

Supplementary Information

CAMF-Zero: Contradiction-Aware Reasoning for Zero-Day Threat Detection in Federated Multi-Cloud Environments

Aligned with the manuscript version dated 16 July 2026

S1. Purpose and accessibility

This document accompanies a clean, unencrypted, standard ZIP archive prepared in response to the editorial request for an accessible supplementary file. All archive names use portable ASCII characters, and compiled caches, password protection, symbolic links, and executable binaries have been removed. The package can be opened with standard ZIP utilities on Windows, macOS, and Linux.

The supplementary package is aligned with the 16 July 2026 manuscript and provides manuscript-to-code mapping, input schema, configuration files, small synthetic examples, ablation settings, and machine-readable copies of the reported result tables. The public benchmark datasets are not redistributed; official acquisition links and preparation steps are provided in README.md.

S2. Package contents and manuscript alignment

Package item	Purpose	Manuscript correspondence
camf_zero/models/encoders.py	Text, graph, and temporal encoders	Section 3.2
camf_zero/models/retrieval.py	Retrieval abstraction and contextual evidence	Section 3.3.2
camf_zero/models/fusion.py	Analyst agents, consistency arbiter, pairwise contradiction, CACS, fusion	Sections 3.3.3-3.3.6
camf_zero/models/ssl.py	Contrastive alignment, consistency regularization, modality completion	Section 3.4
camf_zero/models/uncertainty.py	Evidential inference and reliability weighting	Section 3.5
camf_zero/models/streaming.py	Windowing and deployment-oriented latency helpers	Section 3.6
configs/ablation_*.yaml	Controlled removal of individual modules	Table 12
results/*.csv	Machine-readable values reported in the manuscript	Tables 10-14
data/*.jsonl	Synthetic examples demonstrating the input schema	Section 4; demonstration only

S3. Data sources and preparation

The evaluation described in the manuscript uses CIC-IDS2017, CSE-CIC-IDS2018, and UGRansome. Raw files are obtained from their original providers. To prevent leakage, chronological ordering or attack-family holdout is established before normalization, vocabulary construction, augmentation, and zero-day designation.

Dataset	Provider	Approximate scale	Robustness settings
CIC-IDS2017	UNB Canadian Institute for Cybersecurity	~2.83M records in the manuscript summary	10-40% missing; Gaussian noise sigma

			0.1-0.3
CSE-CIC-IDS2018	CSE/UNB CIC and AWS Open Data	~16M+ records	10-40% missing; delay up to 30 s; noise
UGRansome	Public UGRansome distribution	~500k+ traces	15-35% missing modalities

Each preprocessed row represents a multimodal time window. Logs are represented as normalized security-event text; temporal features are normalized sequences; and graph windows contain resource, identity, service, or host nodes connected by communication, access, or dependency edges. Class labels use 0 for benign, 1 for known attack, and 2 for unknown/zero-day.

S4. CACS implementation correspondence

The implementation follows the manuscript logic in which each modality produces a structured hypothesis embedding. For each modality pair, semantic disagreement and a symbolic contradiction signal are combined. Pairwise contradictions are aggregated, then integrated with the consistency arbiter non-entailment signal and predictive uncertainty to produce CACS. CACS is included in the final fusion representation and participates in the open-set decision rule.

Configuration field	Role
lambda1	Weight of semantic disagreement in pairwise contradiction
lambda2	Weight of symbolic contradiction in pairwise contradiction
beta1	Weight of aggregate pairwise contradiction in CACS
beta2	Weight of arbiter non-entailment in CACS
beta3	Weight of predictive uncertainty in CACS
decision_cacs_threshold	Threshold contributing to unknown/zero-day assignment
decision_uncertainty_threshold	Uncertainty threshold contributing to unknown/zero-day assignment

S5. Self-supervised robustness and missing modalities

The package includes contrastive cross-modal alignment, consistency regularization, and a completion network that reconstructs a degraded or absent modality from the available modalities. Controlled ablation files disable self-supervised alignment or modality completion independently. The robustness protocol evaluates 0%, 10%, 20%, 30%, and 40% missing telemetry, consistent with Table 13 of the manuscript.

S6. Uncertainty-aware open-set decision

The evidential head produces non-negative evidence, Dirichlet concentration parameters, class probabilities, and predictive uncertainty. Modality contributions are then weighted using evidence quality, uncertainty, and telemetry-quality indicators. The final output contains the predicted class, predictive uncertainty, and CACS, thereby preserving the three decision quantities emphasized in the manuscript.

S7. Reported quantitative results

S7.1 Main comparative results (manuscript Table 10)

Model	Recall@5% FPR	AUC-PR	F1	ECE
-------	---------------	--------	----	-----

Unimodal (Logs)	62.4 +/- 1.6	71.8 +/- 1.3	68.5 +/- 1.4	0.142 +/- 0.009
Unimodal (Graph)	58.7 +/- 1.8	68.9 +/- 1.5	65.2 +/- 1.6	0.158 +/- 0.011
Early Fusion	71.3 +/- 1.4	79.4 +/- 1.2	76.1 +/- 1.3	0.119 +/- 0.008
Late Fusion	74.8 +/- 1.3	82.1 +/- 1.1	78.9 +/- 1.2	0.107 +/- 0.007
ZeroDay-LLM	68.5 +/- 1.7	76.3 +/- 1.4	72.4 +/- 1.5	0.131 +/- 0.010
Wang et al. [3]	77.2 +/- 1.2	84.6 +/- 1.0	80.5 +/- 1.1	0.098 +/- 0.006
Belay et al. [4]	79.6 +/- 1.1	86.2 +/- 0.9	82.7 +/- 1.0	0.089 +/- 0.006
CAMF-Zero	94.1 +/- 0.8	95.7 +/- 0.6	93.8 +/- 0.7	0.041 +/- 0.004

S7.2 Ablation study (manuscript Table 12)

Configuration	Recall@5% FPR	ECE	Delta recall
Full CAMF-Zero	95.3	0.038	-
Without CACS	78.6	0.092	-16.7
Without Agentic Loop	82.4	0.076	-12.9
Without Evidential Weighting	84.1	0.068	-11.2
Without Modality Completion	79.8	0.105	-15.5
Without Self-Supervised Alignment	81.5	0.089	-13.8

S7.3 Missing-modality robustness (manuscript Table 13)

Missing (%)	Unimodal	Early Fusion	Wang [3]	Belay [4]	CAMF-Zero
0	63.2	72.5	78.4	80.1	95.3
10	58.7	68.9	74.2	76.8	92.6
20	52.4	63.1	69.5	71.9	89.4
30	46.8	57.3	64.8	67.2	85.7
40	39.5	49.6	56.7	59.4	79.2

S7.4 Latency-accuracy trade-off (manuscript Table 14)

Configuration	Latency (ms)	Throughput	Recall (%)
15 s, 2 steps	180	5.2	88.7
30 s, 3 steps	320	3.1	95.3
60 s, 5 steps	580	1.7	96.8
Quantized + caching	245	4.0	93.9

S8. Reproduction workflow

1. Create a Python virtual environment and install requirements.txt.
2. Download the original datasets from the links in README.md and comply with provider terms.
3. Construct chronological or attack-family-held-out splits before fitting preprocessing objects.
4. Convert records to the JSONL multimodal window schema documented in docs/DATA_SCHEMA.md.
5. Run scripts/train.py with configs/default.yaml.
6. Run scripts/evaluate.py on the held-out test split and compute Recall@FPR=5%, AUC-PR, F1, ECE, and false-positive reduction.

7. Run `scripts/run_ablation.py` with the five controlled ablation files.
8. Apply the missingness, noise, and delay settings for robustness analysis and use `scripts/stream_infer.py` for deployment-oriented timing.

S9. Verification and integrity

The archive contains `MANIFEST.txt` and `SHA256SUMS.txt`. Running `python scripts/verify_package.py` checks required files, parses JSONL and CSV resources, loads the default YAML configuration, and imports the CAMF-Zero package. Running `sha256sum -c SHA256SUMS.txt` verifies file integrity after extraction.

S10. Limitations of the distributed supplement

The small JSONL files are synthetic examples intended only for schema and code-path validation. They do not reproduce the numerical results in the manuscript. Full benchmark data and trained model checkpoints are not embedded because of dataset size, provider conditions, and submission-file constraints. The reported values are supplied separately as CSV files for transparent cross-checking against the article.