation**

I. Introduction

Flooding poses a significant and growing risk for many urban areas, with the potential to disrupt normal and emergency operations, damage infrastructure, and cause loss of life [1]. One way to adapt traditional stormwater systems to changing weather and climate conditions is to make them larger (e.g. replacing small pipes with larger ones). These capital improvement projects are typically costly and disruptive for the normal operation of a city. In fact, research suggests that such piecewise improvements can degrade total system performance [2], [3]. Instead of increasing the physical capacity of a stormwater system, controlling them in real-time could increase their effective capacity in a more cost efficient way [4].

In current practice, control policies are often predefined simple heuristics and may require expert knowledge or experience [5] [6]. Coupled with the fact that urban areas are constantly evolving (changing the input to stormwater systems over time), it can become increasingly difficult to design effective control policies for larger and more complex stormwater systems.

One approach to automate the learning process of control policies is by using reinforcement learning (RL), in which an agent learns to optimize its behavior by interacting with its environment [7]. Combined with deep learning, RL has achieved great successes in many fields such as Atari games [8], the game of Go [9], and StarCraft II [10].

More recently, different RL methods such as Deep Q-Network (DQN) [8] and Deep Deterministic Policy Gradient (DDPG) [11] have been applied to stormwater control tasks [12]–[15]. However, stormwater systems considered in these studies remain relatively simple and small (with just a few control assets). It is yet unclear how effective RL methods are for more complex systems. In addition, although the learned control policies are able to mitigate flooding or keep the controlled sites at a desired water level, they may be unstable or may not have converged (as in [15]) and often require erratic adjustments (e.g., frequently opening and closing a valve), which could make them difficult to implement in real-world scenarios.

In this study, we develop RL-based control policies for a medium-sized stormwater system that are able to significantly reduce flooding during large storm events. Based on the Proximal Policy Optimization (PPO) algorithm [16], our approach demonstrates good convergence behavior and stability, and achieves robust out-of-sample performance.

II. Methods

A. Stormwater System Simulation

Stormwater system simulations are conducted using the U.S. Environmental Protection Agency's Stormwater Management Model (SWMM), version 5. The NRCS Type II synthetic storms are used as inputs to the SWMM simulations, and a Python wrapper for SWMM – pyswmm [17] is used to enable the step-by-step running of simulations. Each step corresponds

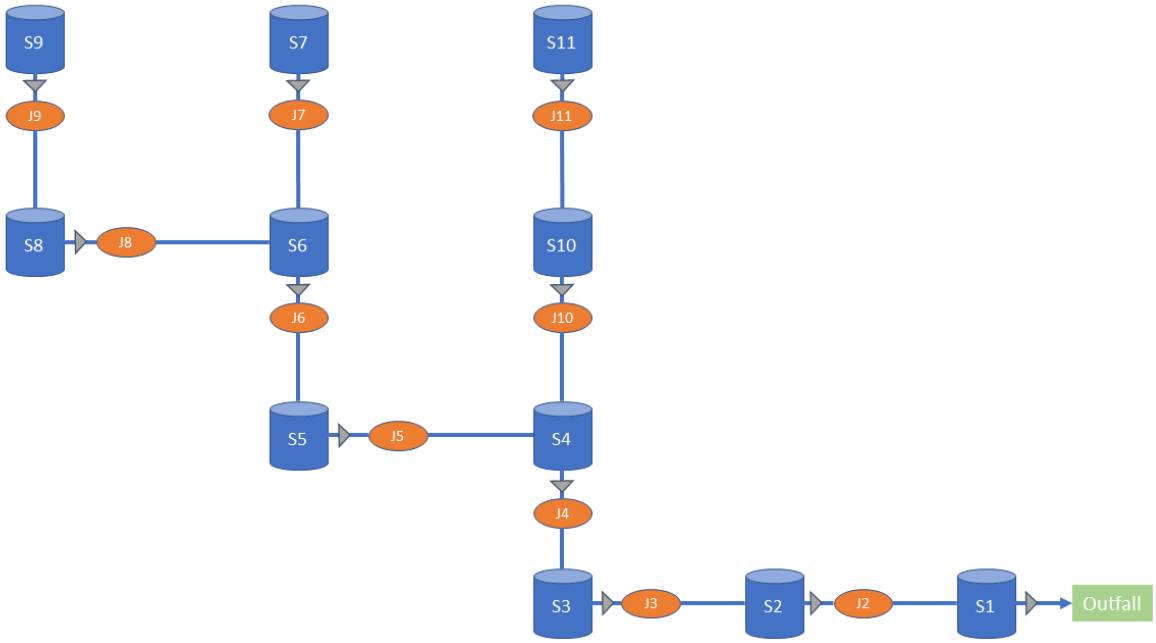

Fig. 1. Abstraction of the stormwater system considered in this study. There are 11 storage units and 10 junction nodes. Flow out of storage units can be regulated by valves (orifices).

to 10 minute simulation time and a full simulation lasts for 48 hours.

The stormwater system used in this study (Fig. 1) is from Scenario Gamma of the pystorms library [18] and is closely related to the system used in [15], which is inspired by an urban watershed in Ann Arbor, Michigan, USA. The system consists of eleven storage units, ten junction nodes, and pipes going to the system outfall that discharges to a waterbody. At the bottom of each storage unit, there is a valve (orifice) that can be used to control the outflow from the corresponding storage unit.

During a storm event, flooding can occur at any storage units or junction nodes. While letting more water flow out of a storage unit may reduce the risk of flooding locally, it could result in too much water going through the storage units and/or junction nodes downstream too quickly. Therefore, mitigating flooding for the whole stormwater system requires coordinated controls in accordance with system conditions.

B. Reinforcement Learning

In Reinforcement learning, an agent learns to optimize its behavior by interacting with its environment [7]. Formally, the environment in RL is defined as a Markov decision process (MDP): (S, A, P, r, γ), where S is the state space, A is the action space, $P : S \times A \to S$ is the transition function, $r : S \times A \to \mathbb{R}$ is the reward function and $\gamma \in [0, 1]$ is the discount factor that is used to determine the present value of future rewards. A policy, π, is a mapping from states to actions. At each time step t, an RL agent observes a state s_t, takes an action a_t and then transitions to a new state s_{t+1}, and receives a reward $r_t = r(s_t, a_t, s_{t+1})$. The return $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is the total discounted future rewards from time t. The goal of an RL agent is to learn an optimal policy that maximizes the expected return: $\mathbb{E}_\pi[G_t]$.

In value-based methods (also known as critic-only methods), the optimal action-value function $Q^*(s, a) = \max_\pi \mathbb{E}_\pi[G_t | s_t = s, a_t = a]$ is learned and is then used to derive an optimal policy $\pi^*(s) = \text{argmax}_a\, Q^*(s, a)$. On the other hand, policy-based methods (or actor-only methods) directly parameterize the policy $\pi(a|s; \theta)$ and optimizes a performance measure $J(\theta)$ (e.g., the expected return) through gradient ascent. To reduce the variance of the estimate of the policy gradient, actor-critic methods use the value function, $V_\pi(s) = \mathbb{E}_\pi[G_t | s_t = s]$, as a baseline in the policy gradient estimator. For example, one commonly used gradient estimator has the form $\nabla J(\theta) = \mathbb{E}_t[\nabla_\theta \log \pi_\theta(a_t|s_t) A_t]$, where $A_t = Q(s_t, a_t) - V(s_t)$ is called the advantage function.

We apply one of the state-of-the-art actor-critic algorithms, Proximal Policy Optimization (PPO) [16], to the stormwater system shown in Fig. 1. To prevent large policy updates that could lead to performance collapse, PPO uses a clipped surrogate objective function

$$\mathbb{E}\Big[\min\big(\rho_t(\theta)A_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)A_t\big)\Big], \qquad (1)$$

where ϵ is a hyperparameter (usually a small positive number) and $\rho_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$ is the probability ratio of the new policy over the old policy. In practice, the objective

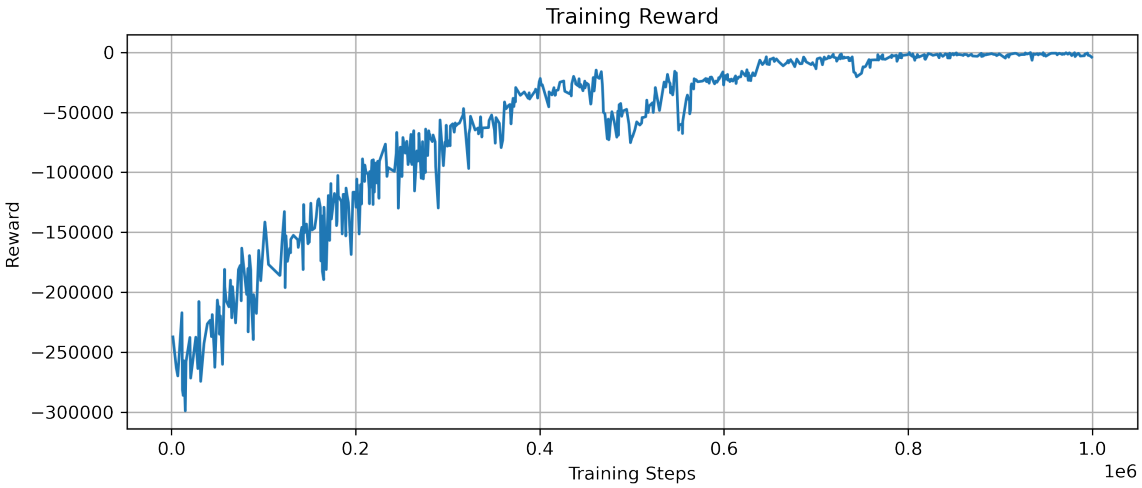

Fig. 2. Rewards received by the RL agent during training.

function in (1) is often augmented by adding an entropy bonus to keep the agent from being stuck in a locally optimal policy.

As our objective is to minimize the total flood volume, the MDP for the stormwater system in Fig.1 is defined as follows:

- State: remaining depths and flooding rates at all storage units and junction nodes;
- Action: valve openings (between 0.0 and 1.0) for all storage units;
- Reward: negative of the increase in cumulative flood volume at storage units and junction nodes.

To deal with the continuous state and action spaces, the value function (the critic) and the control policy (the actor) are approximated by two separate feed-forward neural networks with leaky ReLU as the activation function. Each neural network has 300 and 150 neurons in the first and the second layer, respectively.

III. Results

In this section we report both the training and testing results of RL-based control policies. The RL agent is trained on a single 100-year, 24-hour design storm and tested on 100 25-year, 100-year, and 500-year 24-hour storm events, respectively.

A. Training Performance

We first plot the reward graph during the training process in Fig. 2. Recall that reward from each step is the negative of the flood volume, therefore, Fig. 2 shows that as training progresses, the RL agent is able to greatly mitigate flooding, if not eliminate it altogether. To further confirm that the RL policy has converged and the good training performance is not due to random actions by the agent, the loss functions on the value function, policy, and entropy are also examined in Fig. 3. All loss functions appear to have converged after 800 thousand training steps. The gradual increase on the entropy loss shows that as training progresses, the RL agent is more confident in its actions and its policy is becoming less random.

In order to select the best candidate policy for testing on other storm events, an RL policy is saved after every 10 thousand training steps and is then validated on the same training storm without any exploration (i.e., no random actions by the agent). The total flood volumes from these policies are shown in Fig. 4. The best-performing one, learned after 800 thousand steps, completely eliminates flooding. Its actions during the training storm event are plotted in Fig. 5. In general, changes in valve openings are gradual and smooth, making the policy easy to implement in practice. Still, adjusting valves frequently may pose practical challenges in some real-world scenarios. To further examine the practicability and robustness of the learned RL policy, an additional constraint on control frequency is implemented. For every selected action, the RL agent is required to repeat the same action for the next hour. In other words, once valve openings have been determined, they must be kept at the same settings. Fig. 6 plots actions from the same policy under this constraint and shows that flooding can still be eliminated even with less frequent control actions.

B. Testing Performance

To evaluate the generalization ability of the RL-based control system, the best-performing policy from the training process is tested on a total of 300 storms with different intensities. The results are compared with a baseline policy selected from a set of passive control policies.

TABLE I
Flood volume (m^3) from passive control policies during one hundred 100-year 24-hour storms

Opening Level	0%	25%	50%	75%	100%
Mean	1676	939	923	1099	2211
Std	149	115	103	138	389

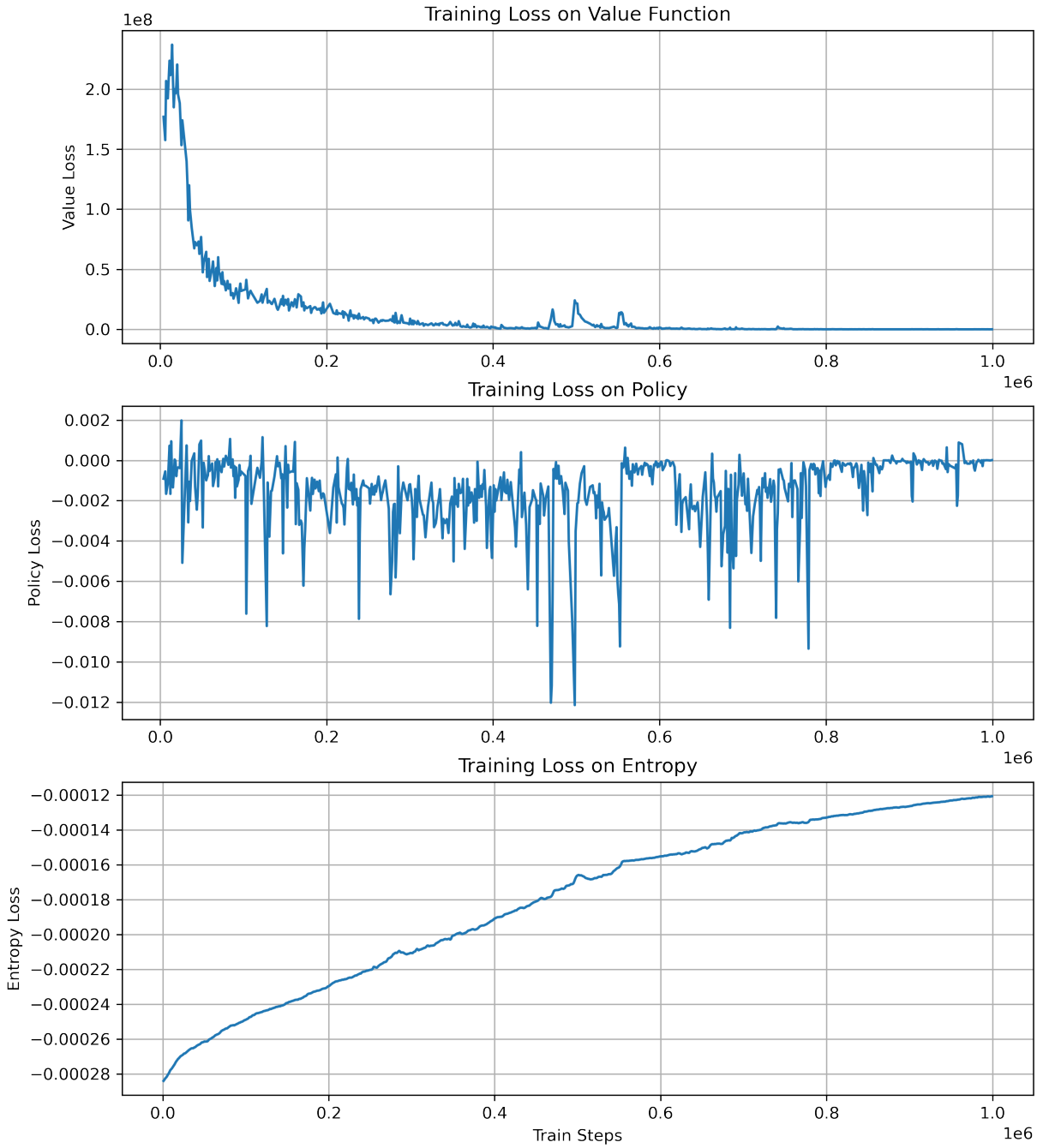

Fig. 3. Training losses on the value function (top panel), policy (middle panel), and entropy (bottom panel).

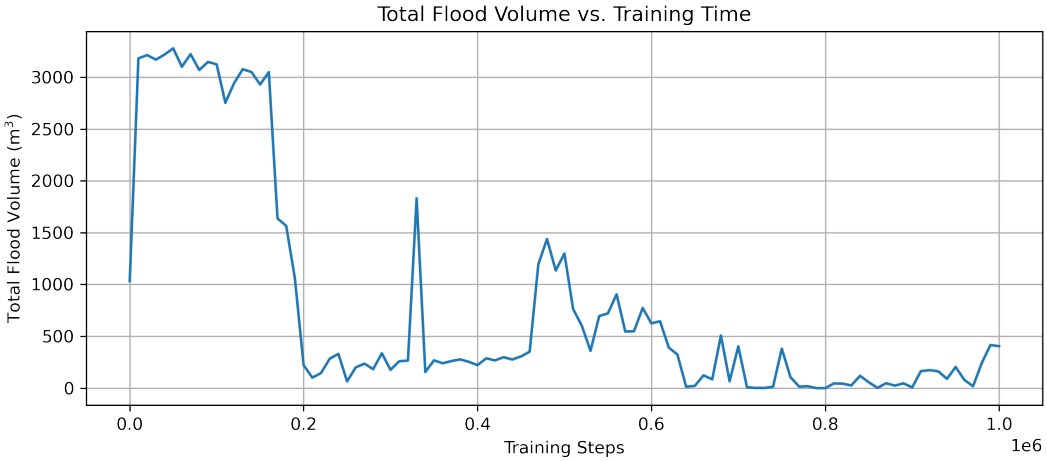

Fig. 4. Total flood volume during training.

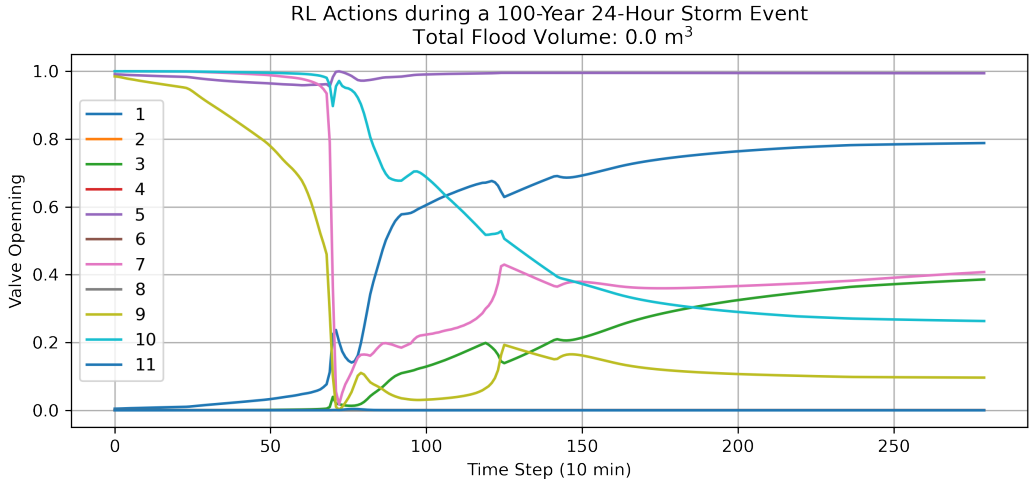

Fig. 5. RL policy during the training storm event.

A commonly used baseline in stormwater control systems is the uncontrolled policy (e.g., in [15]), which keeps valves fully open at all times. It is possible to find better (hence more challenging) baselines by considering different fixed settings. Table I shows the average and standard deviation of flood volumes during one hundred 100-year 24-hour storms from static policies that keep all valves open at 0%, 25%, 50%, 75%, and 100% levels, respectively. It turns out that keeping valves half open results in the fewest average flooding amount, and therefore we will use this policy as the baseline in the following comparisons with RL policies.

Table II reports the statistics on flood volumes during different storm events using the baseline, the RL policy, and the RL policy with repeated actions. It shows that for 25-year and 100-year storms, RL policies are able to completely eliminate flooding, while the baseline policy will always result in flooding at some nodes in the system. For storms with an even larger recurrence interval such as 500-year, flooding can still be greatly reduced by RL-based control policies. On average, total flood volume from the one hundred 500-year 24-hour testing storms is reduced by approximately 90%. And in some scenarios, flooding is almost eliminated by RL policies.

Surprisingly, the constraint on control actions' frequency actually leads to a slightly better performance than the original RL policy during 500-year 24-hour storm events. Although being less adaptive to the changing environment, the extra time may have helped the RL agent evaluate states more accurately and hence make better decisions.

IV. Conclusion

This study explores the possibility of using reinforcement learning to discover effective control policies for flood prevention and mitigation in stormwater systems. Compared with a set of static policies, the RL-based policies demonstrate superior performance during a range of large storm events, significantly reducing the total flood volume for a medium-sized stormwater system. Even after limiting its control frequency,

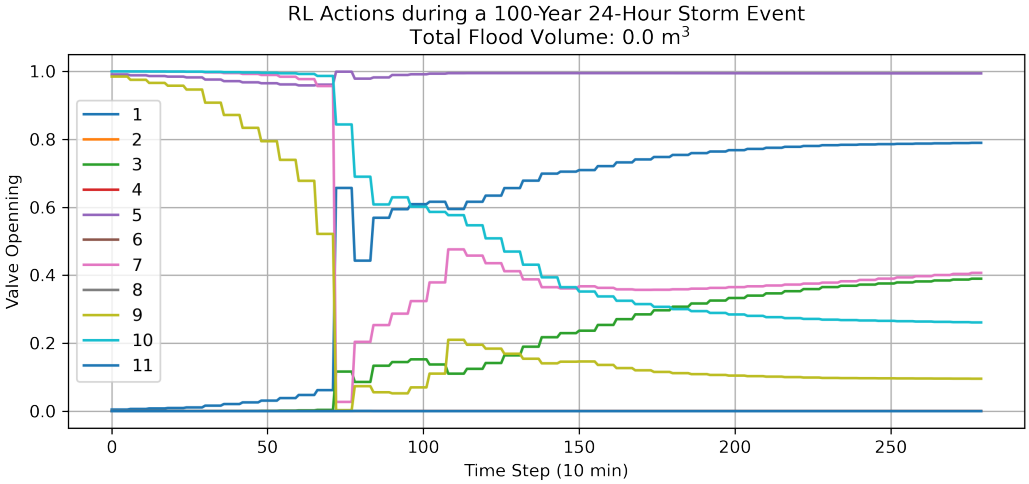

Fig. 6. RL policy with repeated actions during the training storm event.

TABLE II
FLOOD VOLUME (M^3) DURING TESTING STORM EVENTS

Storm	Stat	Baseline	RL	RL-Repeat
25-year 24-hour	Mean	216	0	0
	Max	280	0	0
	Min	149	0	0
100-year 24-hour	Mean	923	0	0
	Max	1129	0	0
	Min	771	0	0
500-year 24-hour	Mean	1649	210	146
	Max	2103	488	395
	Min	1270	3	3

the RL agent is still able to achieve comparable or even better performance, which further indicates the effectiveness of our RL-based approach.

Future work may explore: (i) more complex objective functions such as by considering pollution and desired flow/water levels in combination with flooding risks; (ii) longer storm durations or historical storm events; and (iii) other formulations with expanded state space or different reward functions.

REFERENCES

[1] G. E. Galloway, A. Reilly, S. Ryoo, A. Riley, M. Haslam, S. Brody, W. Highfeld, J. Gunn, J. Rainey, and S. Parker, "THE GROWING THREAT OF URBAN FLOODING: 2018," University of Maryland, Center for Disaster Resilience, and Texas A&M University, Galveston Campus, Center for Texas Beaches and Shores, College Park, Tech. Rep., 2018.

[2] C. H. Emerson, C. Welty, and R. G. Traver, "Watershed-scale evaluation of a system of storm water detention basins," *Journal of Hydrologic Engineering*, vol. 10, no. 3, pp. 237–242, 2005.

[3] G. Petrucci, E. Rioust, J. F. Deroubaix, and B. Tassin, "Do stormwater source control policies deliver the right hydrologic outcomes?" *Journal of Hydrology*, vol. 485, pp. 188–200, apr 2013.

[4] B. Kerkez, C. Gruden, M. Lewis, L. Montestruque, M. Quigley, B. Wong, A. Bedig, R. Kertesz, T. Braun, O. Cadwalader *et al.*, "Smarter stormwater systems," *Environ. Sci. Technol*, vol. 50, pp. 7267–7273, 2016.

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

[6] Y. Abou Rjeily, O. Abbas, M. Sadek, I. Shahrour, and F. H. Chehade, "Model predictive control for optimising the operation of urban drainage systems," *Journal of Hydrology*, vol. 566, pp. 558–565, 2018.

[7] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[9] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[10] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[11] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[12] C. Wang, B. Bowes, A. Tavakoli, S. Adams, J. Goodall, and P. Beling, "Smart stormwater control systems: A reinforcement learning approach," in *Proceedings of the ISCRAM Conference Proceedings—17th International Conference on Information Systems for Crisis Response and Management, Blacksburg, VA, USA*, 2020, pp. 24–27.

[13] B. D. Bowes, A. Tavakoli, C. Wang, A. Heydarian, M. Behl, P. A. Beling, and J. L. Goodall, "Flood mitigation in coastal urban catchments using real-time stormwater infrastructure control and reinforcement learning," *Journal of Hydroinformatics*, 2020.

[14] S. M. Saliba, B. D. Bowes, S. Adams, P. A. Beling, and J. L. Goodall, "Deep reinforcement learning with uncertain data for real-time stormwater system control and flood mitigation," *Water*, vol. 12, no. 11, p. 3222, 2020.

[15] A. Mullapudi, M. J. Lewis, C. L. Gruden, and B. Kerkez, "Deep reinforcement learning for the real time control of stormwater systems," *Advances in Water Resources*, vol. 140, p. 103600, 2020.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[17] B. E. McDonnell, K. Ratliff, M. E. Tryby, J. J. X. Wu, and A. Mullapudi, "Pyswmm: The python interface to stormwater management model (swmm)," *Journal of Open Source Software*, vol. 5, no. 52, p. 2292, 2020.

[18] S. P. Rimer, A. Mullapudi, S. C. Troutman, and B. Kerkez, "A benchmarking framework for control and optimization of smart stormwater networks: demo abstract," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, 2019, pp. 350–351.