PROTOCOL PRIOR GUIDED REINFORCEMENT LEARNING FOR RESILIENT TIME-SENSITIVE NETWORKING ROUTING

Chao Wang[1], Chongwu Dong[2], Mu Yu[3], Wushao Wen[1,*]
[1]*School of Computer Science and Engineering, Sun Yat-sen University*, Guangzhou, China
[2]*Cyberspace Institute of Advanced Technology, Guangzhou University*, Guangzhou, China
[3]*Department of Control Science and Engineering, Tongji University*, Shanghai, China
[*]Corresponding author
Email: wangch556@mail2.sysu.edu.cn, dongchongwu@gzhu.edu.cn, 2332034@tongji.edu.cn, wenwsh@mail.sysu.edu.cn

***Abstract*—Time-Sensitive Networking (TSN) provides deterministic communication, yet the same determinism introduces failure modes that can mislead learning-based routing. When a persistent Gate Control List (GCL) misconfiguration occurs at a port, multiple flows that traverse the faulty port exhibit synchronized delay spikes, so a model-free reinforcement learning (RL) agent can confuse this fault with stochastic congestion if it relies only on end-to-end latency, repeatedly switching routes without resolving the root cause and destabilizing scheduled service. To address this correlation trap, we propose Protocol Prior-Guided Reinforcement Learning (PPRL), where we translate protocol defined determinism into a continuous prior signal. PPRL introduces a Spectral Protocol Prior Module (PPM) that quantifies cross flow synchronicity via the spectral entropy of a delay covariance matrix and outputs a Spectral Synchronicity Index. This spectral evidence constrains Proximal Policy Optimization (PPO) through an adaptive KL regularized objective that biases the policy toward a stability prior under strong fault evidence while naturally reverting to standard PPO updates when the evidence is diffuse. In ns-3 TSN simulations, PPRL reduces route switching by 84.1% under GCL faults and maintains scheduling efficiency comparable to strong baselines.**

***Index Terms*—Time-Sensitive Networking, Reinforcement Learning, Protocol Priors, Network Routing, Fault Tolerant Routing**

I. INTRODUCTION

TSN provides deterministic low latency communication guarantees that underpin mission critical systems such as industrial control and autonomous platforms [1]. Routing in TSN interacts tightly with time-aware scheduling, which turns online adaptation into a high dimensional control problem [2], [3]. Classical formulations such as Integer Linear Programming (ILP) can compute optimal static schedules, yet their computational cost and limited flexibility make them unsuitable when traffic and topology evolve. This gap motivates learning based strategies, particularly RL for TSN routing [4]–[6].

A practical obstacle is that an RL agent may learn spurious correlations from observational signals. High end to end latency can be caused by path level congestion, yet it can also arise from a localized GCL misconfiguration. If these situations are not distinguished, the agent tends to treat latency as a universal rerouting trigger and repeatedly switches routes. This decision churning is not only ineffective under persistent GCL faults, but it can also amplify instability by repeatedly perturbing co-scheduled flows.

The determinism of TSN also provides an opportunity. Standards such as IEEE 802.1Qbv prescribe mechanisms under which a persistent gate schedule error induces aligned delay spikes across multiple flows that traverse the same faulty port [7]. Because the queuing bursts are synchronized by the deterministic schedule, the stochastic delay variations across flows collapse toward a single dominant spectral dimension, which yields a low rank structure that is distinct from the diversified variations induced by stochastic congestion. We treat this structural signature as a protocol prior and use it to guide learning rather than attempting to infer a full root cause model from observational data [8], [9].

Based on this perspective, we propose PPRL. PPRL introduces the PPM to compute a Spectral Synchronicity Index ϕ_t from the eigen spectrum of delay covariances and uses this continuous evidence to regularize policy updates through an adaptive KL penalty that discourages futile path switching under strong fault evidence while retaining standard PPO updates when the evidence is diffuse. This work clarifies a failure mode of symptom driven routing policies in TSN and provides a lightweight protocol prior mechanism that links diagnosis to control. Our contributions consist of characterizing the correlation trap induced by deterministic GCL misconfigurations, designing Spectral PPRL that disentangles synchronized faults from stochastic congestion, and validating the resulting robustness in ns-3 TSN simulations across representative topologies where route switching is reduced by 84.1% under persistent GCL faults while scheduling efficiency remains competitive.

II. RELATED WORK

A. TSN Scheduling, Routing, and GCL Configuration

TSN achieves deterministic latency by combining time aware shaping such as IEEE 802.1Qbv with routing and schedule synthesis for GCLs. A large body of work formulates joint routing and scheduling as ILP problems to guarantee deadlines [10]–[12]. However, exact optimization can be expensive when traffic or topology evolves, motivating scalable approaches for large deployments. Since our fault model is a GCL misconfiguration, recent work on scalable GCL scheduling and synthesis provides context on how schedules are generated and validated in practice. Beyond ILP formulations, recent work studies practical scheduling and shaping realizations, including ultra-reliable TAS variants, multitenancy-aware scheduling, and density-adaptive gate control configuration [13]–[15]. Simulation and testbed studies further characterize cyclic shapers and timing perturbations under realistic jitter [16]–[18]. Multi-topology routing techniques have also been explored to optimize traffic under TSN constraints [19], and learning-based dynamic schedule calculation has recently been investigated with graph-based actor-critic methods [20]. Importantly, persistent timing faults or attacks can create periodic delay bursts that affect traversing flows in a structured pattern, which motivates protocol-grounded fault signatures [21].

B. Deep Reinforcement Learning for Routing and Traffic Engineering

Reinforcement learning has long been studied for adaptive routing [22]. Machine learning for networking has been surveyed extensively [23]. Recent deep reinforcement learning has been applied to routing and traffic engineering (TE) in SDN and WAN settings [24], [25]. To improve robustness under anomalies, FlexDATE incorporates disturbance aware objectives in TE instead of reacting only to end to end delay [26]. For latency critical services, Vitale et al. propose a multi agent deep reinforcement learning framework for joint routing and scheduling, providing a contrasting design point to centralized controllers [27]. In TSN, recent work also explores richer neural architectures for joint routing and scheduling; for example, Li et al. combine graph attention with deep reinforcement learning under vehicular TSN constraints [28]. Nevertheless, many learning-based controllers still rely on observational performance signals and may struggle to disambiguate congestion from protocol-level faults when both manifest as delay spikes.

C. Causal and Prior-Guided Reinforcement Learning for Robust Control

Causal and prior-guided designs inject structured knowledge or auxiliary reasoning to improve robustness and reduce reliance on spurious correlations [29]. Causal reinforcement learning offers a principled way to handle confounding; for example, Q-Cogni integrates causal structure discovery with RL to guide learning under confounded observations [30]. While powerful, causal approaches often require learning or specifying causal models and can be heavy for time critical networking control loops. Unlike prior works that rely on heavy causal graph discovery, our work bridges Graph Signal Processing (GSP) with RL by using the spectral entropy of traffic flows as a lightweight and theoretically grounded protocol prior that regularizes PPO updates through an adaptive KL penalty.

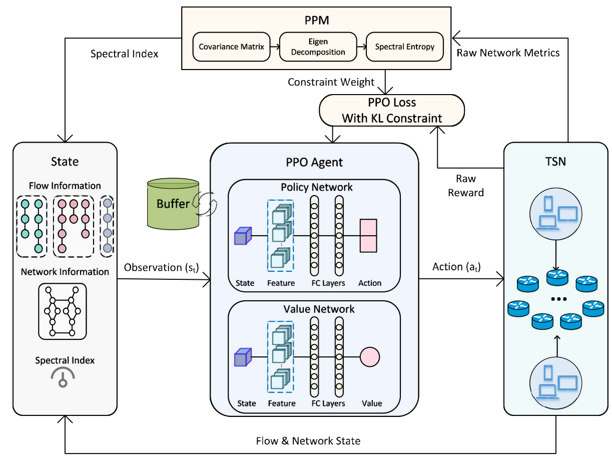


Fig. 1. PPRL framework architecture. The PPM processes network telemetry and produces a Spectral Synchronicity Index ϕ_t that augments the policy input and defines a constraint weight through $\psi[\phi_t]$, which regularizes PPO updates via an adaptive KL penalty.

III. Method

This section formalizes PPRL under a TSN control loop. We describe a robust Markov Decision Process (MDP) formulation in which an unobservable fault mode drives the transition dynamics, then we derive the PPM that extracts continuous evidence from delay covariances, and we finally integrate this evidence into PPO through an adaptive KL regularized objective that constrains policy updates when the evidence indicates a persistent scheduling fault.

A. System Model

We model the TSN as a directed graph $G = \langle V, E \rangle$, where V comprises TSN capable switches and end-hosts, and E represents the full-duplex physical links connecting them, operating under the strict timing constraints of IEEE 802.1Qbv and IEEE 802.1AS [31]. Time-Sensitive flows $f \in F$ are defined by the tuple $\langle src, dst, p, s, d_{\text{deadline}} \rangle$, where src and dst denote the source and destination nodes, p denotes the transmission period, s denotes the maximum packet size, and d_{deadline} denotes the strict end to end deadline.

1) GCL and Misconfiguration Model: In IEEE 802.1Qbv, each egress port is governed by a periodic GCL that opens and closes priority queues according to a predefined time-aware schedule. We abstract the gate state of a switch port pair $\langle S, E \rangle$ as a binary function $g_{SE}[t] \in \{0, 1\}$, where $g_{SE}[t] = 1$ indicates that the time-sensitive queue is allowed to transmit and $g_{SE}[t] = 0$ indicates that transmissions are blocked. When $g_{SE}[t] = 0$, packets accumulate in the queue; once the gate reopens, the backlog is drained at line rate, creating a burst of queuing delay that affects all flows traversing the port.

To model a persistent GCL misconfiguration, we introduce an unintended closed interval of duration Δ_f that recurs every scheduling period T_{gcl}. Specifically, for a faulty port $\langle S^{\star}, E^{\star} \rangle$ we set

$$g_{S^{\star}E^{\star}}[t] = 0 \quad \text{if} \ \ [t \bmod T_{\text{gcl}}] \in [t_f, \, t_f + \Delta_f] \,, \tag{1}$$

and $g_{S^{\star}E^{\star}}[t] = 1$ otherwise. Because this closure pattern is deterministic and localized, all flows passing through the faulty port experience delay spikes aligned to the same gate schedule, which yields the cross-flow synchronicity signature leveraged by the PPM.

2) Centralized Control Loop and Decision Interval: We consider a centralized controller that periodically collects network telemetry such as per-port queue occupancy and per-flow latency measurements and applies routing updates at a fixed control interval Δt. This assumption is consistent with TSN management architectures where a central entity configures streams and schedules, and it allows us to quantify both routing performance and end to end control loop overhead under strict timing budgets.

We formulate the dynamic routing problem as a robust MDP, where a centralized network controller acts as the agent. The transition dynamics follow $P\left[s_{t+1} \mid s_t, a_t, z_t\right]$, where z_t denotes an unobservable network mode that distinguishes persistent gate schedule faults from stochastic congestion. The state s_t at time t must capture sufficient information for a decision. We use a base state $s_t = \{s_{\text{net}}, s_{\text{flow}}\}$, where s_{net} includes link utilization and queue lengths and s_{flow} summarizes flow requirements. The PPM computes a Spectral Synchronicity Index ϕ_t, which is concatenated with the base state as a scalar feature and also used to modulate the strength of policy regularization. The action a_t selects the next hop neighbor. The agent policy π_θ is parameterized by θ and optimized to maximize the expected cumulative discounted return, which we denote as $J\left[\theta\right]$.

$$J\left[\theta\right] = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R_{\text{perf}}\left[s_t, a_t\right]\right] \tag{2}$$

where γ is the discount factor. The performance reward is $R_{\text{perf}} = I_{\text{success}} - L_{e2e}/d_{\text{deadline}}$, where I_{success} is a binary indicator for meeting the deadline and L_{e2e} is the end to end latency. However, an agent guided solely by R_{perf} faces the correlation trap [32]. This reward signal is ambiguous because the high latency induced by a localized GCL fault is observationally similar to that of path wide congestion, so the agent can learn an ineffective reactive policy that repeatedly reroutes under persistent faults, leading to decision churning.

B. Spectral Protocol Prior Extraction

PPRL addresses the correlation trap by coupling PPO [33] with the PPM that extracts continuous evidence from delay covariances. Figure 1 illustrates the closed loop, where the module processes network telemetry and outputs a Spectral Synchronicity Index denoted as ϕ_t that summarizes the degree of cross flow delay synchronization induced by deterministic gate scheduling.

Algorithm 1 Spectral Protocol Prior Regularized PPO

Initialize PPO policy parameters θ and value parameters w
Initialize spectral hyperparameters W, β, α, and ϕ_0
for each training episode **do**
  for each decision step t **do**
    Observe network telemetry and construct base state $s_t = \{s_{\text{net}}, s_{\text{flow}}\}$
    Run the spectral module and compute ϕ_t using Eq. (5)
    Concatenate ϕ_t with s_t to form the policy input
    Sample routing action $a_t \sim \pi_\theta\left[\cdot \mid s_t\right]$ and apply it to the TSN
    Compute R_{perf} from deadline satisfaction and end to end latency
    Store transition $\left[s_t, a_t, R_{\text{perf}}, s_{t+1}, \phi_t\right]$
  end for
  Update PPO using Eq. (7)
end for

1) Delay Covariance Matrix: Let $\mathbf{d_t} \in \mathbb{R}^N$ denote the vector of end to end delays measured for the N flows that traverse a candidate port at time t. Over a sliding window of length W, we form an observation matrix $\mathbf{D}_t = [\mathbf{d}_{t-W+1}, \ldots, \mathbf{d}_t] \in \mathbb{R}^{N \times W}$. We compute the empirical covariance matrix as

$$\mathbf{\Sigma}_t = \frac{1}{W-1}\left[\mathbf{D}_t - \boldsymbol{\mu}_t \mathbf{1}^\top\right]\left[\mathbf{D}_t - \boldsymbol{\mu}_t \mathbf{1}^\top\right]^\top, \tag{3}$$

where $\boldsymbol{\mu}_t = \frac{1}{W}\mathbf{D}_t\mathbf{1}$ is the mean delay vector and $\mathbf{1}$ is an all ones vector.

2) Spectral Synchronicity Index: A persistent GCL misconfiguration forces traversing flows to experience synchronized queuing bursts, which concentrates covariance energy in a dominant eigen direction, whereas stochastic congestion exhibits diverse variations that spread energy across eigen directions. We compute eigenvalues $\lambda_1 \geq \cdots \geq \lambda_N \geq 0$ of $\mathbf{\Sigma}_t$, normalize them as $\tilde{\lambda}_i = \lambda_i / \sum_{j=1}^{N} \lambda_j$, and compute the spectral entropy

$$H_t = -\sum_{i=1}^{N} \tilde{\lambda}_i \ln \tilde{\lambda}_i. \tag{4}$$

We then define the Spectral Synchronicity Index as

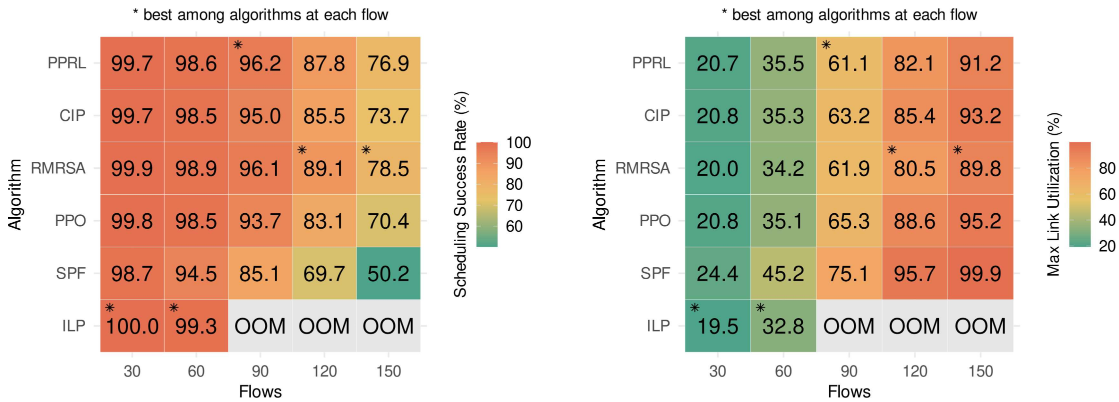
$$\phi_t = 1 - \frac{H_t}{\ln(\max(N, 2))}. \tag{5}$$

Because ϕ_t measures spectral concentration rather than absolute delay magnitude, it provides a direct signature of deterministic synchronization.

C. Prior Regularized Policy Optimization

We incorporate spectral evidence as an adaptive regularization to guide policy updates. We introduce a consistency reference π_{stable} that serves as a soft anchor for decision inertia, favoring the previous action:

$$\pi_{\text{stable}}\left[a \mid s_t\right] = \mathbb{I}\left[a = a_{t-1}\right]. \tag{6}$$



(a) Scheduling Success Rate

(b) Maximum Link Utilization

Fig. 2. Scalability and performance comparison where panel a reports scheduling success rate and panel b reports maximum link utilization versus an increasing number of network flows.

We then optimize a KL regularized PPO objective

$$\mathcal{L}_{\text{PPRL}}(\theta) = \hat{\mathbb{E}}_t\Big[\mathcal{L}_{\text{PPO}}(\theta) - \beta\psi(\phi_t)D_{\text{KL}}\big(\pi_{\text{stable}} \parallel \pi_\theta\big)\Big] \tag{7}$$

Since π_{stable} in Eq. (6) is deterministic, the KL term reduces to a negative log probability of repeating the previous action. To avoid numerical issues when $\pi_\theta\left[a_{t-1} \mid s_t\right]$ becomes extremely small, we compute it with probability clipping:

$$D_{\text{KL}}\big(\pi_{\text{stable}} \parallel \pi_\theta\big) = -\log\left(\max\left(\pi_\theta\left[a_{t-1} \mid s_t\right], \epsilon\right)\right). \tag{8}$$

We set $\epsilon = 10^{-8}$ in all experiments. In Eq. (7), β is a regularization coefficient and $\psi\left[\phi_t\right]$ maps spectral evidence to a regularization strength. We use

$$\psi\left[\phi_t\right] = \frac{1}{1+\exp\left[-\alpha\left[\phi_t - \phi_0\right]\right]}, \tag{9}$$

where α controls the steepness of the transition and ϕ_0 denotes the activation threshold of the spectral evidence, so we set ϕ_0 to 0.5 to establish a balanced boundary that keeps $\psi\left[\phi_t\right]$ close to zero when ϕ_t stays well below ϕ_0 and activates smoothly as synchronized fault evidence emerges. When ϕ_t is high, the penalty term increases and the update is biased toward stability, while when ϕ_t is low the penalty term vanishes and the objective reduces to standard PPO, which preserves adaptation under congestion and uncertainty.

Intuitively, a localized GCL fault induces a concentrated bottleneck with synchronized cross flow effects at a single port, which yields a low entropy spectrum and a high ϕ_t, whereas stochastic congestion yields diversified variations and a low ϕ_t. Algorithm 1 summarizes the overall training loop of PPRL, including how spectral evidence modulates the KL regularization term during policy optimization.

The KL penalty in Eq. (7) acts as a behavior constraint that is activated by spectral evidence. When the delay covariance spectrum is concentrated and ϕ_t becomes high, the update is pulled toward the stability prior, which suppresses futile rerouting under persistent gate faults. Conversely, when the spectrum is diffuse and ϕ_t becomes low, the constraint weakens through $\psi\left[\phi_t\right]$ and the optimization reduces to the standard PPO objective, which preserves adaptation under congestion and uncertainty.

D. Design Considerations

Robustness is enabled by evidence modulated regularization that preserves full decision flexibility. When the spectral evidence is weak or misleading, ϕ_t becomes low and $\psi\left[\phi_t\right]$ approaches zero, so the KL penalty in Eq. (7) becomes negligible and the objective reduces to standard PPO driven by R_{perf}. This fallback avoids over commitment under uncertain evidence while preserving exploration and adaptation.

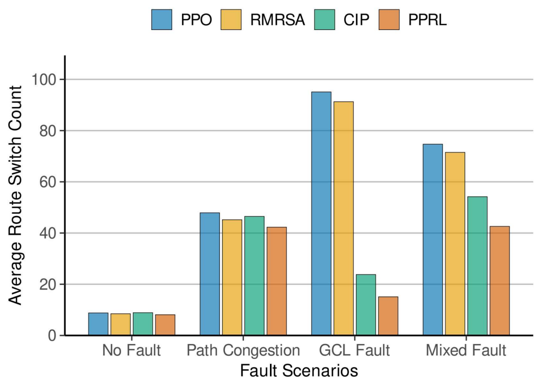
IV. Experiments

We evaluate PPRL through a scalability sweep and fault oriented stress tests.

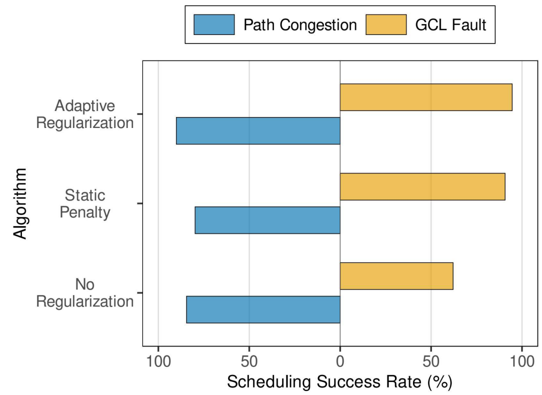
A. Experimental Setup

All experiments are conducted within the ns-3 simulation environment [34], [35]. To assess performance across diverse network structures, we consider three representative TSN topologies that include a deterministic line, a redundant ring, and an unstructured Erdos Renyi random graph. Each topology is instantiated at a moderate scale with 10–50 nodes and 1 Gbps links to approximate industrial TSN deployments and enable fair comparison. The environment implements IEEE 802.1Qbv and IEEE 802.1AS, and workloads comprise 30–150 concurrent TSN flows.

We benchmark PPRL against a diverse set of representative baselines. These include an ILP formulation that serves as an optimistic bound for static schedules, the static Shortest Path First (SPF) algorithm, a standard PPO agent, the specialized routing algorithm RMRSA [36], and the representative causal RL baseline Causal Information Prioritization (CIP) [37]. Key evaluation metrics are scheduling success rate, maximum link utilization, and average route switch count, where route switch



(a) Decision Churning Analysis.



(b) Ablation Study of Regularization.

Fig. 3. PPRL performance under fault conditions where panel a reports average route switch count and panel b reports scheduling success rate in the regularization ablation study.

count quantifies decision churning and thus measures resilience against the correlation trap. As network load increases, ILP can run out of memory in our setting, so we report its results only where feasible.

Unless otherwise stated, we report averaged results for learning-based methods over multiple random seeds, while deterministic baselines such as ILP and SPF are evaluated once per scenario. We keep the policy network and observation window consistent across learning-based methods, and we report the additional controller-side overhead explicitly in Table I.

By default, we set the spectral window size W to 10 and use $\beta = 1.0$, $\alpha = 10$, and $\phi_0 = 0.5$. We keep these values fixed across all scenarios; β and α are chosen as stable defaults supported by the sensitivity sweep in Fig. 4, and ϕ_0 is set to the sigmoid midpoint.

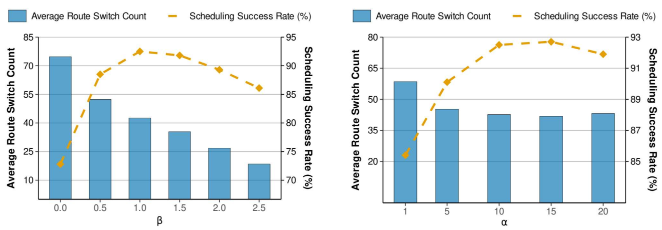
Our evaluation includes a fault free scalability test with an increasing network load. We also conduct a fault resilience study under a fixed load of 90 flows, where we consider four scenarios that include No Fault, Path Congestion, GCL Fault, and Mixed Fault. The GCL fault is engineered to create periodic delays that rerouting cannot eliminate, which directly probes the agent ability to overcome the correlation trap. For the main experiments, we use $\beta = 1.0$ and $\alpha = 10$ as stable defaults suggested by the sensitivity sweep in Fig. 4.

Across the evaluated load regimes, PPRL demonstrates robust scalability with performance comparable to the specialized RMRSA baseline, as shown by the scheduling success rate and the maximum link utilization in Fig. 2a and Fig. 2b. Under high load, it yields a 9.2% relative gain in scheduling success rate over standard PPO, indicating that evidence modulated regularization stabilizes decisions under transient latency fluctuations while preserving the ability to adapt when spectral evidence is diffuse.

As shown in Fig. 3a, PPRL effectively mitigates the correlation trap during GCL faults. PPO and RMRSA, which largely react to end to end latency, exhibit severe route switching under persistent GCL misconfiguration, reflecting a failure of purely associational learning where high delay is spuriously treated as a routing quality signal. *CIP* reduces churning compared with PPO and RMRSA, but it still switches more than PPRL in the GCL fault and mixed fault scenarios. In contrast, PPRL leverages the synchronized structure of delay spikes through ϕ_t and therefore maintains routing stability under persistent scheduling faults. Relative to PPO, PPRL reduces route switching by 84.1% in the GCL fault scenario. In the mixed fault scenario, PPRL still reduces route switching by 43.0% relative to PPO. Importantly, PPRL does not suppress adaptation indiscriminately. Under path congestion, its switch count remains comparable to PPO, RMRSA, and *CIP*, which suggests that regularization is selective rather than overly conservative.

An ablation study in Fig. 3b confirms the benefit of evidence modulated regularization by comparing Adaptive Regularization against Static Penalty and No Regularization. When regularization is removed in No Regularization by setting β to 0, the agent exhibits frequent route switching under GCL faults and the scheduling success rate drops accordingly. When a static penalty is applied, stability improves but adaptation under congestion can be overly constrained, whereas Adaptive Regularization retains selectivity by activating only when spectral evidence strengthens. We further examine the coefficient sweep in Fig. 4a, where increasing β suppresses futile switching and the scheduling success rate peaks around $\beta = 1.0$ before decreasing as the penalty becomes overly restrictive, so we set $\beta = 1.0$ in the main experiments as a stable trade off between decision stability and adaptability.

The sensitivity of the regularizer is governed by the sigmoid slope α in Eq. (9). Small α yields a gradual mapping from ϕ_t to $\psi[\phi_t]$, which can under react to persistent faults, whereas large α yields a near binary response that can over react to measurement noise. In Fig. 4b, scheduling success saturates across a broad range of α, and we therefore use $\alpha = 10$ as a stable default.



(a) Regularization coefficient β. (b) Sigmoid slope α.

Fig. 4. Regularization trade off analysis where panel a reports the effect of the regularization coefficient β and panel b reports the effect of the sigmoid slope α.

TABLE I
MEASURED PER STEP CONTROL LOOP OVERHEAD.

Method	Diagnosis module cost	Policy cost	Total overhead
PPO	–	1.0 ms	1.0 ms
PPRL	0.2 ms	1.0 ms	1.2 ms
CIP	50.0 ms	1.0 ms	51.0 ms

1) Runtime Overhead: Table I highlights a trade-off between accuracy and latency that is particularly acute in TSN. Time-aware scheduling and routing adaptation operate under millisecond scale decision intervals in industrial settings [38], which makes controller-side latency a key constraint rather than a secondary metric. While causality-aware modeling can provide grounded disambiguation, its online inference overhead of 51.0 ms in our representative *CIP* baseline can exceed such control-loop budgets. PPRL instead targets a deployable design point in which spectral evidence is computed from small covariance matrices, adding only 0.2 ms on top of PPO inference in our implementation. While Fig. 4 provides controlled guidance for selecting β and α, practical deployments may face load variation and measurement noise that shift the effective evidence distribution. Simple safeguards such as smoothing ϕ_t over a window of size W and limiting the maximal regularization weight can reduce boundary oscillations and prevent unnecessary constraints. We next examine PPM selectivity under diverse non GCL disturbances to confirm that regularization activates only when the protocol defined fault signature is present.

2) Fault Diversity and Selectivity of PPM: To evaluate whether regularization spuriously activates under non GCL disturbances, we augment the fault resilience scenario with stochastic packet loss and per hop jitter. Unlike a deterministic GCL misconfiguration, these disturbances do not induce aligned cross flow delay spikes at a single port and thus yield a diffuse spectrum with low ϕ_t. We report the distribution of ϕ_t across fault types in Fig. 5, where a dashed line at $\phi_t = \phi_0$ marks the sigmoid center and we set ϕ_0 to 0.5 in all experiments. Consistent with the protocol defined nature of the GCL signature, ϕ_t remains substantially lower under loss and jitter than under GCL faults, which keeps $\psi\left[\phi_t\right]$ small and preserves adaptation.

These selectivity results also clarify the operational scope

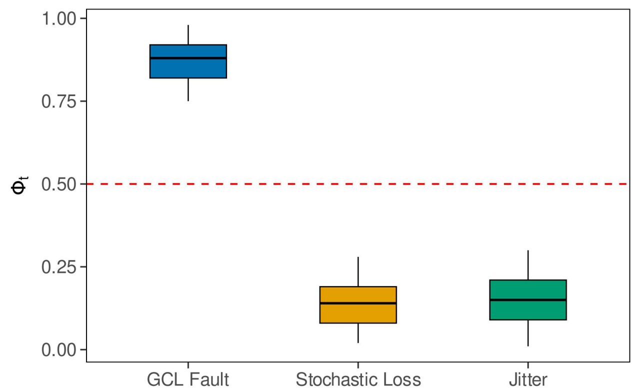


Fig. 5. Fault diversity and selectivity of PPM. We compare the spectral synchronicity index ϕ_t under GCL faults versus stochastic loss and jitter, where the dashed line at $\phi_t = \phi_0$ indicates the sigmoid center and we set ϕ_0 to 0.5.

of the protocol prior exploited by PPM. The spectral signature is strongest when multiple concurrent flows share the faulty port, the GCL misconfiguration persists across several decision intervals, and the controller can reliably observe delay and queueing signals. In low concurrency regimes or highly transient disturbances, the evidence can be weak or noisy, and the regularizer naturally weakens as $\psi\left[\phi_t\right]$ approaches zero, which recovers the standard PPO objective without requiring an explicit ambiguous label. We also note that aligned spikes may arise from exogenous synchronization at the sources or strongly periodic workloads, which motivates evidence modulated regularization rather than hard constraints.

V. Conclusion and Future Work

This paper presents PPRL, a protocol prior-guided reinforcement learning framework for TSN routing that mitigates the correlation trap between congestion driven latency and localized GCL misconfiguration. By extracting spectral evidence through delay covariance analysis and regularizing PPO updates with an adaptive KL penalty, PPRL suppresses futile route switching while preserving adaptation when evidence is diffuse. ns-3 experiments show that PPRL reduces misguided route switching by 84.1% under GCL faults while maintaining scheduling efficiency comparable to strong baselines.

Our results suggest that protocol prior regularization is most effective when the fault signature is persistent and observable across multiple concurrent flows. When concurrency is low or telemetry is noisy and time misaligned, the spectral evidence weakens, ϕ_t becomes low, and $\psi\left[\phi_t\right]$ naturally reduces the KL penalty toward zero. In this regime, PPRL falls back to the standard PPO objective, which preserves adaptability while avoiding over commitment to a potentially spurious prior.

Future work will extend PPM toward a broader taxonomy of TSN failures and will study adaptive calibration strategies for robust operation under load variation. We also plan to explore distributed controllers and tighter coordination between routing and scheduling, which are critical for large scale deployments.

REFERENCES

[1] T. Zhang, G. Wang, C. Xue, J. Wang, M. Nixon, and S. Han, "Time-Sensitive Networking (TSN) for Industrial Automation: Current Advances and Future Directions," *ACM Comput. Surv.*, vol. 57, no. 2, pp. 30:1–30:38, Oct. 2024.

[2] T. Stüber, L. Osswald, S. Lindner, and M. Menth, "A Survey of Scheduling Algorithms for the Time-Aware Shaper in Time-Sensitive Networking (TSN)," *IEEE Access*, vol. 11, pp. 61 192–61 233, 2023.

[3] X. Hong, Y. Xi, and P. Liu, "Flow-Aware Scheduling with Graph Neural Network Routing for Resource Efficiency in Time-Sensitive Networking," in *2024 International Joint Conference on Neural Networks (IJCNN)*, Jun. 2024, pp. 1–9.

[4] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource Management with Deep Reinforcement Learning," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, ser. HotNets '16. New York, NY, USA: Association for Computing Machinery, Nov. 2016, pp. 50–56.

[5] J. Min, Y. Kim, M. Kim, J. Paek, and R. Govindan, "Reinforcement learning based routing for time-aware shaper scheduling in time-sensitive networks," *Computer Networks*, vol. 235, p. 109983, Nov. 2023.

[6] X. He, X. Zhuge, F. Dang, W. Xu, and Z. Yang, "DeepScheduler: Enabling Flow-Aware Scheduling in Time-Sensitive Networking," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, May 2023, pp. 1–10.

[7] S. S. Craciunas, R. S. Oliver, M. Chmelík, and W. Steiner, "Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks," pp. 183–192, Oct. 2016.

[8] S. M. Srinivasan, T. Truong-Huu, and M. Gurusamy, "Machine Learning-Based Link Fault Identification and Localization in Complex Networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6556–6566, Aug. 2019.

[9] B. Arzani, S. Ciraci, L. Chamon, Y. Zhu, H. Liu, J. Padhye, B. T. Loo, and G. Outhred, "007: Democratically finding the cause of packet drops," in *Proceedings of the 15th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'18. USA: USENIX Association, Apr. 2018, pp. 419–435.

[10] S. S. Craciunas, R. S. Oliver, and W. Steiner, "Formal Scheduling Constraints for Time-Sensitive Networks," no. arXiv:1712.02246, Dec. 2017.

[11] A. C. T. dos Santos, B. Schneider, and V. Nigam, "TSNSCHED: Automated Schedule Generation for Time Sensitive Networking," in *2019 Formal Methods in Computer Aided Design (FMCAD)*, Oct. 2019, pp. 69–77.

[12] X. Feng, Y. Wang, J. Lin, W. Li, S. Zhan, Y. Liu, J. Zhang, and J. Wang, "Advancing TSN flow scheduling: An efficient framework without flow isolation constraint," *Comput. Netw.*, Oct. 2024.

[13] G. Li, S. Wang, Y. Huang, X. Zhong, G. Zhang, L. Bai, and T. Huang, "uTAS: Ultra-Reliable Time-Aware Shaper for Time-Sensitive Networks," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, Dec. 2023, pp. 4283–4288.

[14] H. Iwasawa, N. Azuma, H. Masutani, and T. Kuwahara, "Flexible Time-Aware Shaper Scheduling System for a Multitenancy Environment," in *ICC 2023 - IEEE International Conference on Communications*, May 2023, pp. 2571–2576.

[15] R. Yan, M. Liu, Q. Yu, X. Yang, and S. Duan, "Density-Adaptive Gate Control Configuration Method in Time-Sensitive Networking," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2025, pp. 1–7.

[16] R. Debnath, L. Zhao, M. Barzegaran, and S. Steinhorst, "CyclicSim: Comprehensive Evaluation of Cyclic Shapers in Time-Sensitive Networking," in *2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC)*, Jan. 2025, pp. 01–09.

[17] P. Rodriguez-Martin, O. Adamuz-Hinojosa, P. Muñoz, J. Caleya-Sanchez, J. Navarro-Ortiz, and P. Ameigeiras, "Empirical Analysis of the Impact of 5G Jitter on Time-Aware Shaper Scheduling in a 5G-TSN Network," in *2025 IEEE 21st International Conference on Factory Communication Systems (WFCS)*, Jun. 2025, pp. 1–8.

[18] P. Meißner, R. Kirrbach, and F. Bär, "Time Sensitive Networking Testbed with Wireless Real-Time Connectivity via Li-Fi and Energy Efficiency Analysis," in *2025 IEEE 21st International Conference on Factory Communication Systems (WFCS)*, Jun. 2025, pp. 1–4.

[19] Ö. K. Demir and S. Cevher, "Multi-Topology Routing based traffic optimization for IEEE 802.1 Time Sensitive Networking," *Real-Time Systems*, vol. 59, no. 2, pp. 123–159, Jun. 2023.

[20] S. T. Islam and A. B. Muslim, "AI-Based Dynamic Schedule Calculation in Time Sensitive Networks Using GCN-TD3," in *2024 IFIP Networking Conference (IFIP Networking)*, Jun. 2024, pp. 714–719.

[21] F. Luo, Z. Wang, and B. Zhang, "Impact Analysis and Detection of Time-Delay Attacks in Time-Sensitive Networking," *Computer Networks*, vol. 234, p. 109936, 2023.

[22] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Proceedings of the 7th International Conference on Neural Information Processing Systems*, ser. NIPS'93. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Nov. 1993, pp. 671–678.

[23] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 16, Jun. 2018.

[24] B. Lin, Y. Guo, H. Luo, and M. Ding, "TITE: A transformer-based deep reinforcement learning approach for traffic engineering in hybrid SDN with dynamic traffic," *Future Generation Computer Systems*, Jan. 2024.

[25] L. Zhang, H. Cheng, S. Peng, and P. Dong, "Flowlet-Level Routing Optimization with GNN-Based Multi-Agent Deep Reinforcement Learning," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, Dec. 2023, pp. 5214–5219.

[26] M. Ye, J. Zhang, Z. Guo, and H. J. Chao, "FlexDATE: Flexible and Disturbance-Aware Traffic Engineering With Reinforcement Learning in Software-Defined Networks," *IEEE/ACM Transactions on Networking*, vol. 31, no. 4, pp. 1433–1448, Aug. 2023.

[27] V. N. Vitale, A. M. Tulino, A. F. Molisch, and J. Llorca, "A Flexible Multi-Agent Deep Reinforcement Learning Framework for Dynamic Routing and Scheduling of Latency-Critical Services," no. arXiv:2510.11535, Oct. 2025.

[28] Z. Li, J. Ma, and M. Qiu, "GAT-Enhanced DRL Decision Framework for Joint Routing and Scheduling Under Vehicular Time-Sensitive Networking," in *Wireless Artificial Intelligent Computing Systems and Applications: 19th International Conference, WASA 2025, Tokyo, Japan, June 24–26, 2025, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, Jun. 2025, pp. 1–12.

[29] T. Eliyahu, Y. Kazak, G. Katz, and M. Schapira, "Verifying learning-augmented systems," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, ser. SIGCOMM '21. New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 305–318.

[30] C. da Costa Cunha, W. Liu, T. French, and A. Mian, "Q-Cogni: An Integrated Causal Reinforcement Learning Framework," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 12, pp. 6186–6195, Dec. 2024.

[31] M. D. Johas Teener and G. M. Garner, "Overview and timing performance of IEEE 802.1AS," in *Control and Communication 2008 IEEE International Symposium on Precision Clock Synchronization for Measurement*, Sep. 2008, pp. 49–53.

[32] Z. Deng, J. Jiang, G. Long, and C. Zhang, "Causal Reinforcement Learning: A Survey," no. arXiv:2307.01452, Nov. 2023.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," no. arXiv:1707.06347, Aug. 2017.

[34] G. F. Riley and T. R. Henderson, *The Ns-3 Network Simulator*. Berlin, Heidelberg: Springer, 2010, pp. 15–34.

[35] A. Candia and C. Cappo, "TinyICS: An Industrial Control System Simulator Based on NS-3," in *2024 43rd International Conference of the Chilean Computer Science Society (SCCC)*, Oct. 2024, pp. 1–8.

[36] H. Cheng, L. Yang, Q. Zhang, and W. Zhu, "Reliable Routing and Scheduling in Time Sensitive Networks Based on Reinforcement Learning," *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 4, pp. 2415–2427, Jul. 2025.

[37] H. Cao, F. Feng, T. Yang, J. Huo, and Y. Gao, "Causal information prioritization for efficient reinforcement learning," in *International Conference on Learning Representations*, 2025.

[38] C. Xue, T. Zhang, Y. Zhou, M. Nixon, A. Loveless, and S. Han, "Real-Time Scheduling for 802.1Qbv Time-Sensitive Networking (TSN): A Systematic Review and Experimental Study," in *2024 IEEE 30th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, May 2024, pp. 108–121.