Neuro-Symbolic Pump Scheduling for Safe and Cost-efficient Water Distribution Networks

Summary of changes to the EMAS Post-Proceedings version

Significant changes throughout the paper are highlighted in blue.
The following is a summary of the changes:

- We have added more details about the DRL (D3QN) in Sec 3.2.
- We have considerably expanded the BDI section (Sec 3.3) to provide more detail.
- Concerns raised about BDI over-reliance were addressed by providing a qualitative comparison against recent DRL-only WDNs-optimisation controllers, constrained-RL systems, and similar safety-optimisation approaches in non-WDNs environments in the new Related Work.
- We have clarified that the reduced safety performance for Tank 6 is due to the shared control of Tanks 6 and 7 via pump group 5. The generalised BDI plans used by Hydro-MAS prioritise generalisability over tank-specific control granularity, leading to less optimal performance for Tank 6 but improved performance for Tank 7.
- The disparity in operating costs between the two approaches has been addressed at the end of Section 4. We have clarified that the marginal increase in operating costs may be due to the BDI agent prioritising safety over cost, occasionally overriding DRL-selected OFF pump actions by switching them ON.
- Additional details were added about the experiments (in particular, the DRL parameters) and the results in Sec 4. Evaluation.

Neuro-Symbolic Pump Scheduling for Safe and Cost-efficient Water Distribution Networks

Mahirul Islam[1], Fulong Yao[2], Wanqing Zhao[1], Rafael C. Cardoso[3], and Mengwei Xu[1]

[1] School of Computing, Newcastle University, Newcastle upon Tyne, United Kingdom
{H.M.M.Islam1, Wanqing.Zhao, Mengwei.Xu}@newcastle.ac.uk
[2] School of Computer Science and Informatics, Cardiff University, Cardiff, United Kingdom yaof@cardiff.ac.uk
[3] School of Natural and Computing Sciences, University of Aberdeen, Aberdeen, United Kingdom
rafael.cardoso@abdn.ac.uk

Abstract. The Pump Scheduling Problem is a highly challenging real-world control task in Water Distribution Networks (WDNs) that aims to minimise operational costs while meeting safety requirements (e.g., minimum and maximum allowable tank levels). The latest Deep Reinforcement Learning (DRL) techniques are effective for cost optimisation but can still violate safety constraints at deployment, despite explicit safety considerations during training. Furthermore, evolving safety requirements (e.g., due to seasonal considerations) make retraining for minor changes to safety specifications disproportionately expensive. To address these challenges, we present a neuro-symbolic framework that pairs a pre-trained DRL agent with a symbolic Belief-Desire-Intention (BDI) agent to supervise WDNs safety. Our implementation and preliminary empirical results demonstrate improved safety compliance over a DRL-only baseline while maintaining comparable cost performance.

Keywords: Water Distribution Networks · Neuro-Symbolic Artificial Intelligence · Deep Reinforcement Learning · BDI Agents.

1 Introduction

Water Distribution Networks (WDNs) are large-scale cyber-physical systems comprising physical assets (such as pipes, junctions, tanks, and pumps) and digital control decisions (e.g., pump schedules) that deliver an essential public service. While their purpose is clear, their operation is non-trivial and energy-intensive, accounting for approximately 2–3% of national electricity consumption in the UK [14], largely because pumping requires significant energy.

As a result, the operation of pumps motivated the *Pump Scheduling Problem* (PSP): an optimisation problem that seeks pump control strategies to minimise operational cost while maintaining hydraulic safety. Recent work has treated PSP as a sequential decision-making problem and has adopted Deep Reinforcement

Learning (DRL) as an effective approach to learn control policies from simulation or operational data, as exemplified in [24], to optimise long-horizon energy cost.

However, even when safety is explicitly considered during training (e.g., via penalty for water-level violations), DRL policies can still breach safety constraints at deployment. This is because learned policies are inherently stochastic and may still produce occasional safety violations. In addition, safety requirements in WDNs can be time-varying (e.g., across seasons or under operator policies), and accommodating small changes in safety specifications may require full retraining or extensive retuning, making updates disproportionately costly.

This paper proposes a Neuro-Symbolic (NeSy) framework that combines data-driven DRL for cost-efficient pump scheduling with a Belief-Desire-Intention (BDI) [6,7] symbolic agent that monitors safety compliance and intervenes in high-risk situations. To achieve so, we first train a DRL pump scheduler using the latest DRL technique, e.g., Double Duelling Deep Q-Network (D3QN) [22], chosen for its stability in discrete action spaces and its reduced Q-value overestimation. Then, we deploy the successfully trained DRL on a de facto standard WDNs benchmark using the EPANET hydraulic simulator[1]. At runtime, the DRL controller proposes pumping control actions, which are checked by a BDI agent implemented in Jason [5] for safety compliance, and can be overridden before execution in the EPANET simulator to prevent potentially unsafe pumping actions. The resulting framework is fully implemented[2] to coordinate state exchange and action execution across EPANET simulator, the D3QN controller, and the BDI agent. Preliminary results show a considerable reduction in the frequency of tank-level violations, with only a modest increase in operational cost relative to DRL-only.

The remainder of the paper is organised as follows. Section 2 presents a discussion of related work. Section 3 introduces our NeSy framework. Section 4 provides the preliminary empirical results, and Section 5 concludes.

2 Related Work

The PSP has been widely studied as a constrained optimisation problem, with the objective of cost minimisation while maintaining hydraulic safety. DRL has emerged as a promising alternative to classical optimisation by learning control policies directly through interaction with WDNs. This section reviews existing work against our work in three areas: (1) Safe Reinforcement Learning (RL), (2) DRL-based pump scheduling, and (3) NeSy approaches.

2.1 Safety in Reinforcement Learning

Safe RL has emerged as a major research direction for safety-critical autonomous systems, which includes runtime safety mechanisms that monitor or modify the

[1] `https://www.epa.gov/water-research/epanet/` (Accessed: 27/04/2026)
[2] `https://github.com/Mahir-Islam/hydro_mas` (Accessed: 27/04/2026)

agent's actions during deployment [18]. Examples include shielding [2] and barrier functions [20]. Safety may also be directly incorporated into policy optimisation through constrained RL [8]. For example, shielded RL has been applied in improving safety in physical robotic systems [10]. Although promising, safe RL has received little attention in WDNs pump-scheduling settings. We refer interested readers to [18] for more details.

2.2 DRL for Pump Scheduling

Early work on the PSP employed deterministic optimisation methods, including linear [17,11], nonlinear [19,23], and mixed-integer programming [17,25]. Though they can produce high-quality schedules, they often require large datasets of hydraulic simulations, making real-time deployment infeasible [24].

As a result, recent work instead applies DRL to learn pump-scheduling policies offline before deployment, in particular using EPANET simulations of WDNs. For example, the work [16] uses Proximal Policy Optimisation (PPO) controllers, training the agents on offline policies prior to the scheduling period. Meanwhile, the work [9] employs Exploration-Enhanced PPO (E-PPO), which introduces an exploration mechanism that improves energy efficiency while reducing long-term tank-level variation compared to conventional PPO. These studies demonstrate the capability of DRL to optimise operational costs while maintaining acceptable hydraulic behaviours. However, safety is learned implicitly through the reward function. Reward-based safety provides no guarantee that learned policies will remain safe under changing demand patterns or out-of-distribution operating conditions, which motivates the NeSy approaches.

2.3 Neuro-Symbolic Decision Making

NeSy approaches combine data-driven learning techniques with symbolic reasoning to exploit the complementary strengths of both paradigms [13]. NeSy architectures can be represented using dual-process theory [12,4]: neural models (System 1) provide rapid inference and generalisation from experience under routine conditions, while symbolic systems (System 2) provide explicit knowledge representation and logical reasoning under less foreseeable conditions.

While NeSy frameworks have been limited in WDNs pump optimisation problems, the effectiveness of this paradigm has been demonstrated in other safety-critical applications, and interest in NeSy systems has been increasing rapidly [3]. For example, BDI agents have been combined with Machine Learning (ML) systems for autonomous driving, in which a BDI agent supervises ML controllers to ensure safety [1]. Unlike the work [1] that employs an orchestrator to arbitrate between NeSy systems, our architecture centralises decision-making within the BDI agent itself. The DRL controller proposes actions optimised for operational efficiency, while the BDI agent performs symbolic reasoning before execution, allowing unsafe actions to be corrected using explicit domain knowledge.

3 DRL–BDI Framework

Our proposed NeSy approach balances two competing objectives: maintaining safe tank levels and minimising the energy cost of pump operation. Our framework couples three components: (i) a DRL controller that proposes cost-efficient pump actions, (ii) a symbolic BDI supervisor that monitors hydraulic safety, and (iii) a hydraulic simulation environment that executes actions and returns the evolving WDNs state. Figure 1 overviews the resulting DRL–BDI architecture.

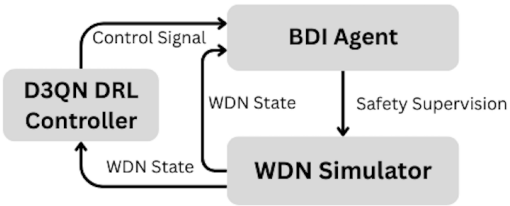

Fig. 1. DRL–BDI framework applied to WDNs.

3.1 Water Distribution Network Environment

We apply our framework to *D-Town* [15], a widely used benchmark WDNs that approximates the water supply for a small city, as depicted in Fig. 2. D-Town comprises 400 junctions, 11 pumps, and 7 tanks arranged into 5 demand zones. Each demand zone contains a pump group (a cluster of pumps operated as a single unit), a primary storage tank, and a subset of junction demands. Hydraulic dynamics are simulated using EPANET, which advances the network state at hourly timesteps. At each timestep, the simulator (i) receives pump actions, (ii) performs the hydraulic computation for the next hour, and (iii) returns the updated WDNs state, including tank levels and energy usage.

3.2 DRL Agent

The DRL agent uses a Double Duelling Deep Q-Network (D3QN) [22] that proposes pump-group actions to minimise energy cost while considering safety by keeping tank levels within safe operating ranges. It employs a holistic data-driven control scheme for sequential decision-making problems. For improved learning efficiency and accuracy in reward estimation, D3QN consists of a Q-Network and a Target Network, both of which contain a state-value function $V(s)$ and an advantage function $A(s, a)$. Here, the target network is a copy of the Q network used to compute future rewards, and it is updated only every few timesteps. The

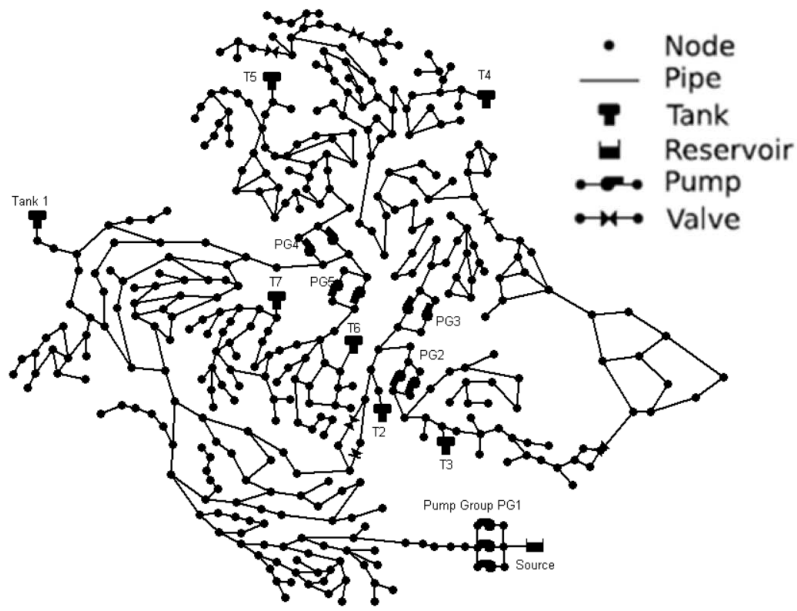

Fig. 2. Topological graph of the D-Town benchmark WDNs.

Q-value determining the expected reward of a control action is thereby calculated as: $Q(s,a) = V(s) + A(s,a) - mean(A(s,a))$; we refer the interested readers to full details that can be found in e.g., [21].

At each timestep, the agent observes a state vector s summarising recent demand/price history and current tank levels from the EPANET simulator, then returns an action a that optimises the control of each pump group. We now very briefly describe how we train a DRL agent using D3QN for the D-Town scenario.

There are five pump groups in D-Town, each with two operational settings: on (1) or off (0). An action is therefore a 5-bit activation vector. Following common practice for discrete Q-learning, we encode actions as integers in $[0, 31]$ and decode them to binary vectors, e.g., $31_{10} = 11111_2 \mapsto [1, 1, 1, 1, 1]$. All pumps within a group are operated together rather than independently. Meanwhile, the state comprises demand, price, and tank levels. We use a context window of 6 timesteps (i.e., 6 hours)[3] to expose the short-term temporal structure to the Q-network. All tanks (including those indirectly controlled by a pump) account for violations of water-level constraints to provide a complete hydraulic context. Consequently, the training objective is a weighted reward combining cost and safety where for timestep t, $w_1, w_2 \in [0, 1]$ and $w_1 + w_2 = 1$:

$$r_t = w_1 \cdot \text{cost_value}(t) + w_2 \cdot \text{safety_value}(t) \tag{1}$$

[3] Optimal value found after hyperparameter tuning during training.

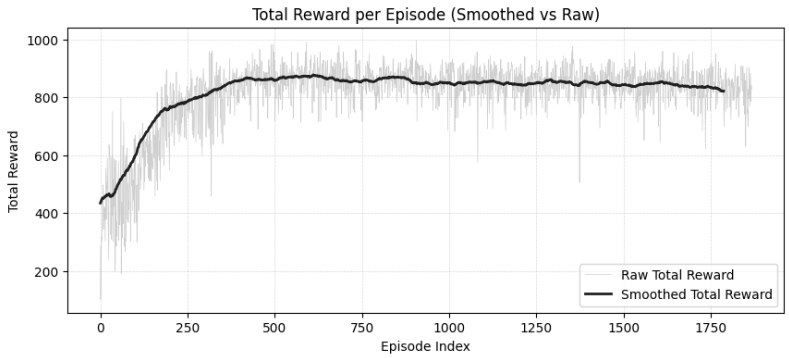

Fig. 3. Training reward over episodes for D3QN controller. Convergence occurs around episode 600–700.

Safety is assessed by normalised tank levels $[0, 1]$: the *safe bounds* are $[0.2, 0.8]$, while the simulator itself enforces *hard bounds* $[0.1, 0.9]$ to avoid infeasible hydraulics. Deviations from safe bounds incur penalties using a scaled trapezoidal membership function with parameters $\{0, 0.2, 0.8, 1\}$ (minimum value -1). Cost is computed as total pump energy usage multiplied by the electricity price, and normalised using an inverse-logit transformation to stabilise learning.

We train the D3QN controller on historical demand and price profiles until the return stabilises and the policy consistently achieves low operating cost with few safety violations. Figure 3 confirms the reward convergence during training.

3.3 BDI Agent

The symbolic component of our NeSy system is implemented as a BDI agent in *Jason*, serving as a symbolic safety supervisor for the DRL controller. The DRL agent uses safety and cost-efficiency as metrics for learning optimal control policies, while the BDI agent provides explicit, rule-based safety enforcement. At each timestep, the BDI agent evaluates the DRL-proposed action against the current system state before allowing it to be executed in the EPANET simulator. This separation ensures the explicit decoupling between efficiency-dominated learning and safety-focused control.

Beliefs Designs. Beliefs in our BDI agent are partitioned into three categories: static beliefs, perceptual beliefs, and derived (inferred) beliefs.
(1) Static beliefs encode underlying domain knowledge about the WDNs. These are initialised once and remain unchanged:

- pump_group(PG) declares valid pump group identifiers used in the WDNs.
- tank(T) declares valid tank identifiers.
- water_network_causal_zone(PG, TANK, DZ) encodes the primary causal dependency among the controlling pump group, a tank, and its demand zone,

thereby restricting the scope of reasoning to physically meaningful interactions.
- tank_level_warning_range(LB, UB) defines a soft anticipatory safety bound used for early intervention before hard constraint violation.
- tank_level_safe_range(LB, UB) the safety constraints as defined in the WDNs.

(2) Perceptual beliefs are updated every timestep T from EPANET simulator and encode the current system state as follows:

- tank_level(TANK, LEVEL, T) records the normalised water level in the range $[0, 1]$ of tank TANK at timestep T.
- tank_delta(TANK, DELTA, T) records the change in the water level of tank TANK between timestep T-1 and timestep T (i.e., DELTA $= L_T - L_{T-1}$).
- demand(Z, D, T) encodes aggregated demand D at zone Z in timestep T.
- pump_setting(PG, A, T) records the pump action A executed by pump group PG at timestep T.
- proposed_action(PG, A, T) records the pump action A proposed by the DRL policy for pump group PG at timestep T.

(3) Derived beliefs are transient facts produced by reasoning rules, for example: correct_action(PG, A, T), is the action A output of the reasoning cycle for pump group PG at timestep T. It registers the action the BDI agent is proposing to correct the DRL action. These are not persistent and are removed after each reasoning cycle. Due to space constraints, we refer the interested readers to our implementation for more details.

Plan Library Designs. We now discuss the designs of core BDI plans for safety reasoning in WDNs. Firstly, our BDI agent in Jason uses a strictly prioritised plan library, executed according to a decision hierarchy:

$$\text{Safety Correction} \succ \text{Anticipatory Control} \succ \text{Demand-Aware Optimisation} \succ \text{DRL Fallback}$$

This order reflects a safety-first reasoning hierarchy in our BDI agents: immediate safety correction takes precedence over preventing future risks, which in turn takes precedence over performance optimisation. The DRL action is used only as a fallback when no BDI plan is applicable. This priority is supported by Jason's plan selection mechanism, in which applicable plans that appear earlier in the plan library are considered before those that appear later. At each timestep, the agent executes at most one applicable plan according to this ordering.

Safety Correction. This set of plans enforces immediate correction when a tank violates physical constraints. For example, the following plan is triggered only under explicit constraint violation and overrides all lower-priority reasoning.

```
+!reason(PG, A, TANK, LEVEL, _, _, _, T)
  : tank_safe_range(TANK, LB, UB) ∧ LEVEL < LB
  ← +correct_action(PG, 1, T).
```

This reactive correction plan checks whether a safe range tank_safe_range exists for tank TANK, and whether the tank level falls below the lower bound (i.e., the tank level is too low). If this condition is met, the BDI agent proposes action 1 on the corresponding pump group (to switch it ON).

Anticipatory Control. The plans for anticipatory reasoning intervene when the system is still within safety bounds but is trending towards a constraint violation. For example, the following plan employs short-horizon prediction using simple temporal extrapolation, $\hat{L}_{T+1} \approx L_T + \Delta_T$, enabling pre-emptive intervention before a constraint violation occurs.

```
+!reason(PG, A, TANK, LEVEL, _, _, _, T) :
  tank_warning_range(TANK, _, WUB)
  ∧ T > 0
  ∧ tank_delta(TANK, DELTA, T)
  ∧ (LEVEL > WUB | LEVEL + DELTA > WUB)
← +correct_action(PG, 0, T).
```

This plan checks if at least 1 timestep has passed (to ensure a frame of reference for anticipation). $\hat{L}_{T+1}$ is a short-horizon prediction of the tank level at timestep T+1, represented as TANK+DELTA. It is then checked whether it exceeds the upper bound, indicating that the BDI agent anticipates the tank level will soon become too high. In response, it proposes switching the corresponding pump group OFF.

Demand-Aware Reasoning. When no safety-related condition is active, the agent incorporates demand information to improve operational efficiency. It will balance efficiency and constraint satisfaction.

```
+!reason(PG, A, TANK, LEVEL, DEMAND, _, D_UB, T) :
  tank_level_safe_range(LB, UB)
  ∧ LEVEL ≥ LB ∧ LEVEL ≤ UB ∧ DEMAND > D_UB
← !predict_demand(PG, T); +correct_action(PG, 1, T).
```

This plan checks if the tank level is within normal operating bounds, and if the demand is high. If so, it switches the corresponding pump group ON, to ensure junction demands are met.

DRL Fallback. If no BDI plan is applicable, the agent defaults to the DRL-proposed action. This preserves the cost efficiency from the learned policy whenever there are no applicable safety concerns, given the design of the BDI agent.

To conclude, our BDI agent considers how the DRL controller's output affects WDNs safety and structures its own output accordingly. This design makes the symbolic component both interpretable and transparent, as each intervention can be directly traced to a specific rule in the plan library.

Table 1. Tank Level Constraint Compliance over 168 hours (Tank 2 excluded)

Metric	Tank 1	Tank 3	Tank 4	Tank 5	Tank 6	Tank 7	Overall
DRL (% Compliance)	84	64	61	80	98	69	76
Hydro-MAS (% Compliance)	97	94	62	92	83	81	85

3.4 DRL–BDI Implementation

We have implemented the overall framework, and the source code is publicly available online[4]. In detail, the Python-based DRL controller, the Jason BDI agent, and the EPANET simulator are integrated in a closed loop. At each timestep, the simulator provides the current WDNs state to the DRL agent. The DRL controller proposes an action, which will also be passed to BDI agents as a proposed action, by evaluating its Q-network on the corresponding state vector. Meanwhile, the EPANET simulator also passes the network state to BDI agents as knowledge. The BDI agent then deliberates and either (i) overrides the proposal to enforce safety/anticipation rules or (ii) accepts it via the fallback mechanism. The selected action is finally applied to the EPANET simulator to advance the network to the next timestep. In our current implementation, the DRL–BDI bridge uses a lightweight file-based inter-process communication mechanism to exchange state and actions across the Python and Java runtimes.

4 Evaluation

We evaluate our NeSy framework by comparing it against a D3QN-only baseline (henceforth referred to as DRL). Specifically, we assess (i) the rate of constraint violations (how often tank levels leave the permitted range), (ii) the severity of safety deviations (how far and for how long tank levels drift outside the safe bounds), and (iii) operational cost. To evaluate, we have run the full 168-timestep D-Town simulation (168 hours) for each system. We then collect time-series data for cost, constraint violations, and violation severity.

The neural network has a batch size of 64, a discount factor of 0.99, a learning rate of 0.001, and an initial exploration rate of 0.9, which decays by 0.5% per timestep until it reaches a floor of 0.1.

Table 1 shows that the NeSy controller improves overall constraint compliance from 76% to 85%, representing a 9 percentage-point increase over DRL-only control. These improvements are consistent across most tanks, except tanks 4 and 6. Tank 6 exhibits reduced performance with Hydro-MAS because the tank level is controlled by pump group 5, which also controls the level of tank 7. Our BDI plans are designed to be generalised and do not include specific plans to address tank 6, as a trade-off between generalisability and control granularity. Consequently, this leads to less optimal performance on tank 6, but provides a better performance for tank 7. Both controllers show lower compliance for Tank

[4] `https://github.com/Mahir-Islam/hydro_mas` (Accessed: 27/04/2026)

4 because rapid level fluctuations with a lag time make control difficult. Tank 2 is not included, since it operates independently of the action space.

Table 2. Wilcoxon Signed-Rank Test Results for Safety Metrics ($n = 168 \times 6 = 1008$).

Metric	Mean (DRL)	Mean (NeSy)	Wilcoxon p-value
Deviation	0.1911	0.1682	0.00027
Violation Severity	0.0814	0.0553	5.0e-08
Violation Occurrence	0.2402	0.1520	3.2e-07

Table 2 shows the severity of safety deviations using three derived metrics computed over all tanks (excluding Tank 2), yielding $n = 168 \times 6 = 1008$ observations per controller. These three metrics can be determined from the tank levels: the *deviation* is defined as the deviation from the middle tank level (0.5); the *violation severity* is the positive excess beyond the safe interval [0.2, 0.8] with negative values set to zero; finally, *violation occurrence* is a binary indicator equal to 1 if the violation severity is greater than 0, otherwise 0. For each metric, a Wilcoxon signed-rank test was conducted to compare paired samples without assuming normality or homogeneity of variances. It is a statistical test used to determine whether two samples differ significantly. This Wilcoxon signed-rank test shows that the safety measures of the NeSy approach differ significantly from those in the DRL-only setting.

Finally, we examine the operational cost of these two approaches. Operational cost is measured using the normalised cost value over 168 hours. Fig. 4 shows a comparison of costs. Overall, there is a modest, statistically significant increase in cost. The DRL-only controller achieves an average normalised cost of 0.168, whereas our NeSy approach achieves the average normalised cost of 0.218.

In summary, our NeSy DRL–BDI framework improves WDNs safety metrics (reduced deviation, reduced violation prevalence, and reduced violation severity) while incurring a modest increase in operational cost relative to DRL-only. One explanation for the marginal increase in operating costs may be that while the DRL agent optimises for both cost and safety, the BDI agent optimises exclusively for safety. The DRL agent may learn to select OFF pump movements to account for cost, while inadvertently affecting safety, whereas in the NeSy model, the BDI agent, which does not account for cost, may flip the OFF action to ON.

5 Conclusion and Future Work

This paper presented a neuro-symbolic DRL–BDI framework for pump scheduling in WDNs. The DRL system provides a continuous action proposal optimised for cost and efficiency, while the BDI agent enforces hard safety constraints and introduces explicit temporal reasoning through anticipatory plan rules. The BDI agent considers how the DRL controller's output affects WDNs safety and

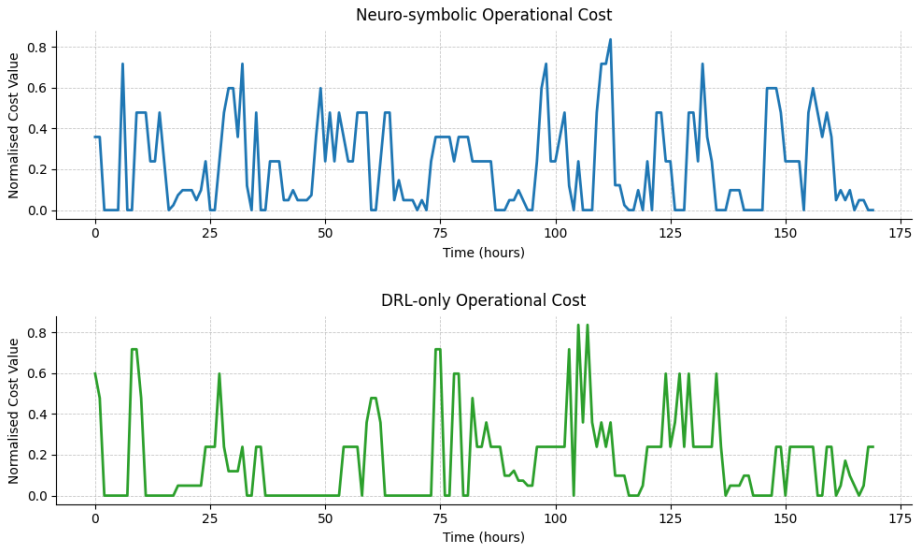

Fig. 4. Cost of operation over time.

structures its own output accordingly. This design makes the symbolic component both interpretable and transparent, as each intervention can be directly traced to a specific rule condition in the plan library. On the standard D-Town benchmark, our preliminary results show that this hybrid supervision improves safety compared to a DRL-only baseline, reducing the frequency and severity of safe-bound deviations while keeping operational cost broadly comparable.

Several directions follow naturally from these findings. First, the evaluation here is intentionally preliminary and should be strengthened with longer horizons and broader operating conditions (e.g., diverse demand/price profiles and stress scenarios). Second, we will extend the supervisor beyond tank-level bounds to encompass additional operational objectives, such as pump-switching frequency (wear), pressure-related constraints, and water quality considerations. We also plan to improve the implementation with a more efficient messaging interface to scale to longer or higher-frequency simulations. Finally, a major avenue for future work is to improve the symbolic supervisor itself. In particular, we will refine systematically the plan rules of the BDI agent and its intervention strategy to further reduce violation rates while preserving the cost-efficiency of the underlying DRL policy.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Al Shukairi, H., Cardoso, R.C.: ML-MAS: A hybrid AI framework for self-driving vehicles. In: Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems. p. 1191–1199. AAMAS '23, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2023). https://doi.org/10.65109/ETRN2911
2. Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., Topcu, U.: Safe reinforcement learning via shielding. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. AAAI'18, AAAI Press (2018). https://doi.org/10.1609/aaai.v32i1.11797
3. Bhuyan, B.P., Ramdane-Cherif, A., Tomar, R., Singh, T.: Neuro-symbolic artificial intelligence: a survey. Neural Computing and Applications **36**(21), 12809–12844 (2024). https://doi.org/10.1007/s00521-024-09960-z
4. Booch, G., Fabiano, F., Horesh, L., Kate, K., Lenchner, J., Linck, N., Loreggia, A., Murgesan, K., Mattei, N., Rossi, F., Srivastava, B.: Thinking fast and slow in ai. Proceedings of the AAAI Conference on Artificial Intelligence **35**(17), 15042–15046 (May 2021). https://doi.org/10.1609/aaai.v35i17.17765
5. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak using Jason. John Wiley & Sons, Chichester, UK (2007). https://doi.org/10.1002/9780470061848
6. Bratman, M.E.: Intention, Plans, and Practical Reason. Harvard University Press, Cambridge, MA (1987)
7. Cardoso, R.C., Ferrando, A.: A review of agent-based programming for multi-agent systems. Computers **10**(2), 16 (2021). https://doi.org/10.3390/computers10020016
8. Feng, J., Shi, Y., Qu, G., Low, S.H., Anandkumar, A., Wierman, A.: Stability constrained reinforcement learning for decentralized real-time voltage control. IEEE Transactions on Control of Network Systems **11**(3), 1370–1381 (2024). https://doi.org/10.1109/TCNS.2023.3338240
9. Hu, S., Gao, J., Zhong, D., Wu, R., Liu, L.: Real-time scheduling of pumps in water distribution systems based on exploration-enhanced deep reinforcement learning. Systems **11**(2) (2023). https://doi.org/10.3390/systems11020056
10. Ji, G., Yan, J., Du, J., Yan, W., Chen, J., Lu, Y., Rojas, J., Cheng, S.S.: Towards safe control of continuum manipulator using shielded multiagent reinforcement learning. IEEE Robotics and Automation Letters **6**(4), 7461–7468 (2021). https://doi.org/10.1109/LRA.2021.3097660
11. Jowitt, P.W., Germanopoulos, G.: Optimal pump scheduling in water-supply networks. Journal of Water Resources Planning and Management **118**(4), 406–422 (1992). https://doi.org/10.1061/(ASCE)0733-9496(1992)118:4(406)
12. Kahneman, D.: Thinking, fast and slow. Farrar, Straus and Giroux, New York (2011)
13. Kautz, H.: The third ai summer: Aaai robert s. engelmore memorial lecture. AI Magazine **43**(1), 105–125 (Mar 2022). https://doi.org/10.1002/aaai.12036
14. Majid, A., Cardenes, I., Zorn, C., Russell, T., Colquhoun, K., Bañares-Alcantara, R., Hall, J.W.: An analysis of electricity consumption patterns in the water and wastewater sectors in south east england, uk. Water **12**(1) (2020). https://doi.org/10.3390/w12010225
15. Marchi, A., Salomons, E., Ostfeld, A., Kapelan, Z., Simpson, A.R., Zecchin, A.C., Maier, H.R., Wu, Z.Y., Elsayed, S.M., Song, Y., Walski, T., Stokes, C., Wu,

W., Dandy, G.C., Alvisi, S., Creaco, E., Franchini, M., Saldarriaga, J., Páez, D., Hernández, D., Bohórquez, J., Bent, R., Coffrin, C., Judi, D., McPherson, T., van Hentenryck, P., Matos, J.P., Monteiro, A.J., Matias, N., Yoo, D.G., Lee, H.M., Kim, J.H., Iglesias-Rey, P.L., Martínez-Solano, F.J., Mora-Meliá, D., Ribelles-Aguilar, J.V., Guidolin, M., Fu, G., Reed, P., Wang, Q., Liu, H., McClymont, K., Johns, M., Keedwell, E., Kandiah, V., Jasper, M.N., Drake, K., Shafiee, E., Barandouzi, M.A., Berglund, A.D., Brill, D., Mahinthakumar, G., Ranjithan, R., Zechman, E.M., Morley, M.S., Tricarico, C., de Marinis, G., Tolson, B.A., Khedr, A., Asadzadeh, M.: Battle of the water networks ii. Journal of Water Resources Planning and Management **140**(7), 04014009 (2014). https://doi.org/10.1061/(ASCE)WR.1943-5452.0000378

16. Pei, S., Hoang, L., Fu, G., Butler, D.: Real-time pump scheduling in water distribution networks using deep reinforcement learning. Journal of Water Resources Planning and Management **151**(6), 04025012 (2025). https://doi.org/10.1061/JWRMD5.WRENG-6476
17. Shao, Y., Zhou, X., Yu, T., Zhang, T., Chu, S.: Pump scheduling optimization in water distribution system based on mixed integer linear programming. European Journal of Operational Research **313**(3), 1140–1151 (2024). https://doi.org/10.1016/j.ejor.2023.08.055
18. Su, T., Wu, T., Zhao, J., Scaglione, A., Xie, L.: A review of safe reinforcement learning methods for modern power systems. Proceedings of the IEEE **113**(3), 213–255 (Mar 2025). https://doi.org/10.1109/jproc.2025.3584656
19. Vieira, B.S., Mayerle, S.F., Campos, L.M., Coelho, L.C.: Optimizing drinking water distribution system operations. European Journal of Operational Research **280**(3), 1035–1050 (2020). https://doi.org/10.1016/j.ejor.2019.07.060
20. Wang, Y., Zhan, S.S., Jiao, R., Wang, Z., Jin, W., Yang, Z., Wang, Z., Huang, C., Zhu, Q.: Enforcing hard constraints with soft barriers: safe reinforcement learning in unknown stochastic environments. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023)
21. Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N.: Dueling network architectures for deep reinforcement learning. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. p. 1995–2003. ICML'16, JMLR.org (2016)
22. Yao, F., Zhao, W., Forshaw, M., Song, Y.: A holistic power optimization approach for microgrid control based on deep reinforcement learning. Neurocomputing **654**, 131375 (2025). https://doi.org/10.1016/j.neucom.2025.131375
23. Yu, G., Powell, R.S., Sterling, M.J.H.: Optimized pump scheduling in water distribution systems. J. Optim. Theory Appl. **83**(3), 463–488 (Dec 1994). https://doi.org/10.1007/BF02207638
24. Zhao, W., Beach, T.H., Rezgui, Y.: Optimization of potable water distribution and wastewater collection networks: A systematic review and future research directions. IEEE Transactions on Systems, Man, and Cybernetics: Systems **46**(5), 659–681 (2016). https://doi.org/10.1109/TSMC.2015.2461188
25. Zhao, W., Beach, T.H., Rezgui, Y.: A systematic mixed-integer differential evolution approach for water network operational optimization. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **474**(2217), 20170879 (09 2018). https://doi.org/10.1098/rspa.2017.0879