14th International Conference on Structural Safety and Reliability - ICOSSAR'25
1-6 June, 2025 - Los Angeles California, USA
R. Ghanem, E. Johnson, S. Masri and A. Olivier (Eds)

GRAPH-BASED MULTI-AGENT REINFORCEMENT LEARNING FOR RAILWAY INFRASTRUCTURE DECISION SUPPORT

GIACOMO ARCIERI[1], GREGORY DUTHÉ[1], CHRISTOPHE MULLER[1], DAVID HAENER[2], KONSTANTINOS G. PAPAKONSTANTINOU[3], DANIEL STRAUB[4] AND ELENI CHATZI[1]

[1] Department of Civil, Environmental and Geomatic Engineering, ETH Zürich
Zürich 8093, Switzerland
{garcieri, gduthe, echatzi}@ethz.ch, mullec@student.ethz.ch

[2] Swiss Federal Railways
Bern 3000, Switzerland
david.haener@sbb.ch

[3] Dept. of Civil and Environmental Engineering, The Pennsylvania State University
University Park, 16802, PA, USA
kpapakon@psu.edu

[4] Engineering Risk Analysis Group & Munich Data Science Institute, Technical University of Munich
Munich, 80333, Germany
straub@tum.de

Key words: Graph learning, Gaussian Processes on Graphs, Multi-Agent Reinforcement Learning, Graph Neural Networks, Maintenance Planning, Railway network

Abstract. Data-driven methods have advanced infrastructure management and policy planning, but often overlook system-level interactions critical to networked systems like railways. In railway maintenance planning, for example, the topology of the network plays a central role in system dynamics and optimal decision-making. In this work, we propose a hierarchical Bayesian model that leverages a latent Gaussian Process on Graph kernel to embed topological features and enable accurate inference of a deteriorating railway network environment from real-world data. To address the complexity of network-level optimization, we integrate multi-agent reinforcement learning with graph-based deep learning, enabling topology-aware intelligent agents to collaboratively optimize maintenance strategies across the network.

1 INTRODUCTION

Modern infrastructure asset management forms a complex sequential decision-making problem. The difficulty of deriving an optimal solution is compounded by the problem's scale, the long-term planning horizon, the inherent stochasticity of the system evolution, and the uncertainty of the observations that trigger corrective actions [1, 2].

Data-driven methods have been successfully implemented to improve decision-making in the management of infrastructure and planning of intelligent policies. These approaches typically involve i)

inference methods for building a simulation environment of the problem from available data [3, 4], and ii) solution algorithms, e.g., Reinforcement Learning (RL), for learning optimal policies based on the inferred environment [5, 6]. Previous work by Arcieri et al. [7] has demonstrated the successful inference of a simulation environment of a deteriorating railway track and applied RL techniques to plan optimal maintenance actions that outperform current human policies.

However, the inference of a reliable simulation environment for complex engineering systems still poses severe challenges that can hinder a successful implementation of these mathematical methods. These can be further aggravated in settings where *system-level interactions* play a critical role. For instance, railway infrastructure networks present system-level interactions that rely on network topology, such as the correlation in the deterioration of interconnected track sections and the economies of scale that arise from coordinated maintenance of different tracks. Accurately modeling and leveraging topological features can significantly improve the fidelity of the simulation environment and the effectiveness of the resulting solutions.

This work proposes a graph-based framework that directly addresses these challenges by integrating graph learning techniques into both the modeling of the environment and the decision support process for railway maintenance planning. We demonstrate this approach on the problem of optimal maintenance and renewal planning for railway infrastructure networks, based on observations of track geometry indicators. First, to enable a realistic simulation environment that reflects network effects, we employ Gaussian Processes on Graphs (GPG) [8] within a hierarchical Bayesian model. This allows us to encode topological dependencies and infer the spatio-temporal dynamics of track deterioration and the impact of maintenance actions directly from real-world data, provided by the Swiss Federal Railways (SBB). Second, leveraging this inferred graph-based environment, we formulate the maintenance and renewal planning problem as a Multi-Agent Reinforcement Learning [9] (MARL) task to achieve policies that jointly and cooperatively optimize the railway tracks. To this end, we develop a novel topology-aware MARL agent architecture based on Graph Neural Networks (GNNs). This architecture, designed for decentralized execution with centralized learning on graph representations, enables agents to learn policies that explicitly account for the state of neighboring components and the overall network structure, facilitating coordinated actions that leverage system-level interactions. We demonstrate the superior performance of the proposed topology-aware agent in optimizing long-term maintenance costs compared to standard MARL baselines and heuristic policies on the inferred simulation environment.

2 MAINTENANCE AND RENEWAL PLANNING OF THE RAILWAY NETWORK

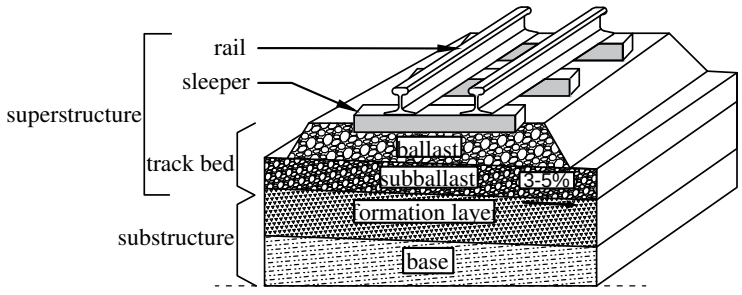


Figure 1: Structure of the railway track. Figure reproduced from [10].

Effective asset management of the railway network is essential for ensuring safe, reliable, and cost-efficient rail operations. A critical component of this management process is the maintenance and renewal planning of railway tracks, which directly impacts the network's operational performance and longevity. Railway tracks are subjected to continuous mechanical and environmental stresses, leading to gradual wear, degradation, and potential safety hazards if not properly maintained. Modern approaches increasingly incorporate predictive maintenance strategies, using data from track diagnostic vehicles [3] (e.g., track geometry measurements), sensor technologies [11], and historical maintenance records to forecast future defects and schedule timely interventions [12].

Among the key indicators monitored through track geometry measurements is the longitudinal level, which describes the vertical alignment of the rails along the track length. Irregularities in longitudinal level develop over time due to factors such as ballast settlement, subgrade instability, ballast and subgrade deterioration, and the mechanical loading from passing trains. These irregularities can be characterized by their wavelength: shorter wavelengths (D1, 3–25 m) typically result from degradation of superstructure components, while longer wavelengths (D2, 25–70 m) are associated with substructure deformations (Figure 1). Understanding the nature and scale of these deviations is essential for selecting appropriate maintenance interventions. Tamping, for example, is a corrective maintenance activity that restores track geometry by compacting and redistributing ballast beneath the sleepers. In cases where degradation exceeds maintenance limits or track components reach the end of their service life, track renewal actions become necessary. By integrating longitudinal level measurements into maintenance planning, infrastructure managers can effectively schedule tamping and renewal activities, ensuring continued track performance and extending asset life.

2.1 Railway data description

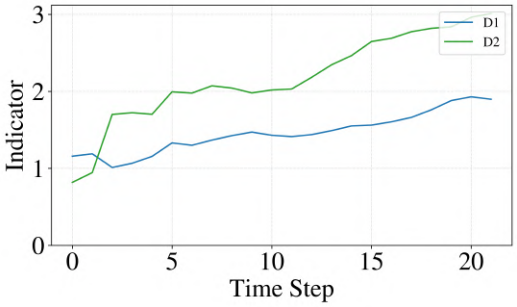
This study draws on a real-world railway dataset collected by the Swiss Federal Railways (SBB) in the Zurich metropolitan area over a 10-year period. The data comprises the following key components:

- **Track geometry measurements**, which serve as indicators of track condition, recorded at 2.5-meter intervals across the network every six months.
- **Geospatial location** data associated with each measurement point, enabling the reconstruction of the network topology and spatial analysis of the measurements.
- **Maintenance records** detailing the interventions performed on the railway tracks throughout the considered time horizon.

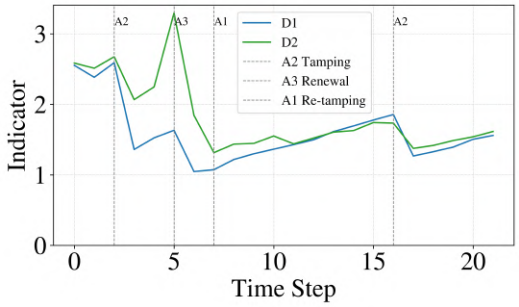
Amongst available track geometry data, this study specifically focuses on the longitudinal level as a primary indicator of track condition. In particular, the analysis employs the moving standard deviations calculated over a 100-meter window for the D1 and D2 filtered signals. These metrics rank among the operationally relevant indicators currently used by the Swiss Federal Railways to guide maintenance planning in response to track geometry irregularities.

Among the various maintenance interventions that can be performed on railway tracks, this study focuses on those typically instigated by track geometry irregularities — specifically, tamping and track renewal activities. According to maintenance regulations, a tamping intervention, here referred to as *re-tamping*, is required approximately 12 months following a track renewal to stabilize the geometry of the newly installed track structure. As a result, four possible maintenance actions are considered in this analysis: $\{$*do-nothing*; *tamping*; *re-tamping*; *track renewal*$\}$.

Figure 2 representatively shows the two indicators from two different network nodes (see Section 3.1), specifically without (left) and with (right) actions performed. The filters D1 and D2 appear highly, but not perfectly, correlated, as they provide indication over the deterioration of the track with a focus on superstructure and substructure, respectively. In absence of any intervention, the natural track deterioration proceeds, which is reflected by an increasing, albeit noisy, trend of the indicators. Repairing actions, when successfully implemented, reduce the value of the indicators (measured at the subsequent time step), with a generally more significant effect of the renewal, and a limited change induced by the re-tamping, as this later action is typically only performed in relatively good conditions. It is important to note that, on top of the inherent noise present in the indicators, human errors can also affect the logged data, both for the measurements and the actions.



(a) Without actions (pure deterioration process).



(b) With maintenance and renewal actions.

Figure 2: The indicators D1 and D2 (moving standard deviations over a 100-meter window) from two different nodes of the network over the decade collection horizon (one measurement every 6 months).

3 INFERENCE

3.1 Graph modeling of the railway network

A railway network can be naturally modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. In this work, the network is divided into track sections of 150 meters, with the data aggregated and averaged within each section. This approach has a twofold benefit: it reduces noise in the recorded measurements and mitigates the influence of potential logging errors and outliers. Each resulting track section is modeled as a node in the graph, with edges reflecting the direct physical connections between adjacent sections. For computational efficiency, the analysis focuses on a subset of the network located to the east of Zurich, comprising a total of 87 nodes (also referred to as *components*). The adjacency matrix corresponding to the final graph structure is presented in Figure 3. Each node is additionally augmented with node features representing the time series of the longitudinal level observations D1 and D2 $[\boldsymbol{z}_{t,i}^{(1)}, \boldsymbol{z}_{t,i}^{(2)}]$ and the implemented actions $[\boldsymbol{a}_{t,i}]$ over all nodes $i \in [1, N_g]$ and for all time steps $t \in [0, T]$.

3.2 Gaussian Processes on Graphs

Gaussian Processes (GPs) are a powerful class of non-parametric models widely used for regression, classification, and spatio-temporal modeling [13]. Standard GPs typically assume input domains residing in Euclidean spaces $\mathbb{R}^d$. However, many real-world datasets are more efficiently captured

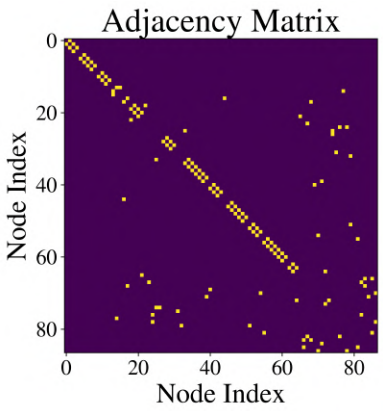


Figure 3: Adjacency matrix of the railway network graph. The yellow points indicate edge connections between nodes.

via graph structures, which are non-Euclidean. Applying traditional GP kernels directly to graph-structured data is often inappropriate as they fail to capture the underlying topology and connectivity.

To address this limitation, GPs on Graphs (GPG) have been developed to extend the GP framework to non-Euclidean domains represented by graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Borovitskiy et al. [8] define GP kernels based on the spectral properties of the graph Laplacian $\mathbf{\Delta}$. This allows for construction of stationary kernels analogous to well-known Euclidean kernels, such as the Matérn family, directly on the graph structure.

The core idea is to leverage the eigendecomposition of the graph Laplacian $\mathbf{\Delta} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{\Lambda}$ is the diagonal matrix that contains the non-negative eigenvalues and $\mathbf{U}$ contains the corresponding orthonormal eigenvectors. Borovitskiy et al. [8] demonstrate that one can reconstruct the Matérn GPG kernel $\mathbf{K}_{\text{graph}}$ by applying the stochastic partial differential equation characterization of the Matérn kernel $\mathbf{\Phi}$ to the Laplacian:

$$\mathbf{\Phi}(\mathbf{\Delta}) = \mathbf{U}\ \mathbf{\Phi}(\mathbf{\Lambda})\ \mathbf{U}^T \tag{1}$$

The resulting function is controlled by the parameters: $\sigma_f^2 > 0$, representing the overall variance; $\kappa > 0$ is a length-scale parameter; $\nu > 0$ controls the smoothness of the underlying function on the graph; and $\mathbf{K}_{\text{point}}$ represents a Matérn output initialization of the node features.

By learning the hyperparameters $(\sigma_f^2, \kappa, \nu)$ from data using standard GP inference techniques (e.g., MCMC inference), this GPG model can effectively capture correlations and dependencies between observations located at different nodes (railway track sections) within the network, respecting the connectivity defined by the railway system.

3.3 Bayesian inference

The real-world dataset is employed to infer a simulated environment of the maintenance and renewal planning problem of the railway network. This is achieved via MCMC sampling of a hierarchical Bayesian model that leverages a latent GP-on-Graph kernel to capture and reproduce correlations in the deterioration and repair effects of different portions of the rail infrastructure network.

The average change in the observations over time is modeled via a linear auto-regressive mean model conditioned on actions:

$$\mu_{t,i}^{(j)} = k_{a_{t,i},i}^{(j)} z_{(t-1),i}^{(j)} + \mu_{a_{t,i}}^{(j)}, \quad j \in \{1, 2\}, i \in \{1, \ldots, N_g\} \tag{2}$$

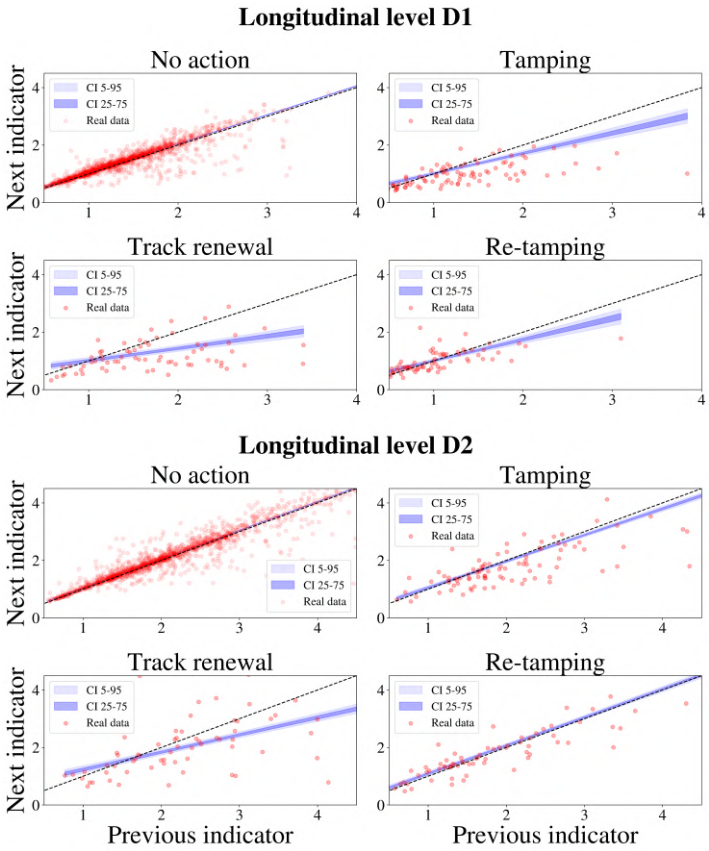


Figure 4: Average change of the indicators D1 and D2 learned by the linear auto-regressive model.

The covariance of the observations is modeled through the latent GPG kernel:

$$\mathbf{K}_{\text{point}} = \text{Matérn}(X_t; [l_{\text{action}}, l_{\text{indicator}}], 5/2)$$
$$\mathbf{K}_t = \mathbf{K}_{\text{graph}}(\nu, \sigma_f, \kappa, \mathbf{K}_{\text{point}}, (\mathbf{\Lambda}, \mathbf{U}))$$

which integrates the node features X_t to capture the covariance across nodes and the interdependence between the two indicators conditioned on the actions, namely $\mathbf{K}_t$ is a $2N_g \times 2N_g$ matrix.

The hyperparameters of the mean and covariance models, namely $(k^{(j)}_{a_{t,i}}, \mu^{(j)}_{a_{t,i}})$ and $(l_{\text{action}}, l_{\text{indicator}}, \nu, \sigma_f, \kappa)$, respectively, are assigned regularizing hyper-priors. Finally, the mean and covariance models, along with a learned degree-of-freedom parameter ν_{df}, parameterize a multivariate Student's t that computes the joint likelihood of the real observations:

$$[z^{(1)}_t, z^{(2)}_t] \sim \text{MvStudentT}([\mu^{(1)}_t, \mu^{(2)}_t], \mathbf{K}_t, \nu_{\text{df}}) \tag{3}$$

The choice of the Student's t instead of the common Gaussian likelihood is motivated by its greater robustness to outliers, expected in the real-world measurements [3]. The model is inferred via use of the Hamiltonian Monte Carlo (HMC) sampler NUTS [14].

3.4 Results

The HMC inference results for the mean model are reported in Figure 4, which demonstrates that the linear auto-regressive assumption fits well the real data points and is able to learn explainable

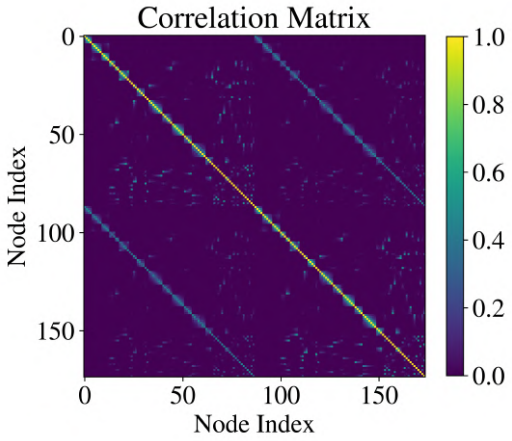


Figure 5: Correlation matrix inferred by the covariance model for the real data actions and graph.

relationships that fully respect the real-world effects. For example, when no maintenance action is executed, the indicators exhibit a slight increase after 6 months on average. The renewal action has a higher effect than the tamping action, and the re-tamping does not generally affect the indicators significantly. It is also possible to note from the figures the considerable inherent noise that affects the indicators.

Figure 5 displays the correlation matrix of the railway network data derived from the output of the covariance model. The matrix is $2N_g \times 2N_g$, as it shows the correlation among all N_g nodes and between the two different indicators (nodes in $[0, N_g - 1]$ refer to indicator D1, while $[N_g, 2N_g - 1]$ refer to the same nodes but for indicator D2), highlighting four possible correlation sub-squares in the matrix. The model is able to learn interesting correlations and capture system-level interactions among different nodes and indicators, and each sub-square closely reflects the graph structure displayed in Figure 3. It should be noted that Figure 5 shows a correlation matrix conditioned on the actions and graph structure of the real-world data, but the learned covariance model is able to generalize beyond this data and can output covariances/correlations for different action time series as well as different graphs. As such, our learned model can be employed as a simulated environment of the maintenance and renewal planning problem of railway networks modeled as a graph.

4 MULTI-AGENT REINFORCEMENT LEARNING SOLUTION

4.1 MARL fundamentals

This work considers MARL problems that can be modeled as *Cooperative Markov games* [9]. The game starts with the system initialized according to a probability distribution p_0. At time $t \in [0, T-1]$, each agent $i \in I = \{1, ..., n\}$ receives some observations $z_{t,i} \in Z$ and executes an action $a_{t,i}$ among a set of available actions $A_{t,i}$ according to the agent policy $\pi_i(a_{t,i}|h_{t,i})$, where $h_{t,i}$ is the input of the policy of agent i (which can differ from $z_{t,i}$, as explained in the following). The joint action $a_t = (a_{t,1}, ..., a_{t,n})$ is executed on the system, which updates its global state according to transition dynamics $\mathcal{T}$. As a result, each agent receives a reward $r_{t,i}$ as a function of the global state of the system and joint actions. The game proceeds with a new decision time step $t + 1$ until a terminal horizon T.

The objective is to learn a joint policy $\pi = (\pi_1, ..., \pi_n)$ that maximizes a common objective. In this work, we are interested in learning a joint policy π^* that maximizes the expected rewards $G_i(\pi)$

Table 1: Representative action costs per node depending on the number of concurrently maintained neighbors (N_c).

N_c	**No action**	**Tamping**	**Renewal**	**Re-tamping**
1	0	7,1	615,0	7,1
2	0	5,6	490,4	5,6
3	0	5,0	433,6	5,0
4	0	4,6	399,2	4,6
5	0	4,4	375,4	4,4

yielded to each agent, also called the *social welfare*:

$$W(\pi^*) = \max_{\pi} \sum_{i \in I} G_i(\pi) \tag{4}$$

Multi-agent Deep RL (MADRL) provides a potent framework to deal with complex real-world problems. Depending on the input observations $h_{t,i}$ that are available to the DRL agents to base their policy π_i, three different families of algorithms can be identified. In *Centralized Training with Centralized Execution* (CTCE), each agent has access to the global information, namely each agent receives the observations of all other agents at every time step, i.e., $h_{t,i} = (z_{t,1}, ..., z_{t,n})$. In *Decentralized Training with Decentralized Execution* (DTDE), each agent only has access to their own observations and, entirely independently from the other agents, learns and executes policy π_i, i.e., $h_{t,i} = z_{t,i}$. While in theory global information can be beneficial to learn better policies, centralized algorithms often do not scale well with the dimensionality of the variables involved and may be outperformed in practice by decentralized algorithms. Finally, by noticing that the inputs used can differ between training and execution times, a third intermediate family of algorithms called *Centralized Training with Decentralized Execution* (CTDE) has arisen. This is typical, for example, in Actor-Critic (AC) algorithms where the critic network is used only during training. It is thus possible to train a *Centralized Critic* with global input information to stabilize the training of the decentralized actor networks.

A further classification of MADRL algorithms depends on *Parameter Sharing* (PS), namely whether or not different agents are represented by the same neural network. For example, in CTCE-PS [15] each agent is represented with a different output of the same network. As such, the network output has dimensionality $|A| \times |I|$. Instead, centralized agents with no PS [16] or decentralized agents [17] can be represented with neural networks with only $|A|$-dimensional output.

4.2 MARL modeling

The problem of maintenance and renewal planning of a railway graph is modeled through the MARL framework presented in the previous section. The transition dynamics $\mathcal{T}$ and initial probability distribution p_0 of the system are simulated via the inferred model based on the GPG kernel described in Section 3.3. The observations $z_{t,i} \in Z \subset \mathbb{R}^2$ are the two-dimensional continuous valued indicators D1 and D2. These are available once every 6 months, hence $t = 6$ months, and we consider a horizon of $T = 50$ time steps. The possible actions $a_{t,i}$ are {*do-nothing*, *tamping*, *re-tamping*, *renewal*}, as described in Section 2.1. At each time step a different set of available actions $A_{t,i}$ can be available to the agents, e.g., 1 year after a renewal action, only re-tamping is allowed. Every agent is assigned to a node of the graph, hence $|I| = |\mathcal{V}| = N_g$, such that the terms *agents* and *nodes* can be used

interchangeably in the rest of the paper. As previously mentioned, our simulated environment is not restricted to the graph of the training data. As such, for this analysis we employ a railway graph composed of $N_g = 52$ components.

The reward structure in our MARL formulation for railway maintenance and renewal planning is defined by minimizing a cost function reflecting operational expenses. This cost function, developed in collaboration with the Swiss Federal Railways to ensure operational realism, incorporates costs associated with maintenance actions and the railway state condition. Importantly, it models system-level interactions, particularly the emergence of *economies of scale* that arise from interdependencies between agent actions. The total cost comprises two primary components:

- **Action Cost per Node:** This represents the cost of performing a specific maintenance action at a given node and includes (i) a *fixed component*, covering the mobilization of personnel, equipment, materials, and other necessary resources for the action, and (ii) a *variable component*, which decreases as the number of neighboring nodes undergoing simultaneous maintenance increases. This structure captures *economies of scale* arising from factors such as improved worker productivity over contiguous work areas or optimized shift utilization. While specific cost values are confidential, Table 1 provides a representative example illustrating this cost reduction for 1 to 5 concurrently maintained neighbors in general unit costs.
- **Condition Cost:** This penalty cost is incurred if track condition indicators (D1 and D2) exceed predefined critical thresholds, derived from official SBB guidelines. Such an exceedance represents an "emergency" state, mandating a renewal action in the subsequent time step and reflecting the cost implications of severe track degradation.

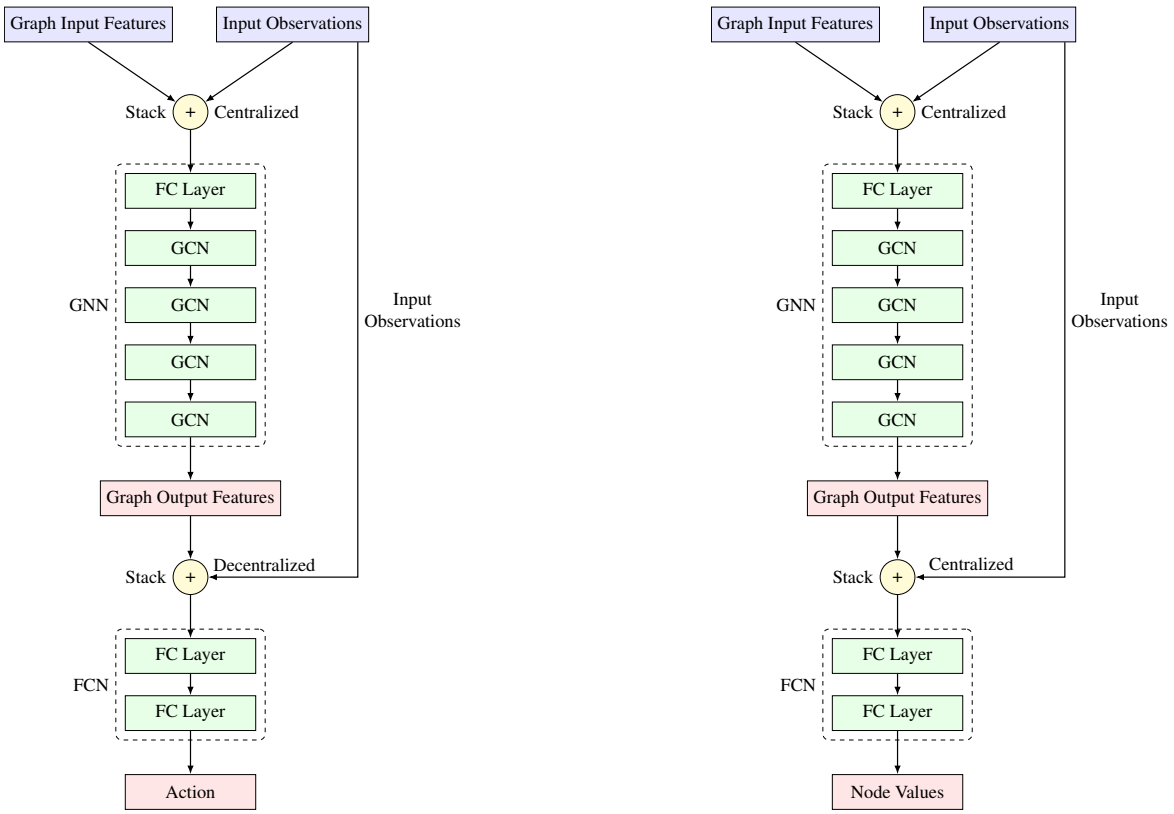
4.3 Topology-aware agents

A key contribution of this work is the development of a topology-aware MARL agent architecture designed to explicitly model system-level interactions inherent in the railway network graph. This architecture, depicted in Figure 6, employs an Actor-Critic design where both networks leverage Graph Neural Networks [18] (GNNs) and Fully-Connected Networks (FCNs) to process graph structure and node-specific observations effectively.

Both the Actor and Critic networks share an initial processing pipeline. Centralized graph input features (representing graph structure) are concatenated with the centralized raw input observations (containing condition indicators D1 and D2 for all nodes). This combined input is fed into a GNN module consisting of an initial Fully-Connected (FC) layer followed by four graph convolutional (GCN) layers. The purpose of this GNN is to generate enriched "Graph Output Features" that encode information about both node states and their topological context within the railway network.

Following the GNN module, the Actor and Critic architectures diverge:

- **Actor Network:** The learned "Graph Output Features" are combined with the raw "Input Observations" in a **decentralized** manner by concatenating the relevant graph features to each node individual observation vector. This node-specific input is then processed by a subsequent FCN module, which outputs the predicted maintenance action for each node *independently*. This decentralized output structure ensures the Actor output dimensionality ($|A|$=4) does not increase with the total number of nodes, allowing the agent to learn centralized graph representations while making localized decisions.



(a) Actor.

(b) Centralized Critic.

Figure 6: Topology-aware agent.

- **(Centralized) Critic Network:** Used only during training, the Critic also takes the "Graph Output Features" from the GNN. However, these are concatenated with the full, **centralized** "Input Observations". This combined centralized state representation is processed by its FCN module to produce a vector of *centralized state values* – one value estimate per node, rather than a single scalar value for the entire system. This per-node value decomposition is designed to mitigate the multi-agent credit assignment problem, potentially improving sample efficiency during learning.

A significant advantage of this GNN-based architecture is its inherent scalability and generalizability. By learning patterns based on local connectivity and graph topology rather than fixed node identities, the agent can potentially be trained on one railway graph configuration and deployed on different, possibly larger, graphs without requiring retraining.

4.4 Results

Figure 7 presents the learning curves comparing the proposed topology-aware (GNN) agent against a CTCE-PS baseline, serving as a standard architecture without explicit graph processing, and heuristic policies. Both GNN and CTCE-PS agents use the (MA)PPO algorithm and centralized critics that learn *per-node state values*. Performance is measured by the cumulative reward over nodes (Equation 4) and horizon T, averaged over 10 test episodes per update and across 5 seeds (smoothed with a rolling window of 10 steps for visualization).

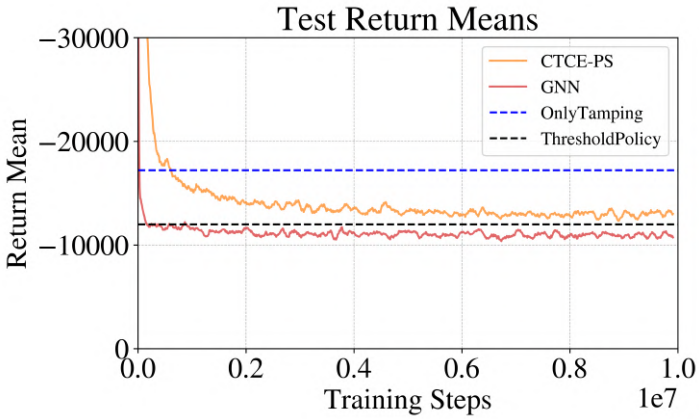


Figure 7: Learning curves of the topology-aware agent (GNN) against a CTCE-PS agent and heuristic policies.

In addition to the CTCE-PS baseline, the MARL agents are benchmarked against two heuristic policies: (i) *OnlyTamping*, a naive strategy applying the tamping action at each step, and (ii) *ThresholdPolicy*, an optimized rule-based approach triggering maintenance actions based on thresholds for the D1 and D2 condition indicators, with thresholds optimized via grid search.

The results presented in Figure 7 show that the proposed topology-aware agent achieves the highest converged test return, demonstrating superior performance compared to the CTCE-PS agent. The optimized ThresholdPolicy establishes a strong baseline, significantly outperforming the OnlyTamping heuristic and even surpassing the converged performance of the CTCE-PS agent. The effectiveness of the ThresholdPolicy likely arises from its indirect exploitation of system-level interactions. Since the deterioration of neighboring nodes is correlated, applying threshold-based maintenance can trigger repairs on adjacent nodes, thus benefiting from the implemented economies of scale.

Despite the strong performance of the optimized heuristic, the GNN agent's ability to explicitly model and leverage the network topology leads to the best overall performance. This highlights the advantages of incorporating graph structure directly into the agent architecture for learning effective, long-term maintenance strategies that minimize system-wide costs.

5 CONCLUSIONS

This work proposes a graph-based multi-agent reinforcement learning framework for optimizing maintenance and renewal planning in railway networks. Leveraging real-world data and Gaussian Processes on Graphs, we first infer a simulation environment capable of capturing complex spatio-temporal correlations and system-level interactions that are inherent in the network deterioration and repair dynamics. We then proceed to develop a novel topology-aware MARL agent architecture based on GNNs. Numerical results demonstrate the superior performance of explicitly incorporating network topology into the agent decision-making process for minimizing long-term costs.

Future research will focus on enhancing the agent representational power by exploring more advanced architectures, such as Graph Transformers [19], potentially capturing longer-range dependencies more effectively. Furthermore, we plan to investigate the scalability and transferability of the topology-aware approach by training agents on smaller graph sections and evaluating their performance when deployed on larger, unseen network structures, thereby assessing the potential for scalable MARL solutions in large-scale infrastructure management.

References

[1] Straub, D. "Value of information analysis with structural reliability methods". In: *Structural Safety* (2014) **49**: 75–85.

[2] Papakonstantinou, K. G. and Shinozuka, M. "Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part I: Theory". In: *Reliability Engineering & System Safety* (2014) **130**: 202–213.

[3] Arcieri, G. et al. "Bridging POMDPs and Bayesian decision making for robust maintenance planning under model uncertainty: An application to railway systems". In: *Reliability Engineering & System Safety* (2023) **239**: 109496.

[4] Arcieri, G. et al. "Deep Belief Markov Models for POMDP Inference". In: *arXiv preprint arXiv:2503.13438* (2025). DOI: 10.48550/arXiv.2503.13438.

[5] Arcieri, G. et al. "Soft Actor-Critic for railway optimal maintenance planning under partial observability". In: *14th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP14)*. Trinity College Dublin. 2023.

[6] Arcieri, G. et al. "A comparison of value-based and policy-based reinforcement learning for monitoring-informed railway maintenance planning". In: *Structural Health Monitoring* (2023).

[7] Arcieri, G. et al. "POMDP inference and robust solution via deep reinforcement learning: An application to railway optimal maintenance". In: *Machine Learning* (2024) **113**: 7967–7995.

[8] Borovitskiy, V. et al. "Matérn Gaussian processes on graphs". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, 2593–2601.

[9] Albrecht, S. V. et al. *Multi-agent reinforcement learning: Foundations and modern approaches*. MIT Press, 2024.

[10] Profillidis, V. *Railway management and engineering*. Routledge, 2016.

[11] Arcieri, G. et al. "Ground penetrating radar for moisture assessment in railway tracks: An experimental investigation". In: *Proceedings of the 11th European Workshop on Structural Health Monitoring (EWSHM)*. Vol. 2024. NDT.net. 2024.

[12] Hoelzl, C. et al. "Fusing expert knowledge with monitoring data for condition assessment of railway welds". In: *Sensors* (2023) **23**: 2672.

[13] Schulz, E. et al. "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions". In: *Journal of Mathematical Psychology* (2018) **85**: 1–16.

[14] Hoffman, M. D. and Gelman, A. "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *J. Mach. Learn. Res.* (2014) **15**: 1593–1623.

[15] Andriotis, C. P. and Papakonstantinou, K. G. "Managing engineering systems with large state and action spaces through deep reinforcement learning". In: *Reliability Engineering & System Safety* (2019) **191**: 106483.

[16] Andriotis, C. P. and Papakonstantinou, K. G. "Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints". In: *Reliability Engineering & System Safety* (2021) **212**: 107551.

[17] Saifullah, M. et al. "Multi-agent deep reinforcement learning with centralized training and decentralized execution for transportation infrastructure management". In: *arXiv preprint arXiv:2401.12455* (2024).

[18] Corso, G. et al. "Graph neural networks". In: *Nature Reviews Methods Primers* (2024) **4**: 17.

[19] Duthé, G. et al. "Graph Transformers for inverse physics: reconstructing flows around arbitrary 2D airfoils". In: *arXiv preprint arXiv:2501.17081* (2025).