Three Reinforcement Learning Approaches for Optimal Chlorine Injection Control in Water Distribution Networks

Talgat Abzanov[1] , **Erik Haße**[1]
[1]Bielefeld University
{tabzanov, ehasse}@techfak.uni-bielefeld.de

Abstract

Ensuring safe and consistent chlorine levels in water distribution networks (WDNs) is a critical yet complex challenge, balancing disinfection efficacy with operational and health constraints. This study explores three reinforcement learning based approaches for managing chlorine injections in a real-world network simulated for the IJCAI-2025 Drinking Water Chlorination Challenge. The models aim to keep chlorine in the network within a safe range under complex dynamics and contamination events, given only limited sensor data. The results show that while all models outperformed a random baseline, they struggled with cyclic demand patterns and often converged to zero-injection policies due to environment complexity and reward conflicts.

1 Introduction

Ensuring the safety and quality of drinking water in water distribution networks (WDNs) is a critical challenge for public health. A widely used method for maintaining microbial safety is chlorine disinfection, where chlorine is injected into a WDN to eliminate pathogens that may enter through contamination events. It is not possible to simply chlorinate the water as much as possible to reduce the number of pathogens to a minimum, as over-chlorinated water is also unsafe. Therefore, maintaining optimal chlorine concentrations across an entire network is a complex, dynamic control problem, influenced by fluctuating water demands, variable source water quality, limited sensor availability, and unpredictable contamination. Furthermore, chlorine decays over time due to reactions with organic matter and other contaminants, and its transport is governed by complex hydraulic dynamics. Too little chlorine compromises disinfection effectiveness, increasing the risk of outbreaks, and too much chlorine leads to undesirable taste, odor, and potentially carcinogenic disinfection by-products. One of the core goals of a safe WDN is to maintain chlorine levels between 0.2 mg/L and 0.4 mg/L at all demand nodes, while minimizing costs and ensuring system robustness.

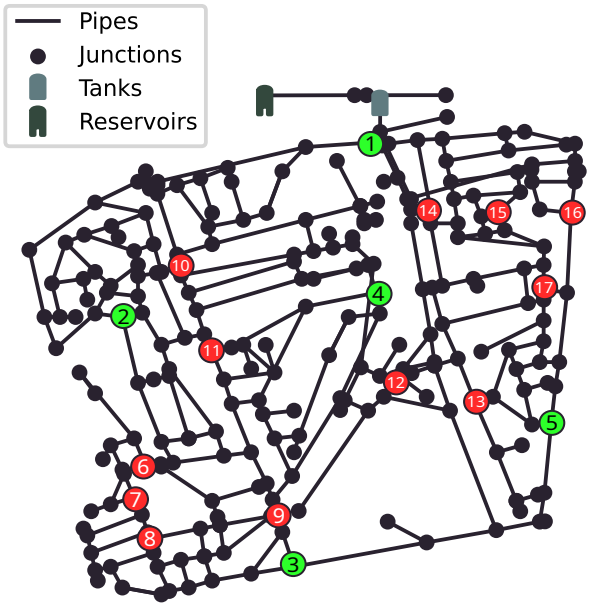

Figure 1: A water distribution network on Cyprus (CY-DBP), that shows water sources, reservoirs, tanks, demand nodes and pipes. Red junctions indicate chlorine sensor placement at that location. Green junctions additionally contain a pump capable of chlorine injection.

In this work, we explore three different reinforcement learning (RL) approaches for managing chlorine injections via a limited set of chlorine booster stations in the *CY-DBP* WDN, which is located on Cyprus. An overview of the WDN's structure is shown in Figure 1. This work is a submission to the "Drinking Water Chlorination Challenge @ IJCAI-2025" [Artelt *et al.*, 2025], which tasks its participants with the development of a model meeting the above criteria to maintain safe drinking water conditions. As an evaluation space, the challenge provides an EPANET-based [US Environmental Protection Agency, 2020] simulation of the CY-DBP WDN, along with several scenarios. The challenge is further complicated by the fact that only a limited number of sensors are available, providing partial observability of the system state. Moreover, contamination events are hidden and occur at random times and locations, making reactive control essential. The control system must act solely based on sparse

sensor feedback to adjust chlorine injection rates at multiple booster stations every five minutes and the flow rate of two specific pipes in the WDN, balancing competing objectives: safe chlorine levels, operational smoothness, fairness across network nodes, and economic efficiency.

2 Related Work

There have been different approaches to chlorine injection management in prior studies. Many of them model this task as classical optimization problems and use convex optimization methods, while others use genetic algorithms.

For instance, Frederick *et al.* proposed a chlorine injection optimization model that leverages water age and breadth-first search (BFS) algorithms to maintain residual chlorine levels within acceptable ranges in WDNs. Their approach identifies water delivery pathways using BFS, estimates required chlorine injection based on water age, and optimizes the injection schedule using a genetic algorithm. The study demonstrated that chlorine requirements exhibit an inverse relationship with demand patterns (higher injection needed during low-demand periods due to longer water retention times) and achieved promising results with average errors below 10% across demand nodes with more lenient chlorine bounds of 0.2-1.0 mg/L. While their method effectively reduces computational complexity compared to previous booster chlorination approaches, it relies on predefined scheduling intervals rather than adaptive decision-making, which presents an opportunity for reinforcement learning approaches to dynamically respond to real-time network conditions.

In another example, Perelman and Ostfeld recently introduced Data-Enabled Predictive Control (DeePC) as a model-free approach for chlorine injection scheduling, demonstrating its effectiveness especially in the Pescara (Italy) WDN. Their method utilizes historical input-output data rather than physical models to optimize chlorine booster injection rates, achieving superior performance compared to the previous state-of-the-art approach based on convex heuristics optimization with a significantly lower mean absolute target error (0.047 vs. 0.065) while maintaining safe chlorine residuals (0.8-1.2 mg/L) across the network. The authors showed that model-free approaches can effectively handle the complex nonlinear dynamics of water quality management without requiring detailed system knowledge.

3 Methodology

To approach the problem of chlorine injection, we have developed three RL-based approaches using the Proximal Policy Optimization (PPO) architecture [Schulman *et al.*, 2017] as a general scaffold. This choice is based on the environment's strong temporal dependencies and linear structure, along with the absence of early termination conditions, which made on-policy learning preferable. Additionally, *Stable-Baselines3* (SB3) [Raffin *et al.*, 2021] offers a robust implementation of PPO, with convenient interfaces for customization.

For training and evaluation of our models, the challenge provides a Gymnasium environment acting as an interface to the simulated dynamics of the WDN.

The simulation itself is implemented in EPyT-Flow [Artelt *et al.*, 2024], which models the WDN as a graph network $G = (V, E)$. Information between the environment and simulation is exchanged via a set of sensors in pipes or junctions $E_f \subset E$, $V_c \subset V$ and chlorine pump junctions $V_p \subset V_c$. This information is obtained in the shape of an observation vector S_t every 5 minutes of simulated time, denoted as a separate time step t. The 17 dimensional observation vector contains readings of all sensor junctions, defined as $S_t = (f(t)_{e_1}, f(t)_{e_2}, c(t)_{v_1}, \ldots, c(t)_{v_{17}})$, where $f(t)$ and $c(t)$ are the flow and chlorine sensor values of a given junction $v_1, ..., v_{17} \in V_c$ or pipe $e_1, e_2 \in E_f$ for time step t. The actor can influence the amount of chlorine inside the network by controlling the injection of chlorine pumps $v_1, v_2, \ldots, v_5 \in V_p$ via the corresponding action vector $A_t = (a_1, a_2, \ldots, a_5)$.

3.1 Modeling Approaches

For accurate modeling of the problem, the initial observation template vector S_t has been enhanced to include additional information. As all scenarios display a periodic pattern of water consumption based on the time of day, an additional time component has been appended to the observation to allow for accurate temporal context of a current state. To map each individual time step to its corresponding time of day, the raw simulation time step has been transformed using the following formula.

$$D = (sin(t \cdot C), cos(t \cdot C)) \text{ with } C = \frac{2\pi}{288} \qquad (1)$$

This creates a continuous mapping which repeats periodically every 24 hours, given a step size of 5 minutes between each observation. It is also worth noting that while an encoding of positional year data is also possible, it was not included as observational data, due to the lack of scenarios spanning extended time frames and the small difference between the encoding of two successive time steps t and $t + 1$. Further enhancements include the previous action of the actor model A_{t-1} which is passed to give the model sufficient data to optimize for the smoothness reward. Additional modeling adjustments have been implemented on the following separate models.

PPO

The *PPO* model uses the data provided by two flow sensors and 17 chlorine sensors, and the encoded time of day to decide how much chlorine should be pumped into the WDN at each of the five chlorine pumps. It uses the default model architecture given by SB3 and is trained using the PPO algorithm implemented by the same library. This model is the simplest approach to this problem and serves as a baseline for the following models.

RNN-PPO

To better model the temporal dependencies of an observation at a given time step on observations of preceding time steps, the *RNN-PPO* model includes a long short-term memory (LSTM) module, a type of recurrent neural network (RNN) component, which can memorize data from previous time steps represented by a hidden state vector. The inputs and outputs are identical to those of the PPO model.

Concurrent-PPO

The *Concurrent-PPO* state definition follows a similar approach to that used for the RNN-PPO model but uses a regular PPO implementation as the model. Like the previous approach, it emphasizes strong temporal dependencies of a given observation on previous states of the environment. This is done by allowing the model to retain information from previous time steps and use the additional data to produce a better prediction by observing a trend in change over multiple time steps. As opposed to the RNN approach however, here the time frame and data are explicitly passed to the model as part of a current observation with equal importance. The observation of a given state S_t is defined as

$$S_t = (f(t\text{-}i)_{E_f}, c(t\text{-}i)_{V_c}, \ldots, f(t)_{E_f}, c(t)_{V_c}, D, A_{t\text{-}1}) \quad (2)$$

where $f(t-i)_{E_f}$ and $c(t-i)_{V_c}$ describe the respective sets of flow and chlorine sensor observations up to i time steps into the past and $D = (sin(t \cdot C), cos(t \cdot C))$.

Context-PPO

The original observation space does not pass any topological information about the location of sensors and pumps, making it more difficult for the model to interpret the influence a specific pump has on a junction and its respective sensor readings. To account for this, the *Context-PPO* approach adds proximity information to each sensor reading, describing its minimal distance to each chlorine pump on the WDN graph. Formally, for each junction $v_i \in V_c$ which produces a sensor reading, we append the minimal distance between the respective junction v_i and all junctions $u_j \in V_p$ to which the model can apply an action a_j. This results in a zipped observation state of the shape

$$s_t = (f(t)_{e1}, f(t)_{e2}, Z_3, Z_4, \ldots, Z_{17}, D, A_{t-1}) \quad (3)$$

where $Z_i = (c(t)_{v_i}, d(v_i, u_1), d(v_i, u_2), \ldots, d(v_i, u_5))$ and $d(u, v)$ describe the minimal distance between two junctions in a graph.

3.2 Reward Function

The reward function consists of a set of partial reward functions whose weighted sum make up the final reward of a given state-action pair. Each of the partial reward functions was designed to reflect the model's evaluation metrics, with further adjustments to emphasize sensitivity to local variations rather than a global mean and consider only the data which is observable by the actor.

Chlorine Reward

The partial reward for chlorine describes a measure on how far the chlorine value of each sensor deviates from the safe range of $[0.2, 0.4]$. Experimentation during training has shown that a stricter reward distribution and continuity at every point resulted in better feedback for the actor as opposed to not deducting any reward for the safe range. Therefore the reward function has been modeled as

$$r_{cl} = \alpha \cdot \sum_{i=0}^{n} (c(t)_{vi} - 0.3)^2 \frac{1}{|V_c|} \quad (4)$$

where 0.3 is set as the midpoint of the safe chlorine range. Small deviations around this center result in mild penalties that increase quadratically as the values move further from the safe zone scaled by a factor α.

Fairness Reward

The fairness reward describes how balanced the distribution of chlorine is across all sensors. To model divergence, the coefficient of variation (CoV) has been computed over all sensors as the following:

$$r_f = \frac{\sigma(c(t)_{V_c})}{\mu(c(t)_{V_c})} \quad (5)$$

where $\sigma(x)$ and $\mu(x)$ describe the standard deviation and mean over the sensor readings at junctions V_c respectively.

Smoothness Reward

The smoothness reward is designed to discourage the model from taking drastic changes in injection rate between two successive time steps. This is done by taking the RMSE between the previous action A_{i-1} and the current action A_i performed by the actor. In addition, experimentation has shown that a target smoothness of zero motivates the model to not inject any chlorine into the network. Therefore instead of rewarding maximum consistency, a small target change of 0.2 for each pump has been added to discourage the model from injecting a constant value. This results in the following final smoothness reward

$$r_s = \sqrt{\frac{\sum_{i=0}^{n} (|a_{i-i} - a_i| - 0.2)^2}{n}} \quad (6)$$

Cost

The cost reward punishes general usage of the pump and is described as the sum over all actions applied

$$r_c = \sum_{i=0}^{n} a_i \quad (7)$$

Final Reward

Every partial reward is weighted by its respective weight w and summed up into a final reward, formally defined as

$$r = -w_{cl} \cdot r_{cl} - w_f \cdot r_f - w_s \cdot r_s - w_c \cdot r_c \quad (8)$$

4 Experiments

4.1 Experimental Setup

Datasets

All experiments were performed on the CY-DBN WDN, which is based on its real-world counterpart located on Cyprus. This WDN contains 256 demand nodes, 335 pipes, one reservoir, which serves as the primary water source, and one elevated tank that helps regulate pressure. Additionally, the simulation contains chlorine measurement sensors at 15 of its nodes as well as two flow sensors. Of the 15 chlorine sensors in the network, five contain chlorine booster stations capable of injecting additional chlorine into the network.

The simulation of the WDN contains a total of eleven different scenarios with varying lengths simulating different parts of the year. The first scenario, labeled "scenario 0", starts on January 1st and lasts a total of six days. The following nine scenarios each continue at the point where its predecessor concludes. The additional eleventh scenario, labeled "scenario 10", simulates the entire duration of a year.

At each simulated second, the simulation calculates new demands across the network, updates water flow rates, simulates chlorine decay as a result of chemical reactions with organic matter and decays, feeds new water into the network and injects additional chlorine at the chlorine booster stations controlled by the actor. As the simulation emulates real world scenarios, water demands vary throughout the day and depending on the season with each node following a unique demand profile which follows an approximate periodic pattern.

To provide realistic starting conditions for an actor controlling the booster stations, the first three days of all scenarios are hidden from the model and are used to initialize a plausible starting state and chlorine levels at all nodes. After the third day, the actor is able to interact with the model by controlling the five booster stations.

Water that enters the WDN from the reservoir has a random chlorine concentration between 0.4 mg/L and 0.6 mg/L and has a variable amount of organic matter, that reacts with and reduces chlorine concentration. This randomness makes the chlorine decay unpredictable and adds additional uncertainty to the disinfection procedure. The organic matter levels vary depending on the month and are based on seasonal patterns, which is especially relevant for the 365-day scenario.

In the six-day scenarios, one contamination event occurs at a random time and random node and lasts between one and eight hours. During this event, pathogens and additional organic matter are injected into the system which react with chlorine. In the 365-day scenario, 15 of these contamination events happen during its complete time span. It is part of a booster station actor's task to adapt to those sudden changes.

Model Structure

For the baseline *PPO* model, we use the default SB3 *MlpPolicy* which consists of two hidden fully-connected (FC) layers containing 64 neurons each paired with Tanh layers. This is replicated for the actor and critic such that they have separate hidden layers. It is followed by an output layer for the actor with five values, one for each action $a_i \in A_t$, and an output layer for the critic describing the value of performing A_t at state S_t.

The *RNN-PPO* model uses the default SB3 recurrent policy *MlpLstmPolicy* which is based on the *MlpPolicy* above, with an added LSTM unit of hidden state size 256 placed before the hidden layers of both the actor and the critic.

Context-PPO largely inherits the same implementation as the baseline PPO model, but has been extended to include an additional hidden layer for both the actor and critic containing an additional 64 neurons.

For *Concurrent-PPO* the same structure as in *Context-PPO* is used, with an additional adjustment for the two time step memory variant of the model. To adjust to the large increase in observational data, the first hidden layer of the function decoder has been increased to 128 neurons.

For every model instance, all observations S_t, prior to their input to the network, were normalized to be zero-centered with a standard deviation of 1 using running statistics collected over the corresponding training runs.

Parameters

For every implementation of the model, the reward weights have been set to the same values: prioritizing chlorine boundary violations with $w_{cl} = 0.6$, followed by a focus on smooth changes with $w_s = 0.2$. Higher weights for fairness tended to encourage the model to remain idle (see section 5.2), and thus have been reduced to $w_f = 0.1$ to mitigate the issue. The weight for cost has similarly been decreased to $w_c = 0.1$ for the same reason.

For the *Concurrent-PPO* model, two instances were trained with varying context windows. One used a single past time step, while the other incorporated data from two time steps back.

All models were trained using the default training parameters given by SB3 PPO implementation, with a learning rate of $lr = 0.003$, a gamma value of $\gamma = 0.99$, a value coefficient of 0.5 and the entropy coefficient set to 0.0.

All models were trained in two separate instances, each on a different set of scenarios. As *PPO* is computed entirely on the CPU and the base implementation of EPANet only utilizes a single CPU core, the first instance was trained in parallel on scenarios 0 through 9 utilizing SB3's vectorized environment to benefit from a linear performance increase. As scenario 10 simulates a drastically longer time frame at higher computational cost, but contained more unique training data points on its own, it was used to train a second separate model. All models were trained for at least 300,000 steps, with the best model of each variation chosen for evaluation.

4.2 Evaluation Procedure and Metrics

Evaluation Measures

The evaluation measures of the models were given by the challenge conditions and value the same general rating as-

pects as the reward function in a more global scope. Evaluation is performed on collected data from each time step of a given scenario which is acted on by its respective model. A model should archive a score as low as possible for each evaluation metric. All models were evaluated on every six-day scenario with five complete episodes each.

Chlorine Violations

The chlorine violation evaluation score quantifies the models ability to keep the chlorine value of all junction in the safe bounds of $[0.2, 0.4]$ It is defined as the average count over all junctions and time steps which are inside the range of safe chlorine values. Formally described as

$$\frac{1}{T|V|} \sum_{v \in V} \sum_{t=1}^{T} \mathbf{1}_{[a,b]}(c_v(t)) \tag{9}$$

where V is the set of all junctions in the network, T describes the total number of time steps in a given scenario and $\mathbf{1}_{[a,b]}$ is defined as $\mathbf{1}_{[a,b]}(x) = 1$ iff $x \in [0.2, 0.4]$.

Fairness

Fairness measures how consistently a model distributes chlorine across different junctions. It is defined as the maximal average difference in chlorine boundary violations between any two pumps, computed over all time steps in a scenario.

$$\max_{(v1,v2) \in V} \frac{1}{T} \sum_{t=0}^{T} \mathbf{1}_{[a,b]}(c_{v1}(t)) - \mathbf{1}_{[a,b]}(c_{v2}(t)) \tag{10}$$

Smoothness

Smoothness rates the local difference of a model's actions. A correctly trained model should aim for small adjustments in-between time steps and not apply sharp changes in injection. It is evaluated as the largest mean difference in action values between two successive time steps.

$$\max_{a} \frac{1}{T-1} \sum_{t=1}^{T-1} |a(t) - a(t+1)| \tag{11}$$

Infection Risk

Infection Risk quantifies the probability of an individual becoming ill from a given amount of pathogens in the water. Infection risk is calculated over all contamination events in a scenario, taking into consideration the number of people accessing a specific junction. The infection risk for a single person is defined as

$$\text{Risk} = 1 - \exp(-r * \text{Dose}) \tag{12}$$

where r describes the risk constant of a specific pathogen, which is multiplied by its dose in a given junction. The simulation models a global risk by performing an approximation of the number of people that have access to a single junction and the number of days they are exposed to a pathogen. The full calculation consists of a weighted summation over all junctions to obtain a final percentage value which describes the risk of an infection. The complete implementation has been omitted for brevity and can be seen in the original source code [Artelt *et al.*, 2025].

Cost

Cost measures how much chlorine has been injected into the network throughout the duration of the simulation. The model should use as little chlorine as necessary to keep all junctions in bounds. It is defined as the sum of all chlorine values inside the WDN for every time step of the simulation.

$$\sum_{t=1}^{T} \sum_{i=0}^{|V|} u_{v_1}(t) + u_{v_2}(t) \tag{13}$$

5 Results

In the following, the evaluation metrics for each model are averaged over all episodes and scenarios. The averages over the episodes for every scenario are provided in Appendix A.1.

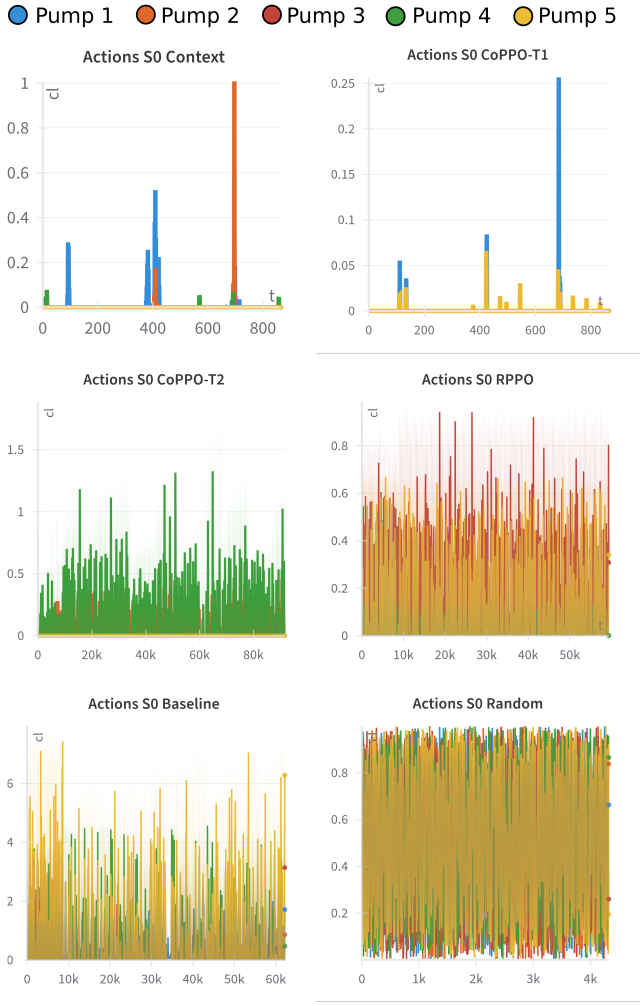

Figure 2: Evaluation plots of action outputs for each model over a complete run on scenario 0. The x axis depicts a single time step S_t, while the y axis shows the chlorine injections A_t at a specific point in time.

Metric	Random	PPO	RNN-PPO	Concurrent-PPO T1	Concurrent-PPO T2	Context-PPO
Chlorine Violations	0.348	**0.174**	0.175	0.178	0.178	0.177
Fairness	0.356	0.178	0.178	0.174	**0.173**	0.174
Smoothness	0.687	0.217	0.097	**0.001**	0.022	0.017
Infection Risk	36.640	18.320	18.329	18.327	18.327	**18.255**
Cost	4324.221	1403.023	236.750	**1.277**	113.099	28.216

Table 1: Results of all models averaged over all scenarios. Lower values indicate better performance.

5.1 Evaluation Results

During training, observations of the environment are normalized using a running mean and standard deviation such that the input data for the models are zero-centered with a standard deviation of 1. Due to corruption of normalization statistics that were paired with each model, additional training in form of five warm-up episodes was introduced before evaluation to ensure that the normalized observation values produce a realistic state expected by the model.

In addition to the previously noted *PPO* baseline model, we compared a random baseline that uniformly sampled from a range $[0, 1]$. This specific interval was chosen, because it approximates a plausible amount of chlorine injections.

The evaluation results of all models are presented in Table 1. The table shows, that every model outperforms the random baseline by a large margin. Notably the baseline *PPO* model also outperforms the random baseline on every metric by a factor two to three. All models provide very similar results for chlorine violations, fairness and infection risk. The other two metrics have a larger difference, with *RNN-PPO* being smoother by a factor of two and nearly six times cheaper injection costs than *PPO*. *Context-PPO* and *Concurrent-PPO-T1* both show highly similar behavior with a relatively low cost and, as a result, low value of smoothness. *Concurrent-PPO-T2* diverges from its singe time step counterpart, showing much more activity, which is reflected in its cost value, while still maintaining a comparably low smoothness value.

5.2 Discussion

While the scores from the evaluation metrics indicate a generally good performance, all models show several general flaws in each of their handling of chlorine injections. First, despite prioritization of the chlorine bound reward r_{cl}, the inclusion of time of day data D and specific model adaptations that capture a temporal context, all models fail to sufficiently adapt to the cyclic change in chlorine values corresponding to the time of day. This becomes apparent when plotting chlorine sensors over time of each trained model (see Figure 3). The plots show sudden drops in chlorine concentration over multiple time steps at predictable time intervals and highlight how the models fail to adjust to the change more effectively than the two baseline models, both of which exhibiting similar behavior.

As the Cycle in itself describes a visibly predictable pattern in every scenario, the reason for the lack of adaptation in the models might be the underlying complexity of the network itself, where slight chlorine propagation delays combined with flow direction changes might lead to inconsistent relations between the sensor readings of V_p and D, which in terms would require the model to receive more topological data than is available in the *Context-PPO* model combined with additional training time. In the context of *Concurrent-PPO*, this may also suggest that the number of past time steps observed by the model is too limited.

The second central problem of the models might also contribute to this issue, where each implementation tends to predict actions of no injection for either most or all pumps in later stages of the learning process, only responding with chlorine bursts to specific observations, as can be observed in figure Figure 2. Here, both *Context-PPO* as well at the single time step implementation of *Concurrent-PPO* show no injections for extended periods before suddenly spiking at specific pumps instead of a smooth injection behavior. *Concurrent-PPO* and *RPPO* display similar behavior. While they adjust a subset of pumps at a constant rate, they generally maintain zero injection across most others.

Further signs of insufficient adaptation can be seen in figure Figure 3, which shows that, while most sensors are within bounds, several sensors stay far below the minimum chlorine concentration throughout the duration of the scenario. Comparison with the baseline models also shows a high similarity and indicates, that the chlorine levels are a property of the scenario itself and not a result of the models actions, further demonstrating the models limited influence on the environment.

A partial reason is the modeling of reward functions, which in their sum encourage the model to produce an output of zero. More precisely, the reward for fairness and smoothness is both maximized when injections and the chlorine in the network stays mostly constant over multiple time steps. Combining this with the cost reward that punishes any chlorine injection, the model slowly learned to reduce its actions to a local reward maxima, with the compromise of slight reward loss for chlorine bounds treated as acceptable noise. The issues still persisted despite changes in the target smoothness and weight distribution of partial rewards. This could also explain the lack of adaptation to time data, as the model is converging faster to producing zero outputs than it is learning the cyclic nature of the network outputs. To test this hypothesis, additional experimentation has been performed by training a *Context-PPO* model with all weights except for w_{cl} set to 0 on the scenario 10. The resulting model surprisingly still converged toward outputting zero

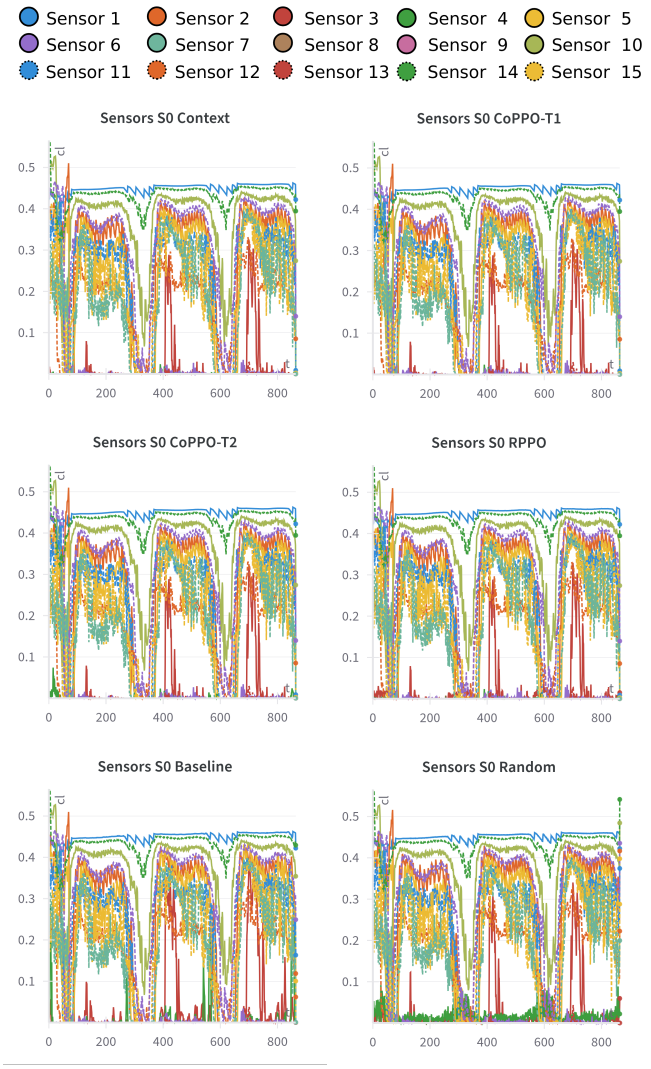

Figure 3: Evaluation plots of sensor readings for each model over a complete run on scenario 0. The x axis depicts a single time step S_t, while the y axis shows the chlorine readings $c(t)_{V_c}$ at a specific point in time.

actions as can be seen in figure Figure 4, pointing towards a more hidden issue.

Another notable observation is the significant difference between the evaluation results of *Concurrent-PPO-T1* and *Concurrent-PPO-T2*, with the latter showing a much higher action activity. One reason could be that the increased information input leads to a more stable action output. However, when observing at the models structure of it's wider function decoder, it is more likely, that the second models behavior is attributed to it's lesser trained state. In addition, as described previously, convergence towards zero action happens as a slow rate with figure Figure 4 showing that a less trained model shows higher activity, further strengthening this assumption.

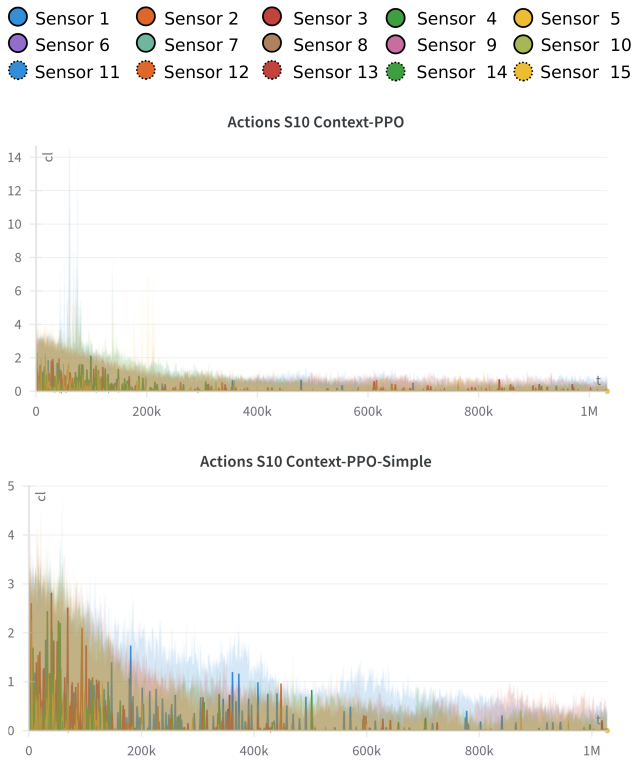

Figure 4: Comparison between action outputs of *Context-PPO* over the course of 1 Mio training steps trained on scenario 10. The upper plot depicts the actions of the model using the final reward function while the lower plot shows a model trained on a simple function with all weights except w_{cl} set to zero. Both eventually converge to producing zero vectors, with sparse burst outputs as outliers.

Furthermore, despite the model's insufficient adaptation, evaluation of the randomized baseline has shown valuable insight into the intensity of the environment staticity. As the plots in Figure 2 and Figure 3 show, despite significant differences in injection behavior between the randomized baseline and trained model, the resulting sensor data shows mostly identical behavior. This demonstrates that, for some sensors, it is difficult or impossible to archive a sensor reading within the safe range without injecting an unsafe amount of chlorine and as a result move other sensors into an unsafe range. This could be attributed to the placement of sensors in regions of high network density, as observed with sensor 12, or to large pump distances, such as in the case of sensor 8. Alternatively, these effects could arise from more complex network dynamics influenced by the global architecture of the system. Further possible evaluation in the future would therefore necessitate a weighting of each junction's sensor when it is considered in the reward function.

Finally, analyzing the results and training the model to respond to contamination events has proven to be difficult due to their unpredictability in terms of their length and point in time of their occurrence for an identical scenario. As we have shown that the current models fail to adapt to readily pre-

dictable changes in chlorine however, analysis of the model's performance on contamination events has been considered unjustified due to the insignificance of such evaluation and a high infection risk of 18.32% on all non-baseline models.

6 Conclusion

The evaluation results demonstrate that the models show insufficient performance in their predictions to act as an injection policy in a real live scenario. As discussed previously, this is partly due to incompatible reward function modeling and partly the result of the environment's high complexity combined with a very limited observational space.

We hypothesize however, that the current approach of models and reward functions could produce better results with a step-wise exposure to the environment's dynamics, which would allow the model to pretrain before being applied on the final scenarios as well as allow for easier analysis of the model's performance. For example, additional simpler scenarios could be synthesized, which only perform a constant chlorination of the water, such that there is no unexpected change of chlorine over the duration of a simulation as well as not including any randomized contamination events. Further scenarios could build on that complexity and introduce deterministic contamination events or slight chlorine changes and eventually have full complexity like the scenarios in this challenge.

On a related note, the current scenarios could also be utilized for training in a nonlinear order by performing off-policy training and assembling samples of different sets of conditions from the environment which can then be trained in a step-wise order to archive the same effect. This however could prohibit the model from learning temporal dependencies and would require manual effort to collect and classify different sets of data.

Regarding the model itself, future work could explore a merge of *RPPO* and *Concurrent-PPO* with *Context-PPO* into a larger observation space or explore different exploration coefficients in order to discourage the model from setting into its zero output state.

In conclusion, we have developed and evaluated several reinforcement learning-based approaches for a simulated chlorine injection task. Our results have demonstrated moderate to low performance across the tested methods. We identified areas for improvement and gave an overview of possible directions for future research, utilizing the current model structures as a foundation.

References

[Artelt *et al.*, 2024] André Artelt, Marios S. Kyriakou, Stelios G. Vrachimis, Demetrios G. Eliades, Barbara Hammer, and Marios M. Polycarpou. EPyT-Flow: A Toolkit for Generating Water Distribution Network Data. *Journal of Open Source Software*, 9(103):7104, 2024.

[Artelt *et al.*, 2025] André Artelt, Janine Strotherm Luca Hermes, Barbara Hammer, Stelios G. Vrachimis, Demetrios G. Eliades Marios S. Kyriakou, Marios M. Polycarpou, Sotirios Paraskevopoulos, Stefanos Vrochidis, Riccardo Taormina, Dragan Savic, and Phoebe Koundouri. 1st AI for Drinking Water Chlorination Challenge. https://github.com/WaterFutures/AI-for-Drinking-Water-Chlorination-Challenge-IJCAI-25, 2025.

[Frederick *et al.*, 2024] Flavia D. Frederick, Malvin S. Marlim, and Doosun Kang. Optimization of Chlorine Injection Schedule in Water Distribution Networks Using Water Age and Breadth-First Search Algorithm. *Water*, 16(3), 2024.

[Perelman and Ostfeld, 2025] Gal Perelman and Avi Ostfeld. Data Enabled Predictive Control for Water Distribution Systems Optimization. *Water Resources Research*, 61(4):e2024WR039059, 2025. e2024WR039059 2024WR039059.

[Raffin *et al.*, 2021] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017.

[US Environmental Protection Agency, 2020] US Environmental Protection Agency. EPANET: Application for Modeling Drinking Water Distribution Systems, 2020. Accessed: 2025-08-03.

A Appendix

A.1 Full Result Tables

The following tables show the full results for each presented model. The values for each scenario are averaged over five evaluation episodes. The total average for each evaluation metric is calculated by averaging over all scenario averages. Lower reported values are better.

PPO	Scenarios										
Metric	0	1	2	3	4	5	6	7	8	9	Average
Chlorine Violations	0.114	0.185	0.185	0.175	0.195	0.175	0.195	0.146	0.181	0.192	0.174
Fairness	0.195	0.194	0.194	0.200	0.118	0.200	0.118	0.200	0.200	0.160	0.178
Smoothness	0.442	0.119	0.119	0.143	0.259	0.143	0.259	0.266	0.153	0.265	0.217
Infection Risk	5.160	8.692	8.692	42.953	1.739	42.953	1.739	40.651	24.774	5.846	18.320
Cost	2264.307	1136.254	1136.254	1291.115	1199.036	1291.115	1199.036	1831.062	1340.184	1341.866	1403.023

Table 2: Results of the PPO model for each scenario and averaged over all scenarios. Lower values indicate better performance.

RNN-PPO	Scenarios										
Metric	0	1	2	3	4	5	6	7	8	9	Average
Chlorine Violations	0.115	0.185	0.185	0.175	0.195	0.175	0.195	0.146	0.182	0.192	0.175
Fairness	0.195	0.194	0.194	0.200	0.118	0.200	0.118	0.200	0.200	0.160	0.178
Smoothness	0.093	0.099	0.099	0.093	0.101	0.093	0.101	0.100	0.095	0.099	0.097
Infection Risk	5.160	8.692	8.692	42.957	1.739	42.957	1.739	40.669	24.835	5.847	18.329
Cost	245.830	235.740	235.740	227.559	243.636	227.559	243.636	250.462	231.598	225.7424	236.750

Table 3: Results of the RNN-PPO model for each scenario and averaged over all scenarios. Lower values indicate better performance.

Concurrent-PPO-T1	Scenarios										
Metric	0	1	2	3	4	5	6	7	8	9	Average
Chlorine Violations	0.194	0.194	0.194	0.200	0.118	0.200	0.118	0.200	0.200	0.159	0.178
Fairness	0.114	0.184	0.184	0.174	0.194	0.174	0.194	0.146	0.181	0.192	0.174
Smoothness	0.002	0.001	0.001	0.002	0.001	0.002	0.001	0.001	0.001	0.001	0.001
Infection Risk	5.160	8.690	8.690	42.951	1.738	42.957	1.738	40.665	24.830	5.846	18.327
Cost	1.600	0.987	0.987	1.830	0.792	1.835	0.792	2.230	0.906	0.814	1.277

Table 4: Results of the Concurrent-PPO model for each scenario and averaged over all scenarios. Lower values indicate better performance.

Concurrent-PPO-T2	Scenarios										
Metric	0	1	2	3	4	5	6	7	8	9	Average
Chlorine Violations	0.194	0.194	0.194	0.200	0.118	0.200	0.118	0.200	0.200	0.159	0.178
Fairness	0.114	0.184	0.184	0.174	0.194	0.174	0.194	0.141	0.181	0.192	0.173
Smoothness	0.019	0.024	0.024	0.025	0.018	0.025	0.018	0.021	0.032	0.016	0.022
Infection Risk	5.160	8.690	8.691	42.957	1.730	42.957	1.738	40.667	24.834	5.845	18.327
Cost	61.550	131.340	131.346	132.236	110.042	132.235	110.042	79.774	134.177	108.248	113.099

Table 5: Results of the Concurrent-PPO model for each scenario and averaged over all scenarios. Lower values indicate better performance.

Context-PPO	Scenarios										
Metric	0	1	2	3	4	5	6	7	8	9	Average
Chlorine Violations	0.194	0.194	0.190	0.200	0.118	0.200	0.118	0.200	0.200	0.160	0.177
Fairness	0.115	0.184	0.184	0.175	0.195	0.170	0.195	0.146	0.181	0.192	0.174
Smoothness	0.007	0.022	0.022	0.019	0.018	0.019	0.018	0.013	0.016	0.013	0.017
Infection Risk	5.160	8.691	8.690	42.950	1.738	42.950	1.700	40.000	24.830	5.840	18.255
Cost	8.024	39.620	39.610	34.297	21.366	34.297	21.366	27.530	35.327	20.720	28.216

Table 6: Results of the Context-PPO model for each scenario and averaged over all scenarios. Lower values indicate better performance.