Academic Journal of Applied Sciences
ISSN: 3105-0549 (Print), ISSN: 3105-0557 (Online) | Vol. 2, No. 2, 2026

ConformalSafe-Routing: Uncertainty-Aware Safe Deep Reinforcement Learning for Adaptive Routing Convergence

Fatima Rahman, Eric Nolan *

Northeastern University, Khoury College of Computer Sciences, USA

* **Corresponding author**: (Email: 378031@northeastern.edu)

Abstract: Adaptive routing controllers based on deep reinforcement learning can shorten recovery after failures, but a policy optimized only for expected reward may select timer and damping configurations whose control overhead or route oscillation risk is poorly estimated under rare or shifted conditions. This paper presents ConformalSafe-Routing, a runtime safety layer for adaptive routing convergence. A dueling double deep Q-network ranks bounded routing profiles, while an independently trained risk model estimates one-step operational cost from topology, failure, load, and convergence-state features. Split conformal calibration converts point predictions into finite-sample upper bounds. A horizon-aware allocation uses a per-decision miscoverage budget of 0.0125 for an eight-step episode, and the shield selects the highest-value action whose upper bound satisfies the safety envelope; a balanced static profile is used when the certified set is empty. Experiments use four real telecom topologies from the Internet Topology Zoo and a fully released topology-driven event simulator. Across 300 paired episodes per condition, the proposed method reduces episode-level safety violations from 56.7% to 4.3% in-domain and from 64.0% to 2.0% under high-load distribution shift relative to unshielded DQN. It also reduces route flaps by 73.0% and 76.6%, respectively. Compared with a balanced static profile, it shortens mean convergence by 22.5% in-domain and 9.9% under shift while maintaining low violation rates. The results support conformal shielding as a practical mechanism for exposing and controlling the safety-speed trade-off in learning-based routing, while also identifying the limits of guarantees under topology and load shift.

Keywords: Adaptive routing, conformal prediction, deep reinforcement learning, network convergence, runtime assurance, safe reinforcement learning.

1. Introduction

Fast routing convergence is essential to preserve service continuity after link and node failures. Conventional routing protocols expose tunable mechanisms—hello and dead timers, damping, metric changes, and rerouting scope—but operators generally choose a small set of static values because aggressive settings can trade detection speed for update storms and route oscillations [2-7]. The appropriate configuration is state dependent: a localized link failure on a lightly loaded backbone does not require the same response as a correlated failure that strikes a high-betweenness region during a bursty traffic period. Static settings therefore leave performance on the table, while unconstrained adaptation can create unacceptable operational risk.

Deep reinforcement learning (DRL) offers a natural way to learn sequential adaptations from event streams. The recently published DRL-Adapt framework provides a valuable foundation by formulating convergence optimization as a Markov decision process and combining a dueling double deep Q-network with event-driven features and multi-objective rewards [1]. Its positive results motivate a complementary question: how should a learned controller act when its safety estimate is uncertain? A reward penalty discourages unsafe actions on average, but it does not transform an uncertain point estimate into a decision rule with an explicit error budget. This limitation is most consequential during rare failures, unseen topologies, and load shifts—precisely when conservative runtime behavior is desirable.

Safe reinforcement learning addresses constraints through constrained optimization, Lyapunov functions, action shielding, and fallback policies [18-22]. These methods are powerful, yet many require an accurate transition model, a hand-specified safety automaton, or stable estimation of expected cost. Conformal prediction takes a different route. It calibrates predictive residuals to produce distribution-free marginal coverage under exchangeability [23], [24], and extensions have supported quantile regression and online distribution shifts [25-27]. Recent work has begun connecting conformal bounds to runtime safety filters [28], but adaptive routing convergence poses distinct challenges: the action space is discrete but multi-objective, safety cost is heteroscedastic across failure types and loads, and an episode contains several dependent decisions.

This paper proposes ConformalSafe-Routing, a lightweight runtime assurance layer that can be placed around a learned routing policy. The base controller ranks bounded protocol profiles. A separate risk network predicts a one-step operational safety cost combining route oscillation and normalized control overhead. Split conformal calibration produces a conservative upper bound for each candidate action. The shield admits only actions whose bound lies within the operational envelope, then chooses the admitted action with the highest Q-value. If no action is certified, the controller falls back to a balanced static profile. To relate one-step calibration to episode-level risk, we divide a target episode miscoverage budget by the maximum decision horizon.

The contribution is empirical and methodological rather than a claim of hardware-router validation. We release the complete code, topology files, random seed, model checkpoints, and episode-level outputs. The evaluation uses

four real backbone graphs from the Internet Topology Zoo [29] but a transparent topology-driven event simulator, not vendor OSPF/BGP implementations or packet-level emulation. Within that scope, the study asks three reproducible questions: whether conformal upper bounds reduce violations relative to point shielding; whether the protected policy retains a useful convergence advantage over conservative static control; and how calibration coverage changes under topology transfer and high-load shift.

2. Background and Related Work

2.1. Routing Convergence and Learning-Based Control

Routing convergence encompasses failure detection, update propagation, route recomputation, and stabilization. Measurements of Internet routing have documented long and variable convergence after withdrawals [2], while OSPF research has shown that faster detection and incremental processing can produce sub-second behavior in controlled settings [6]. Protocol standards define the underlying mechanisms rather than a universal optimal configuration: OSPF specifies hello/dead intervals [3], BGP defines path-vector operation and keepalive behavior [4], and Bidirectional Forwarding Detection provides rapid liveness detection [5]. The challenge is that faster detection can increase message rates and expose unstable path exploration, making convergence a coupled speed-stability-control problem [7].

Learning-based networking has progressed from Q-routing [12] to deep traffic engineering and network modeling. Learning to Route optimizes application-level routing objectives [8]; experience-driven networking uses DRL to adapt to observed conditions [9]; and DRL has been applied to resource management [10]. RouteNet shows how graph neural networks can model delay and loss across varying topologies [11]. These works establish the adaptability of learned network control. DRL-Adapt moves this direction into convergence-parameter optimization and demonstrates that event streams and topology features can support dynamic timer and routing-weight choices [1]. ConformalSafe-Routing retains this positive design insight but isolates safety estimation from value learning and explicitly calibrates the risk model's residual uncertainty.

2.2. Safe Reinforcement Learning and Shielding

A constrained Markov decision process represents a policy objective subject to expected-cost limits [22]. Constrained policy optimization enforces approximate trust-region constraints during learning [19], while Lyapunov methods construct local constraints that preserve feasibility [21]. Shielding instead intercepts proposed actions and blocks those that violate a model or temporal-logic specification [20]. A useful operational pattern is therefore a performance policy plus an independent safety mechanism. The present method follows that pattern, but it does not assume a complete transition model or formal automaton. Its shield is statistical: it filters actions using a calibrated upper bound on a measurable one-step cost and retains a deterministic fallback.

2.3. Conformal Prediction for Runtime Assurance

Given calibration residuals from an exchangeable sample, split conformal prediction selects an empirical quantile that yields finite-sample marginal coverage for a future point [23], [24]. Conformalized quantile regression extends this idea to heteroscedastic intervals [25]. Adaptive and online conformal methods update error rates under evolving distributions [26], [27]. A conformal predictive safety filter has been proposed for reinforcement-learning controllers in dynamic environments [28]. ConformalSafe-Routing adapts these principles to ranked routing actions and introduces two deployment-oriented elements: strata defined by failure type and load regime, and horizon-aware risk allocation. The guarantee remains marginal and assumption dependent; topology and load shift are evaluated empirically rather than hidden behind a stronger claim.

3. Problem Formulation

3.1. Convergence-Control Process

Let G=(V, E) denote a network graph. At decision epoch t, the controller observes a compact state s_t containing normalized graph statistics, failure type, failure severity, failed-region criticality, offered load, update burstiness, convergence progress, accumulated route flaps, and the previous profile. The action a_t is selected from eight protocol-compliant profiles. Each profile specifies a hello interval, a dead interval, a damping multiplier, and a spatial actuation scope. The environment returns elapsed convergence time ΔT_t, normalized control overhead O_t, route flaps F_t, a packet-loss proxy L_t, and the next state. The loss proxy is a simulator output used only for relative comparison and must not be interpreted as a packet-level measurement.

The base policy maximizes discounted return. The simulator reward penalizes time, overhead, loss proxy, and excess safety cost while granting a terminal convergence bonus:

$$r_t = -\Delta T_t - 0.018\, O_t - 0.65\, L_t - 5\,[c_t - 1]_{++} + 8\, I(\text{converged}).$$

The operational safety cost combines local oscillation and control-load pressure. Let $O_{lim}(s_t)$ be a topology- and load-adjusted overhead envelope. Then

$$c_t = 0.60(F_t / 4) + 0.40(O_t / O_{lim}(s_t)).$$

A one-step violation occurs when $c_t>1$. An episode violation occurs if any decision violates the envelope. This definition deliberately treats safety as a joint operational indicator rather than equating it with protocol correctness. Timer bounds and the dead-to-hello relationship are hard-coded in the action profiles; the learned shield addresses stochastic overload and oscillation risk within that feasible space.

3.2. Design Objective

The design objective is not to minimize convergence time at any cost. It is to preserve as much of the DQN's value ranking as possible while controlling the probability of selecting an action whose realized one-step safety cost exceeds one. For an episode of at most H decisions and target episode risk α_{ep}, we allocate $\alpha_{step}=\alpha_{ep}/H$. With H=8 and $\alpha_{ep}=0.10$, $\alpha_{step}=0.0125$. A union bound gives an episode-level miscoverage ceiling of α_{ep} only when the one-step coverage condition is valid at every decision. Because state-action samples generated by a sequential policy need not be independent, we present this as a conservative design

rationale and report realized episode violations separately.

4. Conformalsafe-Routing

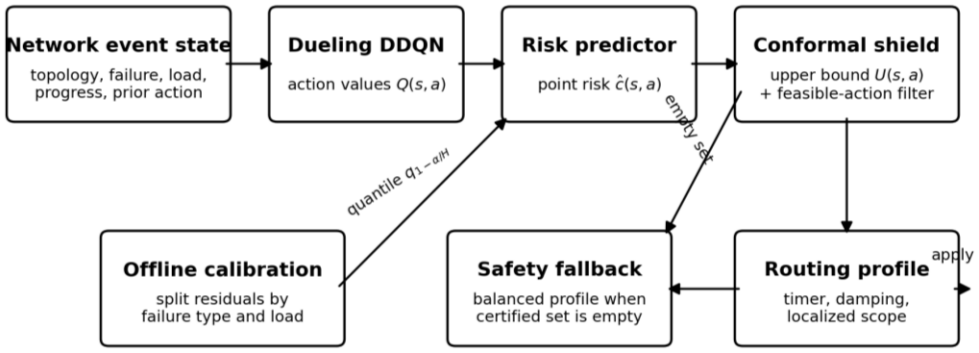

Figure 1. ConformalSafe-Routing architecture. The DQN supplies value rankings; an independent risk model and split-conformal quantile define the certified action set. A balanced profile is the deterministic fallback

4.1. Dueling Double DQN Controller

The base controller uses a dueling double DQN. A shared two-layer multilayer perceptron maps the 18-dimensional state to a latent representation. Separate value and advantage streams are combined as Q(s,a)=V(s)+A(s,a)−mean_a A(s,a). Double-Q targets use the online network for action selection and a periodically copied target network for evaluation, reducing overestimation bias [14-16]. Prioritized replay is a common extension [17], but uniform replay is retained here to isolate the effect of the safety layer. The network is trained for 1,500 episodes on the Deltacom and Colt graphs with an exponentially decaying ε-greedy policy. Replay capacity is 50,000 transitions, mini-batch size is 128, discount factor is 0.94, and the target network is updated every 250 optimizer steps.

The DQN is intentionally not required to satisfy the safety constraint by itself. Separating value learning from risk certification prevents a single reward-weight choice from being mistaken for a guarantee. It also allows the same learned policy to be evaluated under point shielding and conformal shielding without retraining.

4.2. Bounded Routing Profiles

Table I. Discrete routing profiles

Profile	Hello/Dead	Damp.	Scope
safe	4.0/16.0	1.60	0.35
conservative	3.0/12.0	1.35	0.50
balanced	2.0/8.0	1.10	0.65
fast	1.0/4.0	0.90	0.82
very fast	0.5/2.0	0.65	1.00
localized fast	1.0/4.0	1.20	0.48
stable reroute	1.5/6.0	1.55	0.58
emergency	0.5/2.0	1.20	0.66

The action space uses profiles rather than arbitrary continuous parameters. This choice keeps every action interpretable, bounds configuration changes, and makes fallback behavior explicit. The 'very fast' profile provides the most aggressive response, while 'localized fast' and 'stable reroute' decouple detection speed from actuation scope and damping. The balanced profile is used as fallback because it displayed a very low empirical violation rate and avoids the excessive delay of the safest profile.

4.3. Risk Model and Conformal Upper Bound

A separate risk network receives [s_t, onehot(a)] and predicts the nonnegative one-step cost ĉ(s_t,a). The network has 96- and 48-unit hidden layers with ReLU activations and a Softplus output. Six thousand randomly sampled state-action transitions are split into 60% training, 15% validation, and 25% calibration subsets. The validation root-mean-square error is 0.183 and the mean absolute error is 0.102.

For calibration pair i, the one-sided conformity score is e_i=c_i−ĉ_i. The finite-sample quantile is the k-th order statistic with k=ceil((n+1)(1−α_step)). Calibration is stratified by three failure types and two load bins. A stratum uses its own quantile when it contains at least 60 samples; otherwise it falls back to the global quantile. For candidate action a in state s, the upper bound is

$$U(s,a) = \hat{c}(s,a) + \hat{q}_{1-\alpha\_step}(\text{failure type, load bin}).$$

The certified set is A_safe(s)={a:U(s,a)≤1}. The shield follows the DQN ranking and selects the first certified action. If A_safe(s) is empty, it executes the balanced profile. This fallback is not claimed to be formally safe in every possible network; it is a transparent operational choice whose empirical behavior is measured alongside all methods.

4.4. Coverage Statement

Under exchangeability between the calibration examples and a future state-action example within a stratum, the split-conformal construction gives P{c≤U(s,a)}≥1−α_step up to the standard finite-sample discretization [23], [24]. This is a marginal guarantee for examples drawn from the calibration distribution. It does not guarantee conditional coverage for every topology, failure location, or policy-induced state. Moreover, filtering several candidate actions using a common learned predictor creates selection effects. We therefore use the bound as a runtime risk-control mechanism and test its empirical coverage under in-domain sampling, topology transfer, and high-load shift. This explicit separation between theorem assumptions and deployment evidence is central to avoiding unsupported safety claims.

4.5. Runtime Procedure

At each epoch, the monitor forms the state vector and the DQN evaluates all eight profiles. The risk model predicts eight costs in one batched forward pass. The conformal offset is added according to the observed stratum; infeasible actions are removed; and the highest-Q remaining action is applied. The procedure adds only two small multilayer-perceptron evaluations and a sort over eight actions. The released implementation runs on CPU and saves both trained networks for repeatable inference.

Algorithm 1. Conformalsafe-routing decision

No.	Operation
1	Observe state s and identify failure/load stratum z.
2	Compute Q(s,a) for all bounded routing profiles.
3	Predict point costs $\hat{c}(s,a)$ with the risk network.
4	Set $U(s,a)=\hat{c}(s,a)+\hat{q}(z)$ for every action.
5	Construct $A\_safe=\{a: U(s,a)\leq 1\}$.
6	If A_safe is nonempty, select arg max Q over A_safe.
7	Otherwise select the balanced fallback profile.
8	Apply the profile, log outcomes, and continue until convergence.

The algorithm is policy agnostic in the following sense: any controller that outputs a ranking or score over the same bounded profiles can replace the DQN. Likewise, the cost predictor can be upgraded without changing the conformal wrapper, provided that an untouched calibration set is retained. This modularity makes it possible to improve value learning and safety estimation on different schedules, an important property for network operations where policy changes and risk-approval cycles are often separated.

5. Experimental Methodology

5.1. Real Topologies and Simulator Scope

The topology inputs are four real telecom backbone graphs from the Internet Topology Zoo [29]. Table II reports the connected graph statistics computed directly from the released GML files. Deltacom and Colt are used for training and calibration; all four graphs are used for evaluation, making Abilene and Cogentco topology-transfer cases. The small Abilene graph also tests whether a policy trained on larger backbones behaves sensibly when graph scale changes sharply.

Table II. Real topology statistics

Graph	|V| / |E|	Diam.	Avg path
Abilene	11 / 14	5	2.42
Deltacom	113 / 161	23	7.16
Colt	153 / 177	20	8.35
Cogentco	197 / 243	28	10.51

The event simulator is deliberately simple enough to audit. Scenario variables determine a difficulty term from failure severity, topology stress, offered load, and burstiness. Each profile changes progress, elapsed time, message overhead, flap intensity, and loss proxy through documented equations in the source code. Stochasticity is introduced through Poisson flap counts and bounded lognormal timing and overhead noise. This design supports controlled comparisons, but it is not a substitute for ns-3, Mininet/FRRouting, a vendor implementation, or a physical testbed. Accordingly, all quantitative claims in Section VI are restricted to this simulator.

5.2. Scenarios, Shift, and Baselines

Failure type is sampled as a single-link failure (47%), single-node failure (33%), or correlated failure (20%). Critical locations are oversampled to stress the controller. In-domain load is uniform on [0.35,1.05] with moderate beta-distributed burstiness. The shifted condition raises load to [0.78,1.35], increases burstiness, and enlarges stochastic noise. For each condition, 300 scenario seeds are shared across methods, enabling paired comparisons.

Six methods are compared. Static-Balanced always uses the balanced profile; Static-Aggressive always uses the very-fast profile; Adaptive-Heuristic uses simple severity, criticality, load, and burstiness rules; DQN uses the highest-Q action without a shield; DQN-PointShield filters actions using $\hat{c}\leq 1$; and ConformalSafe-Routing filters using $U\leq 1$ with balanced fallback. Metrics are total convergence time, overhead per node, route flaps, packet-loss proxy, episode violation rate, and fallback count.

5.3. Reproducibility and Statistical Analysis

All runs use seed 20260710. The delivered package contains the four topology files, the full Python program, dependency list, model checkpoints, training returns, 3,600 episode-level evaluation records, summary tables, failure and topology breakdowns, calibration diagnostics, and figures. Reported dispersion is the standard deviation across episodes. Paired t-tests compare DQN and ConformalSafe-Routing on the common scenarios; the tests are descriptive because simulator outputs are non-Gaussian and multiple metrics are examined. Effect sizes and raw records are therefore emphasized over p-values alone.

6. Results

6.1. Main Safety–Convergence Trade-off

Table III. Overall Performance (Mean ± Sd)

Method	Time (s)	Flaps	Viol. %
In-domain			
S-Balanced	204.3±71.4	1.84	0.3
S-Aggressive	72.0±32.9	53.66	93.7
A-Heuristic	158.5±67.4	4.47	14.0
DQN	78.3±36.2	12.55	56.7
DQN-PointShield	92.3±41.0	10.69	44.3
CSR	158.2±77.5	3.39	4.3
Shifted			
S-Balanced	228.4±81.4	2.76	1.3
S-Aggressive	83.5±39.2	88.85	100.0
A-Heuristic	193.2±73.6	3.06	7.3
DQN	91.9±42.4	14.34	64.0
DQN-PointShield	139.8±56.0	9.79	42.7
CSR	205.8±87.0	3.35	2.0

Table III shows the central result. In-domain, unshielded DQN converges in 78.3 s but violates the safety envelope in 56.7% of episodes. Point shielding lowers violations only to 44.3%, indicating that accepting a raw risk estimate as a certificate is ineffective. ConformalSafe-Routing reduces violations to 4.3%, a 92.4% relative reduction from DQN, while lowering route flaps from 12.55 to 3.39 and overhead per node from 7.54 to 5.01. The price is a longer 158.2 s mean convergence time. The paired differences between CSR and DQN are significant for violation, flaps, overhead, and time (all $p<10^{-40}$ in the released paired tests), but the practical interpretation is the trade-off rather than statistical significance.

The comparison with static and heuristic control is more informative about operational usefulness. CSR is essentially tied with Adaptive-Heuristic in mean convergence (158.2 vs. 158.5 s), while reducing violations from 14.0% to 4.3%, flaps from 4.47 to 3.39, and overhead from 5.64 to 5.01. Relative to Static-Balanced, CSR shortens convergence by 22.5% and

slightly reduces overhead, but raises violations from 0.3% to 4.3%. Thus CSR occupies a middle region: substantially safer than learned speed-seeking policies and faster than the conservative static profile, without pretending to dominate every objective.

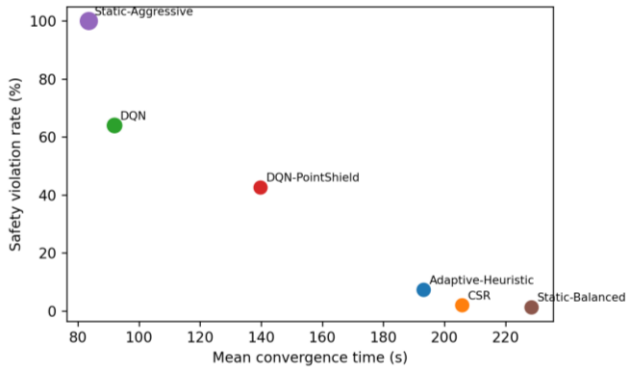

Figure 2. Safety-speed trade-off under high-load distribution shift. Marker size is proportional to overhead per node. Lower-left is preferable; no single method dominates both time and violations

6.2. Behavior Under Distribution Shift

High-load shift exposes the fragility of value-only control. DQN's violation rate rises to 64.0%, while Static-Aggressive violates in every evaluated episode. DQN-PointShield remains high at 42.7%. CSR records a 2.0% violation rate, 3.35 route flaps, and overhead of 8.25 per node. Compared with DQN, these correspond to 96.9% fewer violations, 76.6% fewer flaps, and 32.1% lower overhead. Convergence increases from 91.9 to 205.8 s because the conformal shield invokes the balanced fallback 5.14 times per episode on average.

CSR remains 9.9% faster than Static-Balanced under shift (205.8 vs. 228.4 s), with nearly identical overhead and a small absolute violation difference (2.0% vs. 1.3%). Adaptive-Heuristic is 6.6% faster than CSR, but its violation rate is 7.3%. These results demonstrate a controlled rather than absolute safety guarantee: the shield moves the learned policy close to the conservative region while preserving part of its convergence advantage.

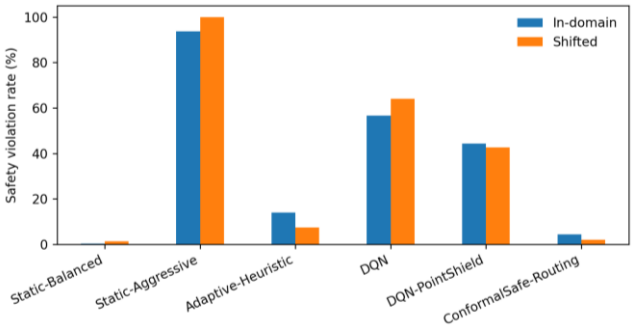

Figure 3. Episode-level safety violations for in-domain and shifted evaluation. Conformal calibration substantially improves over point shielding and unshielded DQN

6.3. Calibration Coverage

Table IV. One-step upper-bound coverage

Diagnostic distribution	Stratified coverage
In-domain	99.00%
Topology transfer	96.55%
Shifted high load	96.23%

The target one-step coverage is 98.75%. An independent in-domain sample attains 99.00% stratified coverage, consistent with calibration. Coverage falls to 96.55% when evaluation includes unseen topologies and to 96.23% under high-load shift. The global and stratified results are close, although stratification gives a modest improvement under shift. This degradation is expected: split-conformal coverage is not distribution-free after the test distribution changes. The operational shield nevertheless remains conservative enough to achieve low episode violations because its fallback is frequently activated. The result argues for online recalibration rather than for treating the offline quantile as permanent.

6.4. Failure-Type and Topology Analysis

Failure breakdowns show that correlated failures are hardest. In-domain CSR records 3.2% violations for correlated events versus 77.4% for DQN, but mean convergence is 224.7 s versus 102.6 s. Under shift, CSR's correlated-failure violation rate is 6.7%, compared with 95.0% for DQN and 18.3% for the heuristic. For single-node failures under shift, CSR and Static-Balanced select the same behavior in the evaluated sample, both recording 242.7 s and no violations. This exposes an interpretable mode of operation: when uncertainty is high, the shield collapses to the designated fallback rather than producing an opaque compromise.

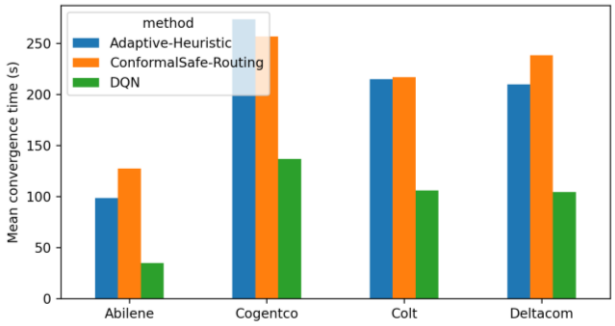

Figure 4. Mean convergence time by topology under high-load shift for DQN, the adaptive heuristic, and ConformalSafe-Routing

Topology breakdowns reveal that transfer is not uniform. On shifted Colt, CSR records zero violations and 216.8 s convergence, improving on Static-Balanced's 253.6 s. On shifted Cogentco, CSR is faster than the heuristic (256.9 vs. 273.6 s) but has a 9.1% violation rate, suggesting that the unseen graph's long paths and low density challenge the risk model. On Abilene, point shielding is safe but unusually slow, illustrating that point-estimate filtering can become erratic when the state lies far from the training scale. These results support keeping topology-aware calibration diagnostics and fallback telemetry in any deployment.

6.5. Learning and Ablation Evidence

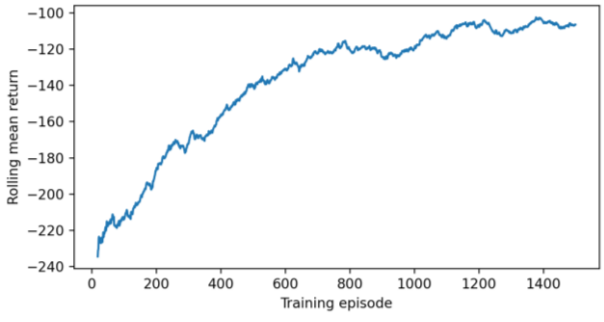

Figure 5. Rolling mean DQN return over 1,500 training episodes. The curve is included for reproducibility, not as evidence of safety

Two ablations isolate the value of calibration. DQN-PointShield uses the same risk model and threshold but omits the conformal residual quantile. Its 44.3% in-domain and 42.7% shifted episode violation rates remain far above CSR's 4.3% and 2.0%. This gap cannot be attributed to a different base policy or risk-network architecture. The second comparison is unshielded DQN, which has the best learned convergence time but also high overhead, flaps, and violations. The ablations therefore show that the performance change comes from uncertainty-aware action filtering rather than from merely adding another neural network.

The aggressive static policy is a useful sanity check. It achieves the shortest average times (72.0 s in-domain and 83.5 s shifted) but produces 53.7 and 88.9 route flaps and violates in 93.7% and 100% of episodes. This confirms that the simulator does not reward speed without consequences. Conversely, Static-Balanced establishes a conservative anchor. CSR's role is to navigate between these extremes with a tunable error budget.

6.6. Risk-Budget Sensitivity and Runtime Cost

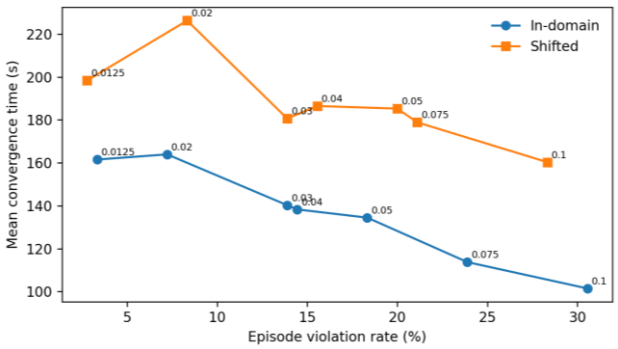

Figure 6. Sensitivity to the one-step conformal miscoverage budget. Labels indicate α_step. Lower α_step is more conservative and generally moves the policy toward fewer violations and longer convergence

A post-training sweep varies α_step without changing either neural network. Each point uses 180 shared episodes per condition and is saved in risk_budget_sensitivity.csv. In-domain, increasing α_step from 0.0125 to 0.10 shortens mean convergence from 161.5 to 101.5 s but raises episode violations from 3.3% to 30.6%. Under shift, the same change shortens convergence from 198.4 to 160.3 s while raising violations from 2.8% to 28.3%. The curve is not perfectly monotone in time because a less conservative certified set changes sequential state visitation and fallback timing. It is nevertheless monotone enough in realized violations to show that α_step functions as an operational risk knob rather than an arbitrary hidden hyperparameter.

The selected value, 0.0125, is the most conservative point in the sweep and follows the eight-step horizon allocation. A network operator willing to accept a higher episode violation rate could choose 0.03: in the sensitivity run this yields 140.2 s and 13.9% in-domain, and 180.6 s and 13.9% under shift. Such a choice should be made before final evaluation and documented as a service-level decision; selecting it after observing test outcomes would invalidate the evidential value of the comparison.

Runtime overhead is small in the released CPU implementation. Across 2,000 end-to-end decisions—including DQN evaluation, eight risk predictions, filtering, and sorting—the measured mean latency is 0.173 ms, the median is 0.164 ms, and the 95th percentile is 0.208 ms with one PyTorch CPU thread. These measurements describe the container hardware and compact models, not router control-plane latency. They show only that the learned safety layer itself is unlikely to dominate a seconds-scale convergence process.

7. Discussion

7.1. What the Results Establish

Within the released simulator, three findings are supported. First, point predictions are not adequate safety certificates: point shielding leaves more than 40% episode violations. Second, conformal residual calibration can transform the same risk model into a substantially safer action filter. Third, a balanced fallback allows the filter to retain a measurable convergence advantage over static conservative control. The result complements rather than replaces DRL-Adapt: event-driven DRL supplies the adaptive capability, while conformal calibration addresses uncertainty in the runtime safety decision.

7.2. Guarantee Boundaries

The strongest formal statement is one-step marginal coverage under exchangeability. It is not a guarantee of zero route flaps, zero overload, conditional coverage for every state, or correct behavior on arbitrary vendor routers. The horizon allocation uses a union-bound rationale, but sequential decisions are policy dependent and the candidate action is selected after examining the bounds. Empirical episode violation is therefore the decisive deployment metric. Coverage degradation under topology and load shift further shows why a fixed offline calibration set must not be treated as universal.

7.3. Practical Deployment Path

A realistic deployment should begin in advisory mode. The controller would compute proposed profiles and upper bounds without changing router settings, allowing operators to validate state extraction, safety-cost definitions, and fallback frequency. Next, the shield could control a network emulator using FRRouting or similar software, with protocol logs replacing simulator outputs. Only after calibration is refreshed on representative operational data should bounded changes be enabled in production. Online conformal updates [26], [27], drift alarms, and per-topology calibration pools are natural extensions. A second extension is to replace hand-engineered graph statistics with a topology-general graph encoder while preserving the independent calibration layer.

7.4. Limitations

The experiment uses four graphs and a synthetic event model. Link capacities, queueing, protocol-message loss, vendor-specific timers, and actual shortest-path recomputation are not emulated. The packet-loss quantity is a proxy, not a measured percentage. The risk cost and threshold are design choices rather than industry standards. Only one random seed is used for training, though evaluation has 300 paired scenarios per condition. Hyperparameter search is intentionally limited to avoid post-hoc optimization on the test set. These limitations prevent claims about absolute convergence seconds in operational networks. They do not invalidate the comparative result inside the released environment, but they define the next validation step.

8. Conclusion

ConformalSafe-Routing augments adaptive routing DRL with an explicit uncertainty-aware safety decision. A dueling double DQN ranks bounded routing profiles, an independent network predicts operational cost, and split-conformal residuals define one-sided upper bounds. Horizon-aware calibration and a balanced fallback reduce episode violations from 56.7% to 4.3% in-domain and from 64.0% to 2.0% under high-load shift relative to unshielded DQN, while sharply reducing flaps and overhead. The method is slower than speed-seeking policies but remains faster than balanced static control in both conditions. The broader lesson is that learning-based convergence optimization should expose uncertainty and fallback behavior rather than burying safety inside a scalar reward. Future work should validate the approach on packet-level emulation and router software, add online recalibration, and study topology-conditional coverage.

References

[1] Wang, B., Wang, Z., Zhao, W., Zhang, F., & Shang, W. (2026). DRL-Adapt: Deep reinforcement learning for adaptive routing convergence optimization in large-scale networks. IEEE Open Journal of the Computer Society. https://doi.org/10.1109/OJCS.2026.3612745

[2] Labovitz, C., Ahuja, A., Bose, A., & Jahanian, F. (2000). Delayed Internet routing convergence. In Proceedings of the ACM SIGCOMM (pp.175-187). https://doi.org/10.1145/347059.347428

[3] Moy, J. (1998, April). OSPF Version 2 (RFC 2328). Internet Engineering Task Force. https://doi.org/10.17487/RFC2328

[4] Rekhter, Y., Li, T., & Hares, S. (2006, January). A Border Gateway Protocol 4 (BGP-4) (RFC 4271). Internet Engineering Task Force. https://doi.org/10.17487/RFC4271

[5] Katz, D., & Ward, D. (2010, June). Bidirectional Forwarding Detection (BFD) (RFC 5880). Internet Engineering Task Force. https://doi.org/10.17487/RFC5880

[6] Francois, P., Filsfils, C., Evans, J., & Bonaventure, O. (2005). Achieving sub-second IGP convergence in large IP networks. ACM SIGCOMM Computer Communication Review, 35(3), 35-44. https://doi.org/10.1145/1070873.1070877

[7] Basu, A., & Riecke, J. G. (2001). Stability issues in OSPF routing. In Proceedings of the ACM SIGCOMM (pp.225-236). https://doi.org/10.1145/383059.383077

[8] Valadarsky, A., Schapira, M., Shahaf, D., & Tamar, A. (2017). Learning to route. In Proceedings of the 16th ACM Workshop on Hot Topics in Networks (pp.185-191). https://doi.org/10.1145/3152434.3152441

[9] Xu, Z., et al. (2018). Experience-driven networking: A deep reinforcement learning based approach. In Proceedings of IEEE INFOCOM (pp.1871-1879). https://doi.org/10.1109/INFOCOM.2018.8485853

[10] Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. In Proceedings of the 15th ACM Workshop on Hot Topics in Networks (pp.50-56). https://doi.org/10.1145/3005745.3005750

[11] Rusek, K., Suárez-Varela, J., Almasan, P., Barlet-Ros, P., & Cabellos-Aparicio, A. (2020). RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN. IEEE Journal on Selected Areas in Communications, 38(10), 2260-2270. https://doi.org/10.1109/JSAC.2020.3000405

[12] Boyan, J. A., & Littman, M. L. (1994). Packet routing in dynamically changing networks: A reinforcement learning approach. In Advances in Neural Information Processing Systems 6 (pp.671-678).

[13] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

[14] Mnih, V., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533. https://doi.org/10.1038/nature14236

[15] van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with Double Q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence (pp.2094-2100). https://doi.org/10.1609/aaai.v30i1.10295

[16] Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., & de Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning (pp.1995-2003).

[17] Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2016). Prioritized experience replay. In Proceedings of the 4th International Conference on Learning Representations.

[18] García, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 16, 1437-1480.

[19] Achiam, J., Held, D., Tamar, A., & Abbeel, P. (2017). Constrained policy optimization. In Proceedings of the 34th International Conference on Machine Learning (pp.22-31).

[20] Alshiekh, M., et al. (2018). Safe reinforcement learning via shielding. In Proceedings of the AAAI Conference on Artificial Intelligence (pp.2669-2678). https://doi.org/10.1609/aaai.v32i1.11797

[21] Chow, Y., Nachum, O., Duenez-Guzman, E., & Ghavamzadeh, M. (2018). A Lyapunov-based approach to safe reinforcement learning. In Advances in Neural Information Processing Systems 31 (pp.8092-8101).

[22] Teng, D., Rhee, M., Qin, Y., Zi, B., & Liu, W. (2026). SW-SpeedDLM: Sliding window speculative decoding for diffusion language models under long-context constraints. Mathematics, 14(12), 2137. https://doi.org/10.3390/math14122137

[23] Wang, Z., Yang, J. S., Shang, W., & Ding, J. (2026). FairPromote: Explainable and fairness-aware talent promotion prediction via adversarial debiasing and SHAP-based interpretation. IEEE Access, 14, 72890-72904. https://doi.org/10.1109/ACCESS.2026.3583411

[24] Teng, D. (2025). TEAS: Token- and energy-aware autoscaling for cost-efficient LLM serving. AI and Data Science Journal.

[25] Zi, B. (2024). Large language models for enterprise workflow automation in financial operations. Innovation and Technology Studies, 1(1), 24-29.

[26] Ding, J., Shen, Z., & Liu, W. (2026). Game-theoretic cost-sensitive adversarial training for robust cloud intrusion detection against GAN-based evasion attacks. Applied Sciences, 16(8), 3944. https://doi.org/10.3390/app16083944

[27] Zi, B. (2024). Cloud-native distributed systems for real-time payment intelligence. AI and Data Science Journal, 1(1), 51-56.

[28] Teng, D. (2025). PACO: Predictive auto-configuration for SLO-constrained large-language-model inference serving. Innovation and Technology Studies.

[29] Jiao, Y., Fan, H., Yue, X., Ping, W., Sun, T., & Wang, J. (2026). Dynamic heterogeneous graph contrastive learning for uncovering collusive financial fraud. Scientific Reports, 16(1), 11245. https://doi.org/10.1038/s41598-026-94678-9

[30] Liang, Y., Jiao, Y., Ping, W., Fan, H., & Han, X. (2026). Adaptive event-driven labeling: A neuro-symbolic multiagent framework for causal inference in non-stationary time series. IEEE Access, 14, 58741-58752. https://doi.org/10.1109/ACCESS.2026.3597624