Safety-Aware Multi-Class QoS Control in SD-WAN via Ensemble-Based Neural CBFs

Ghoshana Bista*, Kamal Singh†, Alain Pégatoquet*, Emmanuel Moulay‡
* Université Côte d'Azur, LEAT, France
† Université Jean Monnet Saint-Étienne, CNRS, Institut d'Optique Graduate School, Laboratoire Hubert Curien, France
‡ XLIM (UMR CNRS 7252), Université de Poitiers, France
Email: *{ghoshana.bista, alain.pegatoquet}@univ-cotedazur.fr,
†kamal.singh@univ-st-etienne.fr, ‡emmanuel.moulay@univ-poitiers.fr

***Abstract*—Software-defined WANs must deliver strict per-class QoS guarantees over heterogeneous paths whose conditions can change rapidly. Reinforcement learning enables adaptive traffic steering, but it can violate SLAs during exploration and under distribution shift. We present an integrated framework that couples a custom Truncated Quantile Critics (TQC) agent with a neural Control Barrier Function (CBF) safety layer. An ensemble dynamics model predicts next-step network state and quantifies uncertainty, allowing the controller to adjust conservatism with network load. When an action is predicted to be unsafe, we project it back into the feasible set using progressive coordinate descent with lexicographic prioritization across traffic classes. In an NS-3 SD-WAN testbed spanning two load scenarios (Scenario A: medium load and Scenario B: high load), our method achieves 90.2% VoIP, 79.0% video, and 73.5% joint VoIP–video compliance at inference on Scenario B, improving joint compliance over unconstrained RL and inference-only filtering. Ablation studies further show that training with ensemble-based CBF-corrected actions is key to these gains.**

***Index Terms*—Software-Defined WAN, Safe Reinforcement Learning, Control Barrier Functions, Quality of Service, Traffic Engineering.**

I. Introduction

Enterprise WANs are being reshaped by cloud migration [1], distributed work [2], and bandwidth-hungry applications [3], stressing static QoS configurations and rigid policy stacks like MPLS/DiffServ [4], [5]. Software-Defined WANs (SD-WANs) address this by pooling multiple underlays (Internet, MPLS, LTE/5G) and enabling centralized, application-aware traffic steering [6]. Maintaining consistent *per-class* QoS, however, remains difficult: VoIP requires low one-way delay and loss [7], video tolerates higher delay but is loss-sensitive [8], while best-effort is elastic, and path conditions vary quickly with congestion and competing flows.

Reinforcement learning (RL) offers a data-driven alternative for adaptive steering and has shown promise in traffic engineering [9]–[11] and congestion control [12], [13]. Yet QoS control is *safety-critical*: during exploration or under distribution shift, an RL policy may violate SLA constraints [14]. Across prior RL-for-TE works, constraint satisfaction is typically a soft objective, and standard "safe RL" techniques Lagrangian methods [15], constrained policy optimization [16], and reward shaping provide only *expected* constraint satisfaction [17], which is unacceptable for real-time traffic.

Control Barrier Functions (CBFs) provide a principled mechanism for enforcing safety constraints online by acting as safety filters that correct a nominal controller or learned policy only when necessary, so that the closed-loop state remains within a forward-invariant safe set [18], [19]. Recent work extends CBF ideas to neural and data-driven settings via predictive safety filters based on learned models [20]–[22], mostly in robotics. Bejarano et al. [23] further show that correcting actions *during training* can improve final performance, which aligns with our design. Applying these ideas to SD-WAN raises two practical challenges: (i) closed-form network dynamics models are not available in practice and must be learned from measurements; (ii) QoS constraints span multiple classes with a natural priority order, where VoIP violations are catastrophic while best-effort (BE) degradation is tolerable. We make this priority trade-off explicit through *lexicographic* constraint ordering.

We propose an integrated framework that combines a custom Truncated Quantile Critics (TQC) agent [24] with a neural CBF-style safety layer[1]. An ensemble of learned dynamics models [25] provides uncertainty-aware next-state predictions, allowing conservatism to adapt with network load. Unsafe actions are projected back toward feasibility via progressive coordinate descent under lexicographic priorities, and *CBF-corrected actions are stored in the replay buffer* during training, letting the policy gradually internalize safe behavior. We evaluate the system on an NS-3/OpenFlow SD-WAN testbed across two load regimes and ablate custom vs. off-the-shelf RL with and without safety filtering, adapting the safety-filter paradigm to networking by (i) quantifying uncertainty via an ensemble dynamics model, (ii) replacing CBF-QP with coordinate-descent projection under nonlinear learned dynamics, and (iii) introducing lexicographic multi-class prioritization to reflect DiffServ-style semantics.

[1]Code and simulation artifacts: https://github.com/safeml/sdwanQoS.

ISBN 978-3-903176-82-9 © 2026 IFIP

TABLE I
PER-CLASS SLA THRESHOLDS AND REWARD WEIGHTS.

Class	ω_c	$w^{(\text{thr})}$	$w^{(\ell)}$	$w^{(d)}$	$w^{(j)}$	$d_c^{\max}$	$\ell_c^{\max}$
VoIP	1.00	0.05	0.15	0.50	0.30	40 ms	1 %
Video	0.40	0.45	0.20	0.25	0.10	60 ms	3 %
BE	0.05	0.90	0.10	0.00	0.00	–	—

II. System Model and Problem Formulation

A. Network Architecture and Traffic

We study a software-defined WAN connecting two enterprise sites through customer-edge (CE) switches S_0, S_1 via two WAN paths: an Internet path through provider-edge (PE) switches and an MPLS path. Each CE steers per-class traffic onto either path. Background hosts inject cross-traffic on the Internet path to emulate competing tenants. The environment is built on NS-3 [26] with an OpenFlow 1.3 control plane [27]. Internet-path bandwidth ranges from 10–50 Mb/s with 15–40 ms propagation delay (BE only); the MPLS path offers 5–30 Mb/s with 8–25 ms latency and per-class priority queueing at the CE switches. These ranges reflect heterogeneity observed in enterprise WAN deployments [6]. Although modest in scale, this dual-path topology captures the structure underpinning most SD-WAN branch-office deployments and lets us interrogate Safe DRL while minimizing confounding variables.

Enterprise traffic is split into three DSCP-marked classes: **VoIP** (DSCP EF, $\leq$ 40 ms / $\leq$ 1% [7]; 64 kb/s CBR, 160 B packets, 15 s mean duration, Poisson arrivals); **Video** (DSCP AF41, $\leq$ 60 ms / $\leq$ 3% [8]; 1 Mb/s, 1200 B packets, 15 s); and **Best Effort** (DSCP 0, no hard SLA; 2 Mb/s, 1500 B packets, 2 s). Offered load follows $\rho(t) = \rho_0 + A\sin(2\pi f t + \phi)$, with ρ_0 drawn uniformly per episode.

B. MDP Formulation

The SDN controller acts at $\Delta t = 1$ s. The agent observes $\mathbf{s} \in \mathbb{R}^{35}$: 14 global metrics (offered load, per-path latency/loss/utilization, aggregate throughput); 15 per-class QoS metrics (normalized throughput, loss, delay, jitter, average throughput); and 6 queue-occupancy values at the two CE switches. Latency and loss are scaled by 10 for feature parity. The action $\mathbf{a} \in [-1, 1]^3$ maps to MPLS routing fractions $\varphi_c = (a_c + 1)/2$ per class c, with $1 - \varphi_c$ on the Internet path. Per-class QoS satisfaction is measured by clipped scores for throughput, loss, delay, and jitter; the per-class composite Σ_c is a weighted average of these, and the aggregate reward is

$$r(\mathbf{s}, \mathbf{a}) = 10\,\frac{\sum_c \omega_c \Sigma_c}{\sum_c \omega_c} - 5, \tag{1}$$

mapping weighted satisfaction to $[-5, +5]$. Table I lists per-class thresholds and weights. We deliberately keep the reward structure standard: when paired with a safety layer, this basic formulation is sufficient for SLA compliance, avoiding ad-hoc reward engineering.

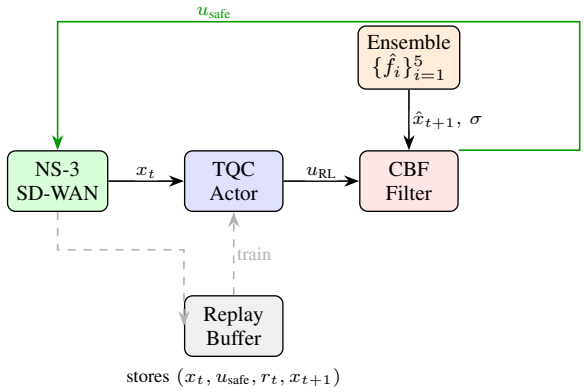

Fig. 1. Training architecture. The TQC actor proposes u_{RL}; the ensemble-backed CBF filter projects it to u_{safe} when predicted QoS violations are detected. The corrected action is executed in NS-3 and the resulting transition is stored in the replay buffer.

C. Safety Constraints and Problem Statement

We define four barrier functions encoding hard QoS bounds for delay-sensitive classes:

$$h_1 = d_{\text{VoIP}}^{\max} - d_{\text{VoIP}}, \qquad h_2 = \ell_{\text{VoIP}}^{\max} - \ell_{\text{VoIP}}, \tag{2}$$

$$h_3 = d_{\text{Video}}^{\max} - d_{\text{Video}}, \qquad h_4 = \ell_{\text{Video}}^{\max} - \ell_{\text{Video}}, \tag{3}$$

with $d_{\text{VoIP}}^{\max} = 0.040$ s, $\ell_{\text{VoIP}}^{\max} = 0.01$, $d_{\text{Video}}^{\max} = 0.060$ s, $\ell_{\text{Video}}^{\max} = 0.03$. The safe set is $\mathcal{S}_{\text{safe}} = \{\mathbf{s} : h_i(\mathbf{s}) \geq 0, \forall i\}$. For discrete-time dynamics $\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t)$, a standard DT-CBF condition [28] is that there exists $\alpha \in (0, 1]$ with

$$h_i(\mathbf{s}_{t+1}) \geq (1 - \alpha)\, h_i(\mathbf{s}_t), \quad \forall i, t, \tag{4}$$

yielding forward invariance by induction when f is known and the condition is enforced exactly. In our setting, f is learned and safety is enforced *predictively* on a conservative one-step risk-aware prediction of $\mathbf{s}_{t+1}$ from an ensemble model. The method thus enforces one-step feasibility under the learned model and is DT-CBF-inspired, but we do not claim formal invariance under model error. The constrained optimization problem is

$$\max_{\pi}\ \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right] \ \text{ s.t. } \ h_i(\mathbf{s}_{t+1}) \geq 0,\ \forall i, t. \tag{5}$$

Penalty-based approaches satisfy (5) only in expectation, permitting transient violations unacceptable for real-time traffic; we instead enforce one-step constraint satisfaction through the safety filter, with an auxiliary penalty used during training to accelerate convergence.

III. Methodology

Our framework has three components: a custom TQC agent that proposes routing actions, an ensemble dynamics model that predicts next-state QoS metrics with calibrated uncertainty, and a neural CBF safety filter that projects unsafe actions onto the feasible set. All three operate in a closed loop in which CBF-corrected actions are stored in the replay buffer (Fig. 1).

A. Truncated Quantile Critics Agent

The original TQC algorithm [24] extends SAC [29] with distributional critics that predict return quantiles, then truncates the highest quantiles to reduce overestimation. We retain this idea but reimplement the agent with SD-WAN-specific choices. The actor is a feedforward network (2×256, ReLU, tanh output to $[-1,1]^3$). Two quantile critics each predict $N=25$ quantiles; during target computation, the top $K=5$ quantiles are dropped from the element-wise minimum of both critics. Training uses Adam ($\text{lr}=3\times10^{-4}$), soft target updates ($\tau=0.005$), $\gamma=0.99$, and gradient clipping (max norm 1.0).

Two modifications enable safety-aware learning. First, the reward is augmented with a QoS violation penalty $r_t = r_{\text{env}} - k_\mu \mu_t$ ($k_\mu = 10$), where $\mu_t = v_{\text{VoIP}}$ if $v_{\text{VoIP}} > 0$ and v_{Video} otherwise, with $v_{\text{VoIP}} = \max(0, d_{\text{VoIP}} - 40\,\text{ms}) + \max(0, \ell_{\text{VoIP}} - 1\%)$ and v_{Video} defined analogously. Second, we store the CBF-corrected action u_{safe} rather than the raw u_{RL} in the replay buffer; this implicit behavioral cloning drives the policy toward the safe manifold and progressively reduces CBF intervention rates over training.

B. Ensemble Neural Dynamics Model

Since closed-form SD-WAN dynamics models are not available in practice, we learn an ensemble of $M=5$ feedforward networks $\{\hat{f}_1, \ldots, \hat{f}_5\}$ (3×256, ReLU). The ensemble prediction and epistemic uncertainty are

$$\begin{aligned} \hat{x}_{t+1} &= \tfrac{1}{M}\sum_i \hat{f}_i(x_t, u_t), \\ \sigma_{t+1} &= \sqrt{\tfrac{1}{M}\sum_i \|\hat{f}_i(x_t, u_t) - \hat{x}_{t+1}\|^2}. \end{aligned} \tag{6}$$

Each member is trained on $N=20{,}000$ transitions collected via mixed exploration (Adam, $\text{lr}=10^{-3}$, early stopping). To produce conservative safety estimates, we evaluate each QoS metric at its worst-case value $m_{\text{WC}} = \hat{m} + \kappa\sigma_m$, where κ adapts to load: $\kappa = 0.15$ under medium load ($L < 60\%$) and $\kappa = 0.3$ at high load ($70\% \le L < 85\%$). This avoids over-conservatism when capacity is abundant while tightening margins near saturation.

C. Neural CBF Safety Filter

When u_{RL} is predicted to violate constraints, the filter solves

$$u_{\text{safe}} = \arg\min_{u\in[-1,1]^3} \|u - u_{\text{RL}}\|^2 \;\text{ s.t. }\; h_i(\hat{f}(x_t,u)) \ge 0,\ \forall i. \tag{7}$$

Because the learned dynamics make this nonlinear, we solve it via *progressive coordinate descent* with lexicographic prioritization. The algorithm performs $P=2$ passes over the three action coordinates; in pass p, a search radius $r_p = 0.35\cdot0.6^{p-1}$ defines the local neighborhood, within which $n_{\text{grid}} = 11$ candidates per coordinate are scored via ensemble prediction of worst-case barrier violations. Before optimization, finite differences ($\pm\epsilon$, $\epsilon = 0.1$) along each action dimension estimate which direction reduces aggregate path-level violations; movements against this direction incur a penalty $w_{\text{dir}} = 0.3$.

Algorithm 1 Neural CBF Action Projection

Require: State x_t, RL action u_{RL}, ensemble $\{\hat{f}_i\}_{i=1}^M$
Ensure: Safe action u_{safe}

```
(x̂_{t+1}, σ) ← EnsemblePredict(x_t, u_RL)
κ ← AdaptiveGain(load(x_t)); v ← WorstCaseViolation(x̂_{t+1}, σ, κ)
if v ≤ ε_safe then return u_RL                ▷ Safe; no correction
end if
u_safe ← u_RL
for j = 1, 2, 3 do
    d_j ← direction reducing violations at ±ε
end for
for p = 1 to P do
    r_p ← 0.35 · 0.6^{p−1}
    for j = 1, 2, 3 do
        Evaluate n_grid candidates in [u_j − r_p, u_j + r_p]
        u_safe[j] ← best (lexicographic, Eq. 8)
    end for
    if all barriers satisfied or no improvement then break
    end if
end for
return u_safe
```

TABLE II
NEURAL CBF HYPERPARAMETERS.

Param.	Sym.	Val.	Param.	Sym.	Val.
Ensemble size	M	5	Max passes	P	2
Hidden dim.	d_h	256	Dir. weight	w_{dir}	0.3
Dynamics lr	η	10^{-3}	Safety tol.	ϵ_{safe}	10^{-6}
Train samples	N	20 000	κ (med / high)	—	0.15 / 0.3
Search radius	r_0	0.35	Safety margin	τ_{margin}	3.2 ms
Radius decay	δ	0.6	Grid res.	n_{grid}	11

Candidates are compared lexicographically:

$$\begin{aligned} u' \succ u \iff\ & v_{\text{VoIP}}(u') < v_{\text{VoIP}}(u) - \epsilon_{\text{tol}} \\ \vee\ & \big(|v_{\text{VoIP}}(u') - v_{\text{VoIP}}(u)| < \epsilon_{\text{tol}} \wedge v_{\text{video}}(u') < v_{\text{video}}(u)\big), \end{aligned} \tag{8}$$

ensuring VoIP is treated as the primary constraint and video as the secondary. To avoid unnecessary intervention, the filter first predicts violations under u_{RL}; if $\Delta_{\text{total}} < \tau_{\text{margin}}$ (3.2 ms, 8% of the VoIP threshold), the action passes through, reducing CBF overhead by 40–60% during converged operation. Worst-case cost is $3\times11\times5 = 165$ ensemble forward passes per action (< 5 ms on GPU, well within the 1 s control interval). If no fully feasible correction is found after P passes, the algorithm returns the least-violating candidate. We use the term "CBF safety filter" to denote this barrier-function-based projection; formal multi-step invariance under fully learned dynamics is left to future work. Algorithm 1 summarizes the procedure; Table II lists hyperparameters.

D. Training Integration

At each step, the TQC actor proposes u_{RL}, the CBF filter projects it to u_{safe} if needed, u_{safe} is executed in NS-3, and

TABLE III
TRAINING AND INFERENCE QOS COMPLIANCE (%) FOR SCENARIOS A AND B. BOLD = BEST **ALL** (JOINT VOIP & VIDEO) PER SCENARIO AND PHASE.

		Training				**Inference**			
Scenario	**Method**	VoIP	Video	BE	All	VoIP	Video	BE	All
A (Med)	Ensemble-CBF	87.9	**93.0**	44.7	**82.1**	**99.8**	**95.4**	41.5	**95.3**
	Without-CBF	89.5	79.2	**93.4**	71.4	93.6	93.4	**96.5**	88.3
	Baseline [31]	**96.7**	74.2	76.7	72.7	99.3	93.0	75.5	92.7
B (High)	Ensemble-CBF	**81.9**	**64.9**	36.2	**57.1**	90.2	**79.0**	54.9	**73.5**
	Without-CBF	77.3	38.9	49.5	26.4	85.1	73.6	52.2	63.2
	Baseline [31]	52.7	55.6	**60.6**	34.9	**95.8**	40.6	**83.9**	40.6

$(x_t, u_{\text{safe}}, r_t, x_{t+1})$ is stored in the replay buffer. Storing the corrected action is critical: the policy learns to stay near the safe manifold, reducing intervention rates from $\sim 35\%$ in early episodes to $\sim 15\%$ at convergence. The ensemble dynamics model is trained offline on data from an initial exploration phase and remains frozen during RL training.

IV. EXPERIMENTAL EVALUATION

The SD-WAN environment uses NS-3 (v3.39) coupled with Python RL and safety-filter modules via NS3-Gym [30]. Topology and traffic follow Section II. Experiments were run on a Linux server using parallel NS-3/Gym environments. Each configuration is trained for 800 episodes of 100 steps and evaluated on 20 held-out inference episodes of 50 steps each. NS-3 seeds are incremented per episode (0–49 train, 50–69 inference). The ensemble is trained on 80 000 transitions from a separate exploration phase. We evaluate two load regimes: **Scenario A** (medium, 45–60% of combined capacity) and **Scenario B** (high, 70–85% with periodic congestion). Three agent configurations are compared: *Ensemble-CBF* (proposed; CBF active during training and inference), *Without-CBF* (same TQC without any safety filter), and *Baseline* [31] (a published DRL+CBF load-balancing method, re-implemented in our environment under identical traffic and SLA thresholds). Joint ("All") compliance requires both VoIP *and* video constraints to hold simultaneously.

A. Safety Across Load Regimes

Table III reports compliance during training and inference, and Fig. 2 shows inference-time CCDFs on Scenario B. The safety filter primarily affects the *distribution tails*. Under Scenario A, all methods satisfy the VoIP latency target with margin and differences appear mainly in loss tails and video latency. Under Scenario B, the separation is clearer: Ensemble-CBF shifts the video latency and loss CCDFs left, reducing exceedance at the 60 ms and 3% thresholds, and reduces the VoIP loss tail relative to Without-CBF. As expected under lexicographic prioritization, BE may absorb residual congestion when the filter is active, reflecting its lowest-priority role. Importantly, gains persist from training to inference: because the agent trains on CBF-corrected actions, it learns to propose safer actions at deployment and requires fewer runtime corrections.

B. Architecture Ablation

Table IV compares all four configurations on Scenario B, where **SB3** denotes the off-the-shelf SB3-Contrib TQC without our custom training modifications and **SB3+CBF** applies the safety layer only at inference. Table IV isolates the effect of our custom training modifications and CBF integration; the cross-method comparison including Baseline [31] is in Table III.

a) Effect of the CBF layer.: The CBF layer improves safety-relevant VoIP–video and joint compliance, although it can reduce BE compliance because BE absorbs residual congestion under lexicographic prioritization. For VoIP at inference, Custom TQC improves from 85.1% to 90.2% with CBF (+5 points), whereas SB3 improves from 76.4% to 81.8% (+5.4 points). A similar trend holds for joint compliance: Custom improves from 63.2% to 73.5%, while SB3 improves from 37.2% to 50.5%. The key difference is training-time integration: our custom agent stores CBF-corrected actions in the replay buffer, so the policy gradually learns to propose actions closer to the feasible set; SB3 trains without corrected-action replay and applies CBF only at inference, so runtime projection must compensate for more unsafe actions. Even without CBF, the custom agent outperforms SB3 (63.2% vs. 37.2% joint compliance at inference), indicating that the reward penalty ($k_\mu = 10$) and architecture choices already encourage QoS-aware behavior; the best result combines reward penalty, corrected-action replay, and the CBF layer.

b) Best-effort trade-off.: Under lexicographic priorities, BE absorbs residual congestion, so Custom TQC+CBF achieves lower BE compliance at inference (54.9%) than unconstrained SB3 (83.7%), which is expected under lexicographic prioritization.

c) Training-time effects.: The advantage of joint CBF integration is even larger during training: Custom+CBF reaches 81.9% VoIP and 64.9% video, vs. 54.6% and 50.7% for SB3+CBF. SB3 trains without safety feedback, so its replay buffer contains uncorrected actions and the policy never learns to avoid unsafe regions. Corrected-action replay acts as an implicit behavioral prior toward feasible actions, complementing the reward penalty and yielding a safer policy both during and after training.

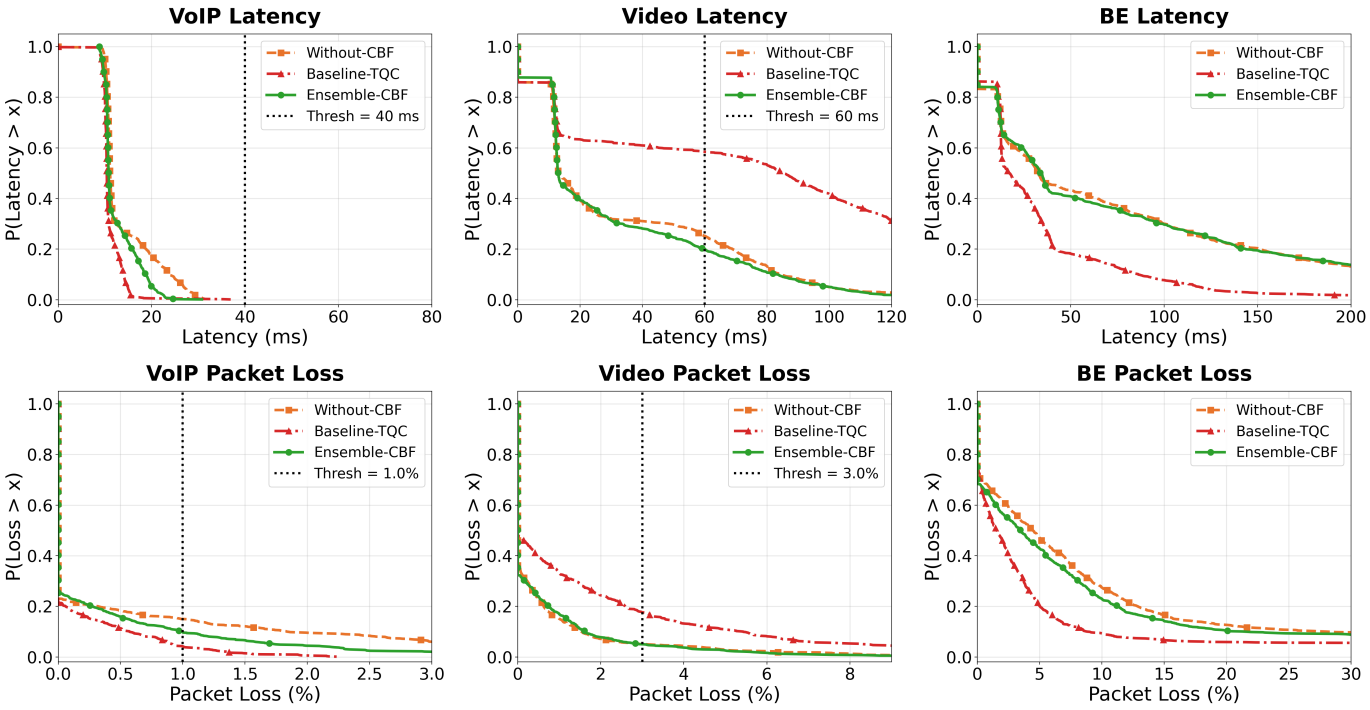

Fig. 2. Inference-time CCDF for Scenario B (high load). Ensemble-CBF reduces tail exceedance at VoIP and video thresholds (dotted lines).

TABLE IV
QoS COMPLIANCE (%) ON SCENARIO B. BOLD = BEST PER COLUMN.

	Inference				Training			
	VoIP	Vid.	BE	All	VoIP	Vid.	BE	All
Custom+CBF (Ens.)	**90.2**	**79.0**	54.9	**73.5**	**81.9**	**64.9**	36.2	**57.1**
Custom	85.1	73.6	52.2	63.2	77.3	38.9	49.5	26.4
SB3+CBF	81.8	52.1	69.0	50.5	54.6	50.7	76.0	35.0
SB3	76.4	43.8	**83.7**	37.2	49.9	49.1	**78.6**	30.9

TABLE V
ENSEMBLE SIZE AND GRID RESOLUTION ABLATION (SCENARIO B).

Config.	VoIP	Video	BE	All
Ensemble size — inference ($n_{grid}=11$)				
$M=1$	89.9	65.7	**65.7**	60.6
$M=5$	**90.2**	79.0	54.9	**73.5**
$M=10$	72.9	**98.6**	58.7	72.9
Ensemble size — training ($n_{grid}=11$)				
$M=1$	**83.4**	44.3	**42.9**	38.3
$M=5$	81.9	64.9	36.2	**57.1**
$M=10$	54.5	**94.3**	33.1	54.2
Grid resolution — inference ($M=5$)				
$n_{\text{grid}}=7$	98.4	43.0	48.3	42.0
$n_{\text{grid}}=11$	90.2	79.0	**54.9**	73.5
$n_{\text{grid}}=15$	**98.6**	**82.2**	39.7	**81.6**
Grid resolution — training ($M=5$)				
$n_{\text{grid}}=7$	**93.4**	26.7	**37.7**	23.4
$n_{\text{grid}}=11$	81.9	64.9	36.2	57.1
$n_{\text{grid}}=15$	88.4	**72.7**	27.4	**63.5**

C. Hyperparameter Sensitivity

Table V varies ensemble size M and grid resolution n_{grid} under Scenario B. For the ensemble sweep at $n_{\text{grid}} = 11$, increasing from $M = 1$ to $M = 5$ substantially improves joint compliance, from 60.6% to 73.5% at inference and from 38.3% to 57.1% during training, indicating that uncertainty quantification from a small ensemble is beneficial. Increasing further to $M = 10$ improves video compliance (98.6% at inference) but reduces VoIP and joint compliance, suggesting over-conservative corrections: the wider uncertainty bands push the filter toward overly cautious actions that, under the learned network dynamics, happen to favor video paths at the expense of VoIP loss compliance — even though the lexicographic ordering in the filter itself still prioritizes VoIP. Thus, $M=5$ offers the best overall trade-off.

For the grid-resolution sweep with $M = 5$, $n_{\text{grid}} = 15$ gives the strongest Scenario B joint VoIP–video compliance among the tested grid sizes, while $n_{\text{grid}} = 11$ is retained as the default because it is the configuration used consistently throughout the main evaluation pipeline. This suggests that finer projection grids can improve the local correction quality, but the best grid resolution remains an implementation-dependent trade-off between QoS performance and computational cost.

V. Conclusion

We presented a safety-aware traffic engineering framework for SD-WAN that pairs a custom TQC agent with a neural CBF safety layer using an ensemble dynamics model with load-adaptive uncertainty margins and lexicographic multi-class prioritization. On NS-3 across two load regimes, training with CBF-corrected actions improves inference-time joint VoIP–video compliance: under Scenario B, the full system reaches 90.2% VoIP, 79.0% video, and 73.5% joint compliance, compared with 85.1%, 73.6%, and 63.2% without CBF (Table III). Ablations confirm that ensemble size and grid resolution both matter, affecting the trade-off between VoIP protection, video compliance, and joint QoS satisfaction. Future work will address online ensemble adaptation, multi-site coordination, and formal verification of the learned barrier conditions.

Acknowledgment

This work has been supported by grant ANR-21-CE25-0005 from the Agence Nationale de la Recherche in France for the SAFE project.

References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology (NIST), Special Publication 800-145, 2011, accessed: 2026-02-08. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[2] Gartner. (2024, May) Gartner forecasts worldwide public cloud end-user spending to surpass $675 billion in 2024. Press release. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2024-05-20-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-surpass-675-billion-in-2024

[3] Cisco, "Cisco annual internet report (2018–2023)," White paper, 2020, accessed: 2026-02-08. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[4] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," RFC 3031, IETF, 2001, accessed: 2026-02-08. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3031

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, IETF, Dec. 1998, accessed: 2026-02-08. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2475

[6] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: Experience with a globally-deployed software defined WAN," in *Proceedings of the ACM SIGCOMM*, 2013, pp. 3–14.

[7] ITU-T, "Recommendation G.114: One-way transmission time," International Telecommunication Union, Tech. Rep., May 2003.

[8] ——, "Recommendation G.1010: End-user multimedia QoS categories," International Telecommunication Union, Tech. Rep., Nov. 2001.

[9] S. Troia, F. Sapienza, L. Varè, and G. Maier, "On deep reinforcement learning for traffic engineering in SD-WAN," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2198–2212, 2021.

[10] X. Pei *et al.*, "Enabling efficient routing for traffic engineering in SDN with deep reinforcement learning," *Computer Networks*, vol. 241, p. 110220, 2024.

[11] 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 Trans. Netw.*, vol. 31, no. 4, pp. 1433–1448, 2023.

[12] N. Jay, N. Rotman, B. Godfrey, M. Schapira, and A. Tamar, "A deep reinforcement learning perspective on internet congestion control," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2019, pp. 3050–3059.

[13] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with Pensieve," in *Proceedings of the ACM SIGCOMM*, 2017, pp. 197–210.

[14] W. Zhao, T. He, R. Chen, T. Wei, and C. Liu, "State-wise safe reinforcement learning: A survey," in *Proc. IJCAI*, 2023, pp. 6814–6822.

[15] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," in *Proc. ICLR*, 2019.

[16] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. ICML*, 2017, pp. 22–31.

[17] S. Gu *et al.*, "A review of safe reinforcement learning: Methods, theory and applications," *arXiv preprint arXiv:2205.10330*, 2024.

[18] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, Jun. 2019, pp. 3420–3431, accessed: 2026-02-08. [Online]. Available: https://arxiv.org/abs/1903.11199

[19] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[20] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods," *IEEE Trans. Robotics*, vol. 39, no. 3, pp. 1749–1767, 2023.

[21] O. So, Z. Serlin, M. Mann, J. Gonzales, K. Rutledge, N. Roy, and C. Fan, "How to train your neural control barrier function: Learning safety filters for complex input-constrained systems," in *Proc. IEEE ICRA*, 2024, pp. 11 532–11 539.

[22] K. P. Wabersich, L. Hewing, A. Carron, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-Jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–164, 2023.

[23] F. P. Bejarano, L. Brunke, and A. P. Schoellig, "Safety filtering while training: Improving the performance and sample efficiency of reinforcement learning agents," *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 788–795, 2025.

[24] A. Kuznetsov, P. Shvechikov, A. Grishin, and D. Vetrov, "Controlling overestimation bias with truncated mixture of continuous distributional quantile critics," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 119, 2020, pp. 5556–5566, accessed: 2026-02-08. [Online]. Available: https://proceedings.mlr.press/v119/kuznetsov20a.html

[25] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 6402–6413, accessed: 2026-02-08. [Online]. Available: https://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles

[26] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Günes, and J. Gross, Eds. Berlin, Heidelberg: Springer, 2010, pp. 15–34.

[27] Open Networking Foundation, "OpenFlow Switch Specification, version 1.3.0," Open Networking Foundation (ONF), Tech. Rep., Jun. 2012, released June 2012. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf

[28] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," in *Robotics: Science and Systems (RSS)*, 2017.

[29] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep rl with a stochastic actor," in *Proc. ICML*, 2018, pp. 1861–1870.

[30] P. Gawłowicz and A. Zubow, "ns-3 meets OpenAI Gym: The playground for machine learning in networking research," in *Proc. ACM MSWiM*, 2019, pp. 113–120.

[31] L. Dinh, P. Tran Anh Quang, and J. Leguay, "Safe load balancing in software-defined-networking," *Computer Communications*, 2025, pII: S0140366424003323.