CTU-FUSION: UTILITY-GUIDED FLOW–TOPOLOGY FUSION FOR ENCRYPTED TRAFFIC CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Strong encrypted traffic classifiers remain constrained by the information within an individual flow. Historical and relational context can provide complementary evidence, but inappropriate fusion can corrupt correct predictions. We propose CTU-Fusion, a utility-guided Flow–Topology framework that separates primary prediction from auxiliary residual correction, estimates the sample-specific benefit and harm of each correction, and selectively controls its influence. Across three traffic-classification settings, CTU-Fusion improves Macro-F1 over the strongest evaluated baseline by 1.12–4.26%. Mapping-aware audits distinguish useful IoT history, negligible CICIDS topology, and severe negative transfer from unreliable USTC timestamps. Component, routing, and robustness studies separate Flow capability, complementary information, and realized fusion gains. These results show that relational evidence should influence a prediction only when its estimated utility is sufficient.

1 INTRODUCTION

Network traffic classification supports application identification, network management, and security monitoring (Velan et al., 2015; Nguyen & Armitage, 2008; Dainotti et al., 2012). In network traffic classification tasks, dynamic ports, tunneling, and encryption restrict the explicit semantics available to port- and payload-based methods (Moore & Papagiannaki, 2005; Moore & Zuev, 2005; Karagiannis et al., 2005; Bernaille et al., 2006; Finsterbusch et al., 2014). Consequently, statistical learning-based methods exploit packet sizes, timing, directions, and aggregate behavior, whereas deep learning-based methods learn representations directly from bytes and traffic sequences (Lotfollahi et al., 2020; Wang et al., 2017a; Aceto et al., 2018; 2019a;b; Liu et al., 2019; Shapira & Shavitt, 2019). Moreover, traffic correlation and fingerprinting extend the observable evidence (Taylor et al., 2016; He et al., 2018; Sirinam et al., 2018), while pretrained and graph-based models strengthen packet- and flow-level representations (Lin et al., 2022; Zhao et al., 2023; Wang et al., 2024; Zhang et al., 2025; Diao et al., 2023; Yu et al., 2024; Li et al., 2025; Okonkwo et al., 2025).

These advances improve the inference capability of single-flow classifiers, yet they cannot recover evidence that is absent from a short or ambiguous observation window. To address this inherent limitation, context-aware methods leverage communication structures and multi-flow relations to improve classification (Klein et al., 2026; Zhao et al., 2025; Nasr et al., 2018). However, Flow and Topology are heterogeneous evidence sources. Flow contains intrinsic observations of the target traffic, whereas the validity of topology depends on historical-data availability, temporal relevance, relation type, and the correspondence between a target sample and its associated context. Consequently, auxiliary context may provide complementary evidence for one sample but become misleading noise that degrades another prediction.

Our auxiliary USTC-TFC2016 experiment exposes this boundary. Its reconstructed timestamps do not support reliable physical-time historical windows. Forcing the resulting topology into the classifier reduces Macro-F1 from the Flow fallback's 95.74% to $71.86 \pm 4.47\%$, an absolute decrease of 23.88%. This result does not establish that topology is generally harmful; instead, it shows why information availability alone cannot justify predictive influence.

To address this problem, we organize our study around three questions: *How strong is the Flow reference? Does topology contain complementary sample-specific information? Can a fusion policy turn that information into reliable gains?* CTU-Fusion preserves a frozen Flow predictor, constructs

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

strictly past relational states, and represents topology as residual logit corrections. An out-of-fold Utility estimator predicts beneficial and harmful transitions, and a calibrated acceptance policy permits a correction only when its estimated utility is sufficient. Rejection retains the original Flow prediction; it never abstains from classification.

In summary, our contributions are as follows:

- We formulate Flow capability, complementary relational information, and fusion effectiveness as separate questions, which makes positive transfer, negligible context, and negative transfer independently measurable.
- We introduce utility-guided residual fusion with a direct Flow fallback. Context-only experts propose corrections, while an out-of-fold Utility estimator decides whether an authorized relation should alter each prediction.
- We evaluate CTU-Fusion on CICIDS2017, IoT-23, and session-level ISCX through mapping, component, degradation, routing, and external-domain studies. Relative to the strongest evaluated baseline, Macro-F1 improves by 2.58%, 4.26%, and 1.12%, respectively.

The three gains arise from different operating modes: CICIDS uses Flow fallback, IoT uses historical Host evidence with Utility routing, and ISCX uses session-set context. This distinction prevents a single aggregate "CTU-Fusion" result from implying that identical relational evidence is available or useful in every dataset.

2 Problem Formulation: When Should We Trust Topology?

For closed-set labels $y_i \in \{1, \dots, C\}$, a flow $e_i = (u_i, v_i, t_i, x_i)$ contains endpoints, arrival time, and intrinsic observations. Historical state $\mathcal{H}_{<t_i}$ includes only evidence available before prediction. The protocol queries this state, predicts, and then updates it; neither future traffic nor current ground-truth labels are available at inference. Endpoint identities are state keys, not learned predictive embeddings.

The frozen Flow predictor gives class scores $z_i^{\text{flow}} = f_F(x_i)$ and prediction $\hat{y}_i^{\text{flow}} = \arg\max_c z_{i,c}^{\text{flow}}$. Each authorized relation $k \in \mathcal{K}$ extracts context $c_{i,k} = \phi_k(\mathcal{H}_{<t_i}, u_i, v_i)$ and a context-only expert proposes $\Delta_{i,k} = M_k(c_{i,k})$. The Flow scores and reliability features are not inputs to M_k. They meet the residual only at the additive candidate anchor

$$z_{i,k} = z_i^{\text{flow}} + \beta \Delta_{i,k}, \tag{1}$$

where β is the policy-controlled correction scale and $\hat{y}_{i,k} = \arg\max_c z_{i,k,c}$ is the candidate prediction. Its effect is measured by

$$b_{i,k} = \mathbb{I}[\hat{y}_i^{\text{flow}} \neq y_i \ \wedge \ \hat{y}_{i,k} = y_i], \qquad h_{i,k} = \mathbb{I}[\hat{y}_i^{\text{flow}} = y_i \ \wedge \ \hat{y}_{i,k} \neq y_i], \tag{2}$$

$$g_{i,k} = \text{CE}(y_i, z_i^{\text{flow}}) - \text{CE}(y_i, z_{i,k}), \tag{3}$$

where $b_{i,k}$ records a wrong-to-correct transition, $h_{i,k}$ records a correct-to-wrong transition, and $g_{i,k}$ measures the reduction in cross-entropy. These quantities use labels only for supervision or evaluation. At inference a Utility estimator predicts these quantities from observable signals. Selecting the best relation k_i^* is distinct from accepting it:

$$z_i^{\text{final}} = z_i^{\text{flow}} + a_i \beta \Delta_{i,k_i^*}, \qquad a_i \in \{0, 1\}, \tag{4}$$

where k_i^* is the selected relation and a_i is the binary acceptance indicator. Together, a_i and β regulate auxiliary influence. This differs from selective prediction or deferral (Geifman & El-Yaniv, 2017; 2019; Mozannar & Sontag, 2020): every sample receives a class prediction, and $a_i = 0$ preserves the Flow decision.

This formulation makes the three evaluation questions operational. Flow-only performance measures the reference capability before relational evidence is introduced. Observed-versus-permuted context audits then test whether a relation contains sample-specific information while holding the model and marginal context distribution fixed. Finally, the difference between an always-applied

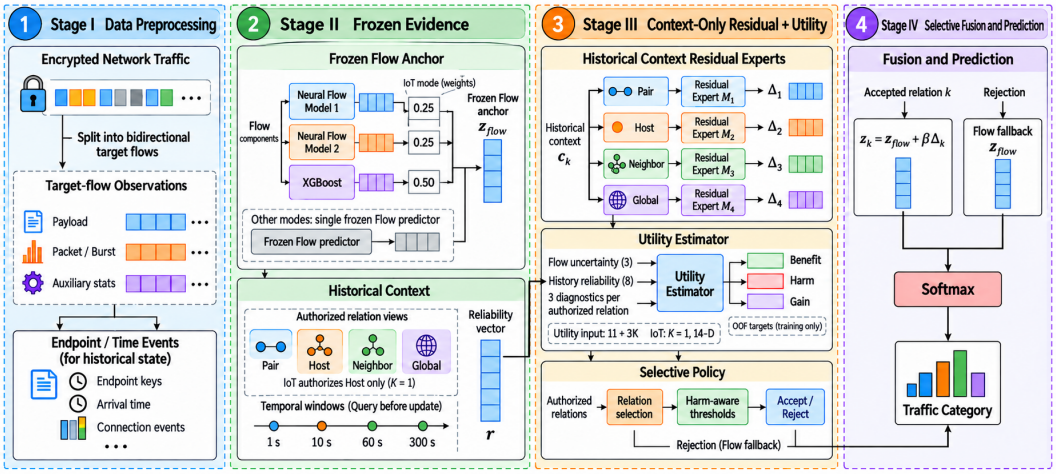

Figure 1: Overall Framework of CTU-Fusion. Context-only experts compute $\Delta_{i,k} = M_k(c_{i,k})$ without Flow scores or reliability as residual inputs. The frozen Flow anchor participates only through $z_{i,k} = z_i^{\text{flow}} + \beta\Delta_{i,k}$. Utility uses $11 + 3K$ observable inputs (IoT: Host only, $K = 1$, 14-D), and rejection retains the Flow anchor. The IoT anchor is the frozen $0.25/0.25/0.50$ ensemble shown; OOF targets require labels only during training.

residual and Utility-controlled fusion measures whether the routing policy converts available information into a reliable prediction gain. Keeping these quantities separate avoids attributing a strong Flow model to topology or treating the mere presence of historical features as evidence that they should be used.

3 CTU-Fusion

Overview. Figure 1 organizes seven modules into preprocessing, separated evidence construction, correction and selection, and prediction. The frozen Flow path provides a common reference for attributing every subsequent change to topology. Appendix A supplies the expanded derivation.

3.1 Preprocessing and Flow Reference

Bidirectional-flow preprocessing separates target-flow observations from events used to maintain history. The Flow branch encodes payload fragments, packet-and-burst rhythm, and auxiliary statistics into h_i^P, h_i^R, h_i^A. Transformer encoders model ordered byte and sequence inputs (Vaswani et al., 2017); auxiliary statistics use an MLP. Modality gates and aggregation give

$$g_i^m = \sigma\big(w_{2,m}^\top \,\mathrm{GELU}(W_{1,m} h_i^m)\big), \quad h_i^F = f_{\text{agg}}(\{g_i^m h_i^m\}_{m=P,R,A}), \quad z_i^{\text{flow}} = W_c h_i^F + b_c, \tag{5}$$

where $m \in \{P, R, A\}$ indexes payload, rhythm, and auxiliary modalities, and g_i^m is the corresponding availability-aware gate. LayerNorm and GELU follow standard definitions (Ba et al., 2016; Hendrycks & Gimpel, 2016). After standalone training, the Flow predictor is frozen during residual and Utility learning. In the frozen IoT implementation, the Flow anchor is an ensemble of two neural Flow predictors and XGBoost on the common label space,

$$z_i^{\text{flow}} = 0.25 z_i^{N_1} + 0.25 z_i^{N_2} + 0.50 z_i^{\text{XGB}}, \tag{6}$$

where N_1 and N_2 are the two neural Flow predictors and XGB denotes XGBoost. The other operating modes use their stated single frozen Flow predictor.

3.2 Historical Topology and Residual Proposals

Pair, Host, Neighbor, and Global scopes summarize endpoint-pair interactions, endpoint activity, communication neighborhoods, and network background. Windows $\mathcal{W} = \{1, 10, 60, 300\}$ seconds

produce $X_i^T \in \mathbb{R}^{4\times 64}$: the first dimension indexes windows, and each window contains $16 + 24 + 12 + 12$ dimensions for the four scopes. An eight-dimensional reliability vector r_i describes availability, maturity, capacity/eviction, and cold start.

For each authorized relation, projected window states $u_{i,w,k} = \mathrm{GELU}(\mathrm{LN}(W_k x_{i,w,k}^T + b_k))$ are combined using context-only attention:

$$\omega_{i,w,k} = \mathrm{softmax}_w\, A_k(u_{i,w,k}), \qquad c_{i,k} = \sum_w \omega_{i,w,k} u_{i,w,k}, \tag{7}$$

$$\Delta_{i,k} = M_k(c_{i,k}), \tag{8}$$

where A_k scores temporal windows, $c_{i,k}$ is the aggregated context for relation k, and M_k maps that context to the residual proposal. Residual training minimizes candidate cross-entropy with an ℓ_2 correction penalty. Neither z_i^{flow} nor r_i enters the expert; z_i^{flow} is used only in Equation 1, where policy scale β controls the proposed change. Relation authorization is dataset specific; the final IoT model authorizes Host only, so $K = 1$.

3.3 Out-of-fold Utility estimation and selective fusion

Purged five-fold residual predictions construct Utility targets without fitting the corresponding residual model on the target sample, following the meta-learning principle of stacked generalization (Wolpert, 1992). With K authorized relations, the Utility input has $11 + 3K$ dimensions: Flow entropy, top-two margin and maximum probability; eight history-reliability values; and three diagnostics per authorized relation (residual ℓ_2 norm, infinity norm, and prediction-change indicator). IoT authorizes only Host, hence $K = 1$ and the actual input is 14-dimensional. It predicts $\hat{b}_{i,k}, \hat{h}_{i,k}, \hat{g}_{i,k}$ and ranks proposals by

$$q_{i,k} = \hat{b}_{i,k} - 2\hat{h}_{i,k}, \qquad k_i^* = \arg\max_{k\in\mathcal{K}} q_{i,k}, \tag{9}$$

where $q_{i,k}$ penalizes estimated harm twice as strongly as estimated benefit and k_i^* is the highest-utility authorized relation. Predicted-class thresholds $T_{\hat{y}_i^{\mathrm{flow}}}$ and scale β are calibrated separately from Utility training. Available context is accepted when $q_{i,k_i^*} \geq T_{\hat{y}_i^{\mathrm{flow}}}$ under a policy selected using a Wilson upper confidence bound on accepted-sample harm (Wilson, 1927). This is an empirical operating criterion, not a distribution-free guarantee (Bates et al., 2021; Angelopoulos et al., 2024). Inference applies Equation 4, returns $\arg\max_c z_{i,c}^{\mathrm{final}}$, then updates history. Context-use coverage is the fraction with $a_i = 1$; all samples still receive predictions.

4 Experiments

We evaluate overall performance (RQ1), Flow and component contributions (RQ2), sample-specific relational information (RQ3), Utility routing (RQ4), and robustness to degraded or shifted information (RQ5).

4.1 Experimental setup

Table 1: Dataset Statistics and Evaluation Protocol. Counts refer to the stated prediction unit. USTC has a separate 92,849-flow calibration partition.

Dataset / role	Unit	Classes	Train-fit	Selection	Test
CICIDS2017 / main	Flow	5	1,132,816	165,465	205,845
IoT-23 / main	Flow	6	544,333	124,566	119,783
ISCX / main	Session	4	64	24	23
USTC / auxiliary	Flow	20	557,112	139,284	139,278

Final modes: CICIDS, Flow fallback; IoT, Host residual + Utility; ISCX, session-attention; USTC, Flow fallback with a forced-topology diagnostic. Split isolation: campaign blocks and purge (CICIDS); within-scenario chronology (IoT); indivisible captures/paired halves (ISCX); within-class/capture session blocks (USTC).

CICIDS2017, IoT-23, and ISCX VPN-nonVPN are the main tasks (Sharafaldin et al., 2018; Garcia et al., 2020; Draper-Gil et al., 2016); USTC is an auxiliary boundary study (Wang et al., 2017b). CICIDS groups complete 600-second campaign blocks and purges split boundaries by 120 seconds. IoT splits chronologically within 23 scenarios and uses a development view designed for class coverage and computational feasibility, rather than the full natural class distribution. ISCX keeps captures and paired recording halves indivisible: each of 111 session bags receives one prediction, and its 19,168 member flows are not independently labeled evaluation units. The test set contains only 23 sessions.

We compare RF (Breiman, 2001), XGBoost (Chen & Guestrin, 2016), 1D-CNN (Wang et al., 2017a), ET-BERT (Lin et al., 2022), MH-Net (Zhang et al., 2025), YaTC (Zhao et al., 2023), NetMamba (Wang et al., 2024), and TrafficScope (Zhao et al., 2025). Adapted models share frozen label spaces, splits, and prediction units, but do not reproduce every original pretraining corpus or pipeline. Macro-F1 is primary. CTU neural results use seeds $\{11, 22, 33, 44, 55\}$ and sample standard deviation; baseline point estimates support descriptive, not significance, comparisons.

Model selection and early stopping use the model-selection partition. The final partition is called test. For the separate Utility-policy analysis, a stable hash divides this partition into policy-fit and disjoint policy-audit halves; only policy-fit selects coverage, β, and thresholds. Half-split audit results are reported separately from full-partition main results. Because the full partition contains policy-fit, its scores are not a separate holdout audit of the fitted policy.

The neural Flow and context components use width 256, eight heads, dropout 0.15, first-five-packet payloads of up to 256 bytes each, 32 packet steps, 20 bursts, and 40 auxiliary dimensions. Normalization is fitted only on training data. Trainable neural components use AdamW (Loshchilov & Hutter, 2019). Historical state is queried before update. ISCX's offline session-set context is a distinct relational setting, not strictly past endpoint-time topology.

All comparisons preserve the dataset-specific prediction unit and frozen split. The mapping-aware audits change only the correspondence between a sample and its context; the model, Flow scores, labels, and context marginals remain fixed. Robustness panels use their own frozen configurations and internal clean anchors, so their curves quantify degradation within a panel rather than re-estimating the main-table operating point. This protocol supports descriptive attribution across heterogeneous settings while avoiding claims of statistical significance from unmatched baseline seeds.

4.2 Overall Classification Performance (RQ1)

Table 2: Main Classification Performance (Macro-F1, %). Baselines are adapted frozen local reproductions reported as point estimates; CTU is mean ± sample standard deviation over five runs, so comparisons are descriptive rather than significance claims. CTU operating modes are CICIDS Flow fallback, IoT Host + Utility, and ISCX session-set context. ISCX metrics are session-level (23 test sessions), and seed variation is not independent-sample uncertainty.

Method	CICIDS2017	IoT-23	ISCX
RF	94.36	74.89	84.84
XGBoost	96.01	78.65	86.15
1D-CNN	94.23	71.86	85.45
ET-BERT adapted	95.78	75.11	89.35
MH-Net adapted	96.51	79.37	90.75
YaTC adapted	96.06	76.34	90.94
NetMamba adapted	96.24	77.58	90.15
TrafficScope adapted	97.02	81.54	92.07
CTU-Fusion	**99.60** ± 0.08	**85.80** ± 3.14	**93.19** ± 3.45

Table 2 shows absolute improvements of 2.58%, 4.26%, and 1.12% over the strongest baseline on CICIDS, IoT, and ISCX. CICIDS's $99.60 \pm 0.08\%$ comes from Flow fallback and must not be attributed to topology. IoT reaches $85.80 \pm 3.14\%$, while ISCX reaches $93.19 \pm 3.45\%$ at session level. Expanded Accuracy, Precision, and Recall results are in Appendix B.

4.3 Flow capability and component contribution (RQ2)

Table 3: Component and Flow-Modality Ablation (Macro-F1, %). (a) Components; the label-aware oracle is diagnostic only. (b) IoT Flow modalities, with standard deviations where supplied.

(a) Component decomposition

Configuration	CICIDS	IoT	ISCX
Flow-only	99.60	79.29	72.19
Auxiliary-context-only	99.40	81.50	93.19
Naive average fusion	99.45	80.80	87.50
Always residual	99.61	84.30	91.00
Confidence-only router	99.60	81.90	91.80
Utility w/o harm constraint	99.61	84.20	92.20
Full CTU	99.60	85.79	93.19
Label-aware oracle	99.66	87.00	96.00

(b) IoT modalities

Configuration	Macro-F1
Payload	70.80 ± 1.80
Rhythm	64.20 ± 2.20
Auxiliary	72.80 ± 1.70
Payload + Rhythm	74.80 ± 1.40
Payload + Auxiliary	76.40 ± 1.10
Rhythm + Auxiliary	74.00 ± 1.50
All three	79.29

On IoT, Flow-only achieves 79.29%, always-residual 84.30%, and Full CTU 85.79% Macro-F1; the label-aware oracle reaches 87.00%. Thus additional features alone do not explain the gain: routing matters. CICIDS is already near saturation, so final CTU retains Flow fallback. ISCX's mean-pooled Flow head achieves 72.19% at the same session-level prediction unit, compared with 93.19% for session-attention; naive average fusion achieves only 87.50%. IoT's three Flow modalities are complementary (Table 3b).

4.4 Sample-specific relational information (RQ3)

Mapping-aware audits fix models, Flow predictions, labels, and sample order while changing only sample-to-context correspondence. Figure 2(a) reports the relation-selection audit: observed Host reaches 83.66% versus 81.50% under Host permutation, an absolute difference of 2.16%. A separate frozen strong-residual counterfactual audit gives 84.51% versus 82.41%, an absolute difference of 2.10%; its paired hierarchical bootstrap 95% interval is approximately [1.96%, 2.23%] (Efron, 1979). Zero history in that counterfactual audit yields 72.19%. CICIDS Pair's observed-minus-permuted gain is below 0.01%, which is also below the 0.10% practical threshold. USTC's invalid timestamps produce an absolute loss of 23.88% under forced fusion. Together, these useful, negligible, and harmful regimes motivate explicit relation authorization.

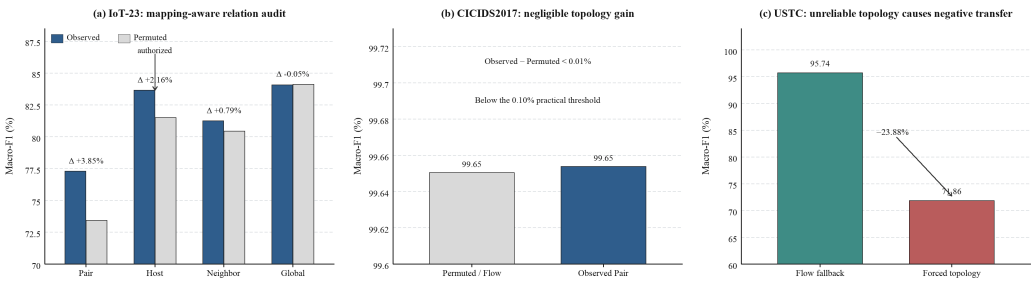

Figure 2: Additional Relational Information Is Not Uniformly Useful. Panel (a) is the IoT relation-selection audit (Host: 83.66% vs. 81.50%, $+2.16\%$), distinct from the frozen strong-residual counterfactual audit reported in the text. Panels (b)–(c) show negligible CICIDS gain and USTC negative transfer.

4.5 Utility-guided selective fusion (RQ4)

On IoT seed-11 policy-fit, increasing requested coverage from 1.00% to 20.00% raises the absolute Macro-F1 gain from about 0.41% to 5.44%; 50.00% reaches 5.89%. Raising requested coverage to 75.00% produces identical argmax predictions, so the lower-coverage point is selected. On disjoint

policy-audit, 29,765 of 59,719 samples are accepted, with 4,785 wrong-to-correct and 224 correct-to-wrong transitions. Accepted-sample harm is 0.75%, its Wilson UCB95 is 0.86%, and Macro-F1 rises from 77.77% to 83.54% (+5.77%). These half-split results quantify routing separately from the full-test comparison; they are not ranked directly against the full-test main result.

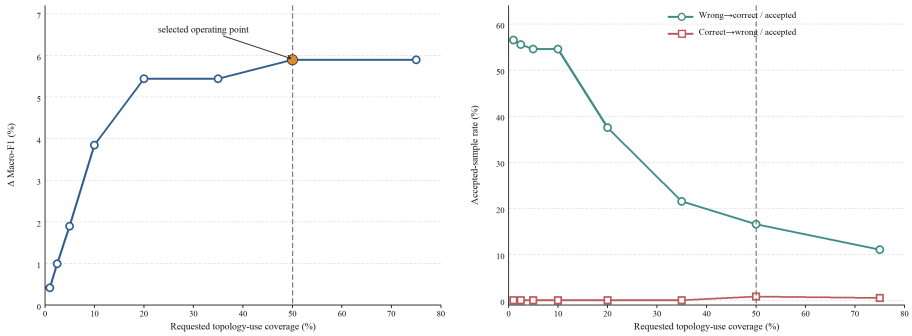

Figure 3: Utility-Guided Routing. Curves describe policy-fit; the text reports the disjoint policy-audit. Subfigure captions are placed below their corresponding panels.

4.6 Robustness under information degradation (RQ5)

At 50.00% IoT packet loss, Flow decreases from 79.29% to 63.80%, while Full CTU decreases from 85.79% to 73.90% (Figure 4(a)). Under 70.00% history dropout, CTU achieves 76.80% versus 65.00% for always-fuse; on ISCX the corresponding scores are 73.50% and 62.00%. At 100.00% incorrect IoT mapping, always-fuse falls to 60.00% and CTU to 77.30%, while topology-use coverage drops to approximately 2.00–5.00%. Selectivity reduces degradation but does not ensure no harm. In the separate cold-start setting, IoT rises from 77.30% without history to 82.40% with eight observations and 85.79% with full history; ISCX rises from 72.20% to 90.20% and 93.20%. Each panel uses its dedicated frozen robustness configuration. Its clean anchor is internal to that panel and is not a cross-panel ablation baseline; for example, a robustness clean anchor may be 84.50%, while Table 3 reports 84.30% for always-residual, and the IoT zero-history value 77.30% is not the 79.29% main Flow result.

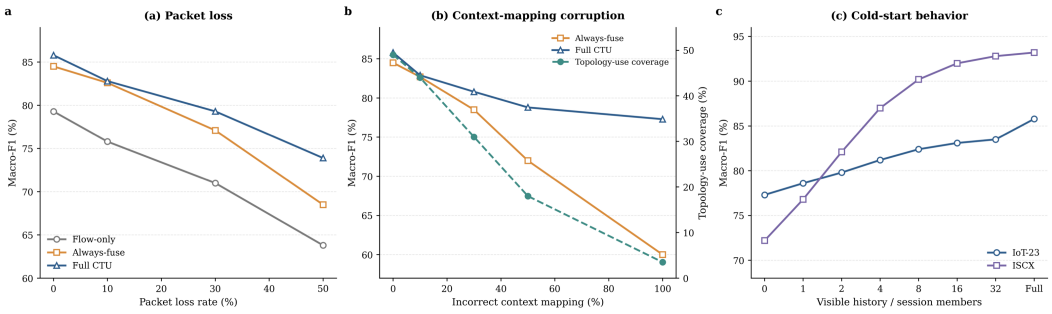

Figure 4: Robustness under Information Degradation. IoT packet loss (a), mapping corruption with context-use coverage on the right axis (b), and cold start on IoT and ISCX (c). Each panel uses a dedicated frozen robustness configuration; clean anchors are panel-specific and are not cross-panel ablation baselines.

4.7 External-domain generalization

Table 4: External-Domain and Scenario-Held-Out Robustness. Weights and routing policies are frozen before external evaluation. F1 values are percentages; ECE is unitless.

Dataset	Model	External F1	Weighted F1	ECE	Worst F1
UNSW	Flow-only	82.00 ± 2.50	88.40 ± 1.90	0.08	76.20
	CTU w/o Utility	84.70 ± 2.20	90.10 ± 1.70	0.07	79.50
	Full CTU	86.00 ± 2.00	91.20 ± 1.50	0.05	82.00
CTU-13	Flow-only	84.00 ± 2.80	89.50 ± 2.10	0.10	78.40
	CTU w/o Utility	86.50 ± 2.40	91.00 ± 1.80	0.08	81.60
	Full CTU	88.00 ± 2.10	92.20 ± 1.60	0.06	84.10

On UNSW-NB15 and scenario-held-out CTU-13 (Moustafa & Slay, 2015; Garcia et al., 2014), Full CTU improves external Macro-F1 by 4.00% over Flow. ECE decreases from 0.08 to 0.05 and from 0.10 to 0.06, respectively. Worst-class F1 improves from 76.20% to 82.00% and from 78.40% to 84.10%. These results support the evaluated shifts, not arbitrary deployment robustness.

5 Related Work

Flow representations. Statistical classifiers use observable flow summaries and ensemble decision rules (Velan et al., 2015; Nguyen & Armitage, 2008; Breiman, 2001; Chen & Guestrin, 2016). Multimodal models combine heterogeneous views of the same traffic observation (Lotfollahi et al., 2020; Aceto et al., 2019b; Liu et al., 2019). Pretrained encoders transfer packet or datagram representations across tasks: ET-BERT uses contextualized datagrams, YaTC uses masked reconstruction, and NetMamba uses state-space sequence modeling (Lin et al., 2022; Zhao et al., 2023; Wang et al., 2024; Gu & Dao, 2024). These methods strengthen the reference against which an auxiliary correction should be measured. CTU-Fusion treats that reference as fixed during relational learning, so improvements from Flow capacity and contextual information can be analyzed separately.

Graphs and cross-flow context. Graph convolutions and attention provide tools for aggregating structured evidence (Kipf & Welling, 2017; Veličković et al., 2018). Traffic models encode multi-scale, chronological, heterogeneous, or higher-order relations (Zhang et al., 2025; Diao et al., 2023; Yu et al., 2024; Li et al., 2025; Okonkwo et al., 2025; Klein et al., 2026); context-aware classification and traffic correlation also exploit information spanning flows (Zhao et al., 2025; Nasr et al., 2018). CTU-Fusion adds an explicit decision about whether such evidence should alter a particular Flow prediction. Its mapping-aware audit tests sample-to-context correspondence, while its residual interface exposes both beneficial and harmful changes.

Selective decisions and calibration. Selective classification and deferral control whether to predict or consult an expert (Geifman & El-Yaniv, 2017; 2019; Mozannar & Sontag, 2020); CTU-Fusion instead accepts or rejects a correction while predicting every sample. Calibration methods improve the interpretation of confidence scores (Guo et al., 2017; Platt, 1999; Niculescu-Mizil & Caruana, 2005), and ensembles or learned confidence support uncertainty estimation (Lakshminarayanan et al., 2017; Hendrycks & Gimpel, 2017; DeVries & Taylor, 2018). A confident Flow prediction can nevertheless admit a useful correction, so our Utility targets incremental benefit and harm rather than confidence alone. Class-imbalance objectives address a different training concern (Lin et al., 2017; Cui et al., 2019; Menon et al., 2021). Unlike risk-controlling or conformal methods (Bates et al., 2021; Angelopoulos et al., 2024), Wilson-bound calibration (Wilson, 1927) provides no distribution-free guarantee. OOF supervision and bootstrap diagnostics connect to stacking and resampling (Wolpert, 1992; Efron, 1979).

6 Discussion and Limitations

Additional information and useful information are not equivalent. Explicit source separation makes correction benefit, harm, and unrealized headroom observable; CICIDS fallback is therefore an

intended outcome rather than a failed use of topology. The mapping-aware audit further shows why a high context-only score is insufficient: a relation can have a useful marginal distribution without being correctly aligned to the target sample. The relevant question is whether the observed correspondence adds information beyond that preserved by permutation.

Utility routing introduces its own dependency. It requires a competent frozen Flow reference, representative out-of-fold residual proposals, and enough beneficial and harmful transitions to estimate sample-level utility. Rare transitions make Utility targets imbalanced, while distribution shift can invalidate thresholds selected on policy-fit. The Wilson-bound criterion is an empirical operating rule for the observed calibration partition; neither it nor the residual scale provides a formal safety guarantee. Rejection only restores the frozen Flow prediction, so the quality of that fallback remains the lower-level performance constraint.

The datasets differ in prediction unit and protocol. IoT is a designed development view with within-scenario chronological splits, ISCX has only 23 test sessions, and CICIDS is near saturation. Seed variability is not uncertainty over new sessions, and adapted local baselines permit descriptive comparisons rather than claims of statistical significance. USTC does not validate physical-time topology; it provides a boundary case in which timestamp semantics are inadequate for historical windows. Query-before-update enforces temporal availability, not causal identification, and persistent endpoint state can still encode environmental shortcuts.

The policy analysis also has a specific evaluation limitation. Policy-fit is drawn from the final test partition, so the full-test classification score is not an untouched holdout estimate of the fitted routing policy. We therefore report full-test main classification results separately from the disjoint policy-audit diagnostic and do not rank half-split audit values against full-test scores. A stronger future protocol would fit all coverage, scale, and threshold choices on validation or on a dedicated calibration partition before a single evaluation on untouched test data. The current model remains closed set, uses predefined relations and windows, and incurs state-maintenance overhead.

7 Conclusion

CTU-Fusion separates Flow prediction from relational correction and estimates when auxiliary evidence should influence the decision. Across the evaluated traffic settings, topology is complementary, negligible, or harmful. Utility-guided selection exploits useful corrections while preserving a direct Flow fallback, making both gains and remaining failure modes measurable.

AI Use Statement

Generative AI tools assisted manuscript organization and drafting, language editing, methodological and experimental-design feedback, code development and implementation, result interpretation, figure and table preparation, and LaTeX conversion. The authors reviewed the AI-assisted text and code and are responsible for the final manuscript, implementation, experimental results, and scientific claims.

Reproducibility Statement

Sections 3–4 and Appendix A specify the model, prediction units, partition sizes, inputs, random seeds, and routing audit. The frozen implementation uses context-only residual experts, a $11 + 3K$ Utility input (IoT: Host only, $K = 1$, 14 dimensions), and an IoT Flow anchor combining two neural predictors and XGBoost with weights $0.25/0.25/0.50$; neural components use AdamW. Appendix B provides expanded metrics. The evaluation uses published traffic datasets (Sharafaldin et al., 2018; Garcia et al., 2020; Draper-Gil et al., 2016; Wang et al., 2017b; Moustafa & Slay, 2015; Garcia et al., 2014) and connection-level processing conventions related to Bro/Zeek (Paxson, 1999). Baseline adaptation, the 23-session ISCX limitation, and the distinction between full-test reporting and disjoint policy-audit diagnostics are disclosed. Because policy-fit is drawn from the final test partition, the full-test result is treated as main classification reporting rather than a separate holdout audit.

Ethics Statement

This work studies classification and security analysis of existing traffic datasets. Encrypted traffic classification can expose behavioral information even without plaintext. Deployment should respect dataset terms, access controls, and applicable privacy requirements; the experiments do not establish suitability for identifying individuals or for unrestricted surveillance.

References

G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè. Mobile encrypted traffic classification using deep learning. In *2018 IEEE/ACM Network Traffic Measurement and Analysis Conference (TMA)*, pp. 1–8, 2018. doi: 10.23919/TMA.2018.8506558.

G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE Transactions on Network and Service Management*, 16(2):445–458, 2019a. doi: 10.1109/TNSM.2019.2899085.

G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè. MIMETIC: Mobile encrypted traffic classification using multimodal deep learning. *Computer Networks*, 165:106944, 2019b. doi: 10.1016/j.comnet.2019.106944.

A. N. Angelopoulos, S. Bates, A. Fisch, L. Lei, and T. Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024.

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

S. Bates, A. N. Angelopoulos, L. Lei, J. Malik, and M. I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021.

L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review*, 36(2):23–26, 2006. doi: 10.1145/1129582.1129589.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. doi: 10.1023/A:1010933404324.

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of ACM SIGKDD*, pp. 785–794, 2016. doi: 10.1145/2939672.2939785.

Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of CVPR*, 2019.

A. Dainotti, A. Pescapè, and K. C. Claffy. Issues and future directions in traffic classification. *IEEE Network*, 26(1):35–40, 2012. doi: 10.1109/MNET.2012.6135854.

T. DeVries and G. W. Taylor. Learning confidence for out-of-distribution detection in neural networks. arXiv preprint arXiv:1802.04865, 2018.

Z. Diao, G. Xie, X. Wang, R. Ren, X. Meng, G. Zhang, K. Xie, and M. Qiao. EC-GCN: A encrypted traffic classification framework based on multi-scale graph convolution networks. *Computer Networks*, 224:109614, 2023. doi: 10.1016/j.comnet.2023.109614.

G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani. Characterization of encrypted and VPN traffic using time-related features. In *Proceedings of ICISSP*, pp. 407–414, 2016. doi: 10.5220/0005740704070414.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

M. Finsterbusch, C. Richter, E. Rocha, J.-A. Müller, and K. Hänßgen. A survey of payload-based traffic classification approaches. *IEEE Communications Surveys & Tutorials*, 16(2):1135–1156, 2014. doi: 10.1109/SURV.2013.100613.00161.

S. Garcia, M. Grill, J. Stiborek, and A. Zunino. An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123, 2014. doi: 10.1016/j.cose.2014.05.011.

S. Garcia, A. Parmisano, and M. J. Erquiaga. IoT-23: A labeled dataset with malicious and benign IoT network traffic. Zenodo, version 1.0.0, 2020.

Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 4878–4887, 2017.

Y. Geifman and R. El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *Proceedings of ICML*, volume 97, pp. 2151–2159, 2019.

A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of ICML*, volume 70, pp. 1321–1330, 2017.

G. He, B. Xu, L. Zhang, and H. Zhu. Mobile app identification for encrypted network flows by traffic correlation. *International Journal of Distributed Sensor Networks*, 14(12), 2018. doi: 10.1177/1550147718817292.

D. Hendrycks and K. Gimpel. Gaussian error linear units (GELUs). arXiv preprint arXiv:1606.08415, 2016.

D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel traffic classification in the dark. In *Proceedings of ACM SIGCOMM*, pp. 229–240, 2005.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

M. Klein, R. Marbel, C. Hajaj, R. Dubin, and A. Dvir. GraphMux: A graph-based framework for encrypted traffic classification. *Computer Networks*, 285:112370, 2026. doi: 10.1016/j.comnet.2026.112370.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Z. Li, H. Zhao, J. Zhao, Y. Jiang, and F. Bu. SAT-Net: A staggered attention network using graph neural networks for encrypted traffic classification. *Journal of Network and Computer Applications*, 233:104069, 2025. doi: 10.1016/j.jnca.2024.104069.

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of ICCV*, 2017.

X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In *Proceedings of the ACM Web Conference 2022*, pp. 633–642, 2022. doi: 10.1145/3485447.3512217.

C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li. FS-Net: A flow sequence network for encrypted traffic classification. In *IEEE INFOCOM 2019 – IEEE Conference on Computer Communications*, pp. 1171–1179, 2019. doi: 10.1109/INFOCOM.2019.8737507.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Computing*, 24(3):1999–2012, 2020. doi: 10.1007/s00500-019-04030-2.

A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

A. W. Moore and K. Papagiannaki. Toward the accurate identification of network applications. In *Passive and Active Network Measurement (PAM)*, pp. 41–54, 2005. doi: 10.1007/978-3-540-31966-5_4.

A. W. Moore and D. Zuev. Internet traffic classification using Bayesian analysis techniques. In *Proceedings of ACM SIGMETRICS*, pp. 50–60, 2005. doi: 10.1145/1064212.1064220.

N. Moustafa and J. Slay. UNSW-NB15: A comprehensive data set for network intrusion detection systems. In *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, 2015. doi: 10.1109/MilCIS.2015.7348942.

H. Mozannar and D. Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of ICML*, volume 119, pp. 7076–7087, 2020.

M. Nasr, A. Bahramali, and A. Houmansadr. DeepCorr: Strong flow correlation attacks on Tor using deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1962–1976, 2018. doi: 10.1145/3243734.3243824.

T. T. T. Nguyen and G. Armitage. A survey of techniques for Internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, 10(4):56–76, 2008. doi: 10.1109/SURV.2008.080406.

A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proceedings of ICML*, pp. 625–632, 2005.

Z. Okonkwo, E. Foo, Z. Hou, Q. Li, and Z. Jadidi. A graph representation framework for encrypted network traffic classification. *Computers & Security*, 148:104134, 2025. doi: 10.1016/j.cose.2024.104134.

V. Paxson. Bro: A system for detecting network intruders in real-time. *Computer Networks*, 31 (23–24):2435–2463, 1999. doi: 10.1016/S1389-1286(99)00112-7.

J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pp. 61–74, 1999.

T. Shapira and Y. Shavitt. FlowPic: Encrypted Internet traffic classification is as easy as image recognition. In *IEEE INFOCOM Workshops*, pp. 680–687, 2019. doi: 10.1109/INFCOMW.2019.8845315.

I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of ICISSP*, pp. 108–116, 2018. doi: 10.5220/0006639801080116.

P. Sirinam, M. Imani, M. Juarez, and M. Wright. Deep Fingerprinting: Undermining website fingerprinting defenses with deep learning. In *Proceedings of ACM CCS*, 2018.

V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic. AppScanner: Automatic fingerprinting of smartphone apps from encrypted network traffic. In *2016 IEEE European Symposium on Security and Privacy*, 2016. doi: 10.1109/EuroSP.2016.40.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

P. Velan, M. Čermák, P. Čeleda, and M. Drašar. A survey of methods for encrypted traffic classification and analysis. *International Journal of Network Management*, 25(5):355–374, 2015. doi: 10.1002/nem.1901.

P. Veličković et al. Graph attention networks. In *International Conference on Learning Representations*, 2018.

T. Wang, X. Xie, W. Wang, C. Wang, Y. Zhao, and Y. Cui. NetMamba: Efficient network traffic classification via pre-training unidirectional Mamba. In *2024 IEEE 32nd International Conference on Network Protocols (ICNP)*, pp. 1–11, 2024. doi: 10.1109/ICNP61940.2024.10858569.

W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 43–48, 2017a. doi: 10.1109/ISI.2017.8004872.

W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng. Malware traffic classification using convolutional neural network for representation learning. In *2017 International Conference on Information Networking (ICOIN)*, pp. 712–717, 2017b. doi: 10.1109/ICOIN.2017.7899588.

E. B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

R. Yu, X. Guo, P. Zhang, and K. Zhang. HGNN-ETC: Higher-order graph neural network based on chronological relationships for encrypted traffic classification. *Computers, Materials & Continua*, 81(2):2643–2664, 2024. doi: 10.32604/cmc.2024.056165.

H. Zhang, H. Yue, X. Xiao, L. Yu, Q. Li, Z. Ling, and Y. Zhang. Revolutionizing encrypted traffic classification with MH-Net: A multi-view heterogeneous graph model. In *Proceedings of AAAI*, volume 39, pp. 1048–1056, 2025. doi: 10.1609/aaai.v39i1.32091.

R. Zhao, M. Zhan, X. Deng, Y. Wang, Y. Wang, G. Gui, and Z. Xue. Yet Another Traffic Classifier: A masked autoencoder based traffic transformer with multi-level flow representation. In *Proceedings of AAAI*, volume 37, pp. 5420–5427, 2023. doi: 10.1609/aaai.v37i4.25674.

Z. Zhao, Z. Song, X. Xie, Z. Li, J. Yu, F. Zhang, and T. Li. Towards context-aware traffic classification via time-wavelet fusion network. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2025. doi: 10.1145/3690624.3709315.

A Expanded Method Derivation

A.1 Data Preprocessing

Given an encrypted network traffic stream, we first partition packets into bidirectional flows and order the resulting flows chronologically. For a target flow e_i, the preprocessing stage extracts two groups of information that are deliberately kept separate.

The first group contains observations intrinsic to the target flow, including payload bytes, packet-length sequences, timing and directional information, burst-level behavior, and auxiliary flow statistics. These observations constitute x_i in Section 2 and are used exclusively by the Flow branch.

The second group contains event-level information required to maintain historical relational states, including communicating endpoints, timestamps, and interaction statistics. These events are used to update the historical memory after prediction, but are not directly concatenated with the Flow representation.

This separation is maintained throughout CTU-Fusion. Flow-level observations determine the standalone prediction, whereas historical events provide auxiliary relational evidence that may subsequently propose a correction to that prediction.

A.2 Flow Encoding: Establishing a Strong Flow Reference

The Flow branch implements the Flow predictor f_F defined in Section 2. Its purpose is to establish a strong standalone prediction before any historical topology is introduced. We model three complementary sources of information within the target flow: **payload**, **packet-and-burst rhythm**, and **auxiliary statistics**.

Payload representation The payload encoder captures local byte-level patterns from the observed payload fragments. Payload bytes from the first packets of the target flow are partitioned into ordered patches and embedded as a token sequence. Packet- and patch-level positional information is incorporated before the sequence is processed by a Transformer encoder (Vaswani et al., 2017), producing

$$h_i^P = f_P(x_i^P),$$

where $h_i^P \in \mathbb{R}^d$ denotes the payload representation.

Packet-and-burst rhythm representation Payload semantics alone may not fully characterize encrypted traffic. We therefore model temporal traffic behavior at both packet and burst levels. The packet sequence contains observable characteristics such as direction, packet length, inter-arrival time, TCP-state information, and relative position. The burst sequence aggregates temporally consecutive packets into higher-level descriptors such as burst direction, packet count, byte volume, duration, and inter-burst timing.

The two temporal sequences are encoded separately and combined to obtain

$$h_i^R = f_R(x_i^R),$$

where h_i^R captures the rhythm of the target flow at complementary temporal resolutions.

Auxiliary representation We additionally construct a compact auxiliary feature vector containing protocol characteristics, port categories, basic traffic statistics, packet-length and timing statistics, traffic ratios, and entropy-related descriptors. An MLP maps these features to

$$h_i^A = f_A(x_i^A),$$

where h_i^A is the auxiliary-statistics representation. The three modalities may differ in availability and informativeness across flows. We therefore introduce **modality reliability gating**; the normalization

and activation components follow standard LayerNorm and GELU formulations (Ba et al., 2016; Hendrycks & Gimpel, 2016):

$$g_i^m = \sigma\left(w_{2,m}^\top \,\mathrm{GELU}\left(W_{1,m} h_i^m\right)\right), \qquad m \in \{P, R, A\},$$

where m indexes payload, rhythm, and auxiliary modalities and g_i^m is the corresponding gate. After accounting for modality availability, the gated representations are treated as modality tokens and aggregated to obtain the Flow representation

$$h_i^F = f_{\mathrm{agg}}\left(g_i^P h_i^P, g_i^R h_i^R, g_i^A h_i^A\right),$$

where f_{agg} aggregates the gated modality tokens. The Flow class scores and probabilities are then

$$z_i^{\mathrm{flow}} = W_c h_i^F + b_c, \qquad p_i^{\mathrm{flow}} = \mathrm{softmax}(z_i^{\mathrm{flow}}),$$

with Flow-only prediction

$$\hat{y}_i^{\mathrm{flow}} = \arg\max_c z_{i,c}^{\mathrm{flow}}.$$

After standalone Flow training, the Flow predictor is frozen during subsequent context modeling. In the frozen IoT implementation, the actual Flow anchor combines the stored class-score outputs of two neural Flow predictors and XGBoost:

$$z_i^{\mathrm{flow}} = 0.25 z_i^{N_1} + 0.25 z_i^{N_2} + 0.50 z_i^{\mathrm{XGB}},$$

where N_1 and N_2 are neural Flow predictors and XGB denotes XGBoost. The other operating modes use their stated single frozen Flow predictor. Consequently, z_i^{flow} remains a fixed **Flow anchor**, so any subsequent change can be explicitly attributed to auxiliary context rather than to joint adaptation of the Flow representation.

A.3 Historical Topology Construction

The topology branch constructs relational evidence from network events that occurred strictly before the target flow. For flow e_i arriving at t_i, CTU-Fusion follows the query-before-update protocol defined in Section 2:

$$\mathcal{H}_{<t_i} \rightarrow \hat{y}_i \rightarrow \mathcal{H}_{\leq t_i}.$$

The historical state is therefore queried before e_i is inserted into memory. This prevents the current event and future traffic from leaking into its topology representation.

We characterize historical topology through four relational scopes,

$$\mathcal{S} = \{\mathrm{pair}, \mathrm{host}, \mathrm{neighbor}, \mathrm{global}\}.$$

Pair-level statistics describe previous interactions between the communicating endpoints. Host-level statistics summarize historical activity associated with the endpoints. Neighbor-level statistics characterize their recent communication neighborhoods, while global statistics summarize broader network activity.

Because relational behavior may manifest over different temporal horizons, each relation is maintained over

$$\mathcal{W} = \{1, 10, 60, 300\}\ \mathrm{s}.$$

For each target flow, CTU-Fusion constructs

$$X_i^T \in \mathbb{R}^{4\times 64},$$

where the first dimension corresponds to the four temporal windows rather than the four relation types. Within each window, the 64-dimensional topology state is partitioned as

$$64 = 16_{\text{pair}} + 24_{\text{host}} + 12_{\text{neighbor}} + 12_{\text{global}}.$$

Historical evidence is not equally mature for every target flow. An established endpoint pair may have rich history, whereas a newly observed pair may have little or no relational evidence. We therefore construct an additional history-reliability vector

$$r_i \in \mathbb{R}^8,$$

where r_i summarizes observable properties of the historical state, including availability, state maturity, capacity or eviction behavior, and cold-start conditions.

The topology construction module can therefore be written as

$$(X_i^T, r_i) = \Phi(\mathcal{H}_{<t_i}, u_i, v_i),$$

where Φ denotes the query of the strictly historical state. Endpoint identifiers are used only as keys for retrieving and updating historical state; they are not embedded as predictive features.

A.4 Relation-Specific Context-Only Residual Experts

Section 2 treats Flow and context as heterogeneous evidence sources. CTU-Fusion therefore avoids directly concatenating historical context with the Flow representation. Each authorized relation instead supplies context to a residual expert. The frozen implementation is *context only*: neither the Flow anchor nor the history-reliability vector is an input to the expert.

For authorized relation $k \in \mathcal{K}$ and temporal window $w \in \mathcal{W}$, the corresponding context statistics are projected into a latent representation:

$$u_{i,w,k} = \text{GELU}\left(\text{LN}\left(W_k x_{i,w,k}^T + b_k\right)\right),$$

where W_k and b_k are relation-specific projection parameters. Different temporal windows may be informative for different samples. We aggregate them using attention computed only from relation context:

$$\omega_{i,w,k} = \text{softmax}_w\left(A_k(u_{i,w,k})\right), \qquad c_{i,k} = \sum_{w\in\mathcal{W}} \omega_{i,w,k} u_{i,w,k},$$

where A_k scores temporal windows and $c_{i,k}$ is their context-only aggregation. The relation expert receives only the aggregated context:

$$\Delta_{i,k} = M_k(c_{i,k}),$$

where M_k maps the aggregated relation context to a residual proposal. The fixed Flow anchor participates only when a candidate is formed with policy scale β:

$$z_{i,k} = z_i^{\text{flow}} + \beta\Delta_{i,k},$$

where β is the policy-controlled correction scale. Thus, authorized relations do not independently replace the Flow classifier. They generate context-only correction proposals $\{\Delta_{i,k} : k \in \mathcal{K}\}$, whose incremental effects can be evaluated relative to exactly the same frozen Flow anchor. Relation

authorization is dataset specific. The final IoT model authorizes only Host, so $\mathcal{K} = \{\text{host}\}$ and $K = 1$.

With the Flow branch frozen, the residual experts are optimized by

$$\mathcal{L}_{\text{res}} = \frac{1}{N|\mathcal{K}|} \sum_{i=1}^{N} \sum_{k \in \mathcal{K}} \left[\text{CE}\left(y_i, z_i^{\text{flow}} + \Delta_{i,k}\right) + \lambda_{\text{res}} \|\Delta_{i,k}\|_2^2 \right],$$

where N is the number of residual-training samples and λ_{res} controls correction regularization. The residual-fitting objective uses the unit-scale candidate (equivalently, $\beta = 1$ during residual fitting); policy-fit selects the inference-time β only after residual and Utility training. The residual regularization discourages unnecessary modifications. The policy scale and acceptance rule limit how the proposal is used, but do not provide a formal no-harm guarantee.

A.5 Out-of-Fold Utility Estimation

The existence of a potentially useful topology correction does not imply that the correction is reliable for every sample. The same relation can repair a Flow error in one case and corrupt an already correct Flow prediction in another. CTU-Fusion therefore explicitly estimates the **sample-specific utility** of each residual proposal.

The Utility module is supervised using the benefit, harm, and cross-entropy gain targets introduced in Section 2. A direct construction of these targets from residual experts evaluated on their own training samples would, however, overestimate their usefulness. We address this issue through **purged five-fold out-of-fold (OOF) residual prediction** during Utility training, following the general principle of generating meta-level supervision from predictions produced without fitting on the corresponding sample (Wolpert, 1992).

For each fold, the residual experts are trained without the samples in that fold and subsequently used to generate their topology corrections. Therefore, every residual correction used to construct Utility supervision is produced by a model that has not been optimized on the corresponding sample. The OOF procedure is used only to construct out-of-fold Utility-training targets; once trained, the Utility estimator operates normally during inference.

At inference time, ground-truth labels are unavailable. Utility must therefore be inferred exclusively from observable quantities. We characterize uncertainty in the frozen Flow prediction using

$$v_i^F = \left[H(p_i^{\text{flow}}),\ p_{i,(1)}^{\text{flow}} - p_{i,(2)}^{\text{flow}},\ p_{i,(1)}^{\text{flow}} \right],$$

where $H(\cdot)$ denotes predictive entropy and the subscripts (1) and (2) identify the largest and second-largest Flow probabilities.

For every authorized relation k, we additionally extract residual diagnostics

$$d_{i,k} = \left[\|\Delta_{i,k}\|_2,\ \|\Delta_{i,k}\|_\infty,\ \mathbb{I}(\hat{y}_{i,k} \neq \hat{y}_i^{\text{flow}}) \right],$$

where $d_{i,k}$ contains the residual norms and the prediction-change indicator. Together with the history-reliability vector, the Utility input is

$$v_i = [v_i^F; r_i; d_{i,k_1}; \ldots; d_{i,k_K}] \in \mathbb{R}^{11+3K},$$

where the fixed part has three Flow-uncertainty features and eight reliability features; each authorized relation contributes three diagnostics. IoT authorizes only Host, so $K = 1$ and the deployed Utility input is 14-dimensional. Reliability is observable to Utility but is not an input to the residual expert.

The Utility estimator predicts the three outcomes for each authorized relation:

$$\{\hat{b}_{i,k}, \hat{h}_{i,k}, \hat{g}_{i,k}\}_{k \in \mathcal{K}} = \psi(v_i),$$

where $\hat{b}_{i,k}$ estimates beneficial correction likelihood, $\hat{h}_{i,k}$ estimates harmful correction likelihood, and $\hat{g}_{i,k}$ estimates continuous cross-entropy gain. Relation selection emphasizes the asymmetric consequences of benefit and harm:

$$q_{i,k} = \hat{b}_{i,k} - 2\hat{h}_{i,k}, \qquad k_i^* = \arg\max_{k \in \mathcal{K}} q_{i,k},$$

where the coefficient 2 assigns greater cost to a predicted harmful intervention than to an equally likely beneficial correction. Importantly, k_i^* identifies only the best authorized proposal; it does not imply that context should necessarily be used.

A.6 Utility-Guided Selective Fusion

The final fusion stage determines whether the selected context proposal is sufficiently trustworthy to modify the Flow anchor. This distinction between **relation selection** and **correction acceptance** is central to CTU-Fusion: even the highest-scoring relation can be rejected when its estimated utility is insufficient.

We use the policy-fit calibration partition (Section 4.1) to determine predicted-class-specific utility thresholds

$$T_c,$$

together with a correction scaling factor β. For target flow e_i, the selected topology correction is eligible for fusion only if

$$q_{i,k_i^*} \geq T_{\hat{y}_i^{\text{flow}}}.$$

The calibration procedure additionally tracks correct-to-wrong transitions among accepted topology corrections. Let UCB_{harm} denote the 95% Wilson upper confidence bound of the empirical harmful-correction rate. The operating policy is selected subject to

$$\text{UCB}_{\text{harm}} \leq \rho,$$

where ρ is a predefined calibration tolerance.

This harm-aware constraint is used as an empirical calibration criterion rather than as a formal distribution-free guarantee (Bates et al., 2021; Angelopoulos et al., 2024; Wilson, 1927).

The resulting context-use decision is

$$a_i \in \{0, 1\}.$$

The final class scores are

$$z_i^{\text{final}} = z_i^{\text{flow}} + a_i \beta \Delta_{i,k_i^*},$$

where a_i is the binary context-acceptance decision. When the correction is rejected,

$$a_i = 0,$$

and consequently

$$z_i^{\text{final}} = z_i^{\text{flow}}.$$

Thus, CTU-Fusion preserves a direct Flow fallback path. Rejecting context does not reject the sample or introduce an unknown class; it simply retains the original Flow prediction. The proportion of

samples for which $a_i = 1$ is reported as **context-use coverage**, which is distinct from conventional classification coverage because every sample receives a prediction.

The available final partition is also the source of the policy-fit/policy-audit split. Policy-fit labels select coverage, β, and thresholds; the disjoint policy-audit half is reserved for the routing diagnostic. Consequently, the full-test main classification score overlaps policy-fit and is reported separately from the half-split routing audit; it is not treated as a separate holdout audit.

A.7 Prediction and Staged Optimization

The final traffic category is obtained as

$$\hat{y}_i = \arg\max_c z_{i,c}^{\text{final}},$$

where c indexes the closed-set traffic classes. The model is optimized in stages to preserve the explicit Flow–context separation established in Sections 1 and 2. The Flow components are first trained independently using only target-flow observations and are subsequently frozen; the IoT class-score anchor is then fixed as the $0.25/0.25/0.50$ ensemble described above. Historical states are constructed under the query-before-update protocol, and context-only residual experts are trained for the authorized relations. Purged OOF residual predictions then supervise the Utility estimator, after which policy-fit determines the selective-fusion thresholds and correction scaling factor. Trainable neural components use AdamW (Loshchilov & Hutter, 2019).

During online inference, these components operate sequentially but require no ground-truth labels. For each incoming target flow, CTU-Fusion obtains the frozen Flow anchor z_i^{flow} and queries the historical state $\mathcal{H}_{<t_i}$. Each authorized expert generates $\Delta_{i,k}$ from context alone. Utility then predicts benefit and harm from Flow uncertainty, reliability, and residual diagnostics. The highest-scoring authorized relation is passed to the calibrated acceptance policy. Its correction is applied only as $z_i^{\text{flow}} + \beta\Delta_{i,k}$ when accepted; otherwise, z_i^{flow} is retained unchanged. Only after prediction is the current event incorporated into the historical state.

This staged architecture directly implements the formulation of Section 2. The Flow branch establishes what can be predicted from the target flow alone; the context-only residual experts expose complementary information; the Utility estimator predicts whether a proposal is likely to improve or harm the existing decision; and selective fusion determines whether auxiliary context receives decision authority. CTU-Fusion therefore treats historical or session-set context as conditional evidence whose influence must be justified relative to a strong frozen Flow anchor.

B EXPANDED CLASSIFICATION METRICS

Table 5: Expanded main classification results (%). AC is Accuracy; P, R, and F1 are macro-averaged Precision, Recall, and F1. Baselines are adapted local reproductions; CTU-Fusion reports mean ± sample standard deviation over five seeds.

CICIDS2017 Method	AC	P	R	F1
RF	96.08	94.93	94.05	94.36
XGBoost	97.17	96.45	95.71	96.01
1D-CNN	95.74	94.81	93.91	94.23
ET-BERT adapted	96.51	96.31	95.39	95.78
MH-Net adapted	97.07	96.91	96.18	96.51
YaTC adapted	96.75	96.59	95.65	96.06
NetMamba adapted	96.88	96.65	95.92	96.24
TrafficScope adapted	97.43	97.48	96.69	97.02
CTU-Fusion	99.81 ± 0.02	99.85 ± 0.13	99.36 ± 0.03	99.60 ± 0.08

IoT-23 Method	AC	P	R	F1
RF	89.40	81.67	75.98	74.89
XGBoost	90.65	85.24	79.63	78.65
1D-CNN	86.80	77.94	72.72	71.86
ET-BERT adapted	88.90	83.18	76.71	75.11
MH-Net adapted	90.71	87.93	80.25	79.37
YaTC adapted	89.36	83.86	77.46	76.34
NetMamba adapted	90.14	85.65	78.63	77.58
TrafficScope adapted	90.95	87.12	82.35	81.54
CTU-Fusion	93.35 ± 2.20	89.65 ± 2.01	86.78 ± 5.12	85.80 ± 3.14

ISCX VPN-nonVPN (session level) Method	AC	P	R	F1
RF	83.48	84.95	86.25	84.84
XGBoost	85.22	86.55	87.68	86.15
1D-CNN	84.35	85.88	86.96	85.45
ET-BERT adapted	88.70	89.74	90.63	89.35
MH-Net adapted	89.57	91.65	91.96	90.75
YaTC adapted	89.57	91.31	92.14	90.94
NetMamba adapted	89.57	90.45	91.52	90.15
TrafficScope adapted	91.30	92.40	93.30	92.07
CTU-Fusion	92.17 ± 3.64	93.07 ± 3.64	94.11 ± 2.83	93.19 ± 3.45

Table 6: Expanded IoT Flow-modality ablation (%). W-F1 denotes weighted F1. Values reproduce the supplied ablation table, including its reported precision.

Configuration	Accuracy	Macro-F1	W-F1
Payload only	75.80 ± 1.30	70.80 ± 1.80	75.00 ± 1.40
Rhythm only	68.40 ± 1.70	64.20 ± 2.20	67.10 ± 1.90
Aux only	74.60 ± 1.50	72.80 ± 1.70	74.00 ± 1.50
Payload + Rhythm	77.00 ± 1.10	74.80 ± 1.40	76.50 ± 1.20
Payload + Aux	78.40 ± 0.90	76.40 ± 1.10	78.00 ± 1.00
Rhythm + Aux	75.90 ± 1.20	74.00 ± 1.50	75.40 ± 1.30
All three	79.00 ± 0.80	79.29	78.50 ± 0.80

Table 7: In-domain reference Macro-F1 for the external evaluations in Table 4 (%).

Dataset	Model	In-domain F1
UNSW	Flow-only	90.50 ± 1.50
UNSW	CTU w/o Utility	92.20 ± 1.30
UNSW	**Full CTU**	92.00 ± 1.20
CTU-13	Flow-only	93.00 ± 1.10
CTU-13	CTU w/o Utility	94.10 ± 1.00
CTU-13	**Full CTU**	94.00 ± 0.90