



From Encrypted Flows to Security Event Signals

An Edge-Native Pipeline for Threat-Aware Traffic Analysis

Yu-Lun Tsai · Yu-Liang Cheng · Yu-Ying Huang · Ching-Hao Mao

Fu Jen Catholic University | International Cybersecurity Talent Development Association



Research Background



HTTPS / TLS

dominates

the modern edge

*Payloads are encrypted.
Defenders see only flow artifacts.*



Bulk decryption is rarely viable

Policy, regulation, or operational cost makes payload inspection infeasible at the edge.



Classifiers stop at a label

An argmax or score vector cannot drive a SOC playbook or replay an incident.



No replayable, auditable evidence

SIEM / SOAR pipelines need IDs, context, calibrated scores, and provenance.

The Core Idea

WHAT PRIOR WORK PRODUCES

argmax label
or score vector

```
{  
  "label": "malicious",  
  "score": 0.87  
}
```

WHAT THIS PAPER EMITS

security event signal

replayable · schema-stable · semantically meaningful

```
{  
  event_id, flow_id, run_id,  
  src_ip, dst_ip, ports, protocol,  
  stage1_score, dispatch_reason,  
  stage2_label, threat_behavior,  
  tactic, technique, // ATT&CK  
  schema_version, model_version,  
  checksum, replay_id  
}
```

Treat encrypted-flow inference as an *event-production problem*, not a leaderboard task.

Architecture: One Pipeline, Four Layers

L0 · Observations

Telemetry, gateways, mirror ports, public + lab traces

L1 · Canonicalization

5-tuple, timing, TLS hints, coverage audit

L2 · Two-Stage Inference

Stage 1 screen → Stage 2 behavior (gated)

L3 · Event Signal

IDs, context, scores, TTP, provenance

L2 DETAIL — TWO-STAGE CASCADE

Stage 1 · ET-BERT

Calibrated benign / malicious screening

$\bar{p}_{mal} \geq \tau ?$

frozen threshold

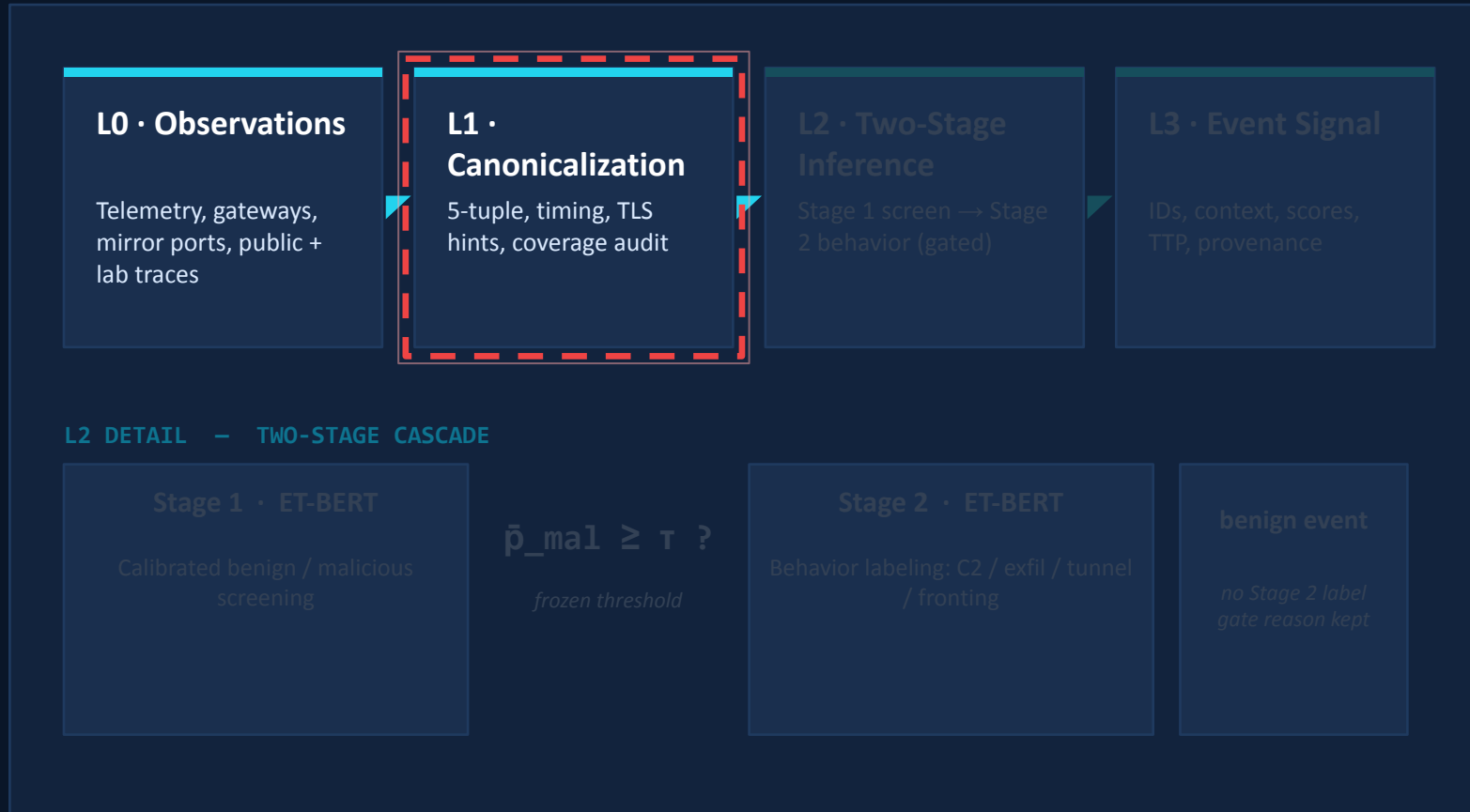
Stage 2 · ET-BERT

Behavior labeling: C2 / exfil / tunnel / fronting

benign event

*no Stage 2 label
gate reason kept*

Step 1 — Protocol-Agnostic Canonicalization



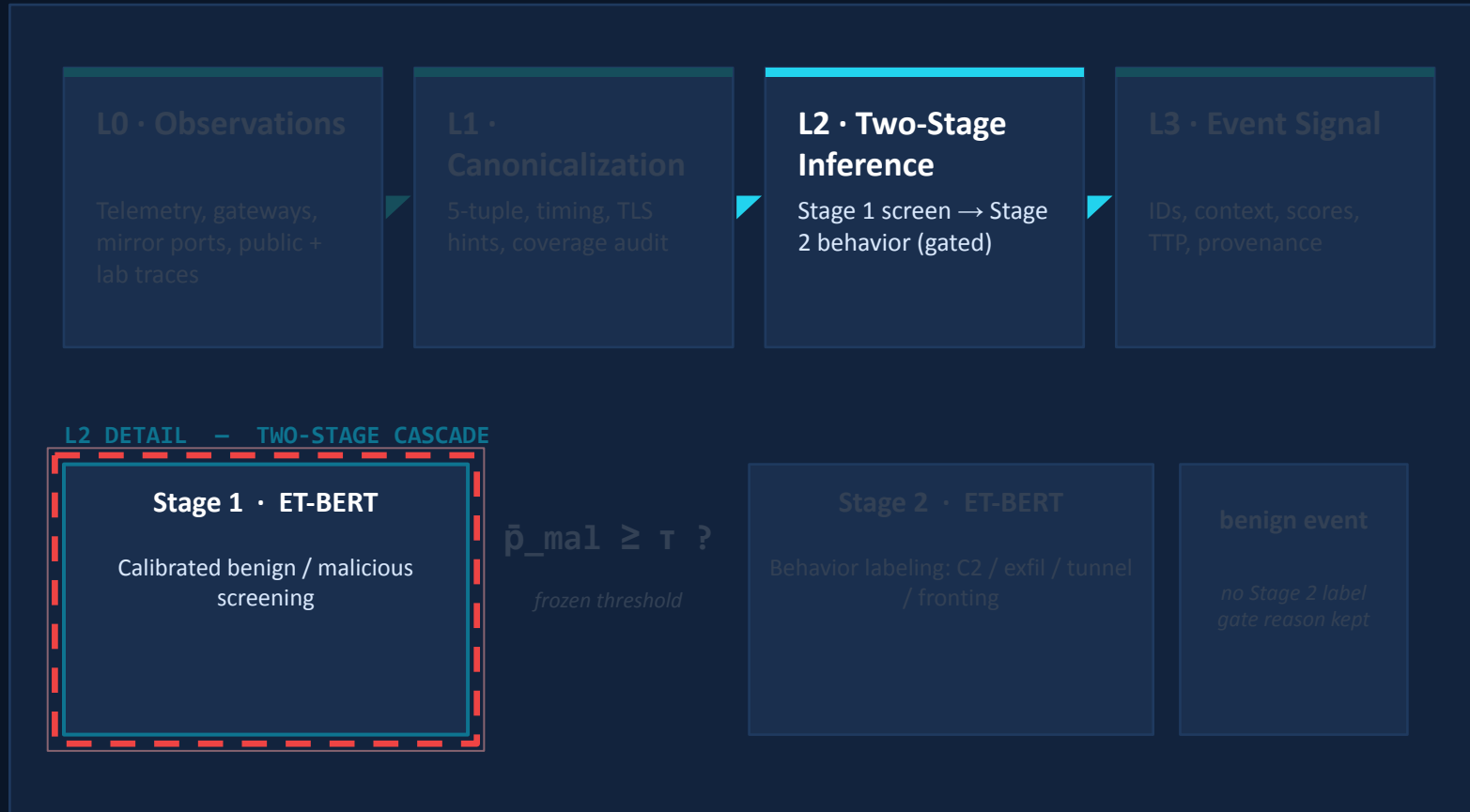
WHY THIS MATTERS

Each flow is mapped into a **protocol-agnostic canonical form**:

- direction tokens
- log-size buckets
- log-IAT buckets
- coarse transport / TLS hints

Coverage gate: if canonical-token overlap < 0.5, flag *adaptation_required* — representation mismatch is **surfaced, not hidden**.

Step 2 — Stage 1: Calibrated Screening



STAGE 1 · THE DISPATCH GATE

ET-BERT outputs a malicious probability $\bar{p}$.

Development-only calibration maps logits to $\bar{p}$ on a held-out split, **disjoint** from OOD and case-study sets.

Class imbalance handled via reweighting / balanced sampling.

Goal: *reliable thresholded dispatch* — not leaderboard accuracy.

Step 3 — One Threshold, Three Phases



THE KEY INVARIANT

A flow escalates iff

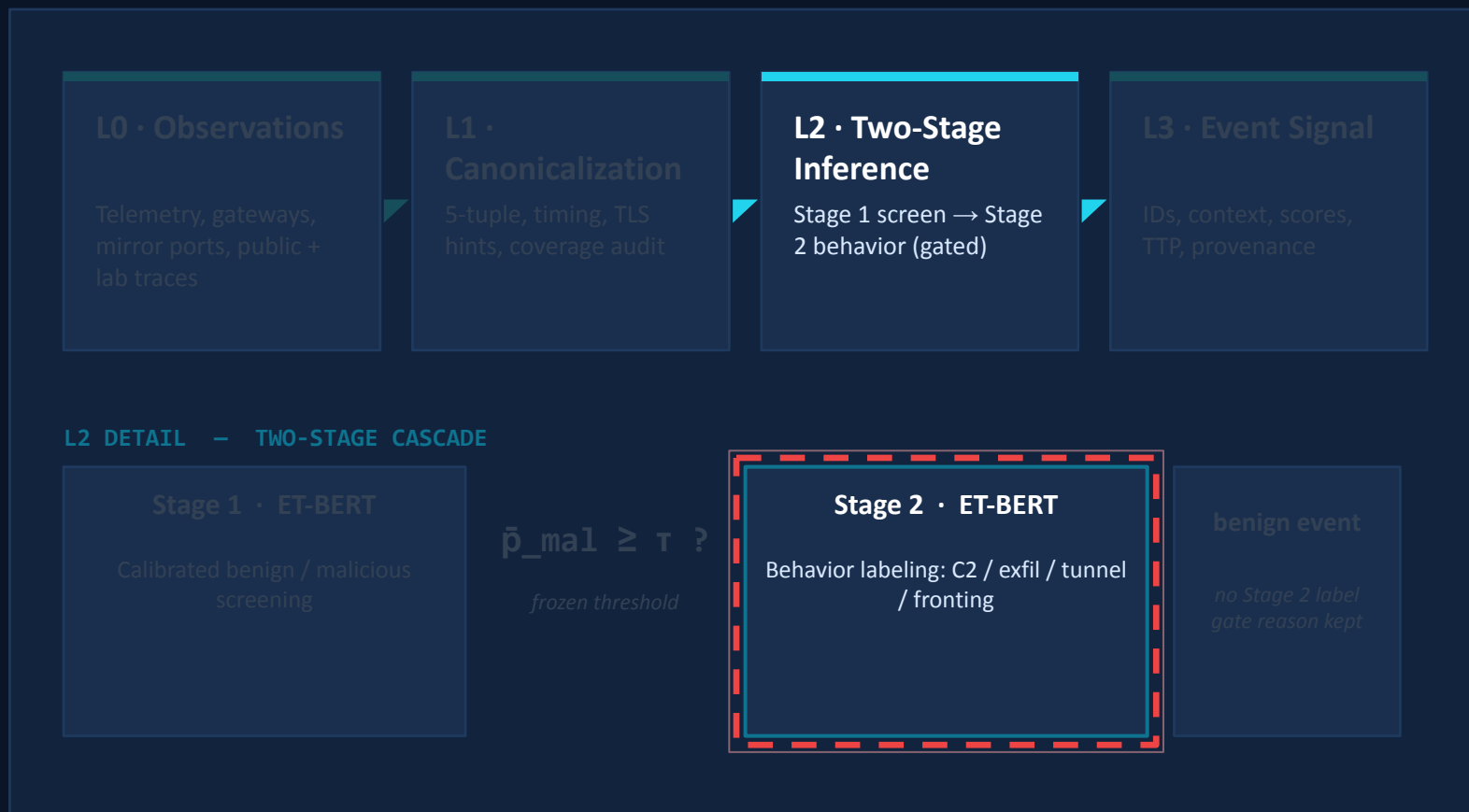
$$\bar{p}_{mal} \geq \tau$$

The SAME inequality governs:

- ① Stage 2 supervision mining
- ② Online inference
- ③ Offline replay

τ is selected on dev replay traces and **frozen before test**. This is what prevents label / dispatch drift.

Step 4 — Stage 2: Conditional Behavior Labeling



BEHAVIOR TAXONOMY

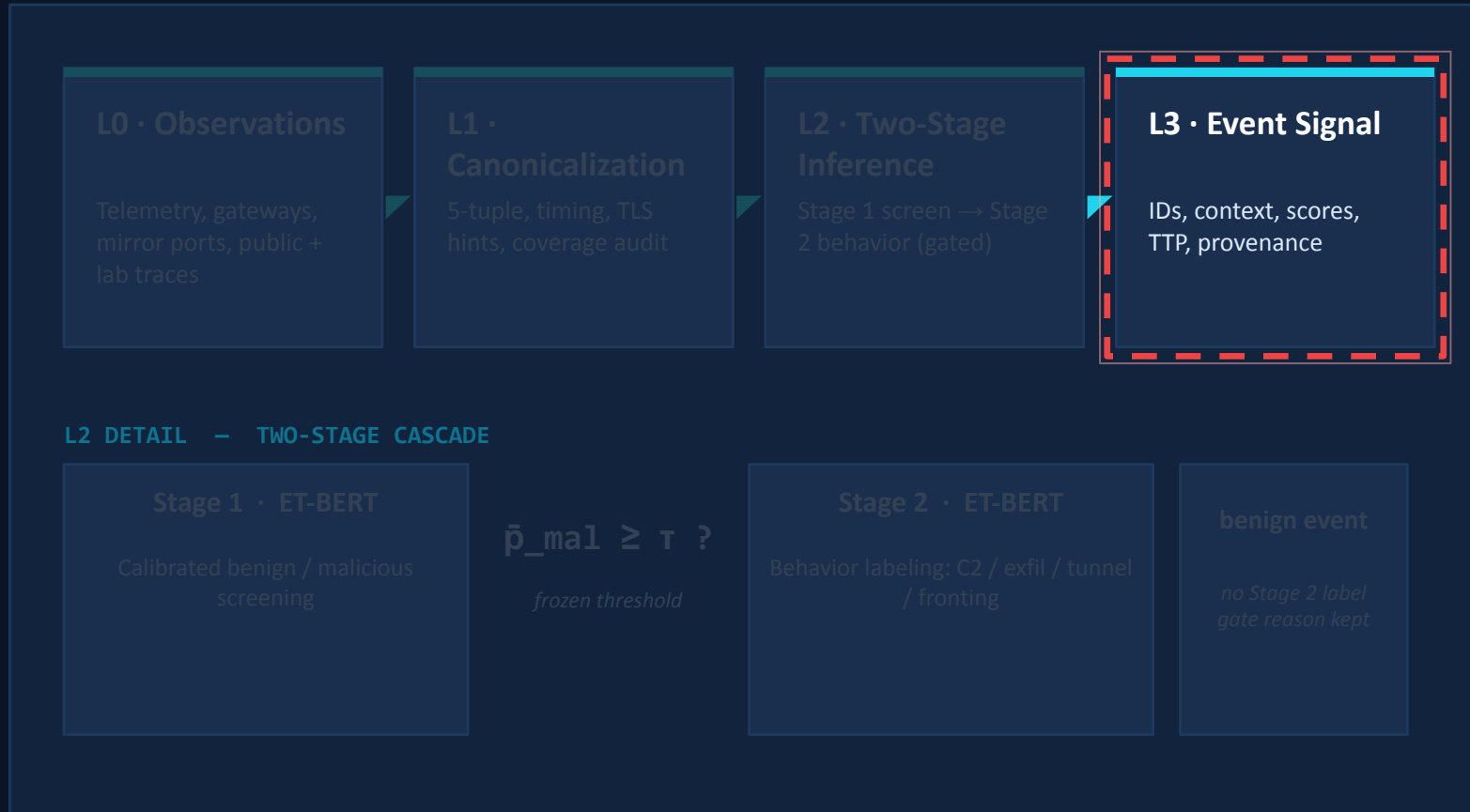
Trained **only on flows admitted by the gate** — matches deployment's conditional distribution.

Four target behaviors:

- C2 — control-channel semantics
- Data exfiltration
- HTTPS tunneling — covert channel
- Domain-fronting-like behavior

Macro-F1 emphasized: two test classes have only 14 examples each.

Step 5 — L3: The Replayable Event Signal



THE OUTPUT INTERFACE

Every flow emits **exactly one signal** — including those suppressed by Stage 1.

$e_i = \{m_i, n_i, x_i, c_i, a_i, g_i\}$

identity · network context · sequences · inference · threat semantics · governance

Deterministic replay invariant:

same features + threshold + model + schema version → same dispatch and same serialized fields.

Putting It All Together

All five design pieces work as one engineering contract. Now: does it actually hold up?

L0 · Observations

Telemetry, gateways, mirror ports, public
+ lab traces

L1 · Canonicalization

5-tuple, timing, TLS hints, coverage audit

L2 · Two-Stage Inference

Stage 1 screen → Stage 2 behavior
(gated)

L3 · Event Signal

IDs, context, scores, TTP, provenance

L2 DETAIL — TWO-STAGE CASCADE

Stage 1 · ET-BERT

Calibrated benign / malicious screening

$$\bar{p}_{\text{mal}} \geq \tau ?$$

frozen threshold

Stage 2 · ET-BERT

Behavior labeling: C2 / exfil / tunnel / fronting

benign event

*no Stage 2 label
gate reason kept*

Validated Results Snapshot

0.9478

Stage 1 OOD macro-F1

calibrated dispatch gate

0.7778

Stage 2 held-out macro-F1

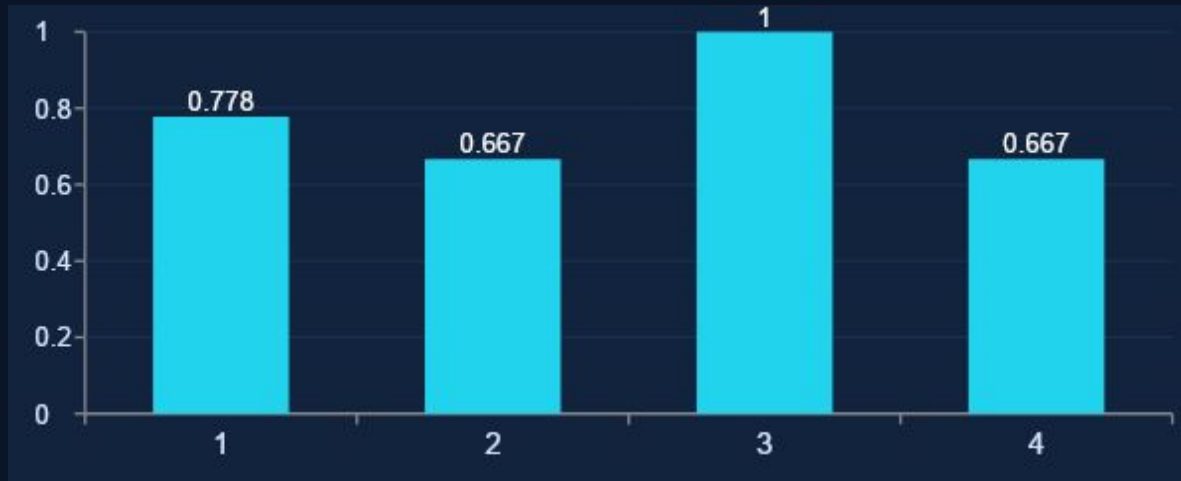
n = 93, four behaviors

460/460

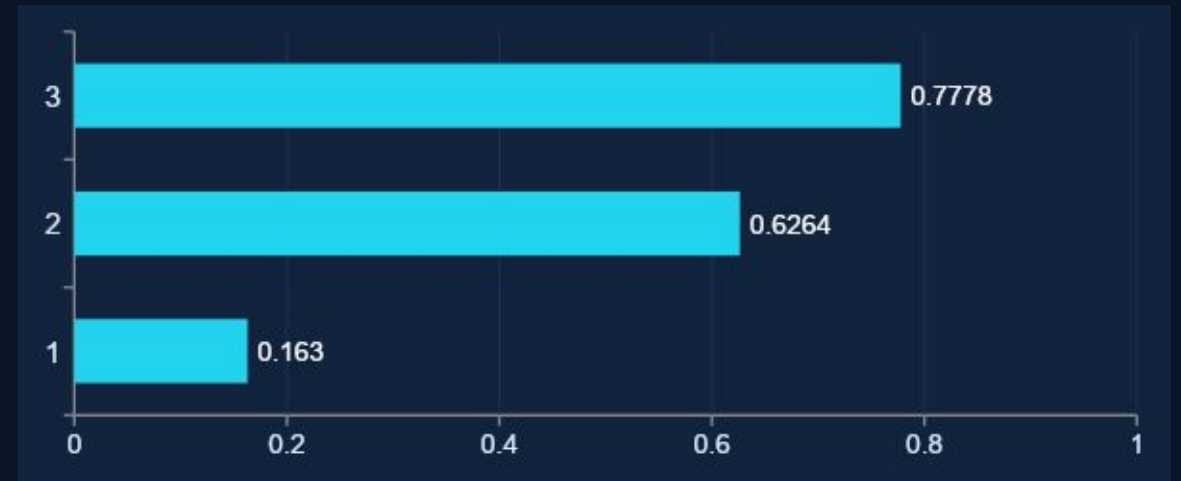
Replay parity

schema-valid event signals

STAGE 2 PER-CLASS F1



BASELINES • STAGE 2 macro-F1



Highlight 1 — Stage 1 Holds Under OOD

0.9478

Stage 1 OOD macro-F1

calibrated dispatch gate

0.7778

Stage 2 held-out macro-F1

n = 93, four behaviors

460/460

Replay parity

schema-valid event signals

WHAT 0.9478 BUYS US

Setup: Train / Dev / Test = 4160 / 450 / 460, class ratio 3744 : 416, **group overlap = 0**.

→ Stage 1 is reliable enough to act as a **calibrated dispatch gate**. Stage 2 only sees flows that already passed this gate, so its training distribution matches deployment.

Why 'group overlap = 0' matters: encrypted-traffic captures are heavily correlated. Zero group overlap forecloses the easy leakage path that inflates many published numbers.

Highlight 2 — Stage 2 Beats Both Baselines

0.9478

Stage 1 OOD macro-F1

calibrated dispatch gate

0.7778

Stage 2 held-out macro-F1

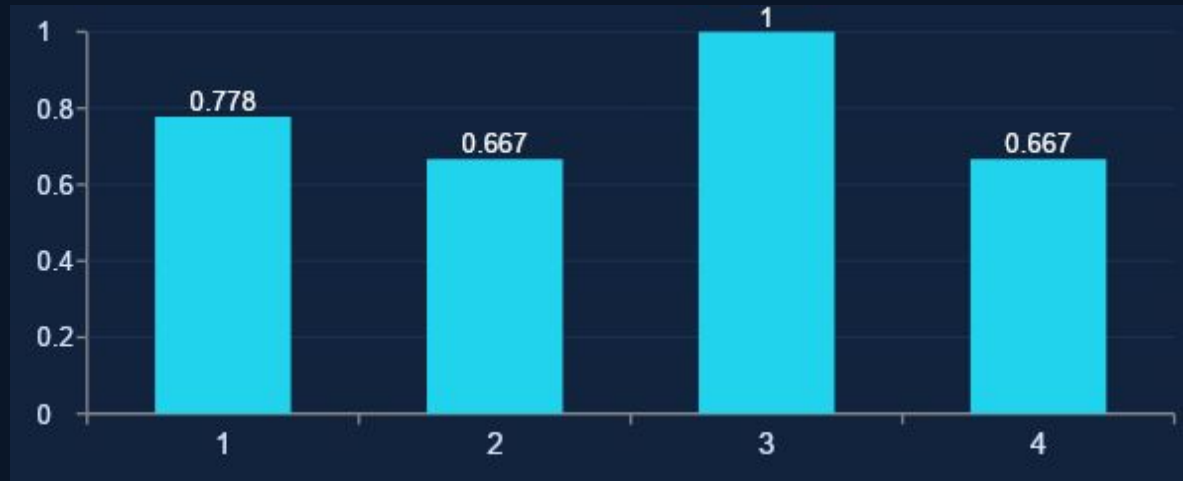
n = 93, four behaviors

460/460

Replay parity

schema-valid event signals

STAGE 2 PER-CLASS F1



READING THE NUMBERS

Stage 2 macro-F1 **0.7778** vs Naive Bayes 0.6264 vs Majority 0.1630

Validates **conditional tagging** in the frozen snapshot — **not** open-world generalization.

Two classes have only 14 test examples — we report this honestly.

Highlight 3 — Replay Parity: The Engineering Claim

0.9478

Stage 1 OOD macro-F1

calibrated dispatch gate

0.7778

Stage 2 held-out macro-F1

n = 93, four behaviors

460/460

Replay parity

schema-valid event signals

460

input flows

→ 460 event signals (1-to-1)

418

Stage 1 benign

still emit auditable signals

42

Stage 2 admissions

all gate-admitted flows tagged

ENGINEERING TAKEAWAY

Scores, dispatch booleans, Stage 2 cardinality, and serialized fields all agree under replay. → **the emitted events are threshold-consistent.**

Boundary Tests — Where Transfer Holds

Frozen checkpoints. None of these data were used for promotion. Strict boundary probes only.

✓ 309 / 360

Public MTA RAT/infostealer → C2

recall proxy 0.8583

✓ 147 / 147

Public DoH tunnels → HTTPS tunneling

complete recall on this corpus

✓ 50 / 51

Benign TLS 1.3 → benign

low false-positive on standard HTTPS

WHAT THESE CASES DO · AND DON'T · PROVE

- ✓ Behavior-specific transfer holds for **single-family** malicious traffic that resembles the training distribution.
- ✓ Benign TLS 1.3 stays benign — the Stage 1 gate does not crumble on a routine protocol upgrade.
- ✗ *But each set is single-label and not balanced.* We use them as boundary evidence — **not as headline benchmarks.**

Boundary Tests — Where Coverage Gates Earn Their Keep

✘ 0.0654

Public → authorized transfer macro-F1

Coverage audit: dev token coverage 1.0000 vs test 0.0000. 93/93 rows fall below the 0.5 threshold and are flagged adaptation_required.

Representation mismatch — surfaced, not hidden.

→ pipeline flags it; downstream consumers can route accordingly.

✘ 105 / 120

QUIC / HTTP3-like UDP/443 → flagged malicious

Stage 1 was trained on TCP/TLS byte patterns. UDP/443 differs in header and timing structure, so Stage 1 misreads it.

Protocol-distribution shift — a known failure mode.

→ pipeline flags it; downstream consumers can route accordingly.

CONCLUSION · TAKEAWAYS

Encrypted threat analysis at the edge should emit
structured, threshold-consistent event signals.

0.9478

Stage 1 OOD macro-F1

calibrated dispatch gate

0.7778

Stage 2 macro-F1

frozen-snapshot conditional tagging

460/460

Schema-valid replay

42 Stage 2 admissions

SCOPE

A **validated input layer** for correlation, ticketing, and future reasoning modules. **Not** production SOC validation, automated containment, or autonomous response.

Thank you. *Questions welcome.*