Efficient Graph Attention-based Learning for Traffic Prediction and Uncertainty-Aware Anomaly Detection in AI-driven O-RAN

Anonymous Authors[1]

Abstract

Traffic load prediction to enable proactive congestion management and resource optimization in Open Radio Access Networks (O-RAN) is challenging, given exhibit distinct temporal patterns, dynamic ranges, and burst behaviors in multiple service slices. To address this challenge, we propose a novelly efficient traffic prediction architecture that processes sixteen radio-level features through three parallel branches: a per-feature shared BiLSTM for temporal dynamics, a Graph Attention Network (GAT) over a feature-as-node graph for cross-feature dependencies, and a dual Transformer over magnitude and phase spectra from the input window. The branch outputs are fused per node, mean-pooled across nodes, and trained per network service slice using a single-step regression objective on granted physical resource blocks. We apply mean squared error for the smoother eMBB slice and show that pure Huber loss is more effective for the burstier mMTC and uRLLC slices than both mean squared error and upper-tail weighted mean squared error. A post-hoc Chebyshev test calibrated on validation residuals converts predictions into a slice-aware anomaly flag without labeled anomalies. On the Colosseum O-RAN dataset, the model achieves promising performance, $R^2 = 0.9011$ on eMBB, $R^2 = 0.3002$ on mMTC, and $R^2 = 0.6624$ on uRLLC, outperforming the existing studies with Chebyshev flag rates below 3.5% across all slices.

1. Introduction

The deployment of network slicing in 5G and Open Radio Access Network (O-RAN) systems has made traffic prediction a load-bearing primitive for radio resource scheduling, admission control, and proactive anomaly response. Since O-RAN exposes programmable control loops and telemetry through open interfaces, slicing maps heterogeneous services to differentiated QoS targets (Polese et al., 2023b; Shah et al., 2025a). Operators are expected to reserve physical resource blocks (PRBs) for three standardized slice types whose statistical signatures differ qualitatively: eMBB carries sustained high-throughput traffic with smooth envelopes, mMTC carries large numbers of small bursts whose aggregate is dominated by short-time-scale noise, and uRLLC carries strict-latency traffic with intermittent spikes. A single-architecture predictor that performs uniformly well across these three regimes has remained elusive in the public O-RAN literature, with mMTC and uRLLC frequently reported as the failure modes. Two design pressures shape this work. First, per-feature radio metrics in an O-RAN datalake (e.g., buffer occupancy, signal quality, packet counts) carry complementary information that purely temporal models miss due to differing units and time scales; a graph view treating each metric as a node enables learned attention without manual interactions (Zhang et al., 2026). Second, periodic patterns in the time domain become explicit under a discrete Fourier transform, motivating a parallel frequency-domain branch to recover regularities the time-domain branch overlooks (Habib et al., 2024).

Unlike prior work, this study's contributions focus on a novel three-branch architecture (Section 3) that combines per-feature-shared BiLSTM, a Graph Attention Network over a feature-as-node graph, and a dual Transformer over the magnitude and phase spectra of the windowed signal. The model runs a Chebyshev-calibrated anomaly test that operates post hoc on validation residuals and yields a tunable, distribution-free flag rate. We report per-slice prediction metrics and a controlled ablation across GAT head merging and loss type on the Colosseum O-RAN dataset (Section 5), showing that upper-tail weighted MSE is harmful in this setting whereas pure Huber loss is the strongest completed configuration for mMTC and uRLLC.

2. Related Work

Traffic prediction in radio access networks has historically been approached with sequence models such as LSTMs and

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Transformers operating on aggregated cell-level traces. For example, DeepCog reframes the problem for network slicing as cost-aware capacity forecasting, where predictions are evaluated by their impact on overprovisioning and service-level violations rather than by symmetric traffic-error metrics alone (Bega et al., 2020). Another complementary direction converts traffic time series or traffic matrices into image representations, allowing pretrained visual feature extractors and recurrent predictors to exploit spatially organized patterns before forecasting future load (Kablaoui et al., 2024). Hybrid deep learning models have also combined attention-based Conv-LSTM modules for spatial-temporal traffic patterns with Bi-LSTM modules, showing the value of separating short-term dynamics from recurring temporal structure in traffic forecasting (Zheng et al., 2021).

Graph-based traffic forecasters provide a closer precedent for dependency modeling: TGC-LSTM and T-GCN to jointly capture road-network topology and temporal evolution (Cui et al., 2020; Zhao et al., 2020). Later graph models make these dependencies more adaptive by learning dynamic graph structures, using spatial-temporal attention, or applying Transformer modules over loop-detector networks, which further motivates treating structural dependence as data-dependent rather than fixed a priori (Zhang, 2025; Huang et al., 2023; Xiao et al., 2025). Further, Visibility-graph structural encoding offers another route to Transformer-compatible multivariate time-series learning by converting temporal visibility relations into sparse attention structure, reducing attention dispersion while preserving interpretable temporal dependencies (Chen et al., 2025). Finally, extreme-value-aware forecasting provides a further complementary perspective: TXtreme combines a mixture-model extreme indicator with LSTM, feed-forward, and Transformer components to model normal and tail observations separately, underscoring why rare traffic surges should not be treated as ordinary symmetric-error cases (Yadav & Thakkar, 2025).

Complementary work on dynamic O-RAN slicing for connected vehicles implements a DRL-based xApp for slice migration, resource allocation, and handover optimization, illustrating why per-slice prediction is operationally relevant for mobile and QoS-sensitive services (Shah et al., 2025a; Lotfi & Afghah, 2023). Recent AI-and-RAN orchestration work further couples workload forecasting and anomaly detection with reinforcement-learning-based resource allocation over O-RAN interfaces, reinforcing the need for predictors whose residuals can be converted into actionable runtime signals (Shah et al., 2025b). Per-slice prediction, in which the predictor must distinguish among slice-specific statistical regimes, is a more recent line of work driven by the standardization of network slicing in 5G and the availability of slice-segregated traces such as the Colosseum O-RAN dataset (Polese et al., 2023a; Jia et al., 2025).

Cross-slice attention-based learning, i.e., combining temporal, spatial (graph), and frequency representations, has been underexplored for general traffic-flow forecasting, where prediction-based anomaly detection with a Chebyshev threshold has been proposed as a way to obtain a calibrated runtime flag without anomaly labels. This work extends the family of attention-based learning methods to the per-slice, per-PRB regression objective relevant to O-RAN scheduling, and treats the radio metrics themselves as graph nodes rather than treating cells or base stations as nodes.

3. Methodology

3.1. Problem Formulation

For each slice $s \in \{\text{eMBB}, \text{mMTC}, \text{uRLLC}\}$ and each one-second timestep t, we observe a vector $x_t^{(s)} \in \mathbb{R}^N$ of N radio-level features. We denote the one-step-ahead granted PRB count for slice s as $y_{t+1}^{(s)}$; in the dataset, this target corresponds to the future value of `sum_granted_prbs`. Given a window of T historical steps $X_t^{(s)} = (x_{t-T+1}^{(s)}, \ldots, x_t^{(s)}) \in \mathbb{R}^{T \times N}$ in the primary configuration, the predictor $f_\theta^{(s)}$ produces an estimate $\hat{y}_{t+1}^{(s)} = f_\theta^{(s)}(X_t^{(s)})$. Each slice is trained independently because the three slice types are collected on disjoint time ranges and there is no cross-slice attention to exploit. The primary configuration uses $N = 16$ feature nodes formed by the fifteen radio metrics together with the historical target.

3.2. Cross-slice traffic prediction architecture

The predictor consists of three parallel branches that share a window-level input and produce per-node embeddings of dimension $H = 64$, followed by a per-node fusion head and a mean readout across nodes. Figure 1 summarizes the complete prediction and post-hoc anomaly-flagging pipeline.

Temporal Branch. Each of the N feature nodes is processed independently by a parameter-shared BiLSTM. The scalar history of length T is first projected to dimension 16, passed through a two-layer BiLSTM with hidden size 64 and dropout 0.2, and the last hidden state of dimension $2 \cdot 64$ is projected to $H = 64$, followed by ReLU and dropout. The result is reshaped to $\mathbb{R}^{B \times N \times H}$, where B denotes the batch dimension.

Spatial Branch. The spatial branch represents the window as a graph in which each of the N feature nodes carries the entire T-step trajectory as its node feature. The adjacency matrix is binary fully-connected with self-loops in the primary configuration, so that the GAT learns the cross-feature interaction structure end-to-end through its attention coefficients. After a linear projection from T to 32 and dropout, the signal passes through two GAT layers of hid-

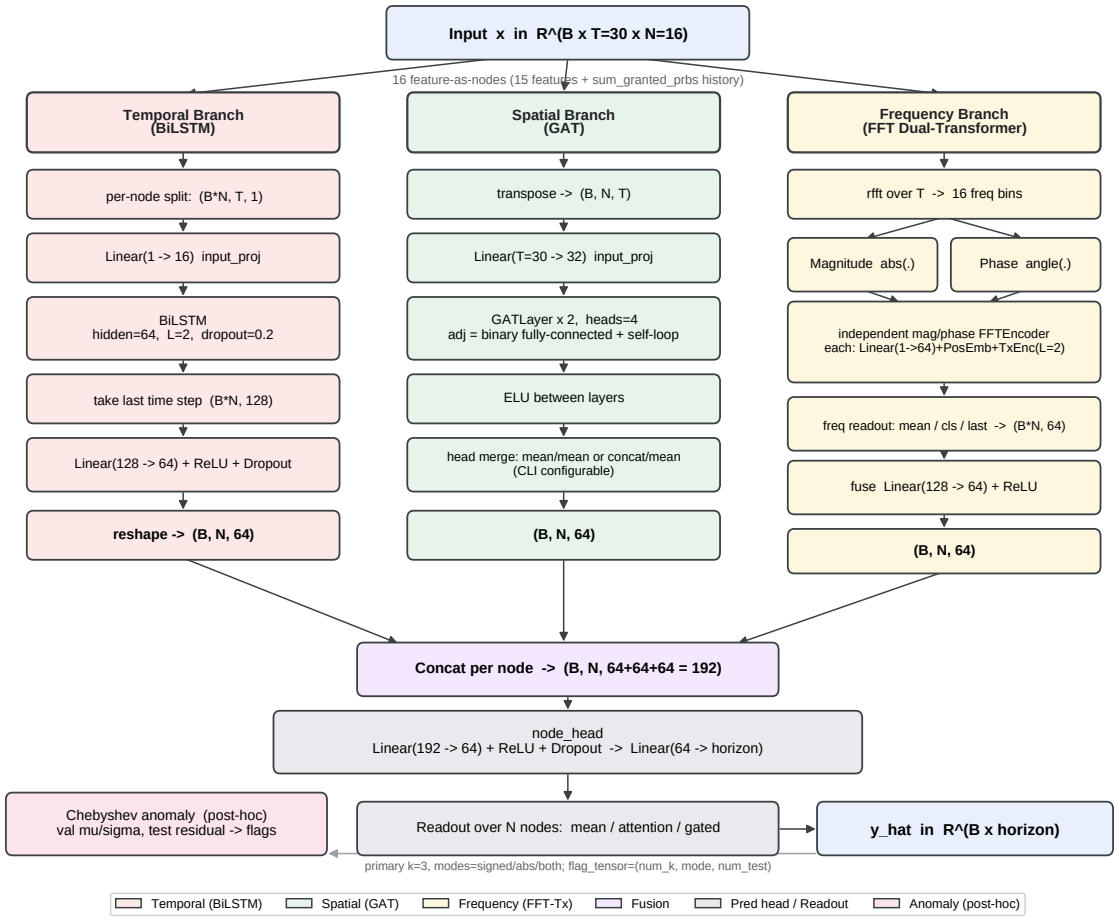

Figure 1. Overall model architecture: a T-step, N-node input window is processed by temporal, spatial, and frequency-domain branches. Their per-node embeddings are concatenated, passed through a shared node-level prediction head, read out across nodes, and then converted into label-free anomaly flags by a post-hoc Chebyshev test on validation-calibrated residuals.

den size 64 with four attention heads, separated by an ELU activation. Heads of the intermediate layer are merged either by averaging or by concatenation while keeping the branch output dimension fixed at 64; in the concatenation setting, each head produces a lower-dimensional subspace whose concatenation restores the hidden size. The heads of the final layer are always averaged so that the output dimension equals $H = 64$.

Frequency Branch. The frequency branch applies the real-input fast Fourier transform (RFFT) along the time axis to each node trajectory in the input window. RFFT computes the discrete Fourier spectrum for real-valued sequences while retaining only the non-redundant frequency components, denoted here as $f = \mathrm{RFFT}(X_t)$, producing $F = \lfloor T/2 \rfloor + 1$ complex bins per node. This corresponds to 16 bins for the primary $T = 30$ configuration and 8 bins for the $T = 15$ sequence-length ablation. The magnitude $|f|$ and the phase $\arg f$ are processed by two separate Transformer encoders with the same configuration: each spectrum is projected to dimension 64, summed with a learnable positional embedding over the frequency bins, and passed through a two-layer Transformer encoder with four heads, feed-forward dimension 256, dropout 0.1, and pre-LayerNorm. The bin dimension is then mean-pooled in each pathway, and each pooled pathway output is independently projected through a linear layer with ReLU and dropout. The resulting magnitude and phase vectors are concatenated and passed through a second linear projection with ReLU and dropout to produce the frequency embedding of dimension $H = 64$.

Fusion and Readout. The three per-node embeddings are concatenated and passed through a linear layer of size $3H \to H$ with ReLU and dropout, followed by a per-node prediction head $H \to 1$ that produces $\hat{y}_{t+1}^{(s,n)}$ for each node n. The window-level prediction is the mean over the N node-level predictions. The mean readout is selected over

attention or gated alternatives because it imposes no slice-specific tuning and matches the symmetric role of the feature nodes under the binary fully-connected adjacency.

3.3. Loss

Training is performed in the scaled target space produced by a RobustScaler fitted on the training data. We evaluate three main regression losses. The first is mean squared error (MSE), which is the primary loss for eMBB. The second is a weighted MSE variant whose sample weight increases linearly above the training-target quantile $q = 0.9$ and saturates at $1 + \alpha$ by $q = 0.99$, with $\alpha = 4$, motivated by the long-tailed nature of the spike traffic in mMTC and uRLLC. The third is a pure Huber loss with $\delta = 1.0$, without any upper-tail weighting. The Huber loss is included because it preserves quadratic sensitivity near zero while making large residuals linear, which can reduce the influence of unpredictable spike outliers on the optimization direction. Optimization uses AdamW with learning rate 10^{-4} and a cosine schedule.

3.4. Chebyshev Anomaly Flag

After training, prediction residuals on the held-out validation slice are used to fit two reference distributions per slice: a signed residual distribution with mean μ_s and standard deviation σ_s, and an absolute residual distribution with mean μ_a and standard deviation σ_a. At test time, a sample is flagged whenever the signed residual exceeds $\mu_s \pm k\sigma_s$ or the absolute residual exceeds $\mu_a + k\sigma_a$, with $k = 3$ in the primary configuration. By the Chebyshev inequality the false-positive rate is bounded above by $1/k^2$ regardless of the residual distribution shape, which yields a closed-form, distribution-free guarantee that is more conservative than the empirical flag rates we observe.

4. Experimental Setup

Experiments are conducted on the Colosseum O-RAN dataset (Polese et al., 2023a), organized into 28 trial directories indexed `tr0` through `tr27`. Each trial contains slice-segregated metrics for the three slice types under the directory naming convention `embb`, `mtc`, and `urllc`. We use a fixed split in which trials `tr0--tr9`, `tr11--tr14`, and `tr16--tr27` form the training set, `tr10` forms the validation set, and `tr15` is reserved for testing. This per-trial leave-out scheme avoids temporal leakage that a within-trial random split would introduce, because consecutive samples within a trial are correlated.

4.1. Training Configuration

All first- and second-round runs use 100 epochs, early-stopping patience of 30 epochs on validation main loss, batch size 1024, learning rate 10^{-4} with a cosine schedule, and seed 42. The Huber runs use the same patience and optimizer settings but extend the maximum budget to 200 epochs; the completed mMTC and uRLLC Huber runs early-stop at epochs 111 and 149, respectively. The temporal branch uses a two-layer BiLSTM with hidden size 64, dropout 0.2, and an input projection to dimension 16. The spatial branch uses two GAT layers of hidden size 64 with four heads, dropout 0.1, and an input projection to dimension 32. The frequency branch uses Transformer encoders (hidden size 64, four heads, two layers, feed-forward 256, dropout 0.1) with mean readout over rFFT bins. The adjacency is binary with self-loops, node readout is mean, and the historical target is included so that $N = 16$. The Chebyshev test uses $k = 3$ as the primary threshold and reports both signed and absolute flag rates.

4.2. Evaluation Metrics

We report mean absolute error (MAE), root mean squared error (RMSE), coefficient of determination (R^2), and mean absolute percentage error (MAPE) on the test trial `tr15`, all computed in the original (unscaled) PRB space. We also report the minimum validation main loss in the scaled space and the epoch at which it occurs, which together summarize convergence behavior. The anomaly-detection statistics are reported as residual mean, residual standard deviation, and flag rate at $k = 3$ for both signed and absolute modes; because the dataset carries no anomaly labels, no precision, recall, or F1 numbers are reported.

4.3. Ablation Axes

Holding the configuration above fixed, we vary three architectural choices to understand their per-slice effect:

GAT head merge. The merge of the four attention heads in the intermediate GAT layer is set either to mean or to concatenation; the final layer is always set to mean so that the branch output dimension equals H.

Sequence length. The history window is set to either $T = 30$ or $T = 15$. The frequency branch applies RFFT to the same window, so this ablation changes both the temporal context and the number of frequency bins.

Loss type. The training loss is set to MSE, weighted MSE with a linear weight ramp from quantile $q = 0.9$ to $q = 0.99$ and $\alpha = 4$, or pure Huber loss with $\delta = 1.0$.

5. Results

5.1. Cross-Slice Test Performance

Table 1 reports the best completed test configuration per slice. The eMBB slice is fit accurately with R^2 =

Table 1. Best completed test configuration per slice on `tr15`. All numbers are computed in the original PRB space. "head merge" refers to the intermediate GAT layer; the final GAT layer is mean.

SLICE	MERGE	SEQ	LOSS	R^2	MAE
EMBB	CONCAT	30	MSE	0.9011	250.21
MMTC	CONCAT	30	HUBER	0.3002	25.34
URLLC	CONCAT	30	HUBER	0.6624	65.68

0.9011 and MAPE of 13.66%, consistent with the smooth, sustained-throughput regime. The mMTC slice is substantially improved by switching from MSE to pure Huber loss, reaching $R^2 = 0.3002$ and reducing MAPE to 24.71%. The uRLLC slice also benefits from Huber loss, reaching $R^2 = 0.6624$ and MAPE of 27.67%. These results indicate that the architecture can support different slice-specific loss choices without changing the feature extractor, and that robust regression is important for the two burstier slices.

5.2. Convergence Behavior

A consistent observation across the MSE configurations is that the validation main loss follows a stable long-term downward trend and the train–validation gap remains small. The eMBB run with mean head merge converges to a final training loss of 0.0239 and a final validation loss of 0.0241, so that the gap is effectively zero. The uRLLC MSE run with mean head merge reaches a final training loss of 0.919 and a validation loss of 0.866, with the validation curve still drifting slowly downward at epoch 100. The mMTC MSE run with concatenated heads bottoms out near a validation loss of 0.985 at epoch 67 and oscillates in a narrow band thereafter. For the Huber runs, the validation-loss scale is not directly comparable with MSE, but the completed mMTC and uRLLC runs select late best checkpoints at epochs 81 and 119, respectively. The best epoch lies well beyond the first few epochs for the useful configurations, indicating that the multi-domain capacity is consumed productively over a long training budget rather than collapsing immediately.

5.3. Ablation: GAT Head Merge

Table 2 compares mean and concatenation merging of the intermediate GAT heads at fixed sequence length $T = 30$ and MSE loss. The two slice types whose targets are dominated by smooth or counting traffic (eMBB, mMTC) prefer concatenation, with ΔR^2 of roughly $+0.006$ and $+0.015$ respectively. The uRLLC slice is the only one in which mean merging dominates under MSE, with $\Delta R^2 \approx +0.010$ and a lower validation loss. Because the completed Huber runs use concatenated intermediate heads, these MSE-only head-merge results should be interpreted as an architectural ablation rather than as a final loss-conditioned claim about the optimal merge rule.

Table 2. Ablation on intermediate GAT head merge at sequence length $T = 30$ with MSE loss. Final-layer head merge is mean.

SLICE	MERGE	R^2	MAE	VAL LOSS
EMBB	CONCAT	0.9011	250.21	0.02610
EMBB	MEAN	0.8949	259.28	0.02372
MMTC	CONCAT	0.0830	34.92	0.9845
MMTC	MEAN	0.0679	35.17	1.0077
URLLC	CONCAT	0.6376	76.67	0.8945
URLLC	MEAN	0.6472	74.95	0.8628

5.4. Ablation: Sequence Length

Reducing the history window from $T = 30$ to $T = 15$ does not improve the MSE configurations. Under concatenated intermediate heads, mMTC drops from $R^2 = 0.0830$ to 0.0703 and uRLLC drops from 0.6376 to 0.6268. Best epoch shifts earlier under the shorter window, from 65–67 to 48–57, so shorter sequences converge faster but generalize slightly worse. Because the frequency branch applies RFFT to the same input, this ablation reduces the time-domain context and the frequency resolution available to the model.

5.5. Ablation: Loss Type

Table 3 shows that the loss function is the dominant lever for the two burstier slices. Replacing MSE with the weighted MSE variant ($\alpha = 4$, $q_{\text{low}} = 0.9$, $q_{\text{high}} = 0.99$) yields a clear negative result. On mMTC the test R^2 collapses from $+0.0830$ to -0.0347, with the residual mean shifting from -1.65 to $+6.31$ PRBs, indicating systematic over-prediction induced by the upper-tail upweighting. On uRLLC the test R^2 drops from 0.6376 to 0.5478 and the residual mean shifts from $+1.27$ to $+36.23$ PRBs, with MAPE rising from 38.7% to 57.4%. In contrast, pure Huber loss improves mMTC to $R^2 = 0.3002$ and uRLLC to $R^2 = 0.6624$. The mMTC gain is especially large, reducing MAE by 27% relative to the MSE run and cutting MAPE from 44.30% to 24.71%. For uRLLC, the R^2 gain is smaller but the MAE and MAPE reductions remain substantial. These results support a robust-regression interpretation: the useful modification is not to increase the weight of the upper tail, but to reduce the gradient leverage of large residuals that are poorly explained by the available feature history.

5.6. Anomaly Detection Statistics

Table 4 reports residual statistics and flag rates at $k = 3$ on the test trial under the best completed configuration for each slice. The signed and absolute flag rates increase across the three slices, from 0.30% and 0.39% on eMBB to 3.40% and 3.48% on uRLLC, mirroring the heavier-tailed character of the bursty residual distributions. All observed flag rates are well below the Chebyshev upper bound $1/k^2 \approx 11.1\%$ at

Table 3. Ablation on loss type at sequence length $T = 30$. "w-mse" denotes weighted MSE with weights ramped from 1 above $q_{\text{low}} = 0.9$ to $1 + \alpha$ at $q_{\text{high}} = 0.99$, with $\alpha = 4$; Huber uses $\delta = 1.0$ without weighting.

SLICE	MERGE	LOSS	R^2	MAE	MAPE
MMTC	CONCAT	MSE	0.0830	34.92	44.30
MMTC	CONCAT	W-MSE	−0.0347	40.20	54.79
MMTC	CONCAT	HUBER	0.3002	25.34	24.71
URLLC	CONCAT	MSE	0.6376	76.67	38.70
URLLC	MEAN	MSE	0.6472	74.95	37.26
URLLC	CONCAT	W-MSE	0.5478	97.92	57.38
URLLC	CONCAT	HUBER	0.6624	65.68	27.67

Table 4. Anomaly-detection statistics at $k = 3$ on the test trial under the best configuration for each slice. "signed" denotes the test on the raw residual; "abs" denotes the test on its magnitude.

SLICE	RES. MEAN	RES. STD	SIGNED	ABS
EMBB	−62.42	402.27	0.30%	0.39%
MMTC	−5.23	47.74	2.12%	2.10%
URLLC	−12.40	90.84	3.40%	3.48%

$k = 3$, confirming that the post-hoc test is conservative on this dataset and that $k = 3$ is a usable default. Because the Colosseum dataset carries no anomaly labels, we do not attempt a precision–recall evaluation; the reported flag rate together with the closed-form Chebyshev bound is the operational quantity that downstream schedulers consume.

6. Discussion

The per-slice picture that emerges from the ablation is that the multi-domain architecture is sufficient to capture the smooth eMBB regime to within roughly 14% relative error and the spike-driven uRLLC regime to $R^2 = 0.6624$, while mMTC requires a robust loss to escape the MSE plateau. Under MSE, mMTC remains near $R^2 = 0.0830$ despite changes to sequence length and head merging. Under pure Huber loss, however, mMTC reaches $R^2 = 0.3002$, with a lower residual standard deviation and much lower MAPE. This does not prove that the entire mMTC difficulty is caused by the loss landscape, but it does show that the earlier MSE plateau should not be interpreted as a hard signal ceiling. The more defensible interpretation is that the mMTC target contains a mixture of predictable mid-range structure and poorly predictable spikes, and that robust regression prevents the latter from dominating optimization.

The negative result on weighted MSE is informative for system design. Heuristic upweighting of the upper tail is common for long-tailed targets, but on this dataset the bias of $+6$ to $+36$ PRBs outweighs gains in tail fit. Pure Huber loss shows the opposite behavior: it reduces the leverage of large residuals and improves both average-error metrics and residual dispersion on mMTC and uRLLC. We treat weighted MSE with $\alpha = 4$ as a negative control and recommend validating any tail-aware modification against both bulk metrics and anomaly flag calibration.

7. Limitations

Three limitations of the current evaluation should be made explicit. First, the test set is a single trial `tr15`, which yields point estimates rather than confidence intervals on the per-slice metrics. All reported configurations use seed 42, so additional seeds are needed before claiming robustness. Second, the anomaly-detection evaluation is limited to flag-rate statistics because the dataset carries no anomaly labels. Third, the eMBB validation–test relationship is asymmetric in the mean-merge configuration, where validation loss is lower but test error is higher than under concatenation; this hints at a residual distributional shift between trials `tr10` and `tr15` that is orthogonal to the architectural choice and that would benefit from explicit per-trial diagnosis.

8. Conclusion

This work presents a novel learning architecture for per-slice O-RAN traffic prediction, combining a per-feature-shared BiLSTM, a Graph Attention Network over a feature-as-node graph, and a dual Transformer over the magnitude and phase spectra from an RFFT, with per-node fusion and mean pooling across nodes. On the Colosseum O-RAN dataset, the best configurations achieve $R^2 = 0.9011$ on eMBB, $R^2 = 0.3002$ on mMTC, and $R^2 = 0.6624$ on uRLLC. A controlled ablation shows that a thirty-step history outperforms a fifteen-step history under MSE, that weighted MSE with $\alpha = 4$ degrades performance on bursty slices, and that pure Huber loss is most effective for mMTC and uRLLC. A Chebyshev test calibrated on validation residuals produces a label-free anomaly flag with empirical rates within the closed-form bound at $k = 3$. The combination of stable training, slice-specific loss selection, and calibrated runtime flagging makes the architecture practical for per-slice O-RAN scheduling workflows.

Software and Data

The Colosseum O-RAN dataset is publicly released by the Colosseum project. Source code, training scripts, and result artifacts will be released with the camera-ready version.

Impact Statement

This paper advances machine cearning for next-generation networks with important societal implications. It enables fine-grained, slice-aware traffic prediction in Open RAN, allowing 6G systems to dynamically allocate radio and com-

pute resources across eMBB, mMTC, and uRLLC services. By combining temporal, graph, and spectral learning with robust loss adaptation, it improves reliability under bursty, non-stationary traffic, while a lightweight, training-free anomaly detector supports real-time monitoring without labeled data.

References

Bega, D., Gramaglia, M., Fiore, M., Banchs, A., and Costa-Perez, X. DeepCog: Optimizing resource provisioning in network slicing with AI-based capacity forecasting. *IEEE Journal on Selected Areas in Communications*, 38 (2):361–376, 2020. doi: 10.1109/JSAC.2019.2959245.

Chen, T., Ren, X., Lai, J., Tan, H., Liu, F., and Chan, W. K. V. Toward Transformer-compatible multivariate time series learning via visibility graph-based structural encoding. *Knowledge-Based Systems*, 329:114389, 2025. doi: 10.1016/j.knosys.2025.114389.

Cui, Z., Henrickson, K., Ke, R., and Wang, Y. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2020. doi: 10.1109/TITS.2019.2950416.

Habib, M. A., Rivera, P. E. I., Ozcan, Y., Elsayed, M., Bavand, M., Gaigalas, R., and Erol-Kantarci, M. Transformer-based wireless traffic prediction and network optimization in O-RAN. In *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6. IEEE, 2024. doi: 10.1109/ICCWorkshops59551.2024.10615438.

Huang, B., Dou, H., Luo, Y., Li, J., Wang, J., and Zhou, T. Adaptive spatiotemporal transformer graph network for traffic flow forecasting by IoT loop detectors. *IEEE Internet of Things Journal*, 10(2):1642–1653, 2023. doi: 10.1109/JIOT.2022.3209523.

Jia, X., Qu, J., Lyu, Y., Guo, M., Zhang, J., and Guo, F. A prediction-based anomaly detection method for traffic flow data with multi-domain feature extraction. *Applied Sciences*, 15(6):3234, 2025.

Kablaoui, R., Ahmad, I., Abed, S., and Awad, M. Network traffic prediction by learning time series as images. *Engineering Science and Technology, an International Journal*, 55:101754, 2024. doi: 10.1016/j.jestch.2024.101754.

Lotfi, F. and Afghah, F. Open RAN LSTM traffic prediction and slice management using deep reinforcement learning. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pp. 646–650. IEEE, 2023. doi: 10.1109/IEEECONF59524.2023.10476972.

Polese, M., Bonati, L., D'Oro, S., Basagni, S., and Melodia, T. Colo-ran: Developing machine learning-based xapps for open ran closed-loop control on programmable experimental platforms. *IEEE Transactions on Mobile Computing*, 22(10):5787–5800, 2023a. doi: 10.1109/TMC.2022.3188013.

Polese, M., Bonati, L., D'Oro, S., Basagni, S., and Melodia, T. Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. *IEEE Communications Surveys & Tutorials*, 25(2):1376–1411, 2023b. doi: 10.1109/COMST.2023.3239220.

Shah, S. D. A., Bashir, A. K., Al-Otaibi, Y. D., Al Dabel, M. M., and Ali, F. Dynamic AI-driven network slicing with O-RAN for continuous connectivity in connected vehicles and onboard consumer electronics. *IEEE Transactions on Consumer Electronics*, 71(1):720–733, 2025a. doi: 10.1109/TCE.2025.3527857.

Shah, S. D. A., Hafeez, M., Salama, A., and Zaidi, S. A. R. Proactive AI-and-RAN workload orchestration in O-RAN architectures for 6G networks. *IEEE Open Journal of the Communications Society*, 6:7939–7954, 2025b. doi: 10.1109/OJCOMS.2025.3608700.

Xiao, H., Zou, B., and Xiao, J. Graph convolution networks based on adaptive spatiotemporal attention for traffic flow forecasting. *Scientific Reports*, 15(1):8935, 2025. doi: 10.1038/s41598-025-88706-w.

Yadav, H. and Thakkar, A. TXtreme: transformer-based extreme value prediction framework for time series forecasting. *Discover Applied Sciences*, 7:98, 2025. doi: 10.1007/s42452-025-06478-4.

Zhang, A. Dynamic graph convolutional networks with temporal representation learning for traffic flow prediction. *Scientific Reports*, 15(1):17270, 2025. doi: 10.1038/s41598-025-01696-7.

Zhang, H., Chen, Y., and Niu, X. ACT: anomaly-aware causal transformer with mixture-of-experts for time-series forecasting. *Knowledge-Based Systems*, 341: 115774, 2026. doi: 10.1016/j.knosys.2026.115774.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., and Li, H. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2020. doi: 10.1109/TITS.2019.2935152.

Zheng, H., Lin, F., Feng, X., and Chen, Y. A hybrid deep learning model with attention-based Conv-LSTM networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(11): 6910–6920, 2021. doi: 10.1109/TITS.2020.2997352.

A. Full testings for non-duplicate completed configurations

Table 5 lists all twelve non-duplicate completed configurations on the Colosseum O-RAN dataset under the fixed training configuration described in Section 4. The variable axes are GAT intermediate-layer head merge, sequence length, loss type, and maximum epoch budget.

Table 5. All twelve non-duplicate completed Model configurations on the Colosseum O-RAN dataset.

run	slice	merge	seq	loss	MAE	RMSE	R^2	MAPE	best ep	val loss
embb_20260423	embb	concat	30	mse	250.21	327.32	0.9011	13.66	82	0.02610
embb_20260424	embb	mean	30	mse	259.28	337.50	0.8949	13.98	74	0.02372
mmtc_20260423	mmtc	concat	30	mse	34.92	53.42	0.0830	44.30	67	0.9845
mmtc_20260424	mmtc	mean	30	mse	35.17	53.86	0.0679	44.60	37	1.0077
mmtc_20260425a	mmtc	concat	15	mse	35.03	53.79	0.0703	44.14	48	1.0093
mmtc_20260425b	mmtc	concat	30	weighted mse	40.20	56.74	−0.0347	54.79	44	2.8069
mmtc_20260427	mmtc	concat	30	huber	25.34	46.67	0.3002	24.71	81	0.2166
urllc_20260423	urllc	concat	30	mse	76.67	118.87	0.6376	38.70	65	0.8945
urllc_20260424	urllc	mean	30	mse	74.95	117.27	0.6472	37.26	82	0.8628
urllc_20260425	urllc	concat	15	mse	77.79	120.62	0.6268	39.26	57	0.9231
urllc_20260426	urllc	concat	30	weighted mse	97.92	132.77	0.5478	57.38	65	2.2598
urllc_20260427	urllc	concat	30	huber	65.68	114.72	0.6624	27.67	119	0.2471

B. Fundamental Formulations of The Proposed Traffic Prediction Architecture

B.1. Problem Setup and Target Scaling

This appendix gives the complete mathematical flow used by the per-slice traffic predictor. For a slice $s \in \{\mathrm{eMBB}, \mathrm{mMTC}, \mathrm{uRLLC}\}$, let $x_t^{(s)} \in \mathbb{R}^N$ denote the scaled radio-feature vector at timestep t, where $N = 16$ in the primary configuration. The input window and one-step prediction target are

$$X_t^{(s)} = \left[x_{t-T+1}^{(s)}, x_{t-T+2}^{(s)}, \ldots, x_t^{(s)}\right]^\top \in \mathbb{R}^{T\times N}, \tag{1}$$

$$y_{t+1}^{(s)} \in \mathbb{R}. \tag{2}$$

All trainable mappings below are slice-specific because each slice is trained independently. When RobustScaler normalization is used, the raw target $\tilde{y}_{t+1}^{(s)}$ is mapped to the scaled training target by

$$y_{t+1}^{(s)} = \frac{\tilde{y}_{t+1}^{(s)} - m_y^{(s)}}{r_y^{(s)}}, \tag{3}$$

where $m_y^{(s)}$ and $r_y^{(s)}$ are the training-set median and interquartile range for the target of slice s. Predictions are transformed back to the original PRB scale by the inverse relation $\tilde{\hat{y}}_{t+1}^{(s)} = r_y^{(s)}\hat{y}_{t+1}^{(s)} + m_y^{(s)}$.

B.2. Temporal, Spatial, and Frequency Branches

B.2.1. BiLSTM Temporal Encoder

For each feature node n, let $\chi_{t,n}^{(s)} = (x_{t-T+1,n}^{(s)}, \ldots, x_{t,n}^{(s)})^\top \in \mathbb{R}^T$ denote its length-T trajectory. The temporal branch processes this single-feature sequence with a parameter-shared BiLSTM. After a scalar-to-vector input projection, the BiLSTM produces a final bidirectional state,

$$u_{\tau,n}^{(s)} = W_{\mathrm{tmp,in}} x_{t-T+\tau,n}^{(s)} + b_{\mathrm{tmp,in}}, \qquad \tau = 1, \ldots, T, \tag{4}$$

$$h_n^{\mathrm{tmp}} = \mathrm{BiLSTM}\left(u_{1,n}^{(s)}, \ldots, u_{T,n}^{(s)}\right) \in \mathbb{R}^{2H}, \tag{5}$$

$$= \mathrm{Dropout}\left(\mathrm{ReLU}\left(W_{\mathrm{tmp}} h_n^{\mathrm{tmp}} + b_{\mathrm{tmp}}\right)\right) \in \mathbb{R}^H. \tag{6}$$

Here, $W_{\mathrm{tmp,in}}$ and $b_{\mathrm{tmp,in}}$ are trainable scalar-to-vector input-projection parameters applied at each timestep and shared across timesteps, while W_{tmp} and b_{tmp} are trainable projection parameters that map the final BiLSTM state to the H-dimensional temporal embedding space. This branch therefore learns a temporal embedding for each radio feature while sharing sequence-model parameters across all feature nodes.

B.2.2. GAT Spatial Encoder

For the spatial branch, the same window is interpreted as a graph $G = (\mathcal{V}, \mathcal{E})$ whose nodes are radio features. In the primary setting, $\mathcal{V} = \{1, \ldots, N\}$ and the adjacency is fully connected with self-loops, so $\mathcal{N}(i) = \mathcal{V}$ for every node i. The initial node state is obtained from the full time trajectory of each feature,

$$q_i^{(0)} = \mathrm{Dropout}\left(W_{\mathrm{gat,in}}\chi_{t,i}^{(s)} + b_{\mathrm{gat,in}}\right) \in \mathbb{R}^{D_0}, \qquad D_0 = 32. \tag{7}$$

Here, $W_{\mathrm{gat,in}}$ and $b_{\mathrm{gat,in}}$ are trainable input-projection parameters that map the T-step feature trajectory to the D_0-dimensional GAT input space.

Let $K = 4$ denote the number of attention heads and let L denote the index of the final GAT layer. For GAT layer ℓ and attention head h, the attention score, normalized coefficient, and pre-merge head output are

$$e_{ij}^{(\ell,h)} = \mathrm{LeakyReLU}\left(a_{\ell,h}^{\top}\left[W_{\ell,h}q_i^{(\ell-1)} \| W_{\ell,h}q_j^{(\ell-1)}\right]\right), \tag{8}$$

$$\alpha_{ij}^{(\ell,h)} = \frac{\exp(e_{ij}^{(\ell,h)})}{\sum_{j' \in \mathcal{N}(i)} \exp(e_{ij'}^{(\ell,h)})}, \tag{9}$$

$$\tilde{q}_i^{(\ell,h)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(\ell,h)} W_{\ell,h} q_j^{(\ell-1)} \in \mathbb{R}^{d_\ell}. \tag{10}$$

In these equations, $W_{\ell,h}$ is the trainable feature transformation for head h in layer ℓ, $a_{\ell,h}$ is the corresponding trainable attention vector, j' is a dummy index ranging over all neighbors of node i in the softmax normalization, and d_ℓ is the per-head output dimension. For the non-final GAT layer, the four attention heads are merged either by averaging or by concatenation, corresponding to the head-merge ablation in the experiments:

$$q_i^{(\ell)} = \mathrm{ELU}\left(\left(\frac{1}{K}\sum_{h=1}^{K}\tilde{q}_i^{(\ell,h)}\right) + b_\ell\right) \in \mathbb{R}^H \qquad \text{(mean merge)}, \tag{11}$$

$$q_i^{(\ell)} = \mathrm{ELU}\left(\left(\|_{h=1}^{K}\tilde{q}_i^{(\ell,h)}\right) + b_\ell\right) \in \mathbb{R}^H \qquad \text{(concat merge)}. \tag{12}$$

For mean merge, each head output has dimension $d_\ell = H$; for concat merge, each head output has dimension $d_\ell = H/K$, so the concatenated representation remains H-dimensional. This matches the custom GAT layer used in the experiments, where the specified layer output dimension is the dimension after head merging. The final GAT layer always uses averaging across heads,

$$z_i^{\mathrm{gat}} = \left(\frac{1}{K}\sum_{h=1}^{K}\tilde{q}_i^{(L,h)}\right) + b_L \in \mathbb{R}^H. \tag{13}$$

The vectors b_ℓ and b_L are trainable output biases added after head merging in the custom GAT layer. Thus, the spatial branch returns one H-dimensional embedding for each feature node i.

B.2.3. RFFT-Transformer Frequency Encoder

For the frequency branch, the real-input fast Fourier transform is applied along the time dimension of each node trajectory,

$$C_t^{(s)} = \mathrm{RFFT}_\tau\left(X_t^{(s)}\right) \in \mathbb{C}^{F \times N}, \qquad F = \left\lfloor \frac{T}{2} \right\rfloor + 1. \tag{14}$$

The complex spectrum is decomposed into magnitude and phase,

$$M_t^{(s)} = \left|C_t^{(s)}\right|, \qquad \Phi_t^{(s)} = \arg C_t^{(s)}. \tag{15}$$

For each node n, the magnitude and phase spectra are embedded separately, augmented by learnable frequency-position embeddings p_k, and passed through two Transformer encoders,

$$g_{k,n}^{M} = W_M M_{t,k,n}^{(s)} + p_k, \qquad k = 1, \ldots, F, \tag{16}$$

$$g_{k,n}^{\Phi} = W_\Phi \Phi_{t,k,n}^{(s)} + p_k, \qquad k = 1, \ldots, F. \tag{17}$$

Here, $M_{t,k,n}^{(s)}$ and $\Phi_{t,k,n}^{(s)}$ denote the RFFT magnitude and phase values at frequency bin k for feature node n in window $X_t^{(s)}$, respectively. The parameters W_M and W_Φ are trainable scalar-to-H projections for the magnitude and phase tokens, and $p_k \in \mathbb{R}^H$ is a learnable frequency-position embedding for bin k. Let $\mathbf{g}_n^M = \mathrm{stack}_{k=1}^{F}(g_{k,n}^M) \in \mathbb{R}^{F\times H}$ and $\mathbf{g}_n^\Phi = \mathrm{stack}_{k=1}^{F}(g_{k,n}^\Phi) \in \mathbb{R}^{F\times H}$ denote the magnitude and phase frequency-token sequences for node n. The Transformer outputs are

$$\mathbf{o}_n^M = \left(o_{1,n}^M, \ldots, o_{F,n}^M\right)^\top = \mathrm{Transformer}_M\left(\mathbf{g}_n^M\right), \tag{18}$$

$$\mathbf{o}_n^\Phi = \left(o_{1,n}^\Phi, \ldots, o_{F,n}^\Phi\right)^\top = \mathrm{Transformer}_\Phi\left(\mathbf{g}_n^\Phi\right). \tag{19}$$

The frequency-bin dimension is mean-pooled, and each pathway applies its own output projection before the magnitude and phase pathways are fused,

$$\bar{o}_n^M = \frac{1}{F}\sum_{k=1}^{F} o_{k,n}^M, \qquad \bar{o}_n^\Phi = \frac{1}{F}\sum_{k=1}^{F} o_{k,n}^\Phi, \tag{20}$$

$$\tilde{o}_n^M = \mathrm{Dropout}\left(\mathrm{ReLU}\left(W_M^{\mathrm{out}}\bar{o}_n^M + b_M^{\mathrm{out}}\right)\right) \in \mathbb{R}^H, \tag{21}$$

$$\tilde{o}_n^\Phi = \mathrm{Dropout}\left(\mathrm{ReLU}\left(W_\Phi^{\mathrm{out}}\bar{o}_n^\Phi + b_\Phi^{\mathrm{out}}\right)\right) \in \mathbb{R}^H, \tag{22}$$

$$z_n^{\mathrm{freq}} = \mathrm{Dropout}\left(\mathrm{ReLU}\left(W_{\mathrm{freq}}\left[\tilde{o}_n^M \| \tilde{o}_n^\Phi\right] + b_{\mathrm{freq}}\right)\right) \in \mathbb{R}^H. \tag{23}$$

Here, $\bar{o}_n^M$ and $\bar{o}_n^\Phi$ are the mean-pooled magnitude and phase Transformer outputs over the F frequency bins. The parameters $(W_M^{\mathrm{out}}, b_M^{\mathrm{out}})$ and $(W_\Phi^{\mathrm{out}}, b_\Phi^{\mathrm{out}})$ are independent trainable $H \to H$ output projections for the magnitude and phase pathways, respectively. The parameters W_{freq} and b_{freq} are trainable fusion parameters that map the concatenated projected magnitude–phase representation from $2H$ to the H-dimensional frequency embedding.

B.3. Fusion and Per-Slice Prediction

The prediction head fuses the three branch embeddings at each node and then aggregates node-level predictions into a single window-level forecast:

$$z_n = \mathrm{Dropout}\left(\mathrm{ReLU}\left(W_{\mathrm{fuse}}\left[z_n^{\mathrm{tmp}} \| z_n^{\mathrm{gat}} \| z_n^{\mathrm{freq}}\right] + b_{\mathrm{fuse}}\right)\right), \tag{24}$$

$$\hat{y}_{t+1}^{(s,n)} = w_{\mathrm{out}}^\top z_n + b_{\mathrm{out}}, \tag{25}$$

$$\hat{y}_{t+1}^{(s)} = \frac{1}{N}\sum_{n=1}^{N} \hat{y}_{t+1}^{(s,n)}. \tag{26}$$

Here, z_n is the fused node embedding, W_{fuse} and b_{fuse} are trainable fusion parameters, and w_{out} and b_{out} are the trainable parameters of the scalar node-level prediction head. Thus, the complete traffic-prediction mapping can be written compactly as

$$\hat{y}_{t+1}^{(s)} = f_\theta^{(s)}(X_t^{(s)}) = \frac{1}{N}\sum_{n=1}^{N} \phi_\theta^{(s)}\left(z_n^{\mathrm{tmp}}, z_n^{\mathrm{gat}}, z_n^{\mathrm{freq}}\right), \tag{27}$$

where $\phi_\theta^{(s)}$ denotes the per-node fusion and prediction head, and θ collects all trainable parameters in the slice-specific model.

B.4. Training Objectives

Training minimizes a slice-specific empirical risk over the training windows $\mathcal{D}_{\mathrm{train}}^{(s)}$. For MSE, the objective is

$$\mathcal{L}_{\mathrm{mse}}^{(s)} = \frac{1}{|\mathcal{D}_{\mathrm{train}}^{(s)}|}\sum_{(X_t, y_{t+1})\in\mathcal{D}_{\mathrm{train}}^{(s)}} \left(\hat{y}_{t+1}^{(s)} - y_{t+1}^{(s)}\right)^2. \tag{28}$$

For the weighted-MSE ablation, let $Q_{0.9}^{(s)}$ and $Q_{0.99}^{(s)}$ denote training-target quantiles and define the saturated upper-tail ramp

$$\rho_t^{(s)} = \min\left\{1, \max\left[0, \frac{y_{t+1}^{(s)} - Q_{0.9}^{(s)}}{Q_{0.99}^{(s)} - Q_{0.9}^{(s)}}\right]\right\}, \tag{29}$$

$$\omega_t^{(s)} = 1 + \alpha\rho_t^{(s)}, \qquad \alpha = 4. \tag{30}$$

The sample weight is therefore 1 below $Q_{0.9}^{(s)}$, increases linearly between $Q_{0.9}^{(s)}$ and $Q_{0.99}^{(s)}$, and saturates at $1+\alpha$ above $Q_{0.99}^{(s)}$. The corresponding objective is

$$\mathcal{L}_{\text{wmse}}^{(s)} = \frac{1}{|\mathcal{D}_{\text{train}}^{(s)}|} \sum_{(X_t, y_{t+1}) \in \mathcal{D}_{\text{train}}^{(s)}} \omega_t^{(s)} \left(\hat{y}_{t+1}^{(s)} - y_{t+1}^{(s)}\right)^2. \tag{31}$$

This normalization averages the weighted per-sample losses, matching the implementation's mean reduction rather than renormalizing by the sum of sample weights.

For Huber training, the residual $r_t^{(s)} = \hat{y}_{t+1}^{(s)} - y_{t+1}^{(s)}$ is penalized by

$$\ell_\delta(r_t^{(s)}) = \begin{cases} \frac{1}{2}\left(r_t^{(s)}\right)^2, & |r_t^{(s)}| \le \delta, \\ \delta\left(|r_t^{(s)}| - \frac{1}{2}\delta\right), & |r_t^{(s)}| > \delta, \end{cases} \qquad \delta = 1.0, \tag{32}$$

and the empirical risk is the mean of $\ell_\delta(r_t^{(s)})$ over training windows.

B.5. Post-Hoc Chebyshev Anomaly Flag

After training, validation residuals in the original PRB scale calibrate the label-free anomaly flag. Let

$$\tilde{r}_t^{(s)} = \tilde{\hat{y}}_{t+1}^{(s)} - \tilde{y}_{t+1}^{(s)} \tag{33}$$

denote the inverse-scaled residual, where $\tilde{\hat{y}}_{t+1}^{(s)}$ and $\tilde{y}_{t+1}^{(s)}$ are the predicted and observed granted PRB counts in the original scale. The signed residual statistics are (μ_s, σ_s), and the absolute residual statistics are (μ_a, σ_a). At test time, the signed and absolute binary flags are

$$A_{\text{signed}}(t) = \mathbf{1}\left[\left|\tilde{r}_t^{(s)} - \mu_s\right| > k\sigma_s\right], \tag{34}$$

$$A_{\text{abs}}(t) = \mathbf{1}\left[\left|\tilde{r}_t^{(s)}\right| > \mu_a + k\sigma_a\right], \tag{35}$$

with $k = 3$ in the primary configuration. The flags use strict thresholds, and each strict event is a subset of the corresponding non-strict event used in Chebyshev's inequality. Thus, Chebyshev's inequality gives $\Pr(A_{\text{signed}} = 1) \le 1/k^2$ for the signed residual test, and the same bound applies to the one-sided absolute-residual event because it is contained in the two-sided event $||\tilde{r}_t^{(s)}| - \mu_a| \ge k\sigma_a$.