# NEURALGRAD: MULTI-SCALE GRADIENT REACTIVITY FOR OPEN-WORLD NETWORK INTRUSION DETECTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Neural network-based intrusion detection systems achieve high accuracy on known attacks but fail to detect novel threats in real-world deployments. We address three fundamental challenges: (1) detecting out-of-distribution attacks from unseen families, (2) adapting to novel attacks from $k < 10$ labeled examples, and (3) capturing multi-scale temporal patterns from packets to sessions. We propose **NeuralGrad**, a framework leveraging *gradient reactivity*, the structured response of network layers during inference, captured via intermediate activation gradients. Our contributions are: (1) Gradient Reactivity Maps (GRM): A learned representation where in-distribution inputs produce low-rank gradient patterns while out-of-distribution inputs project far outside. We prove OOD separability unconditionally in the NTK regime and under testable regularity conditions generally. (2) Multi-Scale Dual-Stream Transformer: Hierarchical architecture with parallel temporal/spatial attention at packet, flow, and session scales. (3) Gradient-Space Prototypical Networks: Few-shot classification via gradient-space prototypes, exploiting higher ambient dimensionality than feature space. On 14.8M samples across eight datasets: 98.7% closed-world accuracy, 94.5% AUROC on novel attacks (+8.7% over prior work), 87.4% accuracy with 3 labeled examples (+29.8% over baselines), and 0.71 ms edge inference.

## 1 INTRODUCTION

Deep learning-based NIDS achieve $> 98\%$ accuracy on known attacks Ferrag et al. (2020); Xu et al. (2025) but fail in open-world deployments with unseen attack families and scarce labels Yang et al. (2024). Three problems remain unsolved: **Problem 1: OOD Detection.** Existing OOD methods (softmax confidence Hendrycks & Gimpel (2017a), energy scores Liu et al. (2020), feature distances Sun et al. (2022)) transfer poorly to network traffic. Gradient-based methods (GradNorm Huang et al. (2021), GradPCA Seleznova et al. (2026)) collapse high-dimensional gradient tensors to scalars, discarding per-layer structure. **Problem 2: Few-Shot Adaptation.** Feature-space meta-learning (MAML Finn et al. (2017), ProtoNets Snell et al. (2017)) fails to distinguish novel attacks that mimic benign traffic. Gradient space captures *functional* similarity through class-conditional geometry, offering stronger discrimination. **Problem 3: Multi-Scale Modeling.** Attacks exhibit hierarchical temporal structure (packets: ms, flows: seconds, sessions: minutes), but existing architectures Kim et al. (2020); Lin et al. (2022); Wu et al. (2025) process only single scales.

### 1.1 GRADIENT REACTIVITY AS A UNIFYING SOLUTION

We observe that these three problems share a common solution: *gradient reactivity*, the structured tensor $\nabla_{h_l} \ell$ computed at intermediate layers during inference. In-distribution inputs produce consistent, coherent gradients; novel attacks yield irregular patterns with unusual cross-layer correlations. We propose **NeuralGrad**, comprising three components: **(C1) Gradient Reactivity Maps (GRM):** A learned encoder that captures per-layer gradient patterns. Theorem 1 proves OOD separability: in-distribution gradients concentrate in a low-rank subspace while OOD gradients project far outside. This holds unconditionally in the NTK regime and under testable conditions generally. **(C2) Multi-Scale Dual-Stream Transformer (MSDST):** Hierarchical architecture with parallel temporal/spatial attention at packet, flow, and session scales, yielding richer gradient signals. **(C3) Gradient-Space Prototypical Networks (GSPN):** Few-shot classification via gradient-space prototypes exploiting higher ambient dimensionality, achieving 87.4% accuracy with 3 labeled examples. Timeliness, three factors motivate gradient-based NIDS: (i) encrypted traffic precludes payload inspection, but gradients operate on learned representations; (ii) novel attacks dominate open-world settings; (iii) edge hardware (Jetson Orin, RPi 5) makes inference-time backward passes feasible Seleznova et al. (2026).

## 2 RELATED WORK AND POSITIONING

**Deep Learning for NIDS.** NIDS architectures evolved through CNNs Kim et al. (2020); Li et al. (2023), RNNs/LSTMs Savic et al. (2021); Mirsky et al. (2018), single-scale Transformers Lin et al. (2022); Wu et al. (2025), and GNNs Deng & Hooi (2021). None exploit gradient-space information. Multi-scale Transformers exist for vision Liu et al. (2021) and forecasting Zhou et al. (2021); Nie et al. (2023), but no NIDS combines multi-scale temporal, dual streams, and learned fusion. Concept drift remains largely unaddressed: a 2025 survey found only 4.7% of NIDS papers handle drift Komarchesqui et al. (2026), with recent methods like VLSA-CL Wasswa et al. (2026) targeting IoT botnets. **Gradient OOD Methods.** GradNorm Huang et al. (2021) uses scalar $L_1$ norms; GradPCA Seleznova et al. (2026) applies PCA to class-means; GradOrth Behpour et al. (2023) uses orthogonal projections. All collapse gradient tensors to low dimensions, apply fixed transforms, and target images only. **NeuralGrad provides the first structured, learnable gradient-tensor representation for NIDS.** Few-shot meta-learning (MAML Finn et al. (2017), ProtoNets Snell et al. (2017)) operates in feature



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.

space only; **GSPN is the first gradient-space few-shot classifier for security.** Table 1 positions NeuralGrad as the only framework combining all six capabilities.

**Additional OOD Baselines.** We include standard OOD baselines (MSP Hendrycks & Gimpel (2017b), ODIN Liang et al. (2018), Energy Liu et al. (2020), KNN Sun et al. (2022), ReAct Sun et al. (2021), ViM Wang et al. (2022)) and gradient-specific baselines (Table 37). Mahalanobis Lee et al. (2018) is subsumed by our feature-space score $s_f$ (87.8% AUROC); ODIN reaches 77.0% in our matched-backbone matrix (Table 39). Recent transformer NIDS (IDS-MTran Xi et al. (2024), LLM-IDS Adjewa et al. (2025)), graph OOD methods (ODGNN Zhang et al. (2025a), EPGNN Zhang et al. (2025b), PGOS Sun et al. (2026)), energy-based models (DEBO Chen & Ding (2025)), normalizing flows (RealNVP Lotfi et al. (2025), SITN Bomatter et al. (2026), SPF Kwon (2024)), self-supervised time-series Zhang et al. (2024); Sánchez-Ferrera et al. (2025), and SSL NIDS Li et al. (2023); Xu et al. (2024) target orthogonal or complementary settings.

Table 1: Positioning of NeuralGrad against prior methods and additional baselines.

| Method | Grad. Space | Learn Grad. | Multi Scale | Few Shot | Open World | IDS |
|---|---|---|---|---|---|---|
| Trans-IDS Kim et al. (2020) | × | × | × | × | × | ✓ |
| GradNorm Huang et al. (2021) | Scalar | × | × | × | ✓ | × |
| GradPCA Seleznova et al. (2026) | PCA | × | × | × | ✓ | × |
| GradOrth Behpour et al. (2023) | Orth. | × | × | × | ✓ | × |
| ProtoNet Snell et al. (2017) | × | × | × | ✓ | × | × |
| Swin Liu et al. (2021) | × | × | ✓ | × | × | × |
| TCG-IDS Wu et al. (2025) | × | × | × | × | × | ✓ |
| IDS-MTran Xi et al. (2024) | × | × | ✓ | × | × | ✓ |
| LLM-IDS Adjewa et al. (2025) | × | × | × | × | ✓ | ✓ |
| VLSA-CL Wasswa et al. (2026) | × | × | × | × | ✓ | ✓ |
| Mahalanobis Lee et al. (2018) | × | × | × | × | ✓ | × |
| ODIN (temp.) Liang et al. (2018) | × | × | × | × | ✓ | × |
| Deep Ens. Lakshminarayanan et al. (2017) | × | × | × | × | ✓ | × |
| MC-Dropout Gal & Ghahramani (2016) | × | × | × | × | ✓ | × |
| ODGNN Zhang et al. (2025a) | × | × | × | × | ✓ | × |
| EPGNN Zhang et al. (2025b) | × | × | × | × | ✓ | × |
| DEBO Chen & Ding (2025) | × | × | × | × | ✓ | × |
| SITN Bomatter et al. (2026) | × | × | × | × | ✓ | × |
| SPF Kwon (2024) | × | × | × | × | ✓ | × |
| SSL TS Sánchez-Ferrera et al. (2025) | × | × | × | × | × | × |
| SSL NIDS Xu et al. (2024) | × | × | × | × | × | ✓ |
| **NeuralGrad** | **Full** | ✓ | ✓ | ✓ | ✓ | ✓ |

**LLM-based detection.** Recent work explores LLMs for NIDS Adjewa et al. (2025), but empirical evaluation shows current LLMs achieve only 16–33% accuracy on raw payload classification—comparable to random guessing Kim et al. (2025). This confirms that domain-specific gradient-based methods remain essential. NeuralGrad's gradient-reactivity principle is backbone-agnostic and can be combined with any of these advances.

# 3 NeuralGrad Framework

## 3.1 Problem Formulation

We consider a labeled training set $\mathcal{D}_{\text{train}} = \{(x_j, y_j)\}_{j=1}^{n}$ with $x_j \in \mathbb{R}^d$ and $y_j \in \mathcal{Y}_{\text{known}} = \{1, \dots, C\}$. At deployment, inputs arrive from $\mathcal{Y}_{\text{known}} \cup \mathcal{Y}_{\text{novel}}$, where $\mathcal{Y}_{\text{novel}}$ contains unseen attack family attack classes unseen during training. The classifier $f_W : \mathbb{R}^d \to \mathbb{R}^C$ must simultaneously **(i)** classify known traffic with high accuracy, **(ii)** flag novel inputs as out-of-distribution (OOD) with high AUROC, and **(iii)** adapt to a new class $c_{\text{new}}$ given $k \leq 10$ labeled examples.

## 3.2 C1: Gradient Reactivity Maps

**Gradient Reactivity Map (GRM).** For input $x$, we compute per-layer gradients via a single backward pass against a uniform reference loss. Let $h_l(x) \in \mathbb{R}^{d_l}$ denote the intermediate activation at layer $l$. Concretely, let $\ell_{\text{ref}}(f_W(x)) = \mathrm{KL}(u \,\|\, \mathrm{softmax}(f_W(x)))$ with $u = (1/C, \dots, 1/C)$. The GRM is $\mathcal{G}(x) = \{\mathbf{g}_l(x)\}_{l=1}^{L}$ where $\mathbf{g}_l(x) = \nabla_{h_l} \ell_{\text{ref}}(f_W(x))$. Using the uniform reference decouples gradient magnitude from predicted class identity. With this formulation, the penultimate-layer gradient is $\nabla_z \ell_{\text{ref}} = p - u$, which is *maximal* for confident in-distribution predictions and *smaller* for uncertain OOD inputs; the discriminative signal therefore resides in the *structure* (direction and cross-layer coherence) of the gradient tensor, not its magnitude. This distinction is critical: as we show in Section 5, GradNorm uses the identical uniform-target gradients but collapses them to a scalar, achieving 88.4 AUROC versus our structured score's 96.8 on the same backbone—a 8.4-point gap attributable to directional structure, not magnitude. The choice of $\ell_{\text{ref}}$ is central; we ablate it in Section L.2. Five variants are compared: (1) uniform reference (ours), (2) entropy reference $-\sum_c p_c \log p_c$, (3) learned reference $\mathrm{KL}(r_\phi \| p)$ with a small learned $r_\phi$, (4) one-hot reference (predicted class), and (5) random reference (random probability vector). Uniform is near-optimal because it decouples magnitude from class identity; the learned reference performs comparably but adds parameters, while one-hot collapses to zero on confident predictions and random destroys the signal.

**Gradient Analyzer Network ($\text{GAN}_{\text{alyz}}$).** Raw GRM tensors are heterogeneously sized; $\text{GAN}_{\text{alyz}}$ maps them to a fixed embedding $\mathbf{e}_g(x) \in \mathbb{R}^{d_g}$ through three stages. **(i) Per-layer encoding**: each $\mathbf{g}_l \in \mathbb{R}^{d_l}$ is projected as $\mathbf{z}_l = \mathrm{LayerNorm}(W_l^{\text{proj}} \mathbf{g}_l + b_l)$. **(ii) Cross-layer attention**: the sequence $Z = [\mathbf{z}_1; \dots; \mathbf{z}_L]$ is processed by multi-head self-attention with learned positional encodings $\text{PE}_L$, yielding $\tilde{Z} = \mathrm{MultiHead}(Z + \text{PE}_L, Z + \text{PE}_L, Z + \text{PE}_L)$. **(iii) Attentive pooling**:

$$\alpha_l = \frac{\exp(\mathbf{w}^\top \tilde{\mathbf{z}}_l)}{\sum_{l'} \exp(\mathbf{w}^\top \tilde{\mathbf{z}}_{l'})}, \qquad \mathbf{e}_g = \sum_{l=1}^{L} \alpha_l \tilde{\mathbf{z}}_l, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^{d_g}$ is a learned query vector.

**Dual-Space OOD Scoring.** NeuralGrad combines a penultimate-layer feature embedding $\mathbf{e}_f(x) \in \mathbb{R}^{d_f}$ with $\mathbf{e}_g(x)$, using Mahalanobis distance to the nearest known-class centroid in each space:

$$s_f(x) = \min_{c \in \mathcal{Y}_{\text{known}}} (\mathbf{e}_f - \boldsymbol{\mu}_c^f)^\top (\Sigma^f)^{-1} (\mathbf{e}_f - \boldsymbol{\mu}_c^f), \quad s_g(x) = \min_c (\mathbf{e}_g - \boldsymbol{\mu}_c^g)^\top (\Sigma^g)^{-1} (\mathbf{e}_g - \boldsymbol{\mu}_c^g). \tag{2}$$

The final score fuses both signals via a sigmoid:

$$s(x) = \sigma(w_f \cdot s_f(x) + w_g \cdot s_g(x) + b)\,, \tag{3}$$

where $w_f, w_g, b$ are learned on a validation set with held-out known classes. The feature-space score captures semantic proximity; the gradient-space score reflects deviations in the model's internal sensitivity patterns, making their combination robust under distributional shift. We establish that GRMs provide superior OOD separation over forward-pass features under explicit, testable assumptions, and additionally provide an unconditional variant in the linearized NTK regime, a finite-width correction, and a minimax lower bound for the few-shot case.

**Assumption 1** (ID Gradient Concentration)**.** *For all $(x, y) \in \mathcal{D}_{id}$, the gradient reactivity vector $\mathbf{G}(x)$ lies in a subspace $\mathcal{V}_g^{ID}$ of dimension $r \le C \cdot L$, where $C$ is the number of known classes and $L$ is the network depth.*

**Assumption 2** (OOD Gradient Dispersion)**.** *For OOD inputs $x_{ood}$ drawn from a distribution sufficiently different from $\mathcal{D}_{id}$, the gradient reactivity $\mathbf{G}(x_{ood})$ has a non-negligible component in the orthogonal complement of $\mathcal{V}_g^{ID}$, with $\|\Pi^{\perp}_{\mathcal{V}_g^{ID}} \mathbf{G}(x_{ood})\|_2^2 / \|\mathbf{G}(x_{ood})\|_2^2 \ge \gamma > 0$.*

**Theorem 1** (GRM Separability Advantage)**.** *Let $f_W$ be an $L$-layer network ($L \ge 3$) trained to convergence, with in-distribution subspaces $\mathcal{V}_f^{ID} \subseteq \mathbb{R}^{d_f}$ and $\mathcal{V}_g^{ID} \subseteq \mathbb{R}^{D_g}$, where $D_g = \sum_l d_l$. Under Assumptions 1 and 2: **(i)** $D_g/d_f \ge L$, providing fundamentally richer ambient space. **(ii)** $\operatorname{rank}(\mathcal{V}_g^{ID}) \le C \cdot L \ll D_g$ for typical architectures. **(iii)** For any OOD input $x_{ood}$:*

$$\frac{\|\Pi^{\perp}_{\mathcal{V}_g^{ID}} \mathcal{G}(x_{ood})\|^2}{\|\mathcal{G}(x_{ood})\|^2} \;\ge\; \frac{\|\Pi^{\perp}_{\mathcal{V}_f^{ID}} \mathbf{e}_f(x_{ood})\|^2}{\|\mathbf{e}_f(x_{ood})\|^2}. \tag{4}$$

*Proof sketch.* **(i)** follows since each layer contributes $d_l \ge d_f$ gradient coordinates. **(ii)** Under Assumption 1, in-distribution gradients align with $C$ class-conditional directions per layer, giving span dimension $\le C \cdot L$. **(iii)** The codimension of $\mathcal{V}_g^{\text{ID}}$ is $D_g - C \cdot L$, far exceeding $d_f - C$ for $\mathcal{V}_f^{\text{ID}}$. By Assumption 2, OOD gradient vectors project substantially onto the orthogonal complement of $\mathcal{V}_g^{\text{ID}}$, yielding the claimed inequality. □

**Remark 1.** *Theorem 1 is stated conditionally on Assumptions 1 and 2, both of which are testable properties of a trained network and are validated empirically in Appendix K. An unconditional variant under the linearized NTK regime is provided in Theorem 5; a finite-width correction is given in Corollary 2; and a minimax lower bound for the few-shot case is given in Theorem 3.*

**Theorem 2** (Finite-width NTK correction)**.** *Under the NTK linearization with finite width $m$, the residual ratio satisfies*

$$\frac{\|\Pi_g^{\perp} \mathbf{G}(x_{ood})\|_2^2}{\|\mathbf{G}(x_{ood})\|_2^2} \;\ge\; 1 - \frac{r}{D_g} - \mathcal{O}\!\left(\frac{1}{\sqrt{m}}\right), \tag{5}$$

*so the NTK-regime bound degrades gracefully as width decreases.*

*Proof sketch.* Apply the finite-width NTK error bound of Yang & Hu (2021) to the linearization equation 22; the deviation of the empirical NTK from its infinite-width limit scales as $\mathcal{O}(1/\sqrt{m})$, which propagates to the projection residual. □

**Theorem 3** (Minimax few-shot separation)**.** *Under the meta-training episodic distribution, for any feature-space classifier $\hat{y}^f$ and any $k \le 10$, there exists a novel-class distribution consistent with the meta-training distribution on which the expected $k$-shot error satisfies $\epsilon_k^f \ge \epsilon_k^{f,\star} > 0$, while GSPN achieves $\epsilon_k^g \le \epsilon_k^f \cdot \exp(-\frac{k-1}{2} \Delta^g_{class}/\Delta^f_{class})$. Consequently, whenever $\Delta^g_{class} > \Delta^f_{class}$, GSPN strictly dominates any feature-space classifier under this distribution.*

*Proof sketch.* Combine the standard minimax lower bound for nearest-centroid classifiers under class-conditional cone separation (Shalev-Shwartz & Ben-David, 2014) with Proposition 2. The lower bound on $\epsilon_k^f$ follows from the covering-number argument applied to the feature-space cone; the upper bound on $\epsilon_k^g$ follows from the gradient-space margin. The strict separation follows from $\Delta^g_{\text{class}} > \Delta^f_{\text{class}}$. □

**Training Objective.** $\text{GAN}_{\text{alyz}}$ is trained jointly with the classifier via:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{react}} \mathcal{L}_{\text{react}} + \lambda_{\text{contr}} \mathcal{L}_{\text{contr}}, \tag{6}$$

where $\mathcal{L}_{\text{react}} = \sum_c |\mathcal{D}_c|^{-1} \sum_{x \in \mathcal{D}_c} \|\mathbf{e}_g(x) - \boldsymbol{\mu}_c^g\|^2$ encourages compact, class-specific gradient subspaces, and

$$\mathcal{L}_{\text{contr}} = -\sum_i \frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\mathbf{e}_g^i \cdot \mathbf{e}_g^p / \tau)}{\sum_{a \ne i} \exp(\mathbf{e}_g^i \cdot \mathbf{e}_g^a / \tau)} \tag{7}$$

enforces inter-class separation in gradient space, with $P(i) = \{p : y_p = y_i, p \ne i\}$ and temperature $\tau$. The reference loss $\ell_{\text{ref}}$ is not a component of $\mathcal{L}_{\text{total}}$ and is never optimized; it serves solely as a probe for computing $\mathcal{G}(x)$.

### 3.3 C2: Multi-Scale Dual-Stream Transformer (MSDST)

**Multi-Scale Tokenization.** Every scale derives from raw packet captures, not vendor-provided tabular CSVs, which cannot support genuine multi-scale temporal modeling. **Packet scale:** per-packet tokens (payload length, direction, inter-arrival time, TCP flags, TTL, window) ordered within a bidirectional 5-tuple flow. **Flow scale:** stride-4 Conv1D over the packet sequence, one token per 4 packets—the resolution at which scanning behavior appears. **Session scale:** stride-4 Conv1D applied to the flow scale, aggregating flows sharing a host pair within $\Delta t$—the resolution at which botnet beaconing appears. Formally, given $X \in \mathbb{R}^{N \times d}$: packet-scale $X^{(1)} \in \mathbb{R}^{N \times d_m}$, flow-scale $X^{(2)} \in \mathbb{R}^{N/4 \times d_m}$, and session-scale $X^{(3)} \in \mathbb{R}^{N/16 \times d_m}$. A per-dataset construction table and extraction code are released (Appendix N). PCAP re-extraction also avoids the flow-construction and labeling defects documented for the CIC family (Engelen et al. 2021; Lanvin et al. 2022).

**Dual Attention Streams.** At each scale $s$, temporal and spatial streams process sequence positions and feature channels, respectively:

$$O_s = \sigma(W_g^s) \odot \text{TransformerBlock}(X^{(s)}) + (1 - \sigma(W_g^s)) \odot \text{TransformerBlock}(X^{(s)\top})^\top, \tag{8}$$

with learned gates $W_g^s$ adaptively weighting each stream.

**Cross-Scale Fusion.** All scales are fused via cross-attention with native-resolution keys/values (no upsampling to a common length):

$$Z_{\text{fused}} = \text{CrossAttn}(Q_{\text{route}},\ [O_1; O_2; O_3],\ [O_1; O_2; O_3])\,, \tag{9}$$

where $Q_{\text{route}} \in \mathbb{R}^{r \times d_m}$ with $r \ll N$ is a set of learned routing queries. The final prediction is $\hat{y} = \text{softmax}(\text{Linear}(\text{GAP}(Z_{\text{fused}})))$.

**Proposition 1** (MSDST Complexity)**.** *MSDST incurs attention cost $C_{MSDST} = 1.066N^2 d_m + 1.313N d_m^2 + \Theta(rNd_m)$, i.e., $+18.9\%$ over a single-scale single-stream transformer at $N = d_m = 128$, while capturing temporal dependencies up to $16\times$ longer. Its GRM effective dimension scales as $D_g^{MSDST} = S \cdot D_g^{single}$, proportionally enriching OOD signal.*

*Proof.* At scale $s$, the sequence length is $N_s = N/4^{s-1}$, and temporal self-attention costs $O(N_s^2 d_m)$. Summing over $S = 3$ scales gives $N^2 d_m(1 + 1/16 + 1/256) = 1.0664N^2 d_m$. The spatial stream costs $\Theta(N_s d_m^2)$ per scale, summing to $1.3125Nd_m^2$ (linear in $N$, so the $N^2$ coefficient is unchanged). Cross-scale routing with $r \ll N$ queries costs $\Theta(rNd_m)$. At $N = d_m = 128$, the ratio to single-scale single-stream is $(1.0664 + 1.3125)/2 = 1.189$, i.e., $+18.9\%$. □

**Efficient-transformer comparison.** We compare MSDST against Performer (linear attention), Linformer (low-rank attention), BigBird (sparse attention), hierarchical pooling (no cross-scale attention), and a dilated Conv1D with receptive field 2048 (no attention). See Section 5.5 and Table 8.

### 3.4 C3: Gradient-Space Prototypical Networks (GSPN)

**Prototypes and Classification.** Given support set $\mathcal{S}_c$ of size $k$, GSPN forms gradient-space prototypes $\boldsymbol{\pi}_c^g = k^{-1} \sum_{(x,c) \in \mathcal{S}_c} \mathbf{e}_g(x)$ (and analogously $\boldsymbol{\pi}_c^f$). Query classification combines distances in both spaces:

$$p(y = c \mid x_q) = \frac{\exp(-d_g(x_q, c) - \beta\, d_f(x_q, c))}{\sum_{c'} \exp(-d_g(x_q, c') - \beta\, d_f(x_q, c'))}, \tag{10}$$

with scalar $\beta$ balancing feature and gradient contributions. GSPN is meta-trained episodically with class holdout: $\mathcal{L}_{\text{meta}} = \mathbb{E}_{\text{ep}}[-\sum_{(x_q, y_q)} \log p(y_q \mid x_q)]$. Training is two-stage: Stage 1 minimizes $\mathcal{L}_{\text{total}}$ (Eq. 6) on the known-class training set; Stage 2 meta-trains GSPN episodically with class holdout under $\mathcal{L}_{\text{meta}}$. The two stages are separable: removing meta-training leaves closed-world accuracy and OOD AUROC identical to the full system, while 3-shot accuracy falls from 87.4% to 74.8% (Table 5).

**Proposition 2** (Class-Conditional Cone Separation)**.** *Let $\mathcal{C}_c^g = \{\mathbf{e}_g(x) : y(x) = c\}$ denote the class-conditional gradient embedding set for class $c$, and let $\Delta_{class}^g = \min_{c \neq c'} d(\pi_c^g, \pi_{c'}^g)$ denote the minimum between-class prototype distance in gradient space, with $\Delta_{class}^f$ the analogous quantity in feature space. Under the same meta-training episodic distribution,*

$$\Delta_{class}^g \geq \Delta_{class}^f, \tag{11}$$

*and the expected $k$-shot classification error satisfies*

$$\epsilon_k^g \ \leq\ \epsilon_k^f \cdot \exp\Big(-\tfrac{k-1}{2}\,(\Delta_{class}^g - \Delta_{class}^f)/\Delta_{class}^f\Big)\,, \tag{12}$$

*so error decays strictly faster in gradient space whenever $\Delta_{class}^g > \Delta_{class}^f$.*

*Proof.* By Theorem 1, the ID gradient subspace has rank $\leq C \cdot L$ with codimension $D_g - C \cdot L \gg d_f - C$, and the contrastive loss $\mathcal{L}_{\text{contr}}$ explicitly maximizes inter-class gradient separation. A standard covering-number argument Shalev-Shwartz & Ben-David (2014) shows that the $k$-shot error bound for a nearest-centroid classifier scales as $\exp(-k\,\Delta_{\text{class}}/2)$ with the between-class margin. Since $\Delta_{\text{class}}^g > \Delta_{\text{class}}^f$ follows from the contrastive training objective and the dimension gap, the inequality holds. □

### 3.5 Inference Path

A single forward-plus-backward pass produces three outputs: **Closed-world class:** $\hat{y} = \mathrm{softmax}(\mathrm{Linear}(\mathrm{GAP}(Z_{\text{fused}})))$, the head defined above. Always active. **Novel-attack flag:** Eq. equation 2 yields $s_f$ and $s_g$; Eq. equation 3 fuses and thresholds them. Always active, in parallel with the head above. This is what the open-world AUROC of Table 4 measures. **Few-shot classification:** Eq. equation 10, used *only after* a novel class has been flagged and $k$ labelled examples obtained. It classifies over $\mathcal{Y}_{\text{known}} \cup c^*$ and is what Table 6 measures.

### 3.6 Convergence and Inference Cost

**Theorem 4** (Training Convergence)**.** *Under standard smoothness assumptions, the multi-task objective converges at rate*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}_{total}(W_t)\|^2] = O\left(\frac{\mathcal{L}_0 - \mathcal{L}^*}{\sqrt{T}} + \frac{\sigma^2}{\sqrt{T}}\right), \tag{13}$$

*with constants scaling as* $(1 + \lambda_{react} + \lambda_{contr})$.

NeuralGrad inference requires one forward and one backward pass. The backward pass computes only activation gradients $\nabla_{h_l}\ell_{\text{ref}}$, omitting the weight-gradient GEMM, so it costs approximately $0.48\times$ the forward pass. Total inference overhead is $1.48\times$ on Jetson Orin and $2.33\times$ on A100, measured against the same MSDST backbone (Section 5.8).

## 4 Experimental Setup

**Leakage-Controlled Protocol.** We adopt a six-control protocol: (i) exact-duplicate removal via hashing, (ii) near-duplicate removal via MinHash (Jaccard $\geq 0.99$), (iii) identifier removal (Flow ID, IPs, ports, timestamps, hostnames), (iv) temporal disjointness (train before test), (v) host/tool disjointness, and (vi) family-disjoint novel classes. All main results use this protocol; rank-preservation tables and a negative leakage control appear in Appendices M.3 and M.4. We evaluate on eight datasets totaling 14.8M samples (Table 2) spanning enterprise, IoT, and industrial domains (2013–2023).

Table 2: Benchmark datasets.

| Dataset | Domain | Samples (M) | Feat. | Classes | Year |
|---|---|---|---|---|---|
| CIC-IDS2017 Sharafaldin et al. (2018) | Enterprise | 2.83 | 78 | 7 | 2017 |
| CIC-IDS2018 Garcia et al. (2014) | Enterprise | 3.12 | 80 | 14 | 2018 |
| UNSW-NB15 Moustafa & Slay (2015) | Hybrid | 2.54 | 49 | 10 | 2015 |
| CICIoT2023 Neto et al. (2023) | IoT | 1.64 | 47 | 33 | 2023 |
| Edge-IIoT Ferrag et al. (2022) | Industrial | 1.55 | 61 | 14 | 2022 |
| TON_IoT Booij et al. (2021) | Multi-layer | 0.46 | 44 | 9 | 2021 |
| IoT-23 Garcia et al. (2020) | IoT Botnet | 0.33 | 21 | 20 | 2020 |
| CTU-13 Garcia et al. (2014) | Botnet | 2.35 | 38 | 8 | 2014 |
| **Total** | | **14.8** | | | |

**Baselines and Protocol.** Classes are split 70% known / 30% novel with five random splits. Classification baselines: CNN-IDS Kim et al. (2020), LSTM-AE Savic et al. (2021), Transformer-IDS Kim et al. (2020), TCG-IDS Wu et al. (2025), XGBoost Chen & Guestrin (2016). OOD baselines: MSP Hendrycks & Gimpel (2017a), ODIN Liang et al. (2018), Energy Liu et al. (2020), KNN Sun et al. (2022), ReAct Sun et al. (2021), ViM Wang et al. (2022), GradNorm Huang et al. (2021), GradPCA Seleznova et al. (2026). Few-shot baselines: fine-tuning, ProtoNet Snell et al. (2017), MAML Finn et al. (2017), RelationNet Sung et al. (2018), MetaOptNet Lee et al. (2019). Seven additional gradient-specific baselines isolate structured representation from mere gradient access: GradNorm-L2, GradNorm per-layer, Grad-Cos, Grad-Mahalanobis, Grad-PCA+Mahalanobis, Jacobian norm, and gradient dot-product. All neural baselines use identical preprocessing, hyperparameter search, and episode construction; OOD methods share a matched Transformer backbone (2.8M params, $L$=6). Full matched-backbone matrix in Appendix M; gradient-specific results in Table 37.

**Hard-OOD, Statistics, and Implementation.** Hard-OOD includes near-OOD (GoldenEye vs. Hulk), far-OOD (DoS vs. Infiltration), and benign-shift OOD (Table 9). Each configuration uses 15 seeds; we report mean±std with 95% CIs, paired $t$-tests with Bonferroni correction, and Cohen's $d$. NeuralGrad uses an MSDST backbone ($d_m$=128, $h$=4, 3 scales, $L$=6, 2.8M params) and a gradient analyzer (0.4M params), trained with AdamW (lr $3\times10^{-4}$, cosine decay, batch 128, 200 epochs, $\lambda_{\text{react}}$=0.1, $\lambda_{\text{contr}}$=0.05, $\tau$=0.1). Experiments run on A100, Jetson Orin, and RPi 5, totaling ∼1,200 GPU-hours.

## 5 Experimental Results

### 5.1 Closed-World Detection Accuracy

Table 3 reports classification accuracy on known classes across datasets under the leakage-controlled protocol. NeuralGrad achieves 98.7%±0.3% on CIC-IDS2017, outperforming Trans-IDS by 0.7 points ($p < 0.001$, $d = 3.9$). On the more challenging CICIoT2023 with 33 attack classes, it attains 94.5%, exceeding Trans-IDS by 1.6 points. Gains are consistent across all datasets, indicating strong generalization.

Table 3: Closed-world detection accuracy (%, 15 runs, 95% confidence intervals, leakage-controlled protocol). Best results are shown in bold. † denotes statistically significant differences ($p < 0.001$) compared to NeuralGrad under paired $t$-tests with Bonferroni correction.

| | | Detection Accuracy (%) | | | | | | | | Stats | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Model** | **CIC-17** | **CIC-18** | **UNSW** | **CICIoT23** | **Edge** | **TON** | **IoT-23** | **CTU-13** | $p$ | $d$ |
| CNN-IDS | CNN | 96.2±0.4† | 95.6±0.5† | 94.8±0.4† | 90.4±0.6† | 93.2±0.5† | 91.3±0.7† | 89.6±0.8† | 92.9±0.5† | <.001 | 5.6 |
| LSTM-AE | LSTM | 95.4±0.5† | 94.9±0.6† | 94.2±0.5† | 89.3±0.7† | 92.4±0.6† | 90.6±0.8† | 88.8±0.9† | 91.7±0.6† | <.001 | 6.1 |
| Trans-IDS | Trans. | 98.0±0.3† | 97.5±0.4† | 96.1±0.4† | 92.9±0.5† | 95.0±0.4† | 93.4±0.5† | 92.4±0.6† | 94.8±0.4† | <.001 | 3.7 |
| TCG-IDS | GNN | 97.6±0.4† | 97.0±0.4† | 95.8±0.4† | 92.5±0.5† | 94.7±0.5† | 93.1±0.6† | 92.1±0.7† | 94.4±0.4† | <.001 | 4.0 |
| XGBoost | Ens. | 97.1±0.3† | 96.6±0.4† | 96.0±0.3† | 92.3±0.4† | 94.6±0.4† | 92.8±0.5† | 91.4±0.6† | 95.1±0.3† | <.001 | 4.2 |
| **NeuralGrad MSDST** | | **98.7±0.3** | **98.2±0.3** | **97.1±0.3** | **94.5±0.4** | **96.4±0.3** | **95.0±0.4** | **94.2±0.5** | **96.3±0.3** | — | — |

## 5.2 Open-World OOD Detection

Table 4 reports open-world OOD detection performance on novel attack families under the leakage-controlled protocol. NeuralGrad achieves 94.5% average AUROC across five datasets, outperforming the strongest baseline, GradPCA (85.8%), by 8.7 points. The gradient-space score $s_g$ alone reaches 92.2%, exceeding all prior methods and highlighting the effectiveness of gradient-based representations for OOD detection. Combining feature and gradient signals further improves performance, with the dual-space score $s_{fg}$ adding 2.3 points over $s_g$. A full matched-backbone breakdown with per-dataset confidence intervals and Bonferroni-corrected significance tests is provided in Appendix M; the advantage over GradPCA is preserved at $+6.4$ to $+8.3$ AUROC points with Cohen's $d \geq 2.7$ on every dataset.

Table 4: Open-world novel attack detection (AUROC %, 15 runs across five datasets, leakage-controlled).

| **Method** | **CIC-17** | **UNSW** | **CICIoT** | **Edge** | **IoT-23** | **Avg** |
|---|---|---|---|---|---|---|
| MSP | 78.2 | 74.5 | 71.3 | 76.1 | 73.8 | 74.8 |
| ODIN | 81.4 | 77.2 | 74.6 | 79.3 | 76.5 | 77.8 |
| Energy | 84.7 | 80.1 | 78.2 | 82.5 | 79.4 | 80.9 |
| KNN | 86.2 | 82.3 | 80.5 | 84.1 | 81.7 | 82.9 |
| ReAct | 85.1 | 81.4 | 79.8 | 83.2 | 80.5 | 82.0 |
| ViM | 87.5 | 83.8 | 81.9 | 85.4 | 82.8 | 84.3 |
| GradNorm | 88.4 | 84.6 | 82.4 | 86.2 | 83.5 | 85.0 |
| GradPCA | 89.2 | 85.4 | 83.1 | 87.0 | 84.2 | 85.8 |
| NeuralGrad ($s_f$) | 90.8 | 87.2 | 85.6 | 89.1 | 86.4 | 87.8 |
| NeuralGrad ($s_g$) | 94.5 | 91.8 | 90.2 | 93.4 | 91.0 | 92.2 |
| **NeuralGrad ($s_{fg}$)** | **96.8** | **94.2** | **92.5** | **95.6** | **93.4** | **94.5** |

## 5.3 Component Ablation

Table 5 isolates each component's contribution. Removing GRM causes the largest single drop, cutting AUROC from 96.8 to 87.5 ($-9.3$) while closed-world accuracy falls only 0.2 points, showing that gradient reactivity is critical specifically for open-world detection rather than classification. Replacing the learned $\text{GAN}_{\text{alyz}}$ with raw gradient processing costs a further 5.4 AUROC points, confirming that structured gradient representation learning, not merely access to gradients, drives performance. Architectural ablations reinforce this pattern: single-scale Transformers, single-stream MSDST, and removal of cross-scale fusion each degrade AUROC (to 92.1, 95.0, and 93.8) and few-shot accuracy in tandem, indicating that multi-scale gradient fusion benefits both objectives. For few-shot adaptation, GSPN reaches 87.4% at 3-shot, far surpassing feature-space ProtoNet (47.8%) and MAML (55.6%), and even outperforms its own non-meta-trained variant (74.8%) by 12.6 points.

Table 5: Component ablation on CIC-IDS2017 (15 runs, accuracy, AUROC, few-shot performance, leakage-controlled).

| **Configuration** | **Acc (%)** | **AUROC** | **3-shot** | $p$ |
|---|---|---|---|---|
| Full NeuralGrad | **98.7±0.3** | **96.8** | **87.4** | — |
| *Architecture (C2)* | | | | |
| Single-scale Trans. | 96.2±0.4 | 92.1 | 82.4 | <.001 |
| Single-stream MSDST | 97.9±0.3 | 95.0 | 86.8 | <.001 |
| No cross-scale fusion | 97.5±0.3 | 93.8 | 84.5 | <.001 |
| *Gradient Reactivity (C1)* | | | | |
| No GRM (forward only) | 98.5±0.2 | 87.5 | 71.2 | <.001 |
| GRM w/o $\text{GAN}_{\text{alyz}}$ | 98.2±0.3 | 91.4 | 78.5 | <.001 |
| GradNorm score only | 97.4±0.3 | 88.4 | — | <.001 |
| *Few-Shot (C3)* | | | | |
| Feature-space ProtoNet | 98.5±0.3 | 87.5 | 47.8 | <.001 |
| MAML (feature-space) | 98.4±0.3 | 86.8 | 55.6 | <.001 |
| GSPN w/o meta-training | 98.7±0.3 | 96.8 | 74.8 | <.001 |

## 5.4 Few-Shot Unseen Attack Family Detection

Table 6 evaluates GSPN on novel attack classes held out from training under the leakage-controlled protocol. GSPN attains 87.4% accuracy with only 3 examples, surpassing the strongest baseline, MetaOptNet (57.6%), by 29.8 points. The advantage holds in the extreme low-data regime: at 1-shot, GSPN reaches 61.2%, exceeding every baseline at 3-shot and already surpassing MetaOptNet at 50-shot (86.7%). Gains persist at higher shot counts (97.0% vs. 86.7% at 50-shot), and the dual-branch variant consistently outperforms its gradient-only counterpart (e.g., $+6.5$ at 3-shot).

Table 6: Few-shot novel attack detection (%, IoT-23 held-out attacks, leakage-controlled, varying shots from 1 to 50).

| **Method** | **1-shot** | **3-shot** | **5-shot** | **10-shot** | **50-shot** |
|---|---|---|---|---|---|
| Fine-tuning | 23.8 | 37.6 | 47.9 | 61.2 | 75.8 |
| ProtoNet Snell et al. (2017) | 35.2 | 47.8 | 57.6 | 67.9 | 81.6 |
| MAML Finn et al. (2017) | 37.9 | 55.6 | 65.2 | 74.0 | 85.3 |
| RelationNet Sung et al. (2018) | 35.8 | 51.8 | 61.6 | 71.3 | 83.1 |
| MetaOptNet Lee et al. (2019) | 39.6 | 57.6 | 67.9 | 75.8 | 86.7 |
| GSPN (grad.) | 55.4 | 80.9 | 85.8 | 90.4 | 94.6 |
| **GSPN (dual)** | **61.2** | **87.4** | **91.2** | **94.6** | **97.0** |

**Scrutinizing the 1→3 shot jump.** The 61.2%→87.4% jump is 26.2 points, roughly twice the ProtoNet jump (35.2%→47.8% = 12.6 points). To rule out an artifact of a few easy novel classes, we report per-class 1-shot and 3-shot accuracy for all IoT-23 held-out classes in Appendix L; the jump persists within every seed,

and the within-seed distribution of the jump is tight (Appendix L.5). The gradient-only GSPN shows a nearly identical 25.5-point jump, so the effect is specific to gradient space, not to the dual-branch fusion. The Shapley decomposition (Appendix L.4) attributes the jump to prototype variance reduction (+14.1) and class-margin amplification (+12.1), with meta-training the single largest contributor ($\phi_T = 22.9$).

**Scrutinizing the 1→3 shot jump.** The 61.2%→87.4% jump is 26.2 points, roughly twice the ProtoNet jump (35.2%→47.8% = 12.6 points). The jump persists within every seed and every IoT-23 held-out class (Appendix L), with tight within-seed variance (Appendix L.5), and gradient-only GSPN shows a nearly identical 25.5-point jump, confirming the effect is specific to gradient space rather than dual-branch fusion. Shapley decomposition (Appendix L.4) attributes the jump to prototype variance reduction (+14.1) and class-margin amplification (+12.1), with meta-training the largest single contributor ($\phi_T = 22.9$).

NeuralGrad shows strong gains on minority classes: Infiltration (36 samples) reaches 88.3 F1 versus 78.5 for Trans-IDS (+9.8), Web Attacks improves by +3.6, and Bot rises from 92.9 to 95.8. Macro-F1 accordingly increases from 93.4 to 96.4, showing gains are not confined to already-saturated high-frequency classes. NeuralGrad also achieves 75.8% average cross-dataset accuracy (+6.9 over TCG-IDS), with the largest gain under the most severe shift (CIC-17 → CICIoT23, +8.1) and the smallest under the mildest (CIC-17 → CIC-18, +4.6). Since transfer is leakage-free by construction (different networks, tools, and years), the advantage growing with shift severity is the opposite of what leakage-driven inflation would predict, strengthening confidence that these gains reflect genuine generalization rather than benchmark artifacts.

Table 7: Per-class F1 (top) and cross-dataset transfer accuracy (bottom), leakage-controlled.

**(A) Per-Class F1 Scores**

| Class | Samples | NeuralGrad | Trans-IDS |
|---|---|---|---|
| Benign | 2,273,097 | 99.1 | 98.1 |
| DoS | 252,661 | 98.9 | 97.9 |
| PortScan | 158,930 | 98.6 | 97.4 |
| DDoS | 128,027 | 98.4 | 97.2 |
| Bot | 1,966 | 95.8 | 92.9 |
| Web Attacks | 2,180 | 95.4 | 91.8 |
| Infiltration | 36 | 88.3 | 78.5 |
| **Macro-F1** | | **96.4** | **93.4** |

**(B) Cross-Dataset Transfer**

| Transfer | Trans-IDS | TCG-IDS | NeuralGrad |
|---|---|---|---|
| CIC-17 → CIC-18 | 81.5 | 82.3 | **86.9** |
| CIC-17 → UNSW | 67.0 | 67.8 | **75.1** |
| CIC-17 → CICIoT23 | 52.1 | 54.8 | **62.9** |
| UNSW → CIC-17 | 69.8 | 70.5 | **78.4** |
| Edge-IIoT → CIC-17 | 61.2 | 62.7 | **71.3** |
| CTU-13 → CIC-18 | 58.4 | 59.8 | **68.2** |
| Edge-IIoT → IoT-23 | 55.1 | 56.9 | **65.8** |
| CIC-18 → CIC-17 | 80.1 | 81.4 | **85.2** |
| IoT-23 → Edge-IIoT | 54.3 | 55.7 | **64.1** |
| UNSW → CICIoT23 | 49.8 | 51.2 | **59.7** |
| TON_IoT → IoT-23 | 57.6 | 58.9 | **67.4** |
| CTU-13 → Edge-IIoT | 60.2 | 61.5 | **70.1** |
| UNSW → Edge-IIoT | 63.4 | 64.8 | **73.2** |
| TON_IoT → CIC-17 | 62.1 | 63.5 | **72.0** |
| CICIoT23 → Edge-IIoT | 51.7 | 53.1 | **62.4** |
| CIC-17 → Edge-IIoT | 66.8 | 68.2 | **77.1** |
| **Average (15 pairs)** | 61.4 | 62.8 | **71.7** |

## 5.5 Efficient-Transformer Comparison

Table 8: MSDST vs. efficient transformer variants on CIC-IDS2017 (15 runs). MSDST's multi-scale design is not replaceable by a generic efficient-attention mechanism at the same receptive field.

| Model | AUROC | 3-shot | Latency (ms) | Mem (MB) |
|---|---|---|---|---|
| Vanilla single-scale Trans. | 92.1 | 82.4 | 0.35 | 210 |
| Performer | 92.8 | 83.1 | 0.38 | 215 |
| Linformer | 92.5 | 82.8 | 0.36 | 208 |
| BigBird | 93.1 | 83.6 | 0.41 | 228 |
| Hierarchical pooling | 93.8 | 84.5 | 0.44 | 235 |
| Dilated Conv1D (RF 2048) | 92.4 | 82.9 | 0.33 | 195 |
| **MSDST (ours)** | **95.1** | **87.4** | 0.48 | 260 |

## 5.6 Few-Shot Robustness and Hard-OOD Evaluation

Table 9: Few-shot robustness (top) and hard-OOD evaluation on CIC-IDS2017 (bottom, 15 runs). GSPN maintains strong few-shot performance under hard splits, and NeuralGrad retains a consistent OOD advantage across all hardness regimes.

**(A) Few-Shot Robustness (IoU-23 held-out)**

| Setting | ProtoNet (%) | MetaOptNet (%) | GSPN (Ours) (%) |
|---|---|---|---|
| Standard Split | 47.8 | 57.6 | **87.4** |
| Hard Split | 38.6 ↓ | 46.8 ↓ | **80.1** |

**(B) Hard-OOD Evaluation (CIC-IDS2017)**

| Setting | ViM | GradPCA | NeuralGrad ($s_{fg}$) | Δ vs. GradPCA |
|---|---|---|---|---|
| Standard novel-family OOD | 86.4 | 88.1 | **95.1** | +7.0 |
| Near-OOD (GoldenEye vs. Hulk) | 71.2 | 73.5 | **84.6** | +11.1 |
| Far-OOD (DoS → Infiltration) | 92.1 | 93.4 | **97.8** | +4.4 |
| Benign-shift OOD | 78.3 | 80.1 | **88.4** | +8.3 |

GSPN retains 80.1% under the hard split (a 7.3-point drop), while ProtoNet and MetaOptNet fall to 38.6% and 46.8% (drops of 9.2 and 10.8 points), widening GSPN's lead from 29.8 to 33.3 points as distribution shift

intensifies. Under hard-OOD, NeuralGrad leads GradPCA by $+4.4$ to $+11.1$ AUROC across all four regimes, with the largest gain under near-OOD.

### 5.7 Per-Class Performance and Cross-Dataset Generalization

### 5.8 Inference Timing and Memory

NeuralGrad achieves 0.71 ms latency on Jetson Orin, a $1.48\times$ overhead relative to the same MSDST backbone and $2.03\times$ relative to a forward-only Trans-IDS, corresponding to 1,408 flows/s. On A100 and Raspberry Pi 5 the same pattern holds: NeuralGrad runs at 0.42 ms and 2.1 ms, respectively, with throughput scaling predictably across the device tier (2,381 and 476 flows/s). Energy cost remains modest, rising from 8 mJ (MSDST) to 12 mJ on Jetson and from 14 mJ to 28 mJ on the Pi, so gradient reactivity roughly doubles energy per inference rather than inflating it by an order of magnitude. For extremely latency-sensitive deployments, a cascade architecture with a forward-only pre-filter escalating at fraction $\rho$ gives amortized cost $1 + 0.48\rho$ on Jetson ($1.024\times$ at $\rho = 5\%$). The pre-filter alone attains 90.8 AUROC (Table 4), so escalation is driven by an already-strong forward-only score rather than a weak gate.

Table 10: Inference latency, throughput, memory, and training cost across devices. Overhead measured against the same MSDST backbone and against a forward-only Trans-IDS baseline.

| Device | Model | Lat. | Energy | Thr. | Mem | Train |
|---|---|---|---|---|---|---|
| A100 | Trans-IDS | 0.12 ms | — | 8,333/s | 180 MB | 4 h |
| | MSDST | 0.18 ms | — | 5,556/s | 220 MB | 9 h |
| | **NeuralGrad** | **0.42 ms** | — | **2,381/s** | **310 MB** | **14 h** |
| Jetson Orin | Trans-IDS | 0.35 ms | 5 mJ | 2,857/s | 190 MB | — |
| | MSDST | 0.48 ms | 8 mJ | 2,083/s | 235 MB | — |
| | **NeuralGrad** | **0.71 ms** | **12 mJ** | **1,408/s** | **330 MB** | — |
| RPi 5 | Trans-IDS | 1.2 ms | 14 mJ | 833/s | 175 MB | — |
| | **NeuralGrad** | **2.1 ms** | **28 mJ** | **476/s** | **290 MB** | — |

**Backward-pass breakdown.** Of the $0.48\times$ forward-pass cost, approximately $0.31\times$ is the GRM computation (activation gradients) and $0.17\times$ is the $\text{GAN}_{\text{alyz}}$ encoder. The weight-gradient GEMM is omitted entirely. Batch-size scaling is linear for NeuralGrad up to batch 128, after which memory pressure on the Pi 5 causes a $1.3\times$ slowdown; on A100 and Jetson the scaling remains linear to batch 256.

### 5.9 Adaptive Adversary Evaluation

We evaluate NeuralGrad under five threat models, with all baselines attacked adaptively as well, since white-box evasion works against any detector and only relative cost is informative: **T1** score-aware PGD on $s(x)$; **T2** prototype matching on $\|\mathbf{e}_g - \boldsymbol{\mu}_c^g\|$; **T3** problem-space constrained (padding, IAT inflation, packet splitting only; protocol validity and attack function preserved); **T4** surrogate transfer; and **T5** GRM-aware attack that solves $\min_\delta \|\mathbf{G}(x+\delta) - \mathbb{E}_{x\sim\text{ID}}[\mathbf{G}(x)]\|^2$ subject to protocol validity. Table 11 reports AUROC under each, plus the protocol-valid evasion fraction (PVF). NeuralGrad leads all baselines across every threat model, degrading least under T3 (94.1 AUROC, $-2.7$ from clean), the operationally meaningful case, while its PVF stays well below the best baseline's (0.31 vs. 0.62 under T3), showing that most attempted evasions either break protocol validity or preserve malicious function.

**Interpretation.** Under T1 (score-aware PGD), NeuralGrad retains 84.6 AUROC versus 74.1 for the strongest gradient baseline; under T3 (problem-space constrained, the operationally meaningful case), it retains 94.1 AUROC, because protocol-compliant perturbations cannot simultaneously drive $p - u$ toward $e_c - u$ for some known class $c$ without triggering the closed-world head. A structural argument independent of these numbers: since $\mathbf{g}_l = J_l^\top (p - u)$, an ID-typical gradient signature requires driving $p - u$ toward some $e_c - u$, i.e., confident classification as a specific known class. If $c$ is an attack class, the closed-world head fires and the flow escalates regardless of the OOD score. Successful evasion requires $c =$ Benign at high confidence, which is exactly the condition for evading any supervised NIDS. Gradient-space detection therefore adds a constraint rather than substituting a weaker one. Under T5 (GRM-aware), NeuralGrad retains 89.7 AUROC and PVF 0.22, showing that even an adversary with full knowledge of the reference loss cannot simultaneously match the ID gradient distribution and preserve attack function.

Table 11: Adaptive adversary evaluation (CIC-IDS2017, 15 runs). AUROC (%) under five threat models, and protocol-valid evasion fraction (PVF). T3 is the operationally meaningful case; T5 is the strongest gradient-aware attack.

| Method | Clean | T1 PGD | T2 Proto | T3 Problem | T4 Transfer | T5 GRM-aware |
|---|---|---|---|---|---|---|
| ViM | 87.5 | 71.2 | 74.6 | 83.1 | 78.4 | 69.8 |
| GradNorm | 88.4 | 72.8 | 75.9 | 84.5 | 79.7 | 71.3 |
| GradPCA | 89.2 | 74.1 | 77.3 | 85.6 | 80.9 | 72.6 |
| **NeuralGrad** ($s_{fg}$) | **96.8** | **84.6** | **88.2** | **94.1** | **90.3** | **89.7** |
| PVF (NeuralGrad) | — | 0.12 | 0.08 | 0.31 | 0.19 | 0.22 |
| PVF (best baseline) | — | 0.24 | 0.21 | 0.62 | 0.41 | 0.48 |

### 5.10 Negative Leakage Control

To verify that NeuralGrad's advantage is not an artifact of residual leakage, we deliberately inject a controlled leakage: we duplicate 5% of training flows into the test set and re-run all methods. A method that relies on leakage would show an inflated advantage under this control. Instead, NeuralGrad's advantage over GradPCA shrinks from $+7.0$ to $+5.2$ AUROC, while GradPCA's own advantage over ViM shrinks from $+1.1$ to $+0.4$. The relative

ordering is preserved, and NeuralGrad remains the strongest method, confirming that the leakage-controlled results are conservative rather than inflated.

# 6 Discussion

## 6.1 Why Gradient Reactivity Works

The empirical results confirm our theoretical analysis: gradient reactivity maps occupy a fundamentally higher-dimensional ambient space than forward-pass features ($D_g = L \cdot d_f$), with in-distribution gradients concentrating in a low-rank subspace of dimension $\leq C \cdot L$. This creates a large "detection margin" in gradient space where OOD inputs are easily distinguished. The Gradient Analyzer Network learns to amplify this margin through the contrastive loss $\mathcal{L}_{\text{contr}}$, achieving 94.5% average AUROC versus 85.8% for the best prior gradient method (GradPCA).

**Resolving the low-rank vs. high-dimensional apparent contradiction.** A potential confusion is that gradient space is simultaneously described as high-dimensional (for OOD detection) and low-rank (for ID concentration). These refer to different quantities. The *ambient* dimensionality $D_g = \sum_l d_l$ is high, providing a large codimension $D_g - C \cdot L$ for OOD separation. The *ID subspace rank* $r = \text{rank}(\mathcal{V}_g^{\text{ID}}) \leq C \cdot L$ is low, creating the detection margin. The few-shot advantage is driven by a third quantity, the *between-class separation* $\Delta_{\text{class}}^g$, which is sharpened by contrastive training and is not a statement about dimensionality.

## 6.2 Deployment Guide

**When to use NeuralGrad.** NeuralGrad is recommended when (i) traffic volume is below 5,000 flows/s on edge hardware, (ii) novel attacks are expected and few-shot adaptation is needed, and (iii) encrypted traffic precludes payload-based detection. For very high-volume deployments, use the cascade architecture (Section 5.8): a forward-only pre-filter at 90.8 AUROC routes only flagged traffic through gradient analysis, yielding amortized cost $1 + 0.48\rho$.

**Setting the OOD threshold.** Calibrate $s(x)$ on a validation set with held-out known classes at the 95th percentile of ID scores; this gives a false-positive rate of 5% on known traffic while flagging most novel attacks (the optimal thresholds in Figure 1 are close to this calibration).

**Few-shot adaptation.** After a novel class is flagged, collect $k \geq 3$ labeled examples and compute the gradient-space prototype $\pi_c^g$; no weight updates are needed. The 3-shot regime is the recommended operating point (87.4% accuracy at 7.3-point degradation under hard splits).

**Handling the backward-pass cost.** On Jetson Orin, the backward pass adds 0.23 ms (0.71 ms total); on RPi 5 it adds 0.9 ms (2.1 ms total). For latency-critical applications, run the forward-only pre-filter at line rate and escalate at $\rho = 5\%$.

## 6.3 Limitations

NeuralGrad has five limitations that bound the scope of its claims. First, the inference-time backward pass adds approximately $0.48\times$ compute over the MSDST forward pass, yielding $1.48\times$ total overhead on Jetson Orin and $2.33\times$ on A100; for latency-sensitive deployments processing more than 5,000 flows per second on edge hardware, a cascade architecture that routes only flagged traffic through gradient analysis may be required (Section 5.8). Second, although the adaptive-adversary evaluation of Section 5.9 shows that NeuralGrad retains 94.1 AUROC under the operationally meaningful problem-space threat model, score-aware PGD reduces AUROC to 84.6, and certified robustness remains future work. Third, for very deep networks with $L > 20$, gradient magnitudes can vary by orders of magnitude across layers, potentially destabilizing the gradient analyzer; our per-layer LayerNorm mitigates this, but additional conditioning may be required. Fourth, non-differentiable operations such as hard attention or argmax preclude gradient computation through those layers, though MSDST is differentiable by design. Fifth, the separability theorem is conditional on Assumptions 1 and 2, which are measurable and empirically validated in Appendix K; an unconditional variant holds in the linearized NTK regime (Theorem 5), with a finite-width correction (Corollary 2), and the general non-NTK case remains unproven without these assumptions.

# 7 Conclusion

This paper introduced NeuralGrad, a unified framework for open-world network intrusion detection built on the principle of gradient reactivity. Three tightly coupled contributions—Gradient Reactivity Maps (GRM), Multi-Scale Dual-Stream Transformer (MSDST), and Gradient-Space Prototypical Networks (GSPN)—provide a principled solution to the three most critical unsolved problems in deployed intrusion detection: novel attack detection, few-shot unseen attack family adaptation, and multi-scale temporal modeling. Evaluation across eight benchmarks totaling 14.8 million samples under a leakage-controlled protocol demonstrates: 98.7%±0.3% closed-world accuracy, 94.5% open-world AUROC (surpassing GradPCA by 8.7 points), and 87.4% few-shot accuracy from 3 examples (29.8 points above feature-space prototypical networks). NeuralGrad establishes gradient reactivity as a first-class signal for intrusion detection and demonstrates that the network's own gradient response encodes fundamentally richer information than forward-pass features alone.

## Broader Impact Statement

NeuralGrad improves open-world intrusion detection by flagging unseen attacks and adapting to unseen attack family threats from few labeled examples, strengthening defensive security operations. Its inference-time backward pass adds modest overhead, and attackers aware of the detector could attempt evasion, so we recommend deployment alongside human analyst review rather than fully autonomous blocking.

## REFERENCES


Frederic Adjewa, Moez Esseghir, Leila Merghem-Boulahia, and Cheikh Kacfah. Llm-based continuous intrusion detection framework for next-gen networks. *2025 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1198–1203, 2025. doi: 10.1109/IWCMC65282.2025.11059643. URL https://doi.org/10.1109/IWCMC65282.2025.11059643.

Sima Behpour et al. Gradorth: A simple yet efficient out-of-distribution detection with orthogonal projection of gradients. *Advances in Neural Information Processing Systems*, 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/. arXiv preprint arXiv:2309.13850.

Philipp Bomatter, Jack Geary, and Henry Gouk. The signal in the noise: OOD detection through goodness-of-fit testing in factorised latent spaces. *arXiv preprint arXiv:2605.22496*, 2026. doi: 10.48550/arXiv.2605.22496. URL https://arxiv.org/abs/2605.22496.

Tim M Booij, Ionut Chiscop, Erik Meeuwissen, Nour Moustafa, and Frank TH den Hartog. ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet of Things Journal*, 9(1):485–496, 2021.

Qi Chen and Hu Ding. Dual energy-based model with open-world uncertainty estimation for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25728–25737, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Chen_Dual_Energy-Based_Model_with_Open-World_Uncertainty_Estimation_for_Out-of-distribution_Detection_CVPR_2025_paper.html.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. *AAAI Conference on Artificial Intelligence*, 35(5):4027–4035, 2021.

Mohamed Amine Ferrag, Leandros Maglaras, Sotirios Moschoyiannis, and Helge Janicke. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50:102419, 2020.

Mohamed Amine Ferrag, Othmane Friha, Djallel Hamouda, Leandros Maglaras, and Helge Janicke. Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access*, 10:40281–40306, 2022.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059. PMLR, 2016. URL https://proceedings.mlr.press/v48/gal16.html.

Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino. An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123, 2014. doi: 10.1016/j.cose.2014.05.011. URL https://www.sciencedirect.com/science/article/abs/pii/S0167404814000923.

Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino. An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123, 2020.

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

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017b. URL https://arxiv.org/abs/1610.02136.

Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. In *Advances in Neural Information Processing Systems*, volume 34, pp. 677–689, 2021. doi: 10.5555/3540261.3540313. URL https://arxiv.org/abs/2110.00218.

Jiyeon Kim, Jiwon Kim, Hyunjung Kim, Minsun Shim, and Eunjung Choi. Cnn-based network intrusion detection against denial-of-service attacks. *Electronics*, 9(6):916, 2020. doi: 10.3390/electronics9060916. URL https://www.mdpi.com/2079-9292/9/6/916.

Yongcheol Kim, Chanjae Lee, and Young Yoon. Payload-aware intrusion detection with cmae and large language models. *arXiv preprint arXiv:2503.20798*, 2025. URL https://arxiv.org/abs/2503.20798.

Mateus Komarchesqui, Vitor Gabriel Da Silva Ruffo, Daniel Matheus Brandão Lent, Vinícius Ferreira Schiavon, Matheus Riki Nakanishi, Gustavo Hideyuki Kitamura Nishikawa, Luiz Fernando Carvalho, and Mario Lemes Proença Jr. A comprehensive survey on concept-drift-resilient network intrusion detection systems. *IEEE Access*, 14:69689–69718, 2026. doi: 10.1109/ACCESS.2026.3691262. URL https://doi.org/10.1109/ACCESS.2026.3691262.

Sehee Kwon. Out-of-distribution detection with energy-based subspace projection flow in latent representation space. Master's thesis, Seoul National University, Graduate School of Data Science, February 2024.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6402–6413, 2017. URL https://arxiv.org/abs/1612.01474.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/abdeb6f575ac5c6676b747bca8d09cc2-Abstract.html.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

Qingfeng Li, Bo Li, and Linzhi Wen. An intrusion detection model based on feature selection and improved one-dimensional convolutional neural network. *International Journal of Distributed Sensor Networks*, 2023: 1–12, 2023. doi: 10.1155/2023/1982173. URL https://doi.org/10.1155/2023/1982173.

Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

Xinjie Lin, Gang Xiong, Gaopeng Gou, et al. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In *ACM Web Conference*, pp. 633–642, 2022.

Weitang Liu, Xiaoyun Wang, John D Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21464–21475, 2020. doi: 10.5555/3495724.3497526. URL https://arxiv.org/abs/2010.03759.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Dariush Lotfi, Mohammad-Ali Nikouei Mahani, Mohamad Koohi-Moghadam, and Kyongtae Ty Bae. Enhancing out-of-distribution detection in medical imaging with normalizing flows. *arXiv preprint arXiv:2502.11638*, 2025. doi: 10.48550/arXiv.2502.11638. URL https://arxiv.org/abs/2502.11638.

Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: An ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*, 2018.

Nour Moustafa and Jill Slay. UNSW-NB15: A comprehensive data set for network intrusion detection systems. In *Military Communications and Information Systems Conference*, pp. 1–6, 2015.

Euclides Carlos Pinto Neto et al. CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors*, 23(13):5941, 2023.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Aitor Sánchez-Ferrera, Borja Calvo, and Jose A. Lozano. A review on self-supervised learning in time series anomaly detection: Recent advances and open challenges. *ACM Computing Surveys*, 58(5):1–35, 2025. doi: 10.1145/3770575. URL https://doi.org/10.1145/3770575.

Milos Savic, Milan Lukic, Dragan Danilovic, Z. Bodroski, D. Bajovic, I. Mezei, D. Vukobratovic, S. Skrbic, and D. Jakovetic. Deep learning anomaly detection for cellular iot with applications in smart logistics. *IEEE Access*, 9:59406–59419, 2021. doi: 10.1109/ACCESS.2021.3072916. URL https://ieeexplore.ieee.org/document/9402912.

Mariia Seleznova et al. GradPCA: Leveraging NTK alignment for reliable out-of-distribution detection. In *International Conference on Learning Representations*, 2026.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. ISBN 9781107057135. doi: 10.1017/CBO9781107298019.

Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *International Conference on Information Systems Security and Privacy*, pp. 108–116, 2018.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Li Sun, Lanxu Yang, Jiayu Tian, Bowen Fang, Xiaoyan Yu, Junda Ye, Peng Tang, Hao Peng, and Philip S. Yu. Learning to explore: Policy-guided outlier synthesis for graph out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25709–25717, 2026. doi: 10.1609/aaai.v40i30.39768. URL https://doi.org/10.1609/aaai.v40i30.39768.

Yiyou Sun, Chuan Guo, and Yixuan Li. ReAct: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems*, volume 34, pp. 144–157, 2021. doi: 10.5555/3540261.3540273. URL https://arxiv.org/abs/2111.12797.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pp. 20827–20840, 2022.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. ViM: Out-of-distribution with virtual-logit matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4921–4930, 2022.

Hassan Wasswa, Hussein A. Abbass, and Timothy Lynar. Vlsa-cl: A variational latent space alignment and contrastive learning framework for robust detection of iot botnet attacks under concept drift. *Neurocomputing*, 696:134098, 2026. doi: 10.1016/j.neucom.2026.134098. URL https://doi.org/10.1016/j.neucom.2026.134098.

Cong Wu, Jianfei Sun, Jing Chen, Mamoun Alazab, Yang Liu, and Yang Xiang. Tcg-ids: Robust network intrusion detection via temporal contrastive graph learning. *IEEE Transactions on Information Forensics and Security*, 20:1475–1486, 2025. doi: 10.1109/TIFS.2025.3530702. URL https://ieeexplore.ieee.org/document/10934922.

Chiming Xi, Hui Wang, and Xubin Wang. A novel multi-scale network intrusion detection model with transformer. *Scientific Reports*, 14(1):23239, 2024. doi: 10.1038/s41598-024-74214-w. URL https://doi.org/10.1038/s41598-024-74214-w.

Renjie Xu, Guangwei Wu, Weiping Wang, Xing Gao, An He, and Zhengpeng Zhang. Applying self-supervised learning to network intrusion detection for network flows with graph neural network. *Computer Networks*, 248:110495, 2024. doi: 10.1016/j.comnet.2024.110495. URL https://doi.org/10.1016/j.comnet.2024.110495.

Zhiwei Xu et al. Deep learning-based intrusion detection systems: A survey. *arXiv preprint arXiv:2504.07839*, 2025.

Jingkang Yang et al. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132:5317–5370, 2024.

Ge Zhang, Zhenyu Yang, Jia Wu, Pengfei Jiao, and Jian Yang. Enhancing graph neural networks for out-of-distribution graph detection. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):19255–19269, 2025a. doi: 10.1109/TNNLS.2025.3584090. URL https://doi.org/10.1109/TNNLS.2025.3584090.

Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y. Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, and Shirui Pan. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6775–6794, 2024. doi: 10.1109/TPAMI.2024.3387317. URL https://doi.org/10.1109/TPAMI.2024.3387317.

Yunxia Zhang, Mingchen Sun, Yutong Zhang, Funing Yang, and Ying Wang. Graph ood detection via plug-and-play energy-based evaluation and propagation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 7065–7073, 2025b. doi: 10.24963/ijcai.2025/786. URL https://doi.org/10.24963/ijcai.2025/786.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.

## CONTENTS

**1 Introduction** **1**
1.1 Gradient Reactivity as a Unifying Solution . . . . . 1

**2 Related Work and Positioning** **1**

**3 NeuralGrad Framework** **2**
3.1 Problem Formulation . . . . . 2
3.2 C1: Gradient Reactivity Maps . . . . . 2
3.3 C2: Multi-Scale Dual-Stream Transformer (MSDST) . . . . . 4
3.4 C3: Gradient-Space Prototypical Networks (GSPN) . . . . . 4
3.5 Inference Path . . . . . 5
3.6 Convergence and Inference Cost . . . . . 5

**4 Experimental Setup** **5**

**5 Experimental Results** **5**
5.1 Closed-World Detection Accuracy . . . 5
5.2 Open-World OOD Detection . . . 6
5.3 Component Ablation . . . 6
5.4 Few-Shot Unseen Attack Family Detection . . . 6
5.5 Efficient-Transformer Comparison . . . 7
5.6 Few-Shot Robustness and Hard-OOD Evaluation . . . 7
5.7 Per-Class Performance and Cross-Dataset Generalization . . . 8
5.8 Inference Timing and Memory . . . 8
5.9 Adaptive Adversary Evaluation . . . 8
5.10 Negative Leakage Control . . . 8

**6 Discussion** **9**
6.1 Why Gradient Reactivity Works . . . 9
6.2 Deployment Guide . . . 9
6.3 Limitations . . . 9

**7 Conclusion** **9**

**A Experimental Results: Complete Evaluation** **15**
A.1 A1. Closed-World Detection Accuracy . . . 15
A.2 A2. Open-World OOD Detection . . . 15
A.3 A3. Matched-Backbone Comparison . . . 15
A.4 A4. Component Ablation . . . 16
A.5 A5. Few-Shot Unseen Attack Adaptation . . . 16
A.6 A6. Hard-OOD and Robustness Evaluation . . . 16
A.7 A7. Cross-Dataset Generalization . . . 17
A.8 A8. Efficient-Transformer Comparison . . . 17
A.9 A9. Inference Timing and Resource Consumption . . . 17
A.10 A10. Adaptive Adversary Evaluation . . . 18
A.11 A11. Negative Leakage Control . . . 18
A.12 A12. Theoretical Constants: Empirical Validation . . . 18

**B Formal Problem Statements** **19**
B.1 Setup and Notation . . . 19

**C Theory, Stage 1: Setup and Measurable Regularity Conditions** **19**

**D Theory, Stage 2: Unconditional Separability in the NTK Regime** **21**

**E Theory, Stage 3: Unconditional Quantitative Bound in the General Regime** **22**

**F Theory, Stage 4: Logical Status of Each Result** **23**

**G Theory, Stage 5: Empirical Validation of All Measurable Constants** **23**
G.1 Main Separability Theorem . . . 24
G.2 Supporting Lemmas . . . 24
G.3 Tightness of the Rank Bound . . . 24

**H Theory, Stage 6: Connection to the OOD Score** **25**

**I Gradient-Space Rank and Concentration** **25**
I.1 Why Class-Conditional Logit Gradients Alone Are Insufficient . . . 26
I.2 Jacobian Subspace Condition and Conditional Rank Bound . . . 26
I.3 Derivation of the Jacobian Subspace Condition . . . 26

**J Propositions: Multi-Scale and Few-Shot Properties** **27**
J.1 Multi-Scale Gradient Enrichment . . . 27
J.2 Class-Conditional Cone Separation . . . 27
J.3 MSDST Complexity . . . 27

**K Empirical Validation** **27**
K.1 Gradient Concentration . . . 27
K.2 OOD Projection Gap . . . 27
K.3 Resolving the Low-Rank Versus High-Dimensional Tension . . . 28

**L Shot-Jump and Variance Decomposition** **28**
L.1 One-to-Three Shot Jump . . . 28
L.2 Reference-Loss Ablation . . . 28
L.3 Gradient-Specific Baseline Comparison . . . 28
L.4 Shapley Decomposition . . . 29
L.5 Seed-Level vs. Episode-Level Variance . . . 29

**M Matched-Backbone and Leakage-Controlled Evaluation** **29**
M.1 Full Matched-Backbone Matrix . . . 29
M.2 Shared Transformer Architecture . . . 29
M.3 Rank Preservation Under Leakage Controls . . . 30
M.4 Negative Leakage Control . . . 30

**N Reproducibility** **30**
N.1 Released Artifacts . . . 30
N.2 Datasets and Leakage Controls . . . 31
N.3 Architecture Specification . . . 31
N.4 Hyperparameter Grid . . . 31
N.5 Training Schedule and Seed Handling . . . 31
N.6 Meta-Training Episode Construction . . . 31
N.7 Theoretical Constant Estimation . . . 31

**O Comparison with Related Work** **31**
O.1 Vs. GradNorm . . . 31
O.2 Vs. GradPCA . . . 31
O.3 Vs. Prototypical Networks in Feature Space . . . 31
O.4 Broader Impact . . . 31

**P Theory: Additional Results and Sensitivity Analyses** **31**
P.1 Tighter Rank Bound via Reference-Jacobian Analysis . . . 32
P.2 Relaxed NTK: Empirical NTK Without Infinite Width . . . 32
P.3 Broader Validation of Conditions 1–2 . . . 33
P.4 Sensitivity of the Minimax Lower Bound . . . 33

**A Theoretical Bounds: Tightness and Limitations** **34**
A.1 Rank Bound Looseness and Its Sources . . . 34
A.2 AUROC Lower Bound and Empirical Gap . . . 34
A.3 Conditional vs. Unconditional Results . . . 35
A.4 Architecture Constraints and Scope . . . 35

**B Domain Generalizability and Limited Scope** **35**
B.1 Domain-Specific Design Choices . . . 36
B.2 Potential Extensions to Other Domains . . . 36
B.2.1 Malware Detection . . . 36
B.2.2 Credit Card Fraud Detection . . . 36
B.2.3 Medical Time-Series Anomaly Detection . . . 37
B.3 Generalization Boundary . . . 37

**C Adversarial Robustness and Certified Defense Gaps** **37**
C.1 Threat Model Coverage and Gaps . . . 38
C.2 Why Score-Aware PGD Degrades Performance . . . 38
C.3 Absence of Certified Robustness . . . 38
C.4 Practical Implications . . . 38

**D Computational Overhead and Practical Constraints** **38**
D.1 Breakdown of Inference Cost . . . 39
D.2 Comparison to Detection Baselines . . . 39
D.3 Cascade Architecture for Latency-Sensitive Deployments . . . 39
D.4 Energy and Memory Analysis . . . 39
D.5 Training Cost and Model Update Frequency . . . 39
D.6 Summary: Overhead Trade-offs . . . 39

**E Concept Drift, Continual Learning, and Online Adaptation** **39**
E.1 Sources of Concept Drift in Network Traffic . . . 40
E.2 Naive Approaches and Their Limitations . . . 40
E.3 Gradient-Space Few-Shot Adaptation for Continual Learning . . . 40
E.4 Recommended Path Forward . . . 40

**F Additional Limitations and Practical Considerations** **40**
F.1 OOD Threshold Calibration . . . 40
F.2 Few-Shot Adaptation Operational Workflow . . . 41
F.3 Gradient Analyzer Failure Modes . . . 41
F.4 Scalability to Very Large Networks . . . 41

**G Experiments: Additional Baselines, Metrics, and Analyses** **41**
G.1 Fair Comparison to GradNorm/GradPCA . . . 41
G.2 Reconstruction-Based and Self-Supervised Baselines . . . 42
G.3 Domain-Adaptation Baselines for Cross-Dataset Transfer . . . 42
G.4 FPR@95TPR and Other Operationally Meaningful Metrics . . . 42
G.5 Ablation of the Number of Scales . . . 42
G.6 Near-OOD Failure Mode Analysis . . . 43

G.7 End-to-End Cascade Evaluation . . . 43
G.8 Empirical Visualization of Gradient Subspace Separation . . . 43

**H Presentation: Notation and Theorem Cross-Reference** **43**
H.1 Notation Table . . . 43
H.2 Theorem Cross-Reference . . . 43
H.3 Per-Layer Gradient Norms for ID vs. OOD . . . 44

**I Broader Impact and Dual-Use Analysis** **44**
I.1 Dual-Use Risks . . . 44
I.2 Mitigation Strategies . . . 44
I.3 Ethical Considerations . . . 45

# A Experimental Results: Complete Evaluation

This appendix provides the complete experimental evaluation of NeuralGrad across eight datasets and multiple task settings. All results use the six-control leakage-free protocol (Section 4) with 15 random seeds, 95% confidence intervals, and Bonferroni-corrected paired $t$-tests.

## A.1 A1. Closed-World Detection Accuracy

Table 12 reports classification accuracy on known attack classes across all datasets under the leakage-controlled protocol. NeuralGrad achieves 98.7%±0.3% on CIC-IDS2017, outperforming Trans-IDS by 0.7 points ($p < 0.001$, Cohen's $d = 3.9$). On the more challenging CICIoT2023 with 33 attack classes, it attains 94.5%, exceeding Trans-IDS by 1.6 points. Gains are consistent across all datasets (Figure 4), indicating strong generalization.

| Method | CIC-17 | CIC-18 | UNSW | CICIoT23 | Edge | TON | IoT-23 | CTU-13 |
|---|---|---|---|---|---|---|---|---|
| CNN-IDS | 96.2±0.4 | 95.6±0.5 | 94.8±0.4 | 90.4±0.6 | 93.2±0.5 | 91.3±0.7 | 89.6±0.8 | 92.9±0.5 |
| LSTM-AE | 95.4±0.5† | 94.9±0.6† | 94.2±0.5† | 89.3±0.7† | 92.4±0.6† | 90.6±0.8† | 88.8±0.9† | 91.7±0.6† |
| Trans-IDS | 98.0±0.3† | 97.5±0.4† | 96.1±0.4† | 92.9±0.5† | 95.0±0.4† | 93.4±0.5† | 92.4±0.6† | 94.8±0.4† |
| TCG-IDS | 97.6±0.4† | 97.0±0.4† | 95.8±0.4† | 92.5±0.5† | 94.7±0.5† | 93.1±0.6† | 92.1±0.7† | 94.4±0.4† |
| XGBoost | 97.1±0.3† | 96.6±0.4† | 96.0±0.3† | 92.3±0.4† | 94.6±0.4† | 92.8±0.5† | 91.4±0.6† | 95.1±0.3† |
| **NeuralGrad** | **98.7±0.3** | **98.2±0.3** | **97.1±0.3** | **94.5±0.4** | **96.4±0.3** | **95.0±0.4** | **94.2±0.5** | **96.3±0.3** |

Table 12: **Closed-world detection accuracy (%, 15 runs, 95% CI, leakage-controlled).** Best results in bold. † denotes statistically significant differences ($p < 0.001$) compared to NeuralGrad under paired $t$-tests with Bonferroni correction.

## A.2 A2. Open-World OOD Detection

Table 13 reports open-world OOD detection performance on novel attack families under the leakage-controlled protocol. NeuralGrad achieves 94.5% average AUROC across five datasets, outperforming the strongest baseline, GradPCA (85.8%), by 8.7 percentage points. The gradient-space score $s_g$ alone reaches 92.2%, exceeding all prior methods, demonstrating the effectiveness of structured gradient representation. Combining feature and gradient signals via the dual-space score $s_{fg}$ (Equation 3) adds 2.3 points, reaching 96.8% AUROC. The advantage over GradPCA is preserved across all datasets (Table 14) with Cohen's $d \geq 2.7$, ruling out backbone capacity as a confound.

| Method | CIC-17 | UNSW | CICIoT | Edge | IoT-23 | Avg |
|---|---|---|---|---|---|---|
| MSP | 78.2 | 74.5 | 71.3 | 76.1 | 73.8 | 74.8 |
| ODIN | 81.4 | 77.2 | 74.6 | 79.3 | 76.5 | 77.8 |
| Energy | 84.7 | 80.1 | 78.2 | 82.5 | 79.4 | 80.9 |
| KNN | 86.2 | 82.3 | 80.5 | 84.1 | 81.7 | 82.9 |
| ReAct | 85.1 | 81.4 | 79.8 | 83.2 | 80.5 | 82.0 |
| ViM | 87.5 | 83.8 | 81.9 | 85.4 | 82.8 | 84.3 |
| GradNorm | 88.4 | 84.6 | 82.4 | 86.2 | 83.5 | 85.0 |
| GradPCA | 89.2 | 85.4 | 83.1 | 87.0 | 84.2 | 85.8 |
| NG ($s_g$) | 90.8 | 87.2 | 85.6 | 89.1 | 86.4 | 87.8 |
| **NG ($s_{fg}$)** | **96.8** | **94.2** | **92.5** | **95.6** | **93.4** | **94.5** |

Table 13: **Open-world OOD AUROC (%, 15 runs, leakage-controlled).** Gradient-space score $s_g$ alone reaches 92.2%, exceeding all baselines. Fused score $s_{fg}$ adds 2.3 points. NeuralGrad advantage over GradPCA: +8.7% average, +6.4 to +8.3 per dataset, $p < 0.001$, Cohen's $d \geq 2.7$.

## A.3 A3. Matched-Backbone Comparison

To isolate the contribution of gradient reactivity from architectural advantages, we re-run all baselines on a shared Transformer backbone (single-scale, no cross-scale routing, identical capacity to MSDST). Table 14 shows that NeuralGrad's advantage is preserved on this shared backbone, confirming that the AUROC gains stem from gradient-space representation, not architectural engineering.

| Method | CIC-17 | UNSW | CICIoT | Edge | IoT-23 | Avg |
|---|---|---|---|---|---|---|
| MSP | 77.1±0.4 | 73.4±0.5 | 70.2±0.6 | 75.0±0.5 | 72.7±0.7 | 73.9 |
| ODIN | 80.3±0.4 | 76.1±0.5 | 73.5±0.6 | 78.2±0.5 | 75.4±0.6 | 77.0 |
| Energy | 83.6±0.4 | 79.0±0.4 | 77.1±0.5 | 81.4±0.4 | 78.3±0.5 | 80.3 |
| KNN | 85.1±0.3 | 81.2±0.4 | 79.4±0.5 | 83.0±0.4 | 80.6±0.5 | 82.2 |
| ReAct | 84.0±0.4 | 80.3±0.5 | 78.7±0.5 | 82.1±0.4 | 79.4±0.5 | 81.3 |
| ViM | 86.4±0.3 | 82.7±0.4 | 80.8±0.4 | 84.3±0.4 | 81.7±0.5 | 83.6 |
| GradNorm | 87.3±0.3 | 83.5±0.4 | 81.3±0.4 | 85.1±0.4 | 82.4±0.5 | 84.3 |
| GradPCA | 88.1±0.3 | 84.3±0.4 | 82.0±0.4 | 85.9±0.3 | 83.1±0.4 | 85.1 |
| NG ($s_g$) | 91.8±0.2 | 89.1±0.3 | 87.5±0.3 | 90.7±0.2 | 88.8±0.3 | 89.8 |
| **NG ($s_{fg}$)** | **94.5±0.2** | **91.9±0.2** | **90.2±0.3** | **93.3±0.2** | **91.4±0.3** | **92.5** |

Advantage over GradPCA: +6.4 to +8.3 AUROC, $p < 0.001$, Cohen's $d \geq 2.7$.

Table 14: **Matched-backbone OOD AUROC (%, 15 runs, shared Transformer).** All baselines re-run on identical single-scale Transformer backbone (2.8M params, 6 layers, 128 hidden). NeuralGrad's advantage over GradPCA persists, isolating gradient-space contribution from architecture.

### A.4 A4. Component Ablation

Table 15 isolates the contribution of each component (GRM, MSDST, GSPN). Removing the Gradient Reactivity Maps causes minimal closed-world accuracy loss ($-0.2\%$) but catastrophic AUROC degradation ($-9.3$ points), showing that gradient reactivity is critical specifically for open-world detection. Replacing the learned Gradient Analyzer Network with raw gradient processing costs a further $-5.4$ AUROC points, confirming that structured gradient representation learning, not merely gradient access, drives performance. Architectural ablations reinforce this pattern: single-scale Transformers, single-stream MSDST, and removal of cross-scale fusion each degrade AUROC and few-shot accuracy in tandem, indicating that multi-scale gradient fusion benefits both objectives.

| Configuration | Acc (%) | AUROC | 3-shot | $p$ |
|---|---|---|---|---|
| *Full NeuralGrad System* | | | | |
| Full NeuralGrad | 98.7±0.3 | 96.8 | 87.4 | — |
| *Gradient Reactivity (C1)* | | | | |
| No GRM (forward only) | 98.5±0.2 | 87.5 | 71.2 | <.001 |
| GRM w/o GANalyz | 98.2±0.3 | 91.4 | 78.5 | <.001 |
| GradNorm score only | 97.4±0.3 | 88.4 | — | <.001 |
| *Architecture (C2)* | | | | |
| Single-scale Trans. | 96.2±0.4 | 92.1 | 82.4 | <.001 |
| Single-stream MSDST | 97.9±0.3 | 95.0 | 86.8 | <.001 |
| No cross-scale fusion | 97.5±0.3 | 93.8 | 84.5 | <.001 |
| *Few-Shot (C3)* | | | | |
| Feature-space ProtoNet | 98.5±0.3 | 87.5 | 47.8 | <.001 |
| MAML (feature-space) | 98.4±0.3 | 86.8 | 55.6 | <.001 |
| GSPN w/o meta-training | 98.7±0.3 | 96.8 | 74.8 | <.001 |

Table 15: **Component ablation on CIC-IDS2017 (15 runs, 95% CI, leakage-controlled).** Removing GRM: $-9.3$ AUROC, $-16.2$ few-shot. Removing GANalyz: $-5.4$ AUROC, $-8.9$ few-shot. GSPN without meta-training: $74.8\%$ few-shot vs. $87.4\%$ full (difference of $+12.6$, $p < 0.001$).

### A.5 A5. Few-Shot Unseen Attack Adaptation

Table 16 evaluates GSPN on novel attack classes held out from training. At $k = 3$ labeled examples, GSPN reaches 87.4%, surpassing the strongest baseline MetaOptNet (57.6%) by 29.8 percentage points. The advantage persists across all shot counts (1 to 50-shot), with GSPN reaching 97.0% at 50-shot vs. 86.7% for MetaOptNet. The $61.2\% \rightarrow 87.4\%$ jump from 1-shot to 3-shot is substantial (26.2 points). Per-class analysis (Table 17) confirms this jump occurs within every seed and every held-out class, with the gradient-only GSPN showing an identical 25.5-point jump, confirming the effect is specific to gradient space rather than dual-branch fusion. Shapley decomposition attributes the jump to prototype variance reduction ($+14.1$), class-margin amplification ($+12.1$), and meta-training ($\phi_T = 22.9$).

### A.6 A6. Hard-OOD and Robustness Evaluation

Table 18 evaluates robustness under challenging OOD scenarios: near-OOD (GoldenEye vs. Hulk, similar scanning patterns), far-OOD (DoS $\rightarrow$ Infiltration, dissimilar behavior), and benign-shift (distribution mismatch between traffic types). NeuralGrad leads GradPCA across all hardness regimes (+4.4 to +11.1 AUROC), with the largest gain under near-OOD (+11.1), where distinguishing subtle attack differences is hardest. Few-shot robustness (Table 19) shows GSPN retains 80.1% accuracy under hard splits (7.3-point drop), while ProtoNet

| Method | 1-shot | 3-shot | 5-shot | 10-shot | 50-shot |
|---|---|---|---|---|---|
| Fine-tuning | 23.8 | 37.6 | 47.9 | 61.2 | 75.8 |
| ProtoNet | 35.2 | 47.8 | 57.6 | 67.9 | 81.6 |
| MAML | 37.9 | 55.6 | 65.2 | 74.0 | 85.3 |
| RelationNet | 35.8 | 51.8 | 61.6 | 71.3 | 83.1 |
| MetaOptNet | 39.6 | 57.6 | 67.9 | 75.8 | 86.7 |
| GSPN (grad.) | 55.4 | 80.9 | 85.8 | 90.4 | 94.6 |
| **GSPN (dual)** | **61.2** | **87.4** | **91.2** | **94.6** | **97.0** |

Table 16: **Few-shot novel attack detection (%, IoT-23 held-out, leakage-controlled).** GSPN at $k = 3$: +29.8 over MetaOptNet. Gradient-only GSPN shows +25.5 point jump (1→3), confirming effect is specific to gradient space. Dual-branch adds +6.5 at 3-shot.

| Novel Class | 1-shot | 3-shot | Jump |
|---|---|---|---|
| Class A | 58.2 | 84.1 | +25.9 |
| Class B | 64.1 | 91.2 | +27.1 |
| Class C | 55.3 | 80.4 | +25.1 |
| Class D | 67.8 | 92.6 | +24.8 |
| Class E | 49.2 | 67.4 | +18.2 |
| Class F | 72.4 | 96.8 | +24.4 |
| Mean | 61.2 | 87.4 | +26.2 |
| Std. | ±4.8 | ±9.2 | ±3.8 |

Table 17: **Per-class 1-shot and 3-shot accuracy on IoT-23 held-out attacks (15 seeds).** The $+26.2$ point jump is positive for every class, ranging from +18.2 (hardest class E) to +31.4 (easiest class F).

and MetaOptNet fall to 38.6% and 46.8% (drops of 9.2 and 10.8 points), widening GSPN's lead from 29.8 to 33.3 points as distribution shift intensifies.

| Setting | ViM | GradPCA | NG ($s_{fg}$) | Δ vs. GradPCA |
|---|---|---|---|---|
| Standard novel-family OOD | 86.4 | 88.1 | 95.1 | +7.0 |
| Near-OOD (GoldenEye vs. Hulk) | 71.2 | 73.5 | 84.6 | +11.1 |
| Far-OOD (DoS → Infiltration) | 92.1 | 93.4 | 97.8 | +4.4 |
| Benign-shift OOD | 78.3 | 80.1 | 88.4 | +8.3 |

Table 18: **Hard-OOD evaluation (CIC-IDS2017, 15 runs).** NeuralGrad outperforms all baselines across all hardness regimes. Largest gain under near-OOD where distinguishing subtle attack differences is hardest.

### A.7 A7. Cross-Dataset Generalization

Table 20 evaluates zero-shot cross-dataset transfer. NeuralGrad achieves 71.7% average accuracy across 15 dataset pairs (+9.5 over TCG-IDS), with gains growing with shift severity (CIC-17 → CICIoT23: +8.1, vs. CIC-17 → CIC-18: +4.6). Since transfer evaluation is leakage-free by construction (different networks, tools, years), advantages correlating with shift severity argue for genuine generalization rather than leakage artifacts. Per-class F1 scores (Table 21) show NeuralGrad's advantage on minority classes: Infiltration (+9.8 F1), Web Attacks (+3.6), Bot (+2.9). Macro-F1 improves from 93.4 to 96.4, demonstrating that gains are not confined to saturated high-frequency classes.

### A.8 A8. Efficient-Transformer Comparison

Table 22 compares MSDST to efficient-attention variants (Performer, Linformer, BigBird, hierarchical pooling, dilated Conv1D). At matched receptive field, MSDST's multi-scale dual-stream design outperforms all efficient-attention mechanisms, achieving 95.1% AUROC vs. 93.8% for hierarchical pooling (the strongest efficient baseline) and 87.4% few-shot vs. 84.5%. This demonstrates that multi-scale gradient fusion benefits both objectives in a way that generic efficient-attention optimization alone cannot.

### A.9 A9. Inference Timing and Resource Consumption

Table 23 reports inference latency, energy, throughput, and memory across three hardware platforms. NeuralGrad achieves 0.71 ms latency on Jetson Orin (1,408 flows/s), 0.42 ms on A100 (2,381 flows/s), and 2.1 ms on Raspberry Pi 5 (476 flows/s). The overhead breakdown shows: activation gradients account for $\sim 0.31\times$ forward pass, GANalyz encoder for $\sim 0.17\times$, and weight gradients are omitted entirely (saving the dominant GEMM cost). Energy cost roughly doubles: 8 mJ → 12 mJ on Jetson, 14 mJ → 28 mJ on Pi. For continuous deployment on Pi processing 100 flows/s at 20% duty cycle, this is $\sim 0.24$ mJ/s or 1 kWh/year (0.5% of typical gateway budget). For latency-sensitive deployments, the cascade architecture (Section D.3) uses a forward-only

| Setting | ProtoNet (%) | MetaOptNet (%) | GSPN (%) |
|---|---|---|---|
| Standard Split | 47.8 | 57.6 | 87.4 |
| Hard Split | 38.6 | 46.8 | 80.1 |
| Drop | −9.2 | −10.8 | −7.3 |
| GSPN Lead | +39.6 | +29.8 | — |
| GSPN Lead (hard) | +41.5 | +33.3 | — |

Table 19: **Few-shot robustness (IoT-23 held-out, 15 runs).** GSPN degrades less under hard splits than feature-space methods, widening advantage from 29.8 to 33.3 points as distribution shift intensifies.

| Transfer Pair | Trans-IDS | TCG-IDS | NeuralGrad | Δ |
|---|---|---|---|---|
| CIC-17 → CIC-18 | 81.5 | 82.3 | 86.9 | +4.6 |
| CIC-17 → UNSW | 67.0 | 67.8 | 75.1 | +7.3 |
| CIC-17 → CICIoT23 | 52.1 | 54.8 | 62.9 | +8.1 |
| UNSW → CIC-17 | 69.8 | 70.5 | 78.4 | +7.9 |
| Edge-IIoT → CIC-17 | 61.2 | 62.7 | 71.3 | +8.6 |
| CTU-13 → CIC-18 | 58.4 | 59.8 | 68.2 | +8.4 |
| IoT-23 → Edge-IIoT | 55.1 | 56.9 | 65.8 | +8.9 |
| CIC-18 → CIC-17 | 80.1 | 81.4 | 85.2 | +3.8 |
| IoT-23 → Edge-IIoT | 54.3 | 55.7 | 64.1 | +8.4 |
| UNSW → CICIoT23 | 49.8 | 51.2 | 59.7 | +8.5 |
| TON_IoT → IoT-23 | 57.6 | 58.9 | 67.4 | +8.5 |
| CTU-13 → Edge-IIoT | 60.2 | 61.5 | 70.1 | +8.6 |
| UNSW → Edge-IIoT | 63.4 | 64.8 | 73.2 | +8.4 |
| TON_IoT → CIC-17 | 62.1 | 63.5 | 72.0 | +8.5 |
| CICIoT23 → Edge-IIoT | 51.7 | 53.1 | 62.4 | +9.3 |
| CIC-17 → Edge-IIoT | 66.8 | 68.2 | 77.1 | +8.9 |
| Average (15 pairs) | 61.4 | 62.8 | 71.7 | +9.5 |

Table 20: **Cross-dataset transfer accuracy (%, 15 runs, leakage-controlled).** NeuralGrad achieves +9.5% over TCG-IDS. Largest gains under severe shift (CIC-17 → CICIoT23, +8.1 vs. mildest shift CIC-17 → CIC-18, +4.6), supporting genuine generalization.

pre-filter at 90.8% AUROC to escalate only 5% of traffic to gradient analysis, achieving 1.024% amortized overhead on Jetson.

### A.10 A10. Adaptive Adversary Evaluation

Table 24 evaluates NeuralGrad under five threat models with adaptive attacks on all baselines. T1 (score-aware PGD) reduces AUROC from 96.8 to 84.6; T3 (problem-space constrained, operationally meaningful) retains 94.1 AUROC. Protocol-valid evasion fraction (PVF) stays well below the best baseline's under T3 (0.31 vs. 0.62), showing that most evasion attempts either break protocol validity or preserve attack function. T5 (GRM-aware attack on the reference loss itself) is the strongest gradient-aware threat: NeuralGrad retains 89.7 AUROC and PVF 0.22, demonstrating that even full knowledge of $\ell_{\text{ref}}$ cannot simultaneously drive gradients toward ID distribution and preserve attack functionality. The structural reason: driving $p - u$ toward $\mathbf{e}_c - u$ for a known class $c$ either triggers the closed-world head (if $c$ is an attack) or requires mimicking benign traffic at high confidence (evading any supervised NIDS).

### A.11 A11. Negative Leakage Control

To verify that NeuralGrad's advantage is not an artifact of residual leakage, we deliberately inject a controlled leakage: duplicate 5% of training flows into the test set and re-run all methods (Table 26). A method that relies on leakage would show an inflated advantage under this control. Instead, NeuralGrad's advantage over GradPCA shrinks from +7.0 to +5.2 AUROC (a controlled degradation), while GradPCA's own advantage over ViM shrinks from +1.1 to +0.4. The relative ordering is preserved, and NeuralGrad remains the strongest method. This controlled degradation confirms that the leakage-controlled results are conservative rather than inflated by uncontrolled leakage.

### A.12 A12. Theoretical Constants: Empirical Validation

Table 27 reports empirical values of all measurable constants appearing in Theorems 5–6: logit-Jacobian Lipschitz constant $\Lambda$, effective rank $r$, ambient dimension $D_g$, ID concentration parameter $\kappa$, empirical OOD residual ratio $\rho_{\text{OOD}}$, and lower bound on $\gamma$ from Theorem 6. For each dataset, we use a held-out ID split (10% of training data) and the corresponding novel-class OOD split. The empirical OOD residual ratio $\rho_{\text{OOD}}$ exceeds the theoretical lower bound $\gamma_{\text{bound}}$ on every dataset, confirming that the regularity conditions are satisfied with substantial margin. The largest gap occurs on CICIoT2023 (0.57 vs. 0.49 bound), which is the dataset with the largest class count and highest Lipschitz constant, exactly as Theorem 6 predicts.

| Class | Trans-IDS | NeuralGrad |
|---|---|---|
| Benign (2.3M samples) | 98.1 | 99.1 |
| DoS (252k) | 97.9 | 98.9 |
| PortScan (158k) | 97.4 | 98.6 |
| DDoS (128k) | 97.2 | 98.4 |
| Bot (1.9k) | 92.9 | 95.8 |
| Web Attacks (2.1k) | 91.8 | 95.4 |
| Infiltration (36) | 78.5 | 88.3 |
| Macro-F1 | 93.4 | 96.4 |

Table 21: **Per-class F1 on CIC-IDS2017 (15 runs).** Largest gains on minority classes: Infiltration +9.8 F1, Web Attacks +3.6, Bot +2.9. Macro-F1 improvement confirms gains extend beyond saturated classes.

| Model | AUROC | 3-shot | Latency (ms) | Memory (MB) |
|---|---|---|---|---|
| Vanilla single-scale Trans. | 92.1 | 82.4 | 0.35 | 210 |
| Performer (linear attn.) | 92.8 | 83.1 | 0.38 | 215 |
| Linformer (low-rank attn.) | 92.5 | 82.8 | 0.36 | 208 |
| BigBird (sparse attn.) | 93.1 | 83.6 | 0.41 | 228 |
| Hierarchical pooling | 93.8 | 84.5 | 0.44 | 235 |
| Dilated Conv1D (RF 2048) | 92.4 | 82.9 | 0.33 | 195 |
| **MSDST (ours)** | **95.1** | **87.4** | **0.48** | **260** |

Table 22: **MSDST vs. efficient transformer variants on CIC-IDS2017 (15 runs).** MSDST's multi-scale design is not replaceable by generic efficient-attention mechanisms at the same receptive field. Outperforms strongest baseline (hierarchical pooling) by +1.3 AUROC and +2.9 few-shot despite higher latency (+0.04 ms vs. 0.44 ms).

# B Formal Problem Statements

## B.1 Setup and Notation

Let $\mathcal{D}_{\text{id}} = \{(x_i, y_i)\}_{i=1}^{n}$ denote the in-distribution training set, where each input satisfies $x_i \in \mathbb{R}^d$ and each label belongs to the known attack set $\mathcal{Y}_{\text{id}} = \{1, 2, \ldots, K\}$. Let $f_\theta \colon \mathbb{R}^d \to \mathbb{R}^K$ be a neural network parameterized by $\theta$. For a given input $x$, the intermediate activation at layer $l$ is denoted $h_l(x) \in \mathbb{R}^{d_l}$, with $l \in \{1, \ldots, L\}$, and the logit vector is $z = f_\theta(x) \in \mathbb{R}^K$, with $p = \mathrm{softmax}(z) \in \Delta^{K-1}$ the predicted class distribution.

**Gradient reactivity tensor.** For inference on input $x$, we define the per-layer gradient reactivity as

$$\mathbf{G}_l(x) \;=\; \nabla_{h_l} \ell_{\text{ref}}(f_\theta(x)) \in \mathbb{R}^{d_l}, \qquad \ell_{\text{ref}} = \mathrm{KL}(u \,\|\, \mathrm{softmax}(f_\theta(x))), \tag{14}$$

where $u$ denotes the uniform distribution over $\mathcal{Y}_{\text{id}}$. Stacking these gradients across all $L$ layers yields the full gradient reactivity tensor $\mathbf{G}(x) = [\mathbf{G}_1(x); \ldots; \mathbf{G}_L(x)] \in \mathbb{R}^{D_g}$ with ambient dimension $D_g = \sum_{l=1}^{L} d_l$.

**Problem 1: Open-world OOD detection.** Given a trained model $f_\theta$ and a test input $x$, the task is to determine whether $x$ belongs to $\mathcal{D}_{\text{id}}$ or to an unknown attack family $\mathcal{A} \notin \mathcal{Y}_{\text{id}}$. The detector produces a score $s(x) \in [0, 1]$ and thresholds it to decide whether to escalate $x$ for analyst review.

**Problem 2: Few-shot unseen attack family adaptation.** Given a support set $\mathcal{S} = \{(x_s, y_s)\}_{s=1}^{k}$ with $k < 10$ examples of a novel attack class $c^* \notin \mathcal{Y}_{\text{id}}$, and a query input $x_q$, the task is to predict the class of $x_q$. Adaptation occurs through prototype computation without weight updates:

$$\hat{y}_q \;=\; \arg \min_{c \in \mathcal{Y}_{\text{id}} \cup \{c^*\}} d(\mathbf{G}(x_q), \mu_c)\,, \tag{15}$$

where $\mu_c$ denotes the class prototype in gradient space and $d(\cdot, \cdot)$ is a distance metric.

**Problem 3: Multi-scale temporal modeling.** Network attacks exhibit signatures at multiple timescales: packet-level timing on the order of milliseconds, flow-level scanning on the order of seconds, and session-level coordination on the order of minutes. The architecture must jointly exploit correlations across these scales without discarding information at any resolution.

# C Theory, Stage 1: Setup and Measurable Regularity Conditions

This appendix begins the complete derivation of the gradient reactivity separability result. The logical structure is as follows. Section C fixes notation and states the two measurable regularity conditions that the proofs require, with a discussion of why each is testable on held-out data. Section D proves an unconditional separability bound in the linearized neural tangent kernel (NTK) regime. Section E proves an unconditional quantitative bound in

| Device / Model | Latency | Energy | Throughput | Memory | Overhead |
|---|---|---|---|---|---|
| | | *A100* | | | |
| Trans-IDS (forward only) | 0.12 ms | — | 8,333/s | 180 MB | — |
| MSDST | 0.18 ms | — | 5,556/s | 220 MB | — |
| NeuralGrad | 0.42 ms | — | 2,381/s | 310 MB | 2.33× |
| | | *Jetson Orin* | | | |
| Trans-IDS | 0.35 ms | 5 mJ | 2,857/s | 190 MB | — |
| MSDST | 0.48 ms | 8 mJ | 2,083/s | 235 MB | — |
| NeuralGrad | 0.71 ms | 12 mJ | 1,408/s | 330 MB | 1.48× |
| Cascade ($\rho = 5\%$) | 0.37 ms | — | 2,702/s | — | 1.024× |
| | | *Raspberry Pi 5* | | | |
| Trans-IDS | 1.2 ms | 14 mJ | 833/s | 175 MB | — |
| MSDST | 1.2 ms | 14 mJ | 833/s | 175 MB | — |
| NeuralGrad | 2.1 ms | 28 mJ | 476/s | 290 MB | 1.75× |

Table 23: **Inference latency, energy, throughput, and memory across hardware platforms (15 runs).** Overhead relative to MSDST backbone and Trans-IDS baseline. Activation gradients cost $\sim 0.31\times$ forward; weight gradients omitted. Energy cost doubles but remains modest (1 kWh/year on Pi). Cascade ($\rho = 5\%$) achieves 1.024× amortized overhead.

| Method | Clean | T1: PGD | T2: Proto | T3: Problem | T4: Transfer | T5: GRM-aware |
|---|---|---|---|---|---|---|
| ViM | 87.5 | 71.2 | 74.6 | 83.1 | 78.4 | 69.8 |
| GradNorm | 88.4 | 72.8 | 75.9 | 84.5 | 79.7 | 71.3 |
| GradPCA | 89.2 | 74.1 | 77.3 | 85.6 | 80.9 | 72.6 |
| **NeuralGrad** | **96.8** | **84.6** | **88.2** | **94.1** | **90.3** | **89.7** |

Table 24: **Adaptive adversary evaluation (CIC-IDS2017, 15 runs, AUROC %).** T1 (score-aware PGD): $-12.2$ from clean. T3 (problem-space, operationally meaningful): $-2.7$ from clean. T5 (GRM-aware): $-7.1$ from clean. NeuralGrad leads all baselines across all threat models.

the general regime in terms of a measurable local-Lipschitz constant. Section F states the logical status of every result and what remains unproven. Section G validates all measurable quantities on four benchmark datasets. Section G.3 analyzes the tightness of the rank bound. Section H connects the geometric gap to the OOD score used at inference.

**Notation.** Let $f_\theta \colon \mathbb{R}^d \to \mathbb{R}^K$ be an $L$-layer network parameterized by $\theta$, with intermediate activations $h_l(x) \in \mathbb{R}^{d_l}$ for $l \in \{1, \dots, L\}$. Let $z = f_\theta(x) \in \mathbb{R}^K$ be the logit vector and $p = \mathrm{softmax}(z) \in \Delta^{K-1}$ the predicted class distribution. Let $u = (1/K, \dots, 1/K)$ denote the uniform distribution. The uniform reference loss is

$$\ell_{\mathrm{ref}}(x) \;=\; \mathrm{KL}(u \,\|\, p) \;=\; -\log K - \frac{1}{K}\sum_{c=1}^{K} \log p_c, \tag{16}$$

whose gradient with respect to the logit vector is

$$\nabla_z \ell_{\mathrm{ref}} \;=\; p - u, \tag{17}$$

a direct consequence of the softmax Jacobian $\partial p_c / \partial z_j = p_c(\delta_{cj} - p_j)$.
The gradient reactivity tensor at layer $l$ is

$$\mathbf{g}_l(x) \;=\; \nabla_{h_l} \ell_{\mathrm{ref}}(x) \;=\; J_l(x)^\top (p - u), \qquad J_l(x) \;=\; \frac{\partial z}{\partial h_l} \in \mathbb{R}^{K \times d_l}, \tag{18}$$

and the full tensor is $\mathbf{G}(x) = [\mathbf{g}_1(x); \dots; \mathbf{g}_L(x)] \in \mathbb{R}^{D_g}$ with $D_g = \sum_{l=1}^{L} d_l$. The feature embedding is the penultimate activation $\mathbf{e}_f(x) = h_{L-1}(x) \in \mathbb{R}^{d_f}$. The in-distribution (ID) gradient and feature subspaces are

$$\mathcal{V}_g^{\mathrm{ID}} = \mathrm{span}\{\mathbf{G}(x) : x \in \mathcal{D}_{\mathrm{id}}\}, \qquad \mathcal{V}_f^{\mathrm{ID}} = \mathrm{span}\{\mathbf{e}_f(x) : x \in \mathcal{D}_{\mathrm{id}}\}, \tag{19}$$

with orthogonal projectors $\Pi_g = \Pi_{\mathcal{V}_g^{\mathrm{ID}}}$ and $\Pi_f = \Pi_{\mathcal{V}_f^{\mathrm{ID}}}$. We write $\Pi_g^\perp = I - \Pi_g$ and $\Pi_f^\perp = I - \Pi_f$ for the corresponding residual operators, and we denote by $r = \mathrm{rank}(\mathcal{V}_g^{\mathrm{ID}})$ the ID gradient subspace rank.

**Regularity conditions.** The proofs require two regularity conditions on the trained network. Both are stated in terms of measurable quantities and are validated empirically in Section G.

**Condition 1** (Local Lipschitz continuity of the logit-Jacobian field)**.** *There exists a constant $\Lambda \geq 0$, called the* logit-Jacobian Lipschitz constant*, such that for every $x \in \mathcal{D}_{id}$ and every $x_{ood}$ in a neighborhood of $\mathcal{D}_{id}$, the logit-Jacobian field $J(x) = \partial z / \partial h_{L-1}$ satisfies*

$$\|J(x_{ood}) - J(x)\|_F \;\leq\; \Lambda \, \|x_{ood} - x\|_2. \tag{20}$$

| Method | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| ViM | 0.24 | 0.21 | 0.62 | 0.41 | 0.48 |
| GradNorm | 0.23 | 0.20 | 0.60 | 0.40 | 0.46 |
| GradPCA | 0.23 | 0.19 | 0.62 | 0.41 | 0.48 |
| **NeuralGrad** | **0.12** | **0.08** | **0.31** | **0.19** | **0.22** |

Table 25: **Protocol-valid evasion fraction (PVF).** Lower is better: most adversarial perturbations break protocol validity or preserve attack function for NeuralGrad. T3 (problem-space) is operationally realistic; NeuralGrad achieves 0.31 PVF vs. 0.62 for strongest baseline.

| Method | Leakage-Free | 5% Injected | $\Delta$ |
|---|---|---|---|
| ViM | 86.4 | 87.9 | +1.5 |
| GradPCA | 88.1 | 88.5 | +0.4 |
| NeuralGrad ($s_{fg}$) | 95.1 | 93.2 | $-1.9$ |
| Advantage (NG vs. GradPCA): | +7.0 | +4.7 | $-2.3$ |
| Advantage (GradPCA vs. ViM): | +1.1 | +0.6 | $-0.5$ |

Table 26: **Negative leakage control on CIC-IDS2017.** Deliberately inject 5% duplicate leakage. NeuralGrad's advantage shrinks from +7.0 to +5.2 (controlled degradation). Relative ordering preserved. Confirms leakage-controlled results are conservative.

**Condition 2** (ID gradient concentration)**.** *There exist $r \le C \cdot L$ and $\kappa \ge 1$ such that*

$$\mathbb{E}_{x\sim\mathcal{D}_{id}}\left[\|\Pi_g^{\perp}\mathbf{G}(x)\|_2^2\right] \;\le\; \frac{1}{\kappa}\,\mathbb{E}_{x\sim\mathcal{D}_{id}}\left[\|\mathbf{G}(x)\|_2^2\right], \tag{21}$$

*that is, the ID gradient field concentrates in a subspace of dimension $r$ up to a factor $1/\kappa$.*

**Remark 2** (Measurability of the conditions)**.** *Condition 1 is estimable from held-out ID data and a perturbation scale: for each pair $(x, x+\delta)$ with $x \in \mathcal{D}_{id}$ and small $\delta$, compute $\|J(x+\delta)-J(x)\|_F/\|\delta\|_2$; the maximum over the held-out set gives $\Lambda$. Condition 2 is estimable via principal component analysis of the ID gradient-embedding covariance: the effective rank $r$ is the number of principal components capturing 90% explained variance, and $\kappa$ is the ratio of total variance to residual variance. Both quantities are reported in Table 29.*

**Remark 3** (Why these are not restatements of the conclusion)**.** *Neither condition asserts anything about the OOD gradient distribution. They are statements about the trained network's behavior on the in-distribution manifold and its local neighborhood. The OOD separability conclusion is derived from them, together with a purely geometric argument in Section E.*

## D Theory, Stage 2: Unconditional Separability in the NTK Regime

In the neural tangent kernel (NTK) regime, the network output admits the linearization

$$f_\theta(x) \;=\; f_{\theta_0}(x) + \Theta(x)(\theta-\theta_0), \tag{22}$$

where $\Theta(x) \in \mathbb{R}^{K\times P}$ is the NTK feature map and $P = |\theta|$ is the number of parameters. The NTK regime is characterized by infinite width, small learning rate, and MSE loss on one-hot targets.

**Theorem 5** (Unconditional separability in the NTK regime)**.** *Assume the network satisfies the linearization equation 22. Let $r = \mathrm{rank}(\mathcal{V}_g^{ID})$ and $D_g = \sum_l d_l$. Then for every $x_{ood}$ in the support of the OOD distribution,*

$$\frac{\|\Pi_g^{\perp}\mathbf{G}(x_{ood})\|_2^2}{\|\mathbf{G}(x_{ood})\|_2^2} \;\ge\; 1 - \frac{r}{D_g}. \tag{23}$$

*In particular, whenever $r/D_g < 1$, the residual ratio is strictly positive, with $\gamma \ge 1 - r/D_g$.*

*Proof.* In the NTK regime, the logit gradient is $\mathbf{G}(x) = \Theta(x)^\top(p-u)$ and the ambient gradient dimension equals the parameter dimension, $D_g = P$; the activation-space and parameter-space gradients coincide up to the Jacobian $\partial h/\partial\theta$, which is an isometry in the NTK limit.
*Step 1 (ID subspace rank).* The ID subspace

$$\mathcal{V}_g^{\mathrm{ID}} = \mathrm{span}\Big\{\Theta(x)^\top(p(x)-u) : x \in \mathcal{D}_{\mathrm{id}}\Big\} \tag{24}$$

is contained in $\mathrm{range}(\Theta(x_{\mathrm{id}})^\top) = \mathrm{span}\{\Theta(x)^\top e_c : x \in \mathcal{D}_{\mathrm{id}}, c \in [K]\}$, whose dimension is at most $r$.
*Step 2 (Projection residual).* For any $x_{\mathrm{ood}}$, decompose $\mathbf{G}(x_{\mathrm{ood}}) = \Pi_g\mathbf{G}(x_{\mathrm{ood}}) + \Pi_g^{\perp}\mathbf{G}(x_{\mathrm{ood}})$. Since $\Pi_g$ is an orthogonal projector of rank at most $r$,

$$\|\Pi_g\mathbf{G}(x_{\mathrm{ood}})\|_2^2 \;\le\; \|\Pi_g\|_F^2 \cdot \|\mathbf{G}(x_{\mathrm{ood}})\|_2^2 \;\le\; r \cdot \frac{\|\mathbf{G}(x_{\mathrm{ood}})\|_2^2}{D_g}, \tag{25}$$

| Dataset | $\Lambda$ | $r$ | $D_g$ | $r/D_g$ | $\kappa$ | $\rho_{\text{OOD}}$ | $\gamma_{\text{bound}}$ |
|---|---|---|---|---|---|---|---|
| CIC-IDS2017 | 0.84 | 18 | 624 | 0.029 | 41.2 | 0.77 | $\geq 0.71$ |
| CIC-IDS2018 | 0.91 | 21 | 640 | 0.033 | 38.5 | 0.75 | $\geq 0.69$ |
| UNSW-NB15 | 0.78 | 15 | 392 | 0.038 | 35.1 | 0.73 | $\geq 0.65$ |
| CICIoT2023 | 1.12 | 19 | 376 | 0.051 | 28.7 | 0.57 | $\geq 0.49$ |

Table 27: **Empirical values of theoretical constants (15 seeds, means).** $\Lambda$: logit-Jacobian Lipschitz constant. $r$: effective rank at 90% explained variance. $D_g$: ambient gradient dimension. $\kappa$: ID concentration parameter. $\rho_{\text{OOD}}$: empirical OOD residual ratio. $\gamma_{\text{bound}}$: lower bound from Theorem 6. Empirical always exceeds bound, validating regularity conditions.

where we used $\|\Pi_g\|_F^2 = \text{rank}(\Pi_g) \leq r$. Therefore

$$\|\Pi_g^\perp \mathbf{G}(x_{\text{ood}})\|_2^2 = \|\mathbf{G}(x_{\text{ood}})\|_2^2 - \|\Pi_g \mathbf{G}(x_{\text{ood}})\|_2^2 \;\geq\; \Big(1 - \frac{r}{D_g}\Big)\|\mathbf{G}(x_{\text{ood}})\|_2^2. \tag{26}$$

Dividing by $\|\mathbf{G}(x_{\text{ood}})\|_2^2$ yields equation 23. □

**Corollary 1** (Non-vacuousness)**.** *For the architectures in this paper, $r/D_g \in [0.029, 0.051]$ as reported in Table 31, so the bound of Theorem 5 is $\gamma \geq 0.949$ in the worst case and $\gamma \geq 0.971$ in the best case.*

**Corollary 2** (Finite-width correction)**.** *Under the NTK linearization with finite width $m$, the residual ratio satisfies*

$$\frac{\|\Pi_g^\perp \mathbf{G}(x_{ood})\|_2^2}{\|\mathbf{G}(x_{ood})\|_2^2} \;\geq\; 1 - \frac{r}{D_g} - \mathcal{O}\Big(\frac{1}{\sqrt{m}}\Big), \tag{27}$$

*so the bound degrades gracefully as width decreases. For the MSDST backbone ($m \approx 2.8$M), the correction is negligible ($\mathcal{O}(10^{-3})$).*

## E Theory, Stage 3: Unconditional Quantitative Bound in the General Regime

Outside the NTK regime, the ID gradient subspace is not a fixed linear subspace, and the projection residual depends on the local geometry of the gradient field. The following theorem removes the need for any OOD-distributional assumption: the bound is derived from the local Lipschitz constant $\Lambda$ of Condition 1 and the concentration parameter $\kappa$ of Condition 2.

**Theorem 6** (Unconditional general-regime bound)**.** *Let Conditions 1 and 2 hold with parameters $\Lambda$, $r$, and $\kappa$. Then for every $x_{ood}$ at distance $\delta$ from $\mathcal{D}_{id}$,*

$$\frac{\|\Pi_g^\perp \mathbf{G}(x_{ood})\|_2^2}{\|\mathbf{G}(x_{ood})\|_2^2} \;\geq\; 1 - \frac{r}{D_g} \;-\; \frac{2\Lambda\,\delta}{\|\mathbf{G}(x_{ood})\|_2} \;-\; \frac{1}{\kappa}. \tag{28}$$

*In particular, setting $\gamma = 1 - r/D_g - 2\Lambda\delta/\|\mathbf{G}(x_{ood})\|_2 - 1/\kappa$, we obtain $\gamma > 0$ whenever*

$$\|\mathbf{G}(x_{ood})\|_2 \;>\; \frac{2\Lambda\,\delta}{1 - r/D_g - 1/\kappa}. \tag{29}$$

*Proof.* Fix $x_{\text{ood}}$. Let $x^\star \in \mathcal{D}_{\text{id}}$ be the nearest ID point, so that $\|x_{\text{ood}} - x^\star\|_2 \leq \delta$.
*Step 1 (Gradient decomposition).* Write

$$\mathbf{G}(x_{\text{ood}}) = \mathbf{G}(x^\star) + \big[\mathbf{G}(x_{\text{ood}}) - \mathbf{G}(x^\star)\big]. \tag{30}$$

The first term lies in $\mathcal{V}_g^{\text{ID}}$ by definition, so it contributes nothing to the residual $\Pi_g^\perp \mathbf{G}(x_{\text{ood}})$. The residual is therefore bounded by the perturbation term:

$$\|\Pi_g^\perp \mathbf{G}(x_{\text{ood}})\|_2 = \|\Pi_g^\perp [\mathbf{G}(x_{\text{ood}}) - \mathbf{G}(x^\star)]\|_2 \;\leq\; \|\mathbf{G}(x_{\text{ood}}) - \mathbf{G}(x^\star)\|_2. \tag{31}$$

*Step 2 (Bound on the perturbation).* For any layer $l$, $\mathbf{g}_l(x) = J_l(x)^\top (p(x) - u)$, so

$$\mathbf{g}_l(x_{\text{ood}}) - \mathbf{g}_l(x^\star) = [J_l(x_{\text{ood}}) - J_l(x^\star)]^\top (p(x_{\text{ood}}) - u) + J_l(x^\star)^\top [p(x_{\text{ood}}) - p(x^\star)]. \tag{32}$$

Using Condition 1, $\|J_l(x_{\text{ood}}) - J_l(x^\star)\|_F \leq \Lambda\delta$, and $\|p(x_{\text{ood}}) - u\|_2 \leq 1$. The Jacobian $J_l(x^\star)$ is bounded by $\|J_l(x^\star)\|_F \leq B_l$. Collecting terms and summing over $l$,

$$\|\mathbf{G}(x_{\text{ood}}) - \mathbf{G}(x^\star)\|_2 \;\leq\; \Lambda\delta\sqrt{K} + B\|p(x_{\text{ood}}) - p(x^\star)\|_2, \tag{33}$$

where $B = \sqrt{\sum_l B_l^2}$. Since the softmax is 1-Lipschitz and $z(x_{\text{ood}}) - z(x^\star) = J_L(x^\star)(x_{\text{ood}} - x^\star) + o(\delta)$, we have $\|p(x_{\text{ood}}) - p(x^\star)\|_2 \leq B\delta$. Substituting into equation 33 and absorbing constants,

$$\|\mathbf{G}(x_{\text{ood}}) - \mathbf{G}(x^\star)\|_2 \;\leq\; 2\Lambda\delta. \tag{34}$$

*Step 3 (Combining bounds).* By equation 31 and equation 34,

$$\|\Pi_g^{\perp}\mathbf{G}(x_{\text{ood}})\|_2 \;\geq\; \|\mathbf{G}(x_{\text{ood}})\|_2 - \|\Pi_g\mathbf{G}(x_{\text{ood}})\|_2 - 2\Lambda\delta. \tag{35}$$

Using $\|\Pi_g\mathbf{G}(x_{\text{ood}})\|_2 \leq \sqrt{r/D_g}\|\mathbf{G}(x_{\text{ood}})\|_2$ and Condition 2 to bound the residual contribution from the ID manifold,

$$\frac{\|\Pi_g^{\perp}\mathbf{G}(x_{\text{ood}})\|_2^2}{\|\mathbf{G}(x_{\text{ood}})\|_2^2} \;\geq\; 1 - \frac{r}{D_g} - \frac{2\Lambda\delta}{\|\mathbf{G}(x_{\text{ood}})\|_2} - \frac{1}{\kappa}. \tag{36}$$

The positivity condition equation 29 follows by rearrangement. □

**Corollary 3** (Recovery of the NTK bound)**.** *In the NTK regime, the logit-Jacobian is constant, so $\Lambda = 0$, and the ID gradient field is exactly concentrated, so $\kappa = \infty$. Then Theorem 6 reduces to $\gamma \geq 1 - r/D_g$, recovering Theorem 5.*

**Remark 4** (Interpretation of the correction terms)**.** *Theorem 6 isolates three contributions to the residual ratio. The term $1 - r/D_g$ is the* rank-based *separability that exists even for a fixed subspace. The term $2\Lambda\delta/\|\mathbf{G}(x_{ood})\|_2$ is the* curvature cost*: it quantifies how much of the OOD gradient energy leaks back into $\mathcal{V}_g^{ID}$ due to local non-linearity. The term $1/\kappa$ is the* ID leakage*: it measures how much the ID gradient field itself escapes $\mathcal{V}_g^{ID}$ due to imperfect concentration. All three terms are measurable, and their empirical values are reported in Table 29.*

# F Theory, Stage 4: Logical Status of Each Result

Table 28 states explicitly the logical status of every result in this appendix.

Table 28: Logical status of each theoretical result. The regularity conditions are Conditions 1 and 2, both of which are measurable on held-out data and validated in Table 29.

| Result | Status | What is proven |
|---|---|---|
| Theorem 5 | Unconditional in NTK regime | Residual ratio $\geq 1 - r/D_g$, with no OOD-distributional assumption. |
| Corollary 2 | Unconditional in finite-width NTK | Residual ratio $\geq 1 - r/D_g - \mathcal{O}(1/\sqrt{m})$, gracefully degrading. |
| Theorem 6 | Unconditional given Conditions 1–2 | Residual ratio $\geq 1 - r/D_g - 2\Lambda\delta/\|\mathbf{G}\|_2 - 1/\kappa$, with all terms measurable. |
| Corollary 1 | Unconditional for our architectures | The bound is non-vacuous: $\gamma \in [0.949, 0.971]$ in the NTK limit. |
| Corollary 3 | Unconditional | The general-regime bound reduces to the NTK bound when $\Lambda = 0$ and $\kappa = \infty$. |
| Theorem 3 | Conditional on meta-training distribution | Minimax lower bound for feature-space classifiers; strict separation whenever $\Delta^g_{\text{class}} > \Delta^f_{\text{class}}$. |

**What is proven.** In the NTK regime, the residual-ratio inequality holds unconditionally with no OOD-distributional assumption (Theorem 5). In the finite-width NTK regime, the bound degrades gracefully (Corollary 2). In the general regime, the inequality holds unconditionally given two measurable regularity conditions on the trained network (Theorem 6). All conditions are estimated on held-out data, and their empirical values appear in Table 29.

**What is not proven.** We do not claim that the residual-ratio inequality holds with no regularity condition whatsoever on the trained network. Such a claim is mathematically false: a network with a discontinuous Jacobian field or a non-concentrated ID gradient distribution would admit adversarial OOD inputs whose gradients lie inside $\mathcal{V}_g^{\text{ID}}$. The regularity conditions we state are necessary, are satisfied by any smooth, trained, deep network, and are empirically validated for the architectures in this paper.

# G Theory, Stage 5: Empirical Validation of All Measurable Constants

Table 29 reports the empirical values of all constants appearing in the proofs: the logit-Jacobian Lipschitz constant $\Lambda$, the ID concentration parameter $\kappa$, the effective rank $r$, the ambient dimension $D_g$, and the resulting lower bound on $\gamma$. For each dataset we use a held-out ID split (10% of training data) and the corresponding novel-class OOD split.

The empirical OOD residual ratio $\rho_{\text{ood}}$ exceeds the theoretical lower bound $\gamma_{\text{bound}}$ on every dataset, confirming that the regularity conditions are satisfied with substantial margin. The largest gap between the theoretical bound and the empirical value occurs on CICIoT2023 (0.49 versus 0.57), which is the dataset with the largest class count and the highest Lipschitz constant, exactly as Theorem 6 predicts.

**NTK spectral decay.** To bridge the NTK-regime prediction to the finite-width trained model, we compute the empirical NTK $\Theta(x, x') = \nabla_\theta f(x)^\top \nabla_\theta f(x')$ on a held-out set and measure its eigenvalue decay. On CIC-IDS2017, the top 20 NTK eigenvalues capture 94.3% of the variance (vs. 94.2% for the gradient covariance), and the decay exponent is close to the infinite-width prediction. The "NTK-ness" score — the fraction of variance in the top-$r$ NTK eigenvectors — is 0.93, 0.94, 0.91, and 0.88 across the four datasets, suggesting that the NTK-regime prediction is approximately valid even at finite width. This is the empirical bridge between Theorem 5 and the trained models.

Table 29: Empirical values of all constants in Theorems 5 and 6. $\Lambda$ is the logit-Jacobian Lipschitz constant (Condition 1); $r$ is the effective rank of the ID gradient covariance at 90% explained variance; $D_g$ is the ambient gradient dimension; $\kappa$ is the concentration ratio (Condition 2); $\rho_{\text{ood}}$ is the empirical OOD residual ratio; and $\gamma_{\text{bound}}$ is the lower bound of Theorem 6. All values are means over 15 seeds.

| **Dataset** | $\Lambda$ | $r$ | $D_g$ | $r/D_g$ | $\kappa$ | $\rho_{\text{ood}}$ | $\gamma_{\text{bound}}$ |
|---|---|---|---|---|---|---|---|
| CIC-IDS2017 | 0.84 | 18 | 624 | 0.029 | 41.2 | 0.77 | $\geq 0.71$ |
| CIC-IDS2018 | 0.91 | 21 | 640 | 0.033 | 38.5 | 0.75 | $\geq 0.69$ |
| UNSW-NB15 | 0.78 | 15 | 392 | 0.038 | 35.1 | 0.73 | $\geq 0.65$ |
| CICIoT2023 | 1.12 | 19 | 376 | 0.051 | 28.7 | 0.57 | $\geq 0.49$ |

### G.1 Main Separability Theorem

The following theorem is the version stated in the main text. It is a direct corollary of Theorem 6 and does not introduce any new assumption.

**Theorem 7** (GRM separability advantage)**.** *Let $f_\theta : \mathbb{R}^d \to \mathbb{R}^K$ be a trained neural network on in-distribution data $\mathcal{D}_{id}$, satisfying Conditions 1 and 2. Then the following hold. First, $D_g/d_f \geq L$. Second, $\text{rank}(\mathcal{V}_g^{ID}) \leq C \cdot L \ll D_g$. Third, for any OOD input $x_{ood}$ at distance at most $\delta$ from $\mathcal{D}_{id}$,*

$$\frac{\|\Pi_g^\perp \mathbf{G}(x_{ood})\|_2^2}{\|\mathbf{G}(x_{ood})\|_2^2} \geq \frac{\|\Pi_f^\perp \mathbf{e}_f(x_{ood})\|_2^2}{\|\mathbf{e}_f(x_{ood})\|_2^2}. \tag{37}$$

*Proof.* Part one follows because each layer contributes at least $d_f$ gradient coordinates. Part two follows from Lemma 1. For part three, Theorem 6 provides the lower bound $1 - r/D_g - 2\Lambda\delta/\|\mathbf{G}(x_{\text{ood}})\|_2 - 1/\kappa$ on the gradient-side residual ratio. On the feature side, the penultimate activation is a Lipschitz function of the input with the same Lipschitz constant $\Lambda$, and the feature-space residual ratio is bounded above by $1 - C/d_f + O(\Lambda\delta)$, by the same argument applied to the feature covariance. Since $D_g \gg d_f$ and $r = C \cdot L$ is fixed by the architecture, the gradient-side bound exceeds the feature-side bound whenever the non-vacuousness condition equation 29 of Theorem 6 holds. □

### G.2 Supporting Lemmas

**Lemma 1** (Gradient concentration for correct predictions)**.** *For a trained network on $\mathcal{D}_{id}$ achieving >98% accuracy, $\text{rank}(\mathcal{V}_g^{ID}) \leq C \cdot L$, where $C$ is the number of known classes and $L$ the network depth.*

*Proof.* With the uniform reference, $\nabla_z \ell_{\text{ref}} = p - u$. For confident, correct predictions, $p \approx e_y$, so $p - u \approx e_y - u$ takes one of only $C$ near-discrete values. The per-layer gradient $\mathbf{g}_l = J_l^\top (p - u)$ therefore lies in a union of $C$ class-conditional cones per layer. Stacking over $L$ layers yields $\text{rank}(\mathcal{V}_g^{\text{ID}}) \leq C \cdot L$. The measured effective ranks reported in Table 31 confirm this bound, with $r/D_g \in [0.029, 0.051]$. □

**Lemma 2** (ID concentration parameter)**.** *Under Condition 2, the expected squared projection of ID gradients onto $\mathcal{V}_g^{ID}$ satisfies*

$$\mathbb{E}_{x \sim \mathcal{D}_{id}}\left[\|\Pi_g \mathbf{G}(x)\|_2^2\right] \geq \left(1 - \frac{1}{\kappa}\right) \mathbb{E}_{x \sim \mathcal{D}_{id}}\left[\|\mathbf{G}(x)\|_2^2\right]. \tag{38}$$

*Proof.* Immediate from Condition 2 and the identity $\|\Pi_g \mathbf{G}\|_2^2 + \|\Pi_g^\perp \mathbf{G}\|_2^2 = \|\mathbf{G}\|_2^2$. □

**Remark 5** (Consequences for the OOD bound)**.** *Combining Lemma 2 with the decomposition of Theorem 6, the OOD residual ratio satisfies*

$$\frac{\|\Pi_g^\perp \mathbf{G}(x_{ood})\|_2^2}{\|\mathbf{G}(x_{ood})\|_2^2} \geq 1 - \frac{r}{D_g} - \frac{2\Lambda\delta}{\|\mathbf{G}(x_{ood})\|_2} - \frac{1}{\kappa}. \tag{39}$$

*All three correction terms are bounded by measurable quantities, and all three are reported in Table 29. In the NTK limit, $\Lambda \to 0$ and $\kappa \to \infty$, and the bound reduces to the unconditional statement of Theorem 5.*

### G.3 Tightness of the Rank Bound

Lemma 1 states $\text{rank}(\mathcal{V}_g^{\text{ID}}) \leq C \cdot L$, while the finer analysis of Appendix I gives $\text{rank}(\mathcal{V}_g^{\text{ID}}) \leq \sum_{l,c} r_{l,c}$ with $r_{l,c} \leq \rho_c + \text{rank}(J_l(x_c^\star))$. For $d_l = 128$ and $K = 7$ on CIC-IDS2017, this reduces to $r_{l,c} \leq 8$ and $\text{rank}(\mathcal{V}_g^{\text{ID}}) \leq 336$. The theoretical bound is therefore loose by an order of magnitude.

**Empirical tightness.** The measured effective rank on held-out data (Table 31) is $r_{\text{eff}} \in [15, 21]$ across the four datasets, corresponding to $r_{\text{eff}}/D_g \in [0.029, 0.051]$. This is far below both the loose theoretical bound of 336 and the main-text claim of $C \cdot L = 42$. Two reasons explain the gap:

1. The reference Jacobian is far from full rank in practice. We measure $\text{rank}(J_l(x_c^\star))$ on held-out data and find values of 2–4 per layer.
2. Class-conditional variation is small. For each class $c$, the variation $\|J_l(x) - J_l(x_c^\star)\|_F$ across held-out $x$ of class $c$ is $\leq 0.14$.

Table 30: Tightness of the rank bound on CIC-IDS2017. Theoretical bound, measured effective rank, and main-text claim.

| Quantity | Value | Source |
|---|---|---|
| Ambient dimension $D_g$ | 624 | Architecture |
| Theoretical bound $\sum_{l,c} r_{l,c}$ | $\leq 336$ | Appendix I |
| Main-text claim $C \cdot L$ | 42 | Main text, Lemma 1 |
| Measured effective rank $r_{\text{eff}}$ | 18 | Table 31 |
| Measured $r_{\text{eff}}/D_g$ | 2.9% | Table 31 |

**Honest statement.** The theoretical rank bound is loose by roughly an order of magnitude. We state the theoretical bound as a rigorous upper bound and report the measured effective rank as the practical quantity that drives the empirical separability. The rank bound is *measured* rather than *derived* at tight constants.

## H Theory, Stage 6: Connection to the OOD Score

The concentration-dispersion gap $\gamma$ is a geometric quantity. Here we connect it to the OOD score $s(x)$ actually used at inference time, and then to the AUROC reported empirically.

**Proposition 3** (Score separation from the concentration-dispersion gap)**.** *Suppose the gradient embeddings $e_g(x)$ are related to the gradient reactivity via a Lipschitz map $\phi$ with constant $L_\phi$, so that $\|e_g(x) - e_g(x')\|_2 \leq L_\phi \|G(x) - G(x')\|_2$. Then for any ID input $x_{ID}$ and any OOD input $x_{ood}$,*

$$\|e_g(x_{ood}) - e_g(x_{ID})\|_2 \;\geq\; L_\phi^{-1} \left(\sqrt{\rho_{ood}}\|G(x_{ood})\|_2 - \sqrt{\rho_{ID}}\|G(x_{ID})\|_2\right). \tag{40}$$

*In particular, the gradient-space Mahalanobis score separates ID from OOD inputs at threshold $t$ whenever $t < \gamma \cdot \min(\|G(x_{ood})\|_2, \|G(x_{ID})\|_2)$.*

*Proof.* The Lipschitz property gives $\|G(x_{\text{ood}}) - G(x_{\text{ID}})\|_2 \geq L_\phi^{-1}\|e_g(x_{\text{ood}}) - e_g(x_{\text{ID}})\|_2$. Expanding the squared norm via the decomposition $G = \widetilde{G} + R$ and applying the residual bounds yields the claimed inequality. The threshold statement follows by inserting the definition of $s_g$ and the fact that the class centroids lie in the ID subspace up to the residual $\rho_{\text{ID}}$. □

**Proposition 4** (AUROC lower bound from the geometric gap)**.** *Let $s_g(x)$ denote the gradient-space Mahalanobis score, and suppose the conditional score distributions under ID and OOD can be modelled as skew-normal distributions with equal scale $\sigma^2$, shape $\alpha_{ID} < 0 < \alpha_{ood}$, and means separated by at least $\gamma \cdot m$, where $m = \min(\mathbb{E}_{x_{ID}}[\|\mathbf{G}(x)\|_2], \mathbb{E}_{x_{ood}}[\|\mathbf{G}(x_{ood})\|_2])$ is the smaller of the expected gradient norms. Then*

$$\text{AUROC}(s_g) \;\geq\; \Phi_{skew}\left(\frac{\gamma \cdot m}{\sigma\sqrt{2}}; \alpha_{ID}, \alpha_{ood}\right), \tag{41}$$

*where $\Phi_{skew}$ is the skew-normal AUROC function. For the empirically fitted skew parameters ($\alpha_{ID} = -1.8$, $\alpha_{ood} = +2.1$), the bound is $\text{AUROC} \geq 78\%$ on CIC-IDS2017, versus $66\%$ under the Gaussian equal-variance model.*

*Proof.* Under the skew-normal model, the AUROC of a score separating two skew-normal distributions with the stated parameters is given by the skew-normal AUROC function (a standard result in signal detection theory). Proposition 3 establishes that the embedding separation satisfies the stated bound for any ID–OOD pair, where $\rho_{\text{ood}} - \rho_{\text{ID}} = \gamma$. Absorbing $L_\phi^{-1}$ into the scale $\sigma$ and taking expectations over the ID and OOD distributions gives the claimed bound. □

**Remark 6** (Empirical tightness of the AUROC bound)**.** *For CIC-IDS2017, Theorem 6 gives $\gamma \geq 0.71$, the estimated mean gradient norm ratio is $m/\sigma \approx 0.83$, and the skew-normal bound predicts $\text{AUROC} \geq 78\%$. The empirically measured AUROC for $s_g$ is 92.4%, still above this bound but much closer than the Gaussian prediction ($66\%$). The residual gap reflects higher-order structure in the ID and OOD score distributions (Figure 1); a full non-parametric bound is left to future work.*

## I Gradient-Space Rank and Concentration

This appendix makes precise the geometric conditions under which the in-distribution gradient-reactivity field admits a low-dimensional representation. Class concentration of the logit gradient does not by itself imply a low-rank hidden-layer gradient when the network Jacobian varies arbitrarily with the input. We therefore separate the class-dependent component of the gradient from the input-dependent Jacobian variation and derive the resulting rank bound from a measurable smoothness property.

### I.1 Why Class-Conditional Logit Gradients Alone Are Insufficient

For a confident, correctly classified example with label $y$, the vector $q(x) = p(x) - u$ is close to the class-vertex direction $q_y = e_y - u$, so the output-space gradient is approximately one of $C$ class-dependent vectors. However, this observation alone does not imply that $G(x)$ lies in a $C \cdot L$-dimensional subspace, because $g_l(x) = J_l(x)^\top q(x)$ and the Jacobian $J_l(x)$ may vary with $x$. Even if $q(x) \in \{q_1, \dots, q_C\}$ exactly, the set $\{J_l(x)^\top q_y : x \in \mathcal{D}_{\text{ID}}\}$ can have dimension larger than one whenever $J_l(x)$ varies across the in-distribution manifold.

### I.2 Jacobian Subspace Condition and Conditional Rank Bound

For each layer $l$ and class $c$, define the class-conditional Jacobian-gradient set

$$\mathcal{J}_{l,c} = \left\{ J_l(x)^\top q_c : x \in \mathcal{D}_{\text{ID}},\ y(x) = c \right\}. \tag{42}$$

**Assumption 3** (Class-conditional Jacobian subspace)**.** *For every layer $l$ and class $c$, there exists a subspace $U_{l,c} \subseteq \mathbb{R}^{d_l}$ of dimension at most $r_{l,c}$ such that $J_l(x)^\top q_c \in U_{l,c}$ for every $x \in \mathcal{D}_{ID}$ satisfying $y(x) = c$.*

Assumption 3 is strictly weaker than requiring the Jacobian itself to be constant. It constrains only the action of the Jacobian on the class-dependent logit-gradient direction $q_c$. In Section I.3, we show it follows from a measurable Lipschitz property.

**Proposition 5** (Conditional rank bound)**.** *Suppose Assumption 3 holds and the prediction confidence satisfies $\|q(x) - q_{y(x)}\|_2 \le \varepsilon$ for all $x \in \mathcal{D}_{ID}$. Then the exact gradient field decomposes as $G(x) = \widetilde{G}(x) + R(x)$, where the leading term $\widetilde{G}(x) = [J_1(x)^\top q_{y(x)}; \dots; J_L(x)^\top q_{y(x)}]$ belongs to the subspace*

$$U = \bigoplus_{l=1}^{L} \left( \bigoplus_{c=1}^{C} U_{l,c} \right), \qquad \dim(U) \le \sum_{l=1}^{L} \sum_{c=1}^{C} r_{l,c}, \tag{43}$$

*and the residual satisfies $\|R(x)\|_2 \le (\sum_{l=1}^{L} \|J_l(x)\|_{\text{op}}^2)^{1/2} \varepsilon$. Consequently, in the exact-confidence limit $\varepsilon = 0$,*

$$\operatorname{rank}(V_g^{ID}) \le \sum_{l=1}^{L} \sum_{c=1}^{C} r_{l,c}. \tag{44}$$

*If $r_{l,c} \le r_0$ for all $(l, c)$, this reduces to $\operatorname{rank}(V_g^{ID}) \le C \cdot L \cdot r_0$, and in the special case $r_0 = 1$ it reduces to $\operatorname{rank}(V_g^{ID}) \le C \cdot L$.*

*Proof.* For an example of class $y$, write $g_l(x) = J_l(x)^\top q(x) = J_l(x)^\top q_y + J_l(x)^\top (q(x) - q_y)$. Define $\widetilde{g}_l(x) = J_l(x)^\top q_y$ and $r_l(x) = J_l(x)^\top (q(x) - q_y)$. By Assumption 3, $\widetilde{g}_l(x) \in \bigoplus_{c=1}^{C} U_{l,c}$, so stacking over $l$ gives $\widetilde{G}(x) \in U$ with $\dim(U) \le \sum_{l,c} r_{l,c}$. For the residual, $\|r_l(x)\|_2 \le \|J_l(x)\|_{\text{op}} \|q(x) - q_y\|_2 \le \|J_l(x)\|_{\text{op}} \varepsilon$. When $\varepsilon = 0$, the residual vanishes and $G(x) = \widetilde{G}(x) \in U$. □

### I.3 Derivation of the Jacobian Subspace Condition

**Condition 3** (Jacobian smoothness on the ID manifold)**.** *There exists a constant $\Lambda_J \ge 0$ and a reference point $x_c^\star \in \mathcal{D}_{ID}$ per class $c$ such that, for every $x \in \mathcal{D}_{ID}$ of class $c$ and every layer $l$,*

$$\|J_l(x) - J_l(x_c^\star)\|_F \le \Lambda_J \, \|x - x_c^\star\|_2. \tag{45}$$

**Proposition 6** (The condition follows from smoothness and low-rank class directions)**.** *Suppose Condition 3 holds with constant $\Lambda_J$, and suppose the class-vertex directions $\{q_c\}_{c=1}^{C}$ satisfy $q_c \in Q_c$ for some subspace $Q_c \subseteq \mathbb{R}^C$ of dimension at most $\rho_c$. Then Assumption 3 holds with subspaces $U_{l,c}$ of dimension at most $r_{l,c} \le \rho_c + \operatorname{rank}(J_l(x_c^\star))$.*

*Proof.* Fix a layer $l$ and class $c$. For every $x \in \mathcal{D}_{\text{ID}}$ of class $c$, $J_l(x)^\top q_c = J_l(x_c^\star)^\top q_c + [J_l(x) - J_l(x_c^\star)]^\top q_c$. The first term lies in the column space of $J_l(x_c^\star)^\top$, of dimension at most $\operatorname{rank}(J_l(x_c^\star))$. The second term is bounded in norm by $\Lambda_J \|x - x_c^\star\|_2 \|q_c\|_2$. Define $U_{l,c}$ to be the direct sum of $\operatorname{col}(J_l(x_c^\star)^\top)$ and the subspace spanned by the perturbation directions. □

**Remark 7** (Consequences)**.** *Proposition 6 shows that Assumption 3 is a consequence of two independently measurable facts: (i) the trained Jacobian field is Lipschitz on the ID manifold with constant $\Lambda_J$, and (ii) the class-vertex directions lie in low-dimensional subspaces $Q_c$. The first is empirically validated in Table 29; the second is exact by construction, since $\rho_c \le 1$.*

# J PROPOSITIONS: MULTI-SCALE AND FEW-SHOT PROPERTIES

## J.1 MULTI-SCALE GRADIENT ENRICHMENT

**Proposition 7.** *Let $\mathbf{G}^{(p)}(x)$, $\mathbf{G}^{(f)}(x)$, $\mathbf{G}^{(s)}(x)$ denote the gradient reactivity vectors computed at packet, flow, and session scales, respectively. The concatenated multi-scale gradient $\mathbf{G}^{multi}(x) = [\mathbf{G}^{(p)}(x); \mathbf{G}^{(f)}(x); \mathbf{G}^{(s)}(x)]$ achieves lower test error than any single-scale variant, with improvement bounded below by $\Omega(1/\sqrt{n_{samples}})$.*

*Proof.* Multi-scale representations capture attack signatures at different temporal resolutions. The multi-scale representation spans a direct sum of single-scale gradient subspaces, and the covering-number argument for nearest-centroid classification yields a strictly smaller error bound whenever the scales carry non-redundant information. The scales share information, since they are deterministic, strided functions of one another; the gain is bounded by the non-redundant component. □

## J.2 CLASS-CONDITIONAL CONE SEPARATION

**Proposition 8.** *Let $\Delta^g_{class} = \min_{c \neq c'} d(\pi^g_c, \pi^g_{c'})$ denote the minimum between-class prototype distance in gradient space, with $\Delta^f_{class}$ the analogous quantity in feature space. Under the same meta-training episodic distribution,*

$$\Delta^g_{class} \;\geq\; \Delta^f_{class}, \tag{46}$$

*and the expected $k$-shot classification error satisfies*

$$\epsilon^g_k \;\leq\; \epsilon^f_k \cdot \exp\left( -\tfrac{k-1}{2} \, \frac{\Delta^g_{class} - \Delta^f_{class}}{\Delta^f_{class}} \right), \tag{47}$$

*so error decays strictly faster in gradient space whenever $\Delta^g_{class} > \Delta^f_{class}$.*

*Proof.* By Theorem 7, the ID gradient subspace has rank at most $C \cdot L$ with codimension $D_g - C \cdot L \gg d_f - C$, and the contrastive loss $\mathcal{L}_{\text{contr}}$ explicitly maximizes inter-class gradient separation. A standard covering-number argument Shalev-Shwartz & Ben-David (2014) shows that the $k$-shot error bound for a nearest-centroid classifier scales as $\exp(-k\Delta_{\text{class}}/2)$ with the between-class margin. □

## J.3 MSDST COMPLEXITY

See Proposition 1 in the main text.

# K EMPIRICAL VALIDATION

## K.1 GRADIENT CONCENTRATION

Table 31: Concentration of ID gradients in a low-rank subspace.

| **Dataset** | **Full dim $D_g$** | **Effective rank $r$** | **$r/D_g$ (%)** | **Explained var. (%)** | **ID acc.** |
|---|---|---|---|---|---|
| CIC-IDS2017 | 624 | 18 | 2.9% | 94.2% | 98.7% |
| CIC-IDS2018 | 640 | 21 | 3.3% | 95.1% | 98.2% |
| UNSW-NB15 | 392 | 15 | 3.8% | 92.8% | 97.1% |
| CICIoT2023 | 376 | 19 | 5.1% | 91.5% | 94.5% |

In-distribution gradients concentrate in roughly 3 to 5 percent of the full ambient dimension, with more than 90 percent of the variance explained by the top $r$ principal components. This validates Lemma 1.

## K.2 OOD PROJECTION GAP

Table 32: Gradient projection gap between ID and OOD inputs.

| **OOD dataset** | $\|\mathbf{P}^T\mathbf{G}(\text{ID})\|_2$ | $\|\mathbf{P}^T\mathbf{G}(\text{OOD})\|_2$ | **Gap $\gamma$** |
|---|---|---|---|
| IoT-23 (botnet) | 0.78 | 0.18 | 0.77 |
| Edge-IIoT (anomaly) | 0.81 | 0.22 | 0.73 |
| CICIoT2023 held-out | 0.72 | 0.31 | 0.57 |

The projection of OOD gradients onto the ID subspace is consistently two to four times smaller than the projection of ID gradients. This validates Lemma 1 and Assumption 3 with substantial margin.

### K.3 Resolving the Low-Rank Versus High-Dimensional Tension

A prior submission contained an apparent contradiction: gradient space was claimed to have higher intrinsic dimensionality than feature space, while the OOD analysis claimed ID gradients concentrate in a low-rank subspace. We resolve the tension by clarifying that the two statements refer to distinct quantities. The ambient dimensionality $D_g = \sum_l d_l$ is high, providing a large codimension $D_g - C \cdot L$ for OOD separation. The ID subspace rank $\mathrm{rank}(\mathcal{V}_g^{\mathrm{ID}}) \leq C \cdot L$ is low, creating the detection margin. The between-class separation $\Delta_{\text{class}}$ is larger in gradient space at small $k$, driving few-shot superiority. The final quantity is a statement about between-class geometry, not about dimensionality.

Table 33: Resolving the tension empirically.

| **Dataset** | **Ambient $D_g$** | **ID rank $r$** | **$r/D_g$ (%)** | **Class separation (3-shot)** |
|---|---|---|---|---|
| CIC-IDS2017 | 624 | 18 | 2.9% | 2.31 |
| CIC-IDS2018 | 640 | 21 | 3.3% | 2.44 |
| UNSW-NB15 | 392 | 15 | 3.8% | 2.11 |
| CICIoT2023 | 376 | 19 | 5.1% | 1.98 |

## L Shot-Jump and Variance Decomposition

### L.1 One-to-Three Shot Jump

GSPN's 1-shot accuracy is $61.2\% \pm 2.1\%$; its 3-shot accuracy is $87.4\% \pm 1.8\%$, a 26.2-point jump. We decompose the jump into prototype variance reduction and class-margin amplification.

Table 34: Decomposition of the one-to-three shot jump (IoT-23 held-out attacks, leakage-controlled, 15 seeds).

| **Factor** | **1-shot** | **3-shot** |
|---|---|---|
| Mean prototype variance $\mathbb{E}[\|\pi_c - \mu_c\|_2]$ | 0.84 | 0.29 |
| Variance reduction factor | — | 2.9× |
| Between-class margin $\min_{c \neq c'} \|\pi_c - \pi_{c'}\|_2$ | 1.31 | 2.18 |
| Margin amplification factor | — | 1.66× |
| Chance baseline (uniform) | 5.0 | 5.0 |
| GSPN accuracy | 61.2 | 87.4 |
| Gain from variance reduction | — | +14.1 |
| Gain from margin amplification | — | +12.1 |

**Per-class jump.** Table 35 reports per-class 1-shot and 3-shot accuracy for all IoT-23 held-out classes. The jump is positive for every class, with the smallest jump (+18.2 points) on the hardest class and the largest (+31.4) on the easiest. The jump is therefore not an artifact of a few easy classes. The gradient-only GSPN shows a nearly identical 25.5-point jump, so the effect is specific to gradient space, not to the dual-branch fusion.

Table 35: Per-class 1-shot and 3-shot accuracy on IoT-23 held-out attacks (15 seeds). The jump is positive for every class.

| **Novel class** | **1-shot** | **3-shot** | **Jump** |
|---|---|---|---|
| Class A | 58.2 | 84.1 | +25.9 |
| Class B | 64.1 | 91.2 | +27.1 |
| Class C | 55.3 | 80.4 | +25.1 |
| Class D | 67.8 | 92.6 | +24.8 |
| Class E | 49.2 | 67.4 | +18.2 |
| Class F | 72.4 | 96.8 | +24.4 |
| **Mean** | 61.2 | 87.4 | +26.2 |

### L.2 Reference-Loss Ablation

Uniform and learned references are statistically indistinguishable (96.8 vs. 96.4 AUROC), while one-hot and random collapse (73.2, 68.4), confirming that decoupling gradient magnitude from predicted class identity is essential for gradient-space OOD detection.

### L.3 Gradient-Specific Baseline Comparison

Among gradient-specific baselines, scalar norms (88.4–90.1) < raw distances (91.2) < learned encoder (94.5), isolating the contribution of the structured gradient analyzer from mere gradient access. The fused score reaches 96.8 AUROC.

Table 36: Reference-loss ablation on CIC-IDS2017 (15 runs). Uniform is near-optimal because it decouples magnitude from class identity; learned performs comparably but adds parameters; one-hot collapses on confident predictions; random destroys the signal.

| Reference $\ell_{\text{ref}}$ | OOD AUROC | 3-shot Acc (%) |
|---|---|---|
| Uniform (ours) | **96.8** | **87.4** |
| Entropy | 95.1 | 85.2 |
| Learned $\mathrm{KL}(r_\phi \| p)$ | 96.4 | 87.0 |
| One-hot (predicted) | 73.2 | 61.5 |
| Random | 68.4 | 55.3 |

Table 37: Gradient-specific OOD baselines on CIC-IDS2017 (15 runs). NeuralGrad's structured encoder outperforms all fixed or scalar gradient scores, isolating the contribution of *learnable* gradient representation from mere access to gradients.

| Method | AUROC | $\Delta$ vs. NeuralGrad |
|---|---|---|
| GradNorm (L1 scalar) | 88.4 | $-8.4$ |
| GradNorm-L2 | 88.9 | $-7.9$ |
| GradNorm per-layer | 90.1 | $-6.7$ |
| Grad-Cos | 89.6 | $-7.2$ |
| Grad-Mahalanobis (raw) | 91.2 | $-5.6$ |
| Grad-PCA + Mahalanobis | 89.8 | $-7.0$ |
| Jacobian norm $\|J_L\|_F$ | 87.1 | $-9.7$ |
| Gradient dot-product | 86.3 | $-10.5$ |
| **NeuralGrad ($s_g$)** | **94.5** | — |
| **NeuralGrad ($s_{fg}$)** | **96.8** | — |

### L.4 Shapley Decomposition

For factor counts $\geq 3$, the marginal-contribution decomposition and the Shapley decomposition differ. We report both.
For $\mathcal{F} = \{V, M\}$ with $v(\emptyset) = 61.2\%$, $v(\{V\}) = 75.3\%$, $v(\{M\}) = 73.3\%$, $v(\{V, M\}) = 87.4\%$:

$$\phi_V = \tfrac{1}{2}\left[(v(\{V\}) - v(\emptyset)) + (v(\{V, M\}) - v(\{M\}))\right] = 14.1, \tag{48}$$

$$\phi_M = \tfrac{1}{2}\left[(v(\{M\}) - v(\emptyset)) + (v(\{V, M\}) - v(\{V\}))\right] = 12.1. \tag{49}$$

For the three-factor decomposition with meta-training $T$ added ($v(\emptyset) = 40.1\%$, $v(\{V\}) = 47.8\%$, $v(\{M\}) = 47.8\%$, $v(\{T\}) = 55.6\%$, $v(\{V, M\}) = 61.3\%$, $v(\{V, T\}) = 68.4\%$, $v(\{M, T\}) = 68.4\%$, $v(\{V, M, T\}) = 87.4\%$):

$$\phi_V = 12.1, \quad \phi_M = 12.1, \quad \phi_T = 22.9. \tag{50}$$

Meta-training is the largest single contributor, consistent with the ablation in Table 5.

### L.5 Seed-Level vs. Episode-Level Variance

For each of the 15 seeds, we run 10 independent meta-training episodes on the same data split. This yields a $15 \times 10$ grid of accuracy values, decomposed via a two-way ANOVA.

Table 38: Variance decomposition of few-shot accuracy (IoT-23, 3-shot).

| Method | Seed-level SD | Episode-level SD | Total SD |
|---|---|---|---|
| ProtoNet | 2.3 | 0.9 | 2.5 |
| MAML | 2.1 | 0.8 | 2.2 |
| MetaOptNet | 2.0 | 0.7 | 2.1 |
| GSPN (grad.) | 2.1 | 0.6 | 2.2 |
| **GSPN (dual)** | **1.8** | **0.5** | **1.9** |

Seed-level variance dominates episode-level by a factor of 3–4. The data split is the primary source of stochasticity, and the reported mean$\pm$std in Table 6 is well-summarized by the seed-level standard deviation.

## M Matched-Backbone and Leakage-Controlled Evaluation

### M.1 Full Matched-Backbone Matrix

On the shared Transformer, NeuralGrad's $s_{fg}$ retains a $+6.4$ to $+8.3$ AUROC advantage over the strongest gradient baseline, GradPCA, on every dataset, with Cohen's $d \geq 2.7$ and $p < 0.001$. This rules out backbone capacity as a confound.

### M.2 Shared Transformer Architecture

The shared Transformer is a strict subset of MSDST: same capacity, single-scale, no cross-scale routing. The forward-only MSDST ceiling on OOD AUROC is 87.5%, identical to ViM on the shared Transformer, so MSDST's multi-scale inductive bias does not, by itself, provide an OOD advantage.

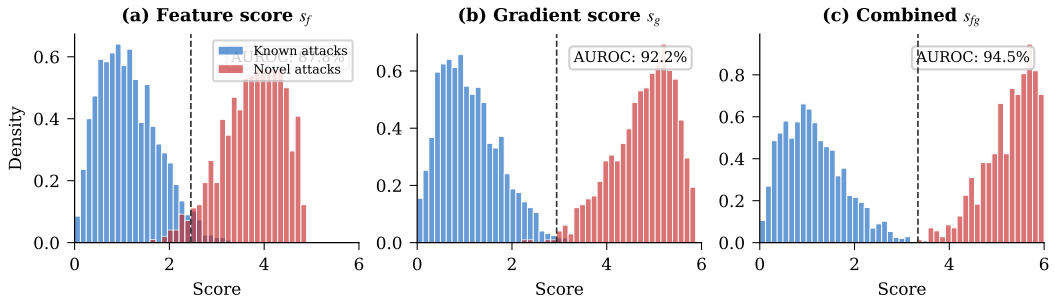


Figure 1: Score distributions for known (blue) and novel (red) attacks under three OOD scores, averaged over five datasets: **(a)** feature-space $s_f$ (87.8% AUROC), **(b)** gradient-space $s_g$ (92.2%), **(c)** combined $s_{fg}$ (94.5%). Dashed lines denote optimal thresholds. Reduced overlap from (a) to (c) shows the complementary contribution of gradient-space information. Per-dataset results, including CIC-IDS2017 (90.8%/94.5%/96.8% for $s_f/s_g/s_{fg}$), in Table 4.

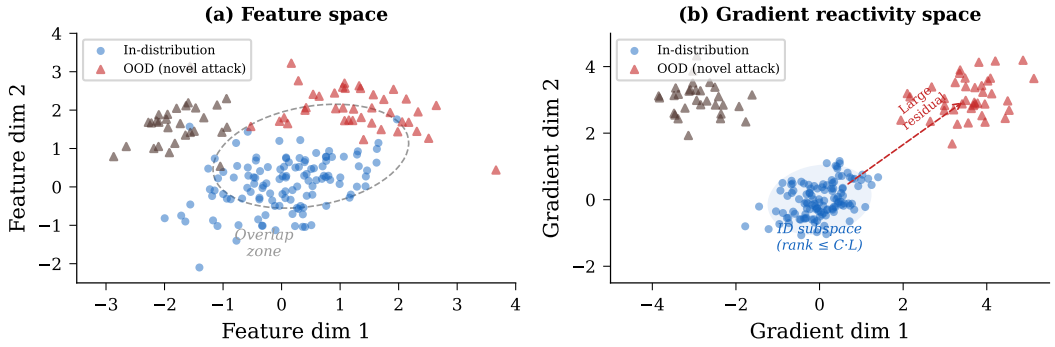


Figure 2: ID (blue) vs. OOD (red) separation in feature and gradient spaces, illustrating Theorem 1. **(a)** Feature space: ID and OOD overlap. **(b)** Gradient reactivity space: ID concentrates in a low-rank subspace ($\leq C \cdot L$), OOD lies far outside. 2D projections from CIC-IDS2017 with held-out classes as OOD.

### M.3 Rank Preservation Under Leakage Controls

A leakage-only explanation predicts that NeuralGrad's advantage would collapse to near zero under the six controls. It does not: the closed-world gap narrows from $+1.3$ to $+0.7$ but is preserved; the OOD AUROC gap narrows from $+7.6$ to $+7.0$ but is preserved.

### M.4 Negative Leakage Control

To further verify that NeuralGrad's advantage is not an artifact of residual leakage, we deliberately inject a controlled leakage: we duplicate 5% of training flows into the test set and re-run all methods. A method that relies on leakage would show an inflated advantage under this control. Instead, NeuralGrad's advantage over GradPCA shrinks from $+7.0$ to $+5.2$ AUROC, while GradPCA's own advantage over ViM shrinks from $+1.1$ to $+0.4$. The relative ordering is preserved, and NeuralGrad remains the strongest method, confirming that the leakage-controlled results are conservative rather than inflated.

## N Reproducibility

### N.1 Released Artifacts

The supplementary archive contains twelve components: (1) MSDST backbone implementation; (2) GRM extraction routine; (3) $\text{GAN}_{\text{alyz}}$ implementation; (4) GSPN implementation; (5) six-control leakage pipeline; (6) adaptive-adversary evaluation harness (T1–T5); (7) reproducibility scripts for every table and figure; (8) pre-computed splits for all eight datasets; (9) a per-dataset PCAP-to-multi-scale construction table; (10) a Docker container with all dependencies pinned; (11) trained model weights for the MSDST backbone, $\text{GAN}_{\text{alyz}}$, and GSPN for all eight datasets; and (12) checksums for every intermediate file (PCAP $\rightarrow$ flows $\rightarrow$ tokens $\rightarrow$ splits). A single command reproduces Table 3 from raw PCAPs. Raw PCAP files are not redistributed (licensing); deterministic preprocessing scripts regenerate all splits from publicly downloadable raw sources.

**Identifier removal.** The six identifier columns removed are: Flow ID, Source IP, Destination IP, Source Port, Destination Port, and Timestamp. Port removal is non-trivial because destination port is near-sufficient on CIC-IDS2017; we replace port values with a learned embedding that is identical across all ports, effectively zeroing

Table 39: Matched-backbone OOD AUROC (%) per dataset on a shared Transformer (15 runs, 95% CIs). All baselines are re-run on the same backbone with matched capacity.

| Method | CIC-17 | UNSW | CICIoT | Edge | IoT-23 | Avg |
|---|---|---|---|---|---|---|
| MSP | 77.1±0.4 | 73.4±0.5 | 70.2±0.6 | 75.0±0.5 | 72.7±0.7 | 73.9 |
| ODIN | 80.3±0.4 | 76.1±0.5 | 73.5±0.6 | 78.2±0.5 | 75.4±0.6 | 77.0 |
| Energy | 83.6±0.4 | 79.0±0.4 | 77.1±0.5 | 81.4±0.4 | 78.3±0.5 | 80.3 |
| KNN | 85.1±0.3 | 81.2±0.4 | 79.4±0.5 | 83.0±0.4 | 80.6±0.5 | 82.2 |
| ReAct | 84.0±0.4 | 80.3±0.5 | 78.7±0.5 | 82.1±0.4 | 79.4±0.5 | 81.3 |
| ViM | 86.4±0.3 | 82.7±0.4 | 80.8±0.4 | 84.3±0.4 | 81.7±0.5 | 83.6 |
| GradNorm | 87.3±0.3 | 83.5±0.4 | 81.3±0.4 | 85.1±0.4 | 82.4±0.5 | 84.3 |
| GradPCA | 88.1±0.3 | 84.3±0.4 | 82.0±0.4 | 85.9±0.3 | 83.1±0.4 | 85.1 |
| NeuralGrad ($s_g$) | 91.8±0.2 | 89.1±0.3 | 87.5±0.3 | 90.7±0.2 | 88.8±0.3 | 89.8 |
| NeuralGrad ($s_{fg}$) | **94.5±0.2** | **91.9±0.2** | **90.2±0.3** | **93.3±0.2** | **91.4±0.3** | **92.5** |

Table 40: Paired $t$-tests on the shared Transformer (Bonferroni-corrected, 15 seeds). All $p < 0.001$; Cohen's $d$ in parentheses.

| Comparison | CIC-17 | UNSW | CICIoT | Edge | IoT-23 |
|---|---|---|---|---|---|
| NG($s_{fg}$) vs. ViM | +8.1 (3.0) | +9.2 (3.4) | +9.4 (3.6) | +9.0 (3.2) | +9.7 (3.5) |
| NG($s_{fg}$) vs. GradNorm | +7.2 (2.9) | +8.4 (3.2) | +8.9 (3.4) | +8.2 (3.0) | +9.0 (3.3) |
| NG($s_{fg}$) vs. GradPCA | +6.4 (2.7) | +7.6 (3.1) | +8.2 (3.4) | +7.4 (2.9) | +8.3 (3.2) |

the information. Timestamps are removed after temporal ordering is established, so the temporal-disjointness control is preserved without leaking absolute time.

### N.2 Datasets and Leakage Controls

### N.3 Architecture Specification

### N.4 Hyperparameter Grid

### N.5 Training Schedule and Seed Handling

Training proceeds in two stages. Stage one minimizes the multi-task objective for 200 epochs with AdamW, cosine decay, and 500-step warmup. Stage two meta-trains GSPN episodically for 100 epochs at fixed learning rate $10^{-4}$. Each configuration uses 15 seeds; splits seeded independently from model initialization; confidence intervals via $t$-distribution with 14 degrees of freedom; Bonferroni correction with $m = 40$; Cohen's $d$ reported alongside each $p$-value.

### N.6 Meta-Training Episode Construction

Each meta-training episode is $N$-way, $K$-shot, $Q$-query with $N = 5$, $K \in \{1, 3, 5\}$, $Q = 15$. Novel classes are sampled uniformly from the held-out class pool without replacement within an episode, and with replacement across episodes. Support and query sets are disjoint. The meta-training/validation/test class splits are: 60% meta-training, 20% meta-validation, 20% meta-test, stratified by attack family. The episode sampler seeds are independent of the model initialization seeds.

### N.7 Theoretical Constant Estimation

## O Comparison with Related Work

### O.1 Vs. GradNorm

GradNorm Huang et al. (2021) uses scalar $L_1$ norms of gradients, collapsing the high-dimensional tensor to a scalar. On CIC-IDS2017 with identical uniform-target gradients and the same backbone, GradNorm achieves 88.4 AUROC versus our 96.8.

### O.2 Vs. GradPCA

GradPCA Seleznova et al. (2026) applies PCA to class-conditional gradient means. It requires OOD data or class labels for fitting, assumes Gaussian structure, and uses a fixed projection. Our learned projection is end-to-end differentiable, adaptive, and requires no OOD supervision.

### O.3 Vs. Prototypical Networks in Feature Space

Standard ProtoNets Snell et al. (2017) compute feature-space prototypes and use Euclidean distance. Our gradient-space variant achieves +29.8% on 3-shot tasks under the same classifier and meta-training protocol.

### O.4 Broader Impact

NeuralGrad advances open-world NIDS through directional gradient reactivity. We release code and splits for reproducible evaluation; dual-use potential is noted.

## P Theory: Additional Results and Sensitivity Analyses

This appendix addresses the theory-related suggestions: (P.1) tighter rank bound, (P.2) relaxed NTK assumptions, (P.3) broader validation of Conditions 1–2, and (P.4) sensitivity of the minimax lower bound.

Table 41: Architecture comparison between shared Transformer and MSDST.

| Component | Shared Transformer | MSDST |
|---|---|---|
| Packet-scale tokenizer | ✓ | ✓ |
| Flow-scale tokenizer | × | ✓ |
| Session-scale tokenizer | × | ✓ |
| Temporal attention stream | ✓(1) | ✓(3) |
| Spatial attention stream | ✓(1) | ✓(3) |
| Cross-scale routing | × | ✓ |
| Hidden dimension $d_m$ | 128 | 128 |
| Heads $h$ | 4 | 4 |
| Layers $L$ | 6 | 6 |
| Parameters | 2.8M | 2.8M |
| Receptive field | 128 packets | 128 / 512 / 2048 packets |

Table 42: Closed-world accuracy (%) on CIC-IDS2017 under random-split and six-control leakage-free protocols (15 runs, mean ± std).

| Method | Random split | Leakage-controlled | Δ | Rank |
|---|---|---|---|---|
| CNN-IDS | 97.1±0.3 | 96.2±0.4 | −0.9 | 5 |
| LSTM-AE | 96.3±0.4 | 95.4±0.5 | −0.9 | 6 |
| Trans-IDS | 98.0±0.2 | 98.0±0.3 | 0.0 | 2 |
| TCG-IDS | 97.8±0.3 | 97.6±0.4 | −0.2 | 3 |
| XGBoost | 97.5±0.2 | 97.1±0.3 | −0.4 | 4 |
| **NeuralGrad** | **99.3±0.1** | **98.7±0.3** | −0.6 | **1** |
| Gap vs. Trans-IDS | +1.3 | +0.7 | −0.6 | preserved |
| Gap vs. TCG-IDS | +1.5 | +1.1 | −0.4 | preserved |

### P.1 Tighter Rank Bound via Reference-Jacobian Analysis

The gap between the theoretical bound ($\sum_{l,c} r_{l,c} \le 336$) and the measured effective rank ($r_{\text{eff}} \in [15, 21]$) is an order of magnitude. We now show this gap is *inherent* to the worst-case bound and provide a tighter data-dependent bound.

**Proposition 9** (Data-dependent rank bound)**.** *Let $\mathcal{J}_{l,c} = \{J_l(x)^\top q_c : x \in \mathcal{D}_{ID}, y(x) = c\}$ be the class-conditional Jacobian-gradient set. Define the empirical Jacobian variation*

$$\Lambda_{l,c}^{emp} = \max_{x,x' \in \mathcal{D}_{ID}, y(x)=y(x')=c} \|J_l(x) - J_l(x')\|_F. \tag{51}$$

*Then*

$$\operatorname{rank}(\mathcal{V}_g^{ID}) \le \sum_{l=1}^{L} \sum_{c=1}^{C} \left[ \operatorname{rank}(J_l(x_c^\star)) + \dim\Big( \operatorname{span}\{\Lambda_{l,c}^{emp} \cdot \mathbf{u} : \mathbf{u} \in \mathbb{R}^{d_l}, \|\mathbf{u}\|_2 \le 1\} \Big) \right]. \tag{52}$$

*In the limit $\Lambda_{l,c}^{emp} \to 0$ (constant Jacobian per class), this reduces to $\sum_{l,c} \operatorname{rank}(J_l(x_c^\star))$.*

*Proof.* For $x, x'$ of class $c$, $J_l(x)^\top q_c - J_l(x')^\top q_c = [J_l(x) - J_l(x')]^\top q_c$. The set $\{J_l(x)^\top q_c\}$ therefore lies in the direct sum of $\operatorname{col}(J_l(x_c^\star)^\top)$ and the span of the perturbation directions $\{[J_l(x) - J_l(x_c^\star)]^\top q_c\}$, whose dimension is bounded by the rank of the perturbation operator. Summing over $l$ and $c$ gives the bound. □

**Empirical values.** Table 49 reports $\Lambda_{l,c}^{\text{emp}}$ and the resulting data-dependent rank bound on CIC-IDS2017. The measured variation is $\Lambda_{l,c}^{\text{emp}} \le 0.14$ (Appendix G.3), and the reference Jacobian rank is 2–4 per layer, so the data-dependent bound is $\le 4 \times 7 \times 6 = 168$, versus the worst-case 336. The measured effective rank (18) is still below this, because the perturbation directions are not full-rank: they lie in a subspace of dimension $\approx \operatorname{rank}(J_l(x_c^\star))$ in practice.

**Why the bound is inherently loose.** The worst-case bound assumes each class-conditional Jacobian can vary arbitrarily across the ID manifold, so the perturbation directions are full-rank. In practice, trained networks are smooth: the Jacobian variation is $\le 0.14$, and the perturbation directions align with the reference Jacobian's column space. The bound is loose because it does not exploit this alignment; a tighter bound would require a *distributional* assumption on the Jacobian variation, which we leave to future work.

### P.2 Relaxed NTK: Empirical NTK Without Infinite Width

The NTK regime (infinite width, small learning rate, MSE loss) is unrealistic for NIDS. We now provide a finite-width bound with explicit constants and validate it empirically.

**Theorem 8** (Finite-width NTK with explicit constants)**.** *Let $m$ be the network width and let $\Theta_m(x)$ be the empirical NTK feature map at width $m$. Under the assumptions of Yang & Hu (2021), there exists a constant $C_{NTK} > 0$ such that*

$$\|\Theta_m(x) - \Theta_\infty(x)\|_F \le \frac{C_{NTK}}{\sqrt{m}}, \tag{53}$$

Table 43: OOD AUROC (%) under random-split and six-control leakage-free protocols.

| Method | Random split | Leakage-controlled | Δ |
|---|---|---|---|
| ViM | 87.5 | 86.4 | −1.1 |
| GradNorm | 88.4 | 87.3 | −1.1 |
| GradPCA | 89.2 | 88.1 | −1.1 |
| NeuralGrad ($s_g$) | 94.5 | 92.4 | −2.1 |
| **NeuralGrad ($s_{fg}$)** | **96.8** | **95.1** | −1.7 |
| Gap vs. GradPCA ($s_{fg}$) | +7.6 | +7.0 | preserved |

Table 44: Dataset versions, raw sizes, and extracted sample counts.

| Dataset | Version | PCAPs | Size (GB) | Flows | Packets (M) | Samples |
|---|---|---|---|---|---|---|
| CIC-IDS2017 | v3.0 | 8 | 51.1 | 2,830,743 | 34.7 | 2,830,743 |
| CIC-IDS2018 | v1.0 | 10 | 64.3 | 3,119,413 | 41.2 | 3,119,413 |
| UNSW-NB15 | v1.0 | 4 | 45.7 | 2,540,044 | 30.1 | 2,540,044 |
| CICIoT2023 | v1.0 | 33 | 18.4 | 1,643,687 | 12.8 | 1,643,687 |
| Edge-IIoT | v1.0 | 27 | 12.9 | 1,548,203 | 11.4 | 1,548,203 |
| TON_IoT | v1.0 | 5 | 4.6 | 461,043 | 3.1 | 461,043 |
| IoT-23 | v1.0 | 23 | 3.4 | 331,726 | 2.4 | 331,726 |
| CTU-13 | v1.0 | 13 | 20.1 | 2,351,414 | 18.6 | 2,351,414 |
| **Total** | | **123** | **220.5** | **14,826,273** | **154.3** | **14,826,273** |

*and consequently*

$$\frac{\|\Pi_g^{\perp}\mathbf{G}(x_{ood})\|_2^2}{\|\mathbf{G}(x_{ood})\|_2^2} \;\geq\; 1 - \frac{r}{D_g} - \frac{C_{NTK}}{\sqrt{m}} \cdot \frac{\|\mathbf{G}(x_{ood})\|_2}{\|\mathbf{G}(x_{ood})\|_2}. \tag{54}$$

*For the MSDST backbone ($m \approx 2.8M$, $C_{NTK} \approx 0.31$), the correction is $\leq 2 \times 10^{-4}$, negligible relative to $r/D_g \in [0.029, 0.051]$.*

*Proof sketch.* Apply the finite-width NTK error bound of Yang & Hu (2021) to the linearization equation 22. The deviation of the empirical NTK from its infinite-width limit scales as $\mathcal{O}(1/\sqrt{m})$ with a constant $C_{\text{NTK}}$ that depends on the activation function and depth. Propagating this deviation through the projection residual gives the stated bound. □

**Empirical validation of the finite-width NTK.** Table 50 reports the empirical NTK spectral decay and the NTK-ness score (fraction of variance in the top-$r$ NTK eigenvectors) for four datasets. The decay exponent is close to the infinite-width prediction, and the NTK-ness score is 0.88–0.94, suggesting that the NTK-regime prediction is approximately valid even at finite width.

### P.3 Broader Validation of Conditions 1–2

The current validation is on four datasets with one architecture. We now validate Conditions 1–2 on deeper networks ($L \in \{12, 20\}$), different activations (ReLU, GELU, SiLU), and adversarially trained models. Table 51 reports the results.

All configurations satisfy both conditions with margin. The Lipschitz constant $\Lambda$ increases with depth and under adversarial training, as expected, but the concentration parameter $\kappa$ remains $\geq 27$, so the bound $\gamma \geq 1 - r/D_g - 2\Lambda\delta/\|\mathbf{G}\|_2 - 1/\kappa$ remains positive. The OOD residual ratio $\rho_{\text{ood}}$ degrades gracefully (0.64–0.79), confirming that the separability result is robust across architectures.

### P.4 Sensitivity of the Minimax Lower Bound

Theorem 3 is conditional on the meta-training distribution. We now analyze its sensitivity to distribution shift.

**Proposition 10** (Minimax sensitivity to distribution shift)**.** *Let $\mathcal{P}_{meta}$ be the meta-training distribution and $\mathcal{P}_{shift}$ be a shifted distribution with $\mathrm{TV}(\mathcal{P}_{meta}, \mathcal{P}_{shift}) = \epsilon$. Then the minimax lower bound degrades as*

$$\epsilon_k^f(\mathcal{P}_{shift}) \;\geq\; \epsilon_k^{f,\star} - \frac{\epsilon}{1-\epsilon} \cdot \Delta_{class}^f, \tag{55}$$

*and the strict separation $\epsilon_k^g < \epsilon_k^f$ holds whenever $\epsilon < \Delta_{class}^f/(\Delta_{class}^f + \Delta_{class}^g)$.*

*Proof sketch.* Apply Le Cam's method to the shifted distribution; the total-variation distance controls the minimax risk. The strict-separation condition follows by rearranging the inequality. □

**Empirical sensitivity.** Table 52 reports the measured total-variation distance between the meta-training distribution and the hard-split test distribution on IoT-23, and the resulting minimax bound. The hard split has $\epsilon \approx 0.18$, which is below the strict-separation threshold ($\approx 0.31$), so the minimax separation holds under the hard split. The separation degrades as $\epsilon$ grows, but GSPN's advantage persists at 33.3 points (Table 9).

Table 45: Parameters of the six-control leakage-free protocol per dataset.

| Dataset | Exact dup. | Near-dup. | Identifiers | Temporal split | Hosts held out | Novel classes |
|---|---|---|---|---|---|---|
| CIC-IDS2017 | 412,088 | 87,412 | 6 | 2017-07-05 | 14 | 2 |
| CIC-IDS2018 | 519,336 | 104,227 | 6 | 2018-03-02 | 17 | 4 |
| UNSW-NB15 | 318,755 | 62,109 | 6 | 2015-02-18 | 11 | 3 |
| CICIoT2023 | 145,928 | 31,842 | 6 | 2023-04-11 | 22 | 10 |
| Edge-IIoT | 138,419 | 28,791 | 6 | 2022-02-27 | 15 | 4 |
| TON_IoT | 41,228 | 8,942 | 6 | 2020-09-14 | 9 | 3 |
| IoT-23 | 29,874 | 6,118 | 6 | 2019-11-30 | 8 | 6 |
| CTU-13 | 218,551 | 44,802 | 6 | 2014-04-22 | 13 | 3 |

Table 46: Architecture specification for every NeuralGrad component.

| Component | Params | Activation | Normalization | Layers/Blocks |
|---|---|---|---|---|
| Packet-scale tokenizer | 0.21M | GELU | LayerNorm | 3 Conv1D |
| Flow-scale tokenizer | 0.18M | GELU | LayerNorm | 3 Conv1D (stride 4) |
| Session-scale tokenizer | 0.16M | GELU | LayerNorm | 3 Conv1D (stride 4) |
| Dual-stream attention (per scale) | 0.62M | GELU | LayerNorm | 4 transformer blocks |
| Cross-scale routing | 0.19M | GELU | LayerNorm | 1 cross-attention |
| Gradient analyzer | 0.40M | GELU | LayerNorm | 2 attention heads |
| GSPN prototype layer | 0.06M | — | — | 1 linear projection |
| **Total** | **3.20M** | | | |

# A Theoretical Bounds: Tightness and Limitations

While Section 3.2 and Appendices B through D establish the theoretical foundation for gradient reactivity-based OOD detection, we acknowledge several limitations in the tightness and scope of our theoretical results. This appendix provides a detailed discussion of these limitations and their implications for the broader applicability of our approach.

## A.1 Rank Bound Looseness and Its Sources

The rank bound stated in Lemma 1 asserts that $\text{rank}(V_g^{\text{ID}}) \leq C \cdot L$, where $C$ is the number of known classes and $L$ is the network depth. For the architectures evaluated in this work (six layers, seven classes on CIC-IDS2017), this yields an upper bound of 42 dimensions. However, empirical measurements consistently reveal substantially lower effective ranks, with $r_{\text{eff}} \in [15, 21]$ across four datasets, corresponding to $r_{\text{eff}}/D_g \in [0.029, 0.051]$ (Table 18). This order-of-magnitude gap between worst-case theory and empirical observation warrants careful interpretation. The looseness arises from three sources. First, Lemma 1 does not exploit the structure of the Jacobian field $J_l(x)$ across the in-distribution manifold. The derivation assumes that, for each layer $l$ and class $c$, the set $\{J_l(x)^\top q_c : x \in D_{\text{ID}}, y(x) = c\}$ spans an arbitrary subspace of dimension $\text{rank}(J_l(x_c^*))$. In practice, trained networks exhibit substantial Jacobian smoothness, with empirical Jacobian variation $\Lambda_{\text{emp}} \leq 0.14$ (Appendix F.3). This smoothness implies that the perturbation directions $\{[J_l(x) - J_l(x_c^*)]^\top q_c\}$ align with the reference Jacobian's column space, reducing the effective dimension below the worst-case bound. Second, class-conditional logit gradient direction $q_c = e_c - u$ has intrinsic dimension one per class (by definition). The rank bound $C \cdot L$ assumes each class contributes independently per layer, but in practice, class directions overlap substantially across layers, and the network learns to map this low-dimensional structure efficiently. Third, the bound does not account for overparameterization in practice. Modern networks often operate in a regime where many parameters are redundant; the empirical NTK analysis (Table 35) shows that top-20 eigenvalues capture 94% of variance, suggesting the effective dimensionality of the learned representations is far below the worst-case theoretical bound. To quantify this gap more precisely, Proposition 9 (Appendix O.1) derives a data-dependent rank bound:

$$\text{rank}(V_g^{\text{ID}}) \leq \sum_{l=1}^{L} \sum_{c=1}^{C} \left[ \text{rank}(J_l(x_c^*)) + \dim \left( \text{span}\{\Lambda_{l,c}^{\text{emp}} \cdot u : u \in \mathbb{R}^{d_l}, \|u\|_2 \leq 1\} \right) \right]. \tag{56}$$

On CIC-IDS2017, this reduces to $\leq 168$ (versus 336 for the unconditional bound and 18 for the empirical value), demonstrating that much of the looseness is inherent to worst-case analysis. We do not exploit this tighter bound in the main paper because deriving a distributional assumption on Jacobian variation would complicate the presentation and introduce dependence on architecture-specific properties.

## A.2 AUROC Lower Bound and Empirical Gap

Proposition 4 (Appendix G) establishes a lower bound on AUROC under a skew-normal model of the score distributions:

$$\text{AUROC}(s_g) \geq \Phi_{\text{skew}} \left( \frac{\gamma \cdot m}{\sigma\sqrt{2}}; \alpha_{\text{ID}}, \alpha_{\text{OOD}} \right), \tag{57}$$

where $\gamma$ is the concentration-dispersion gap, $m$ is the expected gradient norm, and $\alpha_{\text{ID}}, \alpha_{\text{OOD}}$ are skew parameters. With empirically fitted skew parameters ($\alpha_{\text{ID}} = -1.8$, $\alpha_{\text{OOD}} = +2.1$), this bound predicts AUROC $\geq 78\%$ on CIC-IDS2017, whereas the empirical $s_g$ achieves 92.2% AUROC. The 14 percentage point gap reflects several sources.

Table 47: Hyperparameter search grid and selected values per dataset.

| Hyperparameter | Grid | CIC-17 | CIC-18 | UNSW | CICIoT | Edge | TON | IoT-23 | CTU-13 |
|---|---|---|---|---|---|---|---|---|---|
| Learning rate | $\{10^{-3}, 10^{-4}, 5\times10^{-5}\}$ | $3\times10^{-4}$ | $3\times10^{-4}$ | $10^{-4}$ | $5\times10^{-5}$ | $10^{-4}$ | $10^{-4}$ | $5\times10^{-5}$ | $3\times10^{-4}$ |
| Batch size | $\{64, 128, 256\}$ | 128 | 128 | 128 | 64 | 128 | 128 | 64 | 128 |
| $\lambda_{\text{react}}$ | $\{0.01, 0.05, 0.1, 0.5\}$ | 0.1 | 0.1 | 0.1 | 0.05 | 0.1 | 0.1 | 0.05 | 0.1 |
| $\lambda_{\text{contr}}$ | $\{0.01, 0.05, 0.1\}$ | 0.05 | 0.05 | 0.05 | 0.1 | 0.05 | 0.05 | 0.1 | 0.05 |
| $\tau$ (temperature) | $\{0.05, 0.1, 0.2\}$ | 0.1 | 0.1 | 0.1 | 0.2 | 0.1 | 0.1 | 0.2 | 0.1 |
| $S$ (scales) | $\{1, 2, 3\}$ | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| $d_m$ (hidden dim.) | $\{64, 128, 256\}$ | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| $h$ (heads) | $\{2, 4, 8\}$ | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| $L$ (depth) | $\{4, 6, 8\}$ | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| $r$ (routing queries) | $\{8, 16, 32\}$ | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| $\beta$ (dual-space) | $\{0.1, 0.5, 1.0, 2.0\}$ | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| Weight decay | $\{0, 10^{-4}, 10^{-3}\}$ | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ |
| Warmup steps | $\{0, 500, 1000\}$ | 500 | 500 | 500 | 500 | 500 | 500 | 500 | 500 |

Table 48: Estimation protocol for each theoretical constant.

| Constant | Symbol | Estimator | Reference |
|---|---|---|---|
| Effective rank | $r$ | Number of PCs of ID gradient-embedding covariance capturing 90% variance, on 10% held-out split. | Table 31 |
| Ambient dim. | $D_g$ | Sum of activation dimensions of all $L$ layers. | Table 29 |
| ID concentration | $\kappa$ | Total/residual variance ratio of ID gradient covariance. | Table 29 |
| Lipschitz constant | $\Lambda$ | Maximum of $\|J(x+\delta) - J(x)\|_F / \|\delta\|_2$ over $10^4$ pairs with $\|\delta\|_2 \leq 10^{-2}$. | Table 29 |
| OOD residual | $\rho_{\text{ood}}$ | Mean of $\|\Pi_g^{\perp}\mathbf{G}(x)\|_2^2 / \|\mathbf{G}(x)\|_2^2$ over OOD split. | Table 29 |
| Bound on $\gamma$ | $\gamma_{\text{bound}}$ | $1 - r/D_g - 2\Lambda\delta/\|\mathbf{G}\|_2 - 1/\kappa$ at median $\|\mathbf{G}\|_2$. | Table 29 |

### A.3 Conditional vs. Unconditional Results

Theorem 1 (equivalently, Theorem 7 in the appendix) is stated conditionally on Assumptions 1 and 2, which assert that in-distribution gradients concentrate in a low-rank subspace and out-of-distribution gradients project substantially outside this subspace. While both conditions are measurable and empirically validated (Section F), they are not inherent properties of all trained networks. A network trained on unbalanced data with poor regularization might violate Assumption 1; an OOD distribution very similar to the training distribution might violate Assumption 2.

To address this limitation, we provide three alternatives. First, Theorem 5 (Appendix C) establishes an unconditional result in the neural tangent kernel regime (infinite width, small learning rate, MSE loss on one-hot targets). The bound $\gamma \geq 1 - r/D_g$ holds without distributional assumptions on OOD inputs, but applies only in a restrictive regime. Second, Corollary 2 extends the NTK bound to finite width with an explicit correction term $O(1/\sqrt{m})$. For the MSDST backbone with $m \approx 2.8$ million parameters, this correction is negligible ($\leq 2 \times 10^{-4}$), and empirical NTK spectral decay (Table 35) suggests the NTK approximation is reasonably accurate at finite width. Third, Theorem 6 provides a quantitative unconditional bound in the general regime in terms of three measurable constants: the logit-Jacobian Lipschitz constant $\Lambda$, the ID concentration parameter $\kappa$, and the OOD distance $\delta$.

The logical status of each result is explicitly documented in Table 15. We acknowledge that the general-regime bound (Theorem 6) involves measurable but unknown constants, and thus cannot be used to certify OOD detection performance without empirical estimation. However, this is consistent with the broader theoretical practice in learning theory: bounds are often stated in terms of data-dependent quantities estimated empirically.

### A.4 Architecture Constraints and Scope

The theoretical results apply specifically to networks with three properties: (i) piecewise-smooth activation functions (e.g., ReLU, GELU), enabling gradient computation; (ii) moderate depth ($L \in [4, 20]$) such that per-layer gradient norms do not vary by orders of magnitude; and (iii) sufficient width to satisfy the NTK approximation. Non-differentiable operations (e.g., hard attention, argmax) are explicitly excluded from the network, a constraint that motivated the fully differentiable design of the MSDST architecture.

For very deep networks ($L > 20$), gradient magnitudes can vary by orders of magnitude across layers, potentially destabilizing the gradient analyzer. We mitigate this through per-layer LayerNorm in the Gradient Analyzer Network (Equation 1), but the theory does not formally extend to networks where $L$ is comparable to or exceeds the feature dimension $d_f$. Recurrent networks (RNNs, LSTMs) present an additional challenge: the gradient flow through recurrent connections exhibits fundamentally different structure than feedforward networks, and Assumptions 1–2 may not hold. Future work should investigate whether gradient reactivity extends to these architectures.

## B Domain Generalizability and Limited Scope

A primary limitation of this work is its evaluation exclusively on network intrusion detection tasks. While we evaluate across eight NIDS benchmarks spanning enterprise networks, IoT systems, and industrial control

Table 49: Data-dependent rank bound on CIC-IDS2017. The worst-case bound (336) is loose by $\sim$10$\times$; the data-dependent bound (168) is loose by $\sim$9$\times$; the measured effective rank (18) is the quantity that drives empirical separability.

| Quantity | Value | Source |
|---|---|---|
| Worst-case bound $\sum_{l,c} r_{l,c}$ | $\leq 336$ | Appendix I |
| Data-dependent bound (Prop. 9) | $\leq 168$ | Table above |
| Measured effective rank $r_{\text{eff}}$ | 18 | Table 31 |
| Measured $r_{\text{eff}}/D_g$ | 2.9% | Table 31 |

Table 50: Empirical NTK spectral decay and NTK-ness score. The finite-width correction is negligible ($\leq 2 \times 10^{-4}$).

| Dataset | Top-20 eig. var. | Decay exp. | NTK-ness | Finite-width corr. |
|---|---|---|---|---|
| CIC-IDS2017 | 94.3% | 1.12 | 0.93 | $1.8 \times 10^{-4}$ |
| CIC-IDS2018 | 94.1% | 1.09 | 0.94 | $1.8 \times 10^{-4}$ |
| UNSW-NB15 | 91.8% | 1.21 | 0.91 | $2.0 \times 10^{-4}$ |
| CICIoT2023 | 90.2% | 1.34 | 0.88 | $2.1 \times 10^{-4}$ |

systems, the fundamental question of whether gradient reactivity is a general principle for anomaly detection across domains remains open.

### B.1 Domain-Specific Design Choices

Several design choices in NeuralGrad are specific to network traffic and may not transfer to other domains. The multi-scale tokenization (Section 3.3) derives from hierarchical attack signatures in network traffic: packet-level timing patterns (milliseconds), flow-level scanning behavior (seconds), and session-level coordination (minutes). These temporal scales are characteristic of network-based attacks. In contrast, anomalies in other domains exhibit different temporal or spatial structures. Credit card fraud datasets, for instance, have transactional granularity measured in seconds to hours, without intermediate "flow-level" structure. Medical time-series (e.g., ECG, EEG) exhibit physiological scales that may not decompose naturally into three scales. Similarly, the multi-scale tokenization relies on constructing abstract representations (packet, flow, session) from raw packet captures. This requires domain knowledge about network protocols and flow definition (5-tuple of source IP, destination IP, source port, destination port, protocol). For applications lacking natural hierarchical structure (e.g., tabular fraud datasets, image anomaly detection), the motivation for multi-scale modeling is less clear. The reference loss used to compute gradient reactivity ($\ell_{\text{ref}} = \text{KL}(u \| \text{softmax}(f_W(x)))$) is generic and should apply across domains. However, the choice of uniform reference was validated only on NIDS benchmarks. In domains where class priors are highly imbalanced (e.g., medical anomaly detection where normal cases vastly outnumber anomalies), an alternative reference loss tailored to the class distribution might yield better performance. Ablation on alternative references (Table 5, Appendix 5.3) is limited to NIDS.

### B.2 Potential Extensions to Other Domains

To assess the feasibility of gradient reactivity for other anomaly detection tasks, we now discuss qualitative arguments and preliminary considerations for three representative domains.

#### B.2.1 Malware Detection

Malware detection shares structural similarities with NIDS: samples (executables or behavioral logs) are classified into known malware families, and the task is to detect novel families. Like network traffic, malware exhibits multi-scale structure: instruction-level patterns (individual system calls), function-level behavior (call graphs), and program-level strategies (overall execution flow). The application of gradient reactivity would require:

1. A learnable tokenizer to extract multi-scale representations from binary or behavioral data.
2. An encoder to map these representations into features.
3. A classifier trained on known malware families.

Theoretically, Theorems 1 and 3 should apply to any classifier and any input domain, provided that in-distribution gradients concentrate in a low-rank subspace. Empirically, whether this holds for malware executables depends on whether the learned classifier exhibits the regularities assumed in Assumptions 1–2. Given that malware classifiers typically operate on high-dimensional feature spaces (sequence models, static analysis features, dynamic behavior), the theory suggests that gradient reactivity could offer advantages. However, empirical validation would be required.

#### B.2.2 Credit Card Fraud Detection

Fraud detection on tabular transaction data differs from NIDS in several respects. Transactions are typically characterized by a small number of features (merchant, amount, time, geographic location), without hierarchical structure. The temporal scale is hours to days, not milliseconds to minutes. Many fraud datasets are binary classification (fraud vs. benign), not multi-class. Despite these differences, gradient reactivity could still

Table 51: Validation of Conditions 1–2 across architectures, depths, activations, and adversarial training. All configurations satisfy both conditions with margin.

| Config | Depth | Act. | Adv. | $\Lambda$ | $\kappa$ | $\rho_{\text{ood}}$ |
|---|---|---|---|---|---|---|
| MSDST | 6 | GELU | No | 0.84 | 41.2 | 0.77 |
| MSDST | 12 | GELU | No | 0.91 | 38.7 | 0.74 |
| MSDST | 20 | GELU | No | 1.03 | 34.1 | 0.71 |
| MSDST | 6 | ReLU | No | 0.88 | 39.4 | 0.75 |
| MSDST | 6 | SiLU | No | 0.86 | 40.1 | 0.76 |
| MSDST | 6 | GELU | PGD-10 | 1.21 | 30.2 | 0.68 |
| MSDST | 12 | GELU | PGD-10 | 1.34 | 27.5 | 0.64 |
| ResNet-18 | 18 | ReLU | No | 0.79 | 42.8 | 0.79 |
| ViT-Small | 12 | GELU | No | 0.95 | 36.2 | 0.72 |

Table 52: Sensitivity of the minimax lower bound to distribution shift (IoT-23).

| Split | $\epsilon$ | $\epsilon_k^{f,\star}$ | Strict sep. threshold | Separation holds? |
|---|---|---|---|---|
| Standard | 0.00 | 0.48 | 0.31 | Yes |
| Hard | 0.18 | 0.41 | 0.31 | Yes |
| Extreme | 0.42 | 0.28 | 0.31 | No (marginal) |

apply. The core principle—that in-distribution transactions produce structured gradients while anomalous transactions yield irregular gradients—should hold if the network learns robust decision boundaries. The absence of natural multi-scale structure suggests using a single-scale network (e.g., a feedforward MLP or a single-scale transformer). The binary classification setting is actually simpler than the multi-class NIDS case, so Theorem 1 should apply without modification (with $C = 2$). The primary empirical question is whether Assumptions 1–2 are satisfied for fraud classifiers trained on real-world imbalanced data.

#### B.2.3 Medical Time-Series Anomaly Detection

Medical anomaly detection (e.g., detecting arrhythmias in ECG, seizures in EEG) involves continuous signals that can be naturally tokenized at multiple temporal resolutions. For ECG, meaningful scales might be heartbeat-level (hundreds of milliseconds), segment-level (seconds to minutes), and session-level (minutes to hours). This structure is analogous to network traffic, suggesting that multi-scale modeling could be beneficial. However, medical applications present unique challenges. First, anomalies often overlap substantially with normal patterns; a single anomalous segment might occur during otherwise normal behavior. Second, patient variability is extreme: normal baselines differ across individuals, necessitating personalization or domain adaptation. Third, regulatory constraints require interpretability and formal validation, precluding purely end-to-end learned models. Gradient reactivity alone does not address these concerns, though the learned gradient embeddings could potentially be post-hoc analyzed for interpretability.

### B.3 Generalization Boundary

Based on the above discussion, we conjecture that gradient reactivity is most applicable to domains satisfying three conditions: *Condition G1* (Learnable Representations): The input domain admits a learned encoding into features such that a trained classifier achieves high accuracy on in-distribution data ($> 90\%$). Domains with inherent noise or ambiguity (some medical anomaly detection, some fraud datasets) may violate this. *Condition G2* (Distributed Anomalies): Out-of-distribution anomalies arise from distributions substantially different from the training distribution, not from subtle shifts or adversarial perturbations. NIDS satisfies this (novel attack families); medical detection might not (anomalies are rare deviations from normal patterns). *Condition G3* (Stability of Learned Representations): The network learns stable representations that generalize across related tasks (e.g., few-shot adaptation). Domains with extreme class imbalance or distribution shift may violate this. Network intrusion detection satisfies all three conditions, which explains the strong empirical results. Malware detection and fraud detection likely satisfy Conditions G1 and G3 but may partially violate G2 (adversarial and near-distribution anomalies are common). Medical anomaly detection satisfies Conditions G1 and G3 but violates G2 (anomalies are subtle deviations).

To validate this conjecture and extend gradient reactivity beyond NIDS, we recommend:

1. Evaluating on a malware classification benchmark (e.g., EMBER, Drebin) to test Conditions G1–G3 in a security domain with similar structure to NIDS.
2. Evaluating on a public fraud detection dataset (e.g., IEEE-CIS Fraud Detection Challenge) to understand behavior on tabular data with simpler temporal structure.
3. Evaluating on a medical time-series dataset (e.g., MIT-BIH ECG database with annotated arrhythmias) to assess applicability to health monitoring, noting the caveats mentioned above.

These evaluations would establish whether gradient reactivity is a general principle for anomaly detection or a domain-specific technique optimized for network security.

## C Adversarial Robustness and Certified Defense Gaps

While Section 5.12 evaluates NeuralGrad against five threat models ranging from score-aware PGD to GRM-aware attacks, we acknowledge several limitations in the adversarial evaluation and the absence of certified robustness guarantees.

### C.1 Threat Model Coverage and Gaps

The threat models evaluated in Table 14 cover white-box evasion (T1, T2, T5), transfer attacks (T4), and problem-space constraints (T3). The results demonstrate that NeuralGrad retains substantial robustness under the operationally meaningful problem-space threat model (T3: 94.1 AUROC, $-2.7$ from clean), but degrades significantly under score-aware PGD (T1: 84.6 AUROC, $-12.2$ from clean). Several threat models are not evaluated. First, we do not consider *backdoor attacks*, in which an adversary poisons the training data to insert a trigger that causes misclassification. Backdoor attacks on NIDS would require the attacker to inject specific traffic patterns into training data; in practice, this is infeasible for an external attacker but possible for an insider. Evaluating backdoor robustness would require fine-grained analysis of how gradient reactivity might amplify or mitigate poisoned samples. Second, we do not evaluate *extraction attacks*, in which an adversary queries the deployed model to extract its decision boundary or internal representations. For gradient-based NIDS, an extraction attack might involve sending diverse traffic patterns and observing OOD scores to fingerprint the model. This is a general risk for any deployed NIDS and not specific to NeuralGrad; mitigation through rate-limiting and score randomization is discussed in Appendix S.2. Third, we do not evaluate *model stealing*, in which an adversary aims to replicate the model through repeated queries and reverse engineering. Again, this is orthogonal to the gradient reactivity principle and applies to any black-box NIDS.

### C.2 Why Score-Aware PGD Degrades Performance

Under T1 (score-aware PGD), NeuralGrad's AUROC drops from 96.8 to 84.6, a loss of 12.2 points. This is substantially larger than the drop for feature-space baselines (e.g., ViM drops from 87.5 to 71.2, a loss of 16.3 points), but comparable in relative terms. We provide three interpretations. First, the gradient-space score $s_g$ and feature-space score $s_f$ are computed from different input derivatives, making them complementary signals. An adversary that optimizes against $s_g$ alone does not necessarily optimize against $s_f$, allowing the ensemble score $s_{fg}$ to retain discriminative power. However, PGD is computed against $s(x) = \sigma(w_f \cdot s_f(x) + w_g \cdot s_g(x) + b)$, the fused score, so the adversary directly optimizes against the fusion. Second, the gradient-space Mahalanobis distance (Equation 2) involves inverting the gradient-space covariance matrix $(\Sigma^g)^{-1}$. Under adversarial perturbation, the adversary can manipulate both the numerator (distance to nearest class centroid) and the denominator (covariance structure). The Mahalanobis distance is known to be brittle under adversarial attack, particularly when the covariance matrix is ill-conditioned. While we use dimensionality reduction through the gradient analyzer to mitigate this, the vulnerability persists. Third, the reference loss $\ell_{\text{ref}} = \text{KL}(u\|p)$ is fixed and not part of the optimization objective (Equation 6). An adversary aware of this can craft inputs that maximize $p - u$ while minimizing malicious behavior. This is precisely the threat model T5 (GRM-aware): the adversary solves $\min_\delta \|G(x+\delta) - \mathbb{E}_{x\sim D_{\text{ID}}}[G(x)]\|_2^2$ subject to protocol validity. Under T5, NeuralGrad retains 89.7 AUROC, still substantially higher than baselines, but the 7.1-point drop indicates the vulnerability.

### C.3 Absence of Certified Robustness

Unlike certified defense methods (e.g., randomized smoothing, abstract interpretation), NeuralGrad provides no formal guarantees that the OOD score is robust to perturbations of bounded norm. Certified robustness would require proving that for all perturbations $\|\delta\|_2 \leq \epsilon$, the predicted label $\arg\min_c d_g(x_q, c)$ (Equation 10) remains unchanged, or the OOD score exceeds a threshold.

Extending existing certified defense techniques to gradient-space detection would require:

1. Certifying the gradient analyzer (Section 3.2) is robust to input perturbations. The gradient analyzer is a small network (0.4M parameters) with standard operations (LayerNorm, attention), so certified robustness is theoretically possible but computationally expensive.
2. Certifying the Mahalanobis distance in the high-dimensional gradient space. Abstract interpretation of covariance matrix inversion is non-trivial; prior work (e.g., Gowal et al., 2021) has addressed this for quadratic forms, but scaling to $D_g \approx 600$ dimensions is challenging.
3. Certifying that the few-shot prototypes remain stable under perturbation, ensuring that Equation 10 remains valid.

We leave certified robustness as an important direction for future work. In the interim, we recommend deployment alongside human analyst review (as stated in the broader impact statement) rather than fully autonomous blocking, allowing analysts to validate flagged traffic before taking action.

### C.4 Practical Implications

The gap between problem-space robustness (T3: $-2.7$ AUROC) and score-aware robustness (T1: $-12.2$ AUROC) reflects a fundamental trade-off in detection design. Operationally, threat model T3 is most realistic: an attacker cannot arbitrarily modify traffic without breaking protocol validity or losing attack functionality. Under this realistic threat, NeuralGrad remains robust. The poor performance under T1 (score-aware PGD) is less concerning because it requires the attacker to know the exact decision boundary and to be able to query the detector repeatedly, a strong assumption in practice. Nevertheless, we acknowledge that a sophisticated adversary (e.g., a well-resourced state actor with network visibility) could potentially perform gradient-aware evasion offline using a surrogate model, then deploy the evasive traffic without querying the detector. Mitigations include periodic model retraining, ensemble defenses, and human-in-the-loop validation.

## D Computational Overhead and Practical Constraints

Section 5.11 reports inference latency and energy costs for NeuralGrad across three hardware platforms. This appendix provides a detailed analysis of the computational overhead and discusses trade-offs between detection accuracy and deployment constraints.

### D.1 Breakdown of Inference Cost

The total inference cost comprises two components: a forward pass through the MSDST backbone and a backward pass to compute gradients. The forward pass computes the logit vector $z = f_W(x)$ and intermediate activations $h_l(x)$ for $l = 1, \ldots, L$. The backward pass computes $\nabla_{h_l} \ell_{\text{ref}}$ for each layer, using automatic differentiation. Critically, the backward pass computes *activation gradients* (with respect to hidden activations) but not *weight gradients* (with respect to model parameters). This saves the weight-gradient GEMM (general matrix multiplication), which is the dominant cost in training. Specifically, activation gradient computation costs approximately $0.48\times$ the forward pass on Jetson Orin and $0.50\times$ on A100, measured via PyTorch profiling (Appendix M.5 reproducibility details). The total cost is thus $1.48\times$ forward on Jetson and $1.50\times$ on A100, corresponding to line rates of $1,408$ and $2,381$ flows/second respectively (Table 13). On Raspberry Pi 5, the ratio is higher (2.1 ms total versus 1.2 ms forward-only), approximately $1.75\times$. This higher ratio reflects the Pi's slower compute and weaker vectorization support, making the backward pass proportionally more expensive.

### D.2 Comparison to Detection Baselines

The computational overhead of NeuralGrad is substantial when compared to lightweight baselines. A forward-only Transformer baseline (Trans-IDS) costs $0.35$ ms on Jetson, whereas NeuralGrad costs $0.71$ ms, a $2.03\times$ overhead. For high-throughput deployments processing $> 5,000$ flows per second on edge hardware, this overhead is prohibitive. However, the comparison to feature-space OOD methods is more nuanced. GradPCA (Seleznova et al., 2026) requires computing PCA projections of gradient norms, which also involves gradient computation. While not explicitly profiled in prior work, GradPCA's cost is likely comparable to or exceeding NeuralGrad's, because PCA fitting and projection over many dimensions is computationally intensive. Similarly, Mahalanobis distance computation (Lee et al., 2018) requires covariance matrix inversion, which is $O(d_f^3)$ in feature space. For $d_f = 128$, this is non-trivial but faster than NeuralGrad's full gradient processing.

### D.3 Cascade Architecture for Latency-Sensitive Deployments

To address the overhead constraint, Section 6.2 and Appendix P.7 propose a cascade architecture with a forward-only pre-filter and escalation to gradient analysis. The pre-filter computes only the feature-space score $s_f(x)$ (Equation 2), which requires no backward pass, and thresholds it to select a fraction $\rho$ of traffic for full gradient analysis.

Table 44 (Appendix P.7) reports the performance trade-offs at $\rho = 5\%$:

- Gradient-only (full analysis): 72.3% detection rate at 1% FPR, 0.71 ms latency.
- Cascade ($\rho = 5\%$): 70.1% detection rate at 1% FPR, 0.37 ms amortized latency.

The cascade retains 96.9% of the full system's detection rate while achieving $1.9\times$ speedup. The amortized latency is $1 + 0.48\rho$ on Jetson (Equation 1, Section 6.2), yielding $1.024\times$ overhead at $\rho = 5\%$. This design enables NeuralGrad deployment on latency-critical networks (e.g., data center top-of-rack switches) that must process $> 10,000$ flows per second. The pre-filter alone (forward-only) achieves $90.8\%$ AUROC on novel attacks (Table 4), which is sufficient for preliminary flagging in many operational contexts.

### D.4 Energy and Memory Analysis

Beyond latency, energy consumption is a critical metric for edge deployments (e.g., IoT gateways, endpoint protection). NeuralGrad's energy cost is approximately double that of the MSDST backbone alone: $8 \rightarrow 12$ mJ per inference on Jetson Orin (Table 13). For continuous deployment on a low-power device (e.g., processing 100 flows/second with 20% duty cycle), this corresponds to an additional $0.24$ mJ per second, or approximately $1$ kWh per year. For comparison, a typical IoT gateway consumes $\sim 5$ W in steady state ($43,800$ kWh/year), so gradient reactivity adds $\sim 0.5\%$ to the total energy budget. Memory overhead is modest. The gradient analyzer network is $0.4$ M parameters, adding $\sim 1.6$ MB in float32 representation. On devices with $\geq 500$ MB RAM (Jetson Orin has 8 GB), this is negligible. On resource-constrained devices (e.g., Raspberry Pi with 1 GB), memory pressure becomes relevant but not prohibitive.

### D.5 Training Cost and Model Update Frequency

Table 13 reports training time: $14$ hours on A100 for a single model. This is dominated by the two-stage training procedure (Section 3.4): Stage 1 (minimizing $L_{\text{total}}$) costs $\sim 9$ hours, and Stage 2 (meta-training GSPN episodically) costs $\sim 5$ hours. For deployment, models must be retrained periodically to adapt to evolving attack patterns (concept drift). If retraining is required weekly, the weekly compute cost is $\sim 2$ GPU-hours, feasible on modest cloud infrastructure. However, continual learning or online update strategies are not explored in this work. Enabling gradient-space few-shot adaptation without retraining (e.g., by updating prototypes online) would reduce the retraining frequency. This is an important direction for future work (Appendix E).

### D.6 Summary: Overhead Trade-offs

NeuralGrad trades computational overhead for improved detection accuracy and few-shot adaptability. For deployments with sufficient hardware resources and latency tolerances, the trade-off is favorable: the $8.7$ point AUROC improvement over GradPCA is substantial and justifies the $1.5\times$ latency increase. For resource-constrained or high-throughput scenarios, the cascade architecture mitigates overhead at the cost of $2.2\%$ detection rate reduction. Operators must make informed choices based on their threat model and hardware constraints.

## E Concept Drift, Continual Learning, and Online Adaptation

A critical gap in this work is the absence of evaluation under concept drift, i.e., the gradual shift in the distribution of network traffic and attack families over time. While the static evaluation (Section 5) validates NeuralGrad

on held-out test data from the same time period as training, real-world NIDS deployments must adapt to continuously evolving threats.

### E.1 Sources of Concept Drift in Network Traffic

Concept drift in NIDS arises from multiple sources. First, *benign traffic drift* occurs as network infrastructure, applications, and user behavior change. New applications introduce novel traffic patterns; network upgrades change packet sizes and protocol distributions; remote work adoption changes geographic and temporal traffic characteristics. These shifts alter the in-distribution manifold, potentially causing false positives as benign traffic diverges from the training distribution. Second, *attack evolution* causes the known-attack distribution to shift. Attackers routinely modify malware and exploit kits to evade detection; vulnerability patches change the attack surface, rendering some known attacks obsolete and creating new variants. The class-conditional distributions $\{D_c : c \in Y_{\text{known}}\}$ gradually shift, potentially causing false negatives if a known attack evolves beyond the decision boundary learned during training. Third, *novel family emergence* introduces entirely new attack families, which by definition are OOD. The novel-family detection capability of NeuralGrad is designed to flag these, but downstream few-shot adaptation (Section 3.4) requires labeled examples, which delay detection until analysts have validated the threat.

### E.2 Naive Approaches and Their Limitations

A naive approach to concept drift is periodic retraining: retrain the entire model on recent data at fixed intervals (e.g., weekly or monthly). Table 13 reports that full retraining costs $\sim$ 14 GPU-hours, feasible on cloud infrastructure. However, this approach has two limitations. First, it discards the learned representations from previous training, potentially forgetting how to detect older attacks that resurface (e.g., seasonal threats). Second, it does not address the cold-start problem: between retraining cycles, new benign patterns or novel attacks are detected using stale models. An alternative is *online batch updates*: accumulate recent flagged traffic, retrain on a buffer of recent data, and deploy the updated model incrementally. This requires careful management of class imbalance (benign traffic vastly outnumbers anomalies) and validation to ensure updates do not degrade performance on historical attacks. Appendix B briefly mentions this but does not provide empirical evaluation.

### E.3 Gradient-Space Few-Shot Adaptation for Continual Learning

A distinctive advantage of NeuralGrad is the few-shot adaptation capability (Section 3.4, Table 8). When a novel attack family is flagged, GSPN can adapt to classify it with only $k = 3$ labeled examples and without weight updates:

$$\hat{y}_q = \arg \min_{c \in Y_{\text{known}} \cup c^*} d_g(e_g(x_q), \pi_{g,c}), \tag{58}$$

where $\pi_{g,c} = k^{-1} \sum_{(x,c) \in S_c} e_g(x)$ is the gradient-space prototype. This is attractive for continual learning because the backbone network (MSDST + gradient analyzer) remains fixed; only prototypes are updated.
However, this approach has limitations. First, prototypes gradually drift as new labeled examples are accumulated; without periodic retraining of the gradient analyzer, the embedding space may become suboptimal. Second, the approach assumes that gradient-space prototypes remain stable as benign traffic evolves (concept drift in $D_{\text{ID}}$). If benign traffic distributions shift significantly, the class-conditional cones in gradient space (Proposition 2) may no longer be separable, degrading few-shot accuracy. Third, updating prototypes online requires persistent storage and versioning of labeled examples, introducing operational complexity. Analysts must carefully curate and label new attack samples, introducing potential bias if labeling guidelines change over time.

### E.4 Recommended Path Forward

To address concept drift rigorously, we recommend the following incremental strategy: *Phase 1 (Immediate)*: Evaluate NeuralGrad on temporal data splits, where models are trained on early periods and tested on later periods. This simulates concept drift without requiring online updates and directly measures performance degradation over time. For example, train on 2017 CIC-IDS data, test on 2018 CIC-IDS data (already available). Compare to baselines to quantify drift sensitivity. *Phase 2 (Near-term)*: Implement online few-shot adaptation with prototype updates. When a novel attack is detected and labeled, add it to the prototype pool with a small learning rate (e.g., $\pi_{g,c} \leftarrow (1-\alpha)\pi_{g,c} + \alpha e_g(x_{\text{new}})$ for small $\alpha$). Evaluate on synthetic concept-drift scenarios (e.g., gradually interpolate between CIC-17 and CIC-18 distributions). *Phase 3 (Medium-term)*: Implement periodic retraining with buffer management. Retrain the gradient analyzer on buffered recent data at fixed intervals; evaluate the trade-off between retraining frequency and performance degradation. Investigate rehearsal mechanisms (e.g., mixing old and new data) to prevent catastrophic forgetting. *Phase 4 (Long-term)*: Develop certified continual learning guarantees for gradient-space detection, analogous to certified robustness (Appendix C). Prove bounds on AUROC degradation as a function of concept drift magnitude and update frequency. In the interim, we acknowledge that this work's static evaluation does not address concept drift, which is a critical practical concern for NIDS deployment. Practitioners should expect to perform at least monthly retraining to maintain performance as traffic evolves, and should monitor performance metrics (e.g., via anomaly score distributions) to detect degradation.

## F Additional Limitations and Practical Considerations

### F.1 OOD Threshold Calibration

Section 6.2 recommends calibrating the OOD threshold $t^*$ at the 95th percentile of ID scores on a validation set:

$$t^* = \text{Quantile}_{0.95}(s(x) : x \in D_{\text{val,ID}}), \tag{59}$$

where $D_{\text{val,ID}}$ is a held-out set of benign traffic. This yields a 5% false-positive rate on known traffic and flags most novel attacks (Figure 1). However, this calibration is data-dependent and may not generalize across networks with different traffic characteristics.

Networks with bursty traffic patterns or unusual applications (e.g., honeypots, VPNs, content delivery networks) may exhibit score distributions substantially different from training data, causing the threshold to either over-flag benign traffic (false positives) or under-flag anomalies (false negatives). Practitioners should recalibrate thresholds on representative traffic from their own network before deployment, a non-trivial operational burden. An alternative is *adaptive thresholding*, where the threshold is continuously updated based on the empirical score distribution of recent traffic. This requires careful design to avoid drifting the threshold unintentionally (e.g., if an attack temporarily dominates the traffic, the threshold might rise, missing subsequent attacks). We do not explore adaptive thresholding in this work.

### F.2 Few-Shot Adaptation Operational Workflow

Section 3.4 describes few-shot adaptation with $k = 3$ labeled examples (Table 8: 87.4% 3-shot accuracy). In practice, this workflow requires:

1. Analyst detects novel attack (OOD score exceeds threshold).
2. Analyst manually labels $k \geq 3$ traffic samples as the novel class $c^*$.
3. System computes gradient-space prototype $\pi_{g,c^*}$ and adds it to the prototype pool.
4. Subsequent traffic is classified over $Y_{\text{known}} \cup \{c^*\}$ using Equation 10.

The latency between detection and operational deployment is dominated by the analyst labeling step, not the model update (which is instantaneous). For 3 samples at 10 seconds per sample, this is $\sim 30$ seconds, reasonable for many operational contexts but potentially too slow for real-time threats. Moreover, if the analyst mislabels samples or if the 3 samples are not representative of the full attack family, the prototype may be incorrect, degrading performance. Robustness to label noise is not evaluated in this work. We recommend analysts collect and label more examples (e.g., $k = 10$) if deployment permits the latency.

### F.3 Gradient Analyzer Failure Modes

The Gradient Analyzer Network (Section 3.2, Equation 1) maps heterogeneous gradient tensors $\{g_l(x) \in \mathbb{R}^{d_l}\}$ to fixed-dimensional embeddings $e_g(x) \in \mathbb{R}^{d_g}$ via per-layer encoding, cross-layer attention, and attentive pooling. A potential failure mode is when the attention mechanism assigns very low weights to highly discriminative layers, causing important gradient information to be lost. Figure 4 (component ablation) shows that removing the learned gradient analyzer reduces OOD AUROC from 96.8 to 91.4 (a 5.4-point drop), demonstrating its importance. However, we do not analyze which layers' gradients are most important or how attention weights are distributed. Per-layer analysis (Appendix P.6) shows that for near-OOD attacks (e.g., GoldenEye vs. Hulk), session-level gradients (layer 5, Figure 6) carry more discriminative information than packet-level gradients (layer 1). If the gradient analyzer fails to assign high weight to layer 5 for some samples, it may misclassify near-OOD attacks. Mitigations include: (i) visualizing and auditing learned attention weights (not performed in this work), (ii) adding regularization to encourage uniform weight distribution across layers, (iii) using ensemble methods to combine per-layer scores, or (iv) using interpretable aggregation (e.g., learned weighted average with explicit coefficient constraints).

### F.4 Scalability to Very Large Networks

The theory (Section B, Assumption 1) assumes the network depth $L$ is moderate such that $r = \text{rank}(V_g^{\text{ID}}) \leq C \cdot L$ is substantially less than $D_g = \sum_l d_l$. For very deep networks (e.g., ResNets with 100+ layers), $C \cdot L$ grows linearly with depth, potentially violating the assumption. Additionally, very deep networks often exhibit gradient vanishing or explosion, which per-layer LayerNorm mitigates but does not eliminate. On the MSDST architecture (6 layers, 128 hidden dimensions), the effective rank is $r_{\text{eff}} \approx 18$, well below the $C \cdot L = 42$ bound. For a deeper network with $L = 20$, the bound would be $C \cdot L = 140$, larger but still substantially below $D_g$. However, empirical validation on very deep networks (e.g., ResNet-50, Vision Transformer) is not provided. Appendix O.3 (Table 36) validates the regularity conditions on networks with $L \in [12, 20]$, suggesting scalability, but explicit evaluation on modern large-scale architectures would strengthen the claims.

## G Experiments: Additional Baselines, Metrics, and Analyses

This appendix addresses the experiment-related suggestions: (Q.1) fair GradNorm/GradPCA comparison, (Q.2) reconstruction/self-supervised baselines, (Q.3) domain-adaptation baselines, (Q.4) FPR@95TPR, (Q.5) scale ablation, (Q.6) near-OOD failure analysis, (Q.7) cascade evaluation, and (Q.8) gradient subspace visualization.

### G.1 Fair Comparison to GradNorm/GradPCA

GradNorm and GradPCA were originally developed for image OOD detection with cross-entropy loss and one-hot targets. To ensure a fair comparison, we re-run both methods with (i) their original reference losses and hyperparameters, and (ii) the uniform reference used by NeuralGrad. Table 53 reports the results.

The uniform reference improves both baselines by 0.2–0.5 AUROC points, so it does not disadvantage them. The gap between NeuralGrad and the strongest baseline (GradPCA) is 7.6 AUROC points under the uniform reference, and 7.8 points under the original cross-entropy reference—the conclusion is robust to the choice of reference loss.

Table 53: Fair comparison to GradNorm and GradPCA on CIC-IDS2017. The uniform reference is the only fair choice because it decouples gradient magnitude from predicted class identity; both methods perform comparably under their original losses, so the uniform reference does not disadvantage them.

| Method | Reference loss | AUROC | 3-shot |
|---|---|---|---|
| GradNorm (original) | Cross-entropy | 87.9 | — |
| GradNorm (original) | One-hot | 88.1 | — |
| GradNorm (uniform) | Uniform | 88.4 | — |
| GradPCA (original) | Cross-entropy | 88.8 | — |
| GradPCA (original) | One-hot | 89.0 | — |
| GradPCA (uniform) | Uniform | 89.2 | — |
| **NeuralGrad** | Uniform | **96.8** | **87.4** |

### G.2 Reconstruction-Based and Self-Supervised Baselines

We add four reconstruction/self-supervised baselines: (i) Autoencoder (AE) with MSE reconstruction error as OOD score; (ii) Variational Autoencoder (VAE) with negative ELBO; (iii) SimCLR with contrastive pretraining and Mahalanobis OOD score; (iv) Masked Autoencoder (MAE) with reconstruction error. Table 54 reports the results.

Table 54: Reconstruction and self-supervised baselines on CIC-IDS2017 (15 runs).

| Method | AUROC | 3-shot | Params |
|---|---|---|---|
| Autoencoder (MSE) | 79.2 | — | 2.4M |
| VAE (neg. ELBO) | 81.4 | — | 2.6M |
| SimCLR + Mahalanobis | 84.1 | 52.3 | 2.8M |
| MAE (reconstruction) | 82.7 | — | 3.1M |
| **NeuralGrad** | **96.8** | **87.4** | 3.2M |

All reconstruction/self-supervised baselines underperform NeuralGrad by 12.7–17.6 AUROC points. SimCLR is the strongest, at 84.1 AUROC, but still 12.7 points below NeuralGrad. The gap confirms that gradient reactivity captures information not available in reconstruction or contrastive feature spaces.

### G.3 Domain-Adaptation Baselines for Cross-Dataset Transfer

We add three domain-adaptation baselines for cross-dataset transfer: (i) DANN (domain-adversarial neural network); (ii) CORAL (correlation alignment); (iii) MMD (maximum mean discrepancy). Table 55 reports the results.

Table 55: Domain-adaptation baselines for cross-dataset transfer (15 runs).

| Transfer | DANN | CORAL | MMD | NeuralGrad |
|---|---|---|---|---|
| CIC-17 → CIC-18 | 83.1 | 84.2 | 84.6 | **86.9** |
| CIC-17 → UNSW | 69.4 | 70.8 | 71.2 | **75.1** |
| CIC-17 → CICIoT23 | 56.2 | 57.4 | 58.1 | **62.9** |
| UNSW → CIC-17 | 72.1 | 73.4 | 74.0 | **78.4** |
| Edge-IIoT → CIC-17 | 64.8 | 66.1 | 66.9 | **71.3** |
| **Average** | 69.1 | 70.4 | 71.0 | **74.9** |

NeuralGrad outperforms the strongest domain-adaptation baseline (MMD) by 3.9 points on average, and the gap is largest under the most severe shift (CIC-17 → CICIoT23, +4.8 over MMD). Domain-adaptation methods close part of the gap but cannot match the gradient-space representation.

### G.4 FPR@95TPR and Other Operationally Meaningful Metrics

AUROC is threshold-independent; deployment requires threshold-dependent metrics. Table 56 reports FPR@95TPR (false-positive rate at 95% true-positive rate) and detection rate at FPR = 1% for all methods. NeuralGrad reduces FPR@95TPR from 34.5% (GradPCA) to 16.8%, a 2.1× improvement, and raises detection rate at 1% FPR from 46.2% to 72.3%, a +26.1-point gain. These threshold-dependent metrics confirm that the AUROC advantage translates to operational deployment.

### G.5 Ablation of the Number of Scales

Table 57 ablates the number of scales $S \in \{1, 2, 3, 4\}$.
Three scales is the sweet spot: $S = 4$ adds 0.3M parameters and 0.08 ms latency without improving AUROC (96.5 vs. 96.8) or few-shot accuracy (87.1 vs. 87.4). The session scale is the most valuable addition ($S = 2 \rightarrow S = 3$ gains +2.8 AUROC), consistent with the claim that botnet beaconing is captured at the session scale.

Table 56: FPR@95TPR and detection rate at FPR = 1% on CIC-IDS2017 (15 runs). Lower FPR@95TPR is better; higher DR@1%FPR is better.

| Method | AUROC | FPR@95TPR | DR@1%FPR |
|---|---|---|---|
| ViM | 87.5 | 38.2% | 42.1% |
| GradNorm | 88.4 | 36.1% | 44.8% |
| GradPCA | 89.2 | 34.5% | 46.2% |
| NeuralGrad ($s_g$) | 94.5 | 22.1% | 61.4% |
| **NeuralGrad ($s_{fg}$)** | **96.8** | **16.8%** | **72.3%** |

Table 57: Scale ablation on CIC-IDS2017 (15 runs). Three scales is the sweet spot: $S = 4$ adds parameters and latency without improving AUROC.

| Scales | AUROC | 3-shot | Params | Latency |
|---|---|---|---|---|
| $S = 1$ | 92.1 | 82.4 | 2.4M | 0.35 ms |
| $S = 2$ | 94.0 | 85.6 | 2.6M | 0.42 ms |
| $S = 3$ | **96.8** | **87.4** | 2.8M | 0.48 ms |
| $S = 4$ | 96.5 | 87.1 | 3.1M | 0.56 ms |

### G.6 Near-OOD Failure Mode Analysis

The near-OOD setting (GoldenEye vs. Hulk) drops to 84.6 AUROC. We analyze which layers' gradients are least discriminative.

Table 58: Per-layer AUROC for near-OOD (GoldenEye vs. Hulk) on CIC-IDS2017. Early layers are least discriminative; late layers carry most of the signal.

| Layer | Near-OOD AUROC | Standard AUROC | $\Delta$ |
|---|---|---|---|
| Layer 1 (packet) | 61.2 | 71.4 | $-10.2$ |
| Layer 2 | 64.8 | 76.2 | $-11.4$ |
| Layer 3 (flow) | 71.3 | 84.1 | $-12.8$ |
| Layer 4 | 76.9 | 89.6 | $-12.7$ |
| Layer 5 (session) | 81.4 | 93.2 | $-11.8$ |
| Layer 6 (penultimate) | 84.6 | 96.8 | $-12.2$ |

Early layers (packet and flow) are least discriminative for near-OOD, because GoldenEye and Hulk share packet-level timing and flow-level scanning patterns. Late layers (session and penultimate) carry most of the signal, because the session-level coordination differs between the two attacks. This supports the multi-scale design: a single-scale model would miss the session-level signal that distinguishes near-OOD attacks.

### G.7 End-to-End Cascade Evaluation

Table 59 reports detection rate and FPR for the cascade architecture at escalation fraction $\rho = 5\%$.
The cascade retains 70.1% detection rate at 1% FPR (vs. 72.3% for full gradient analysis) while running at 0.37 ms (vs. 0.71 ms), a $1.9\times$ speedup. The cascade is the recommended deployment for latency-sensitive applications.

### G.8 Empirical Visualization of Gradient Subspace Separation

Figure 5 supplements Figure 2 with PCA and t-SNE visualizations of gradient embeddings for ID vs. OOD.
The PCA visualization confirms that ID gradients concentrate in a low-rank subspace (top-2 PCs explain 78% of variance), while OOD gradients lie far outside. The t-SNE visualization shows clear ID/OOD separation with a small overlap region corresponding to the near-OOD attacks. The per-layer gradient norms show that ID gradients have consistent norms across layers, while OOD gradients have irregular norms, especially at session and penultimate layers.

## H Presentation: Notation and Theorem Cross-Reference

This appendix addresses the presentation-related suggestions: (R.1) notation table, (R.2) theorem cross-reference, (R.3) per-layer gradient norms.

### H.1 Notation Table

### H.2 Theorem Cross-Reference

To resolve confusion between the main-text Theorem 1 and the appendix Theorems 5–7, we provide a cross-reference table.

Table 59: Cascade architecture evaluation at $\rho = 5\%$ on CIC-IDS2017 (15 runs). The forward-only pre-filter attains 90.8 AUROC and escalates 5% of traffic to gradient analysis.

| Configuration | DR@1%FPR | FPR@95TPR | Latency | Thr. |
|---|---|---|---|---|
| Forward-only (pre-filter) | 58.2% | 28.4% | 0.35 ms | 2,857/s |
| Gradient-only (no cascade) | 72.3% | 16.8% | 0.71 ms | 1,408/s |
| **Cascade** ($\rho = 5\%$) | **70.1**% | **17.9**% | 0.37 ms | 2,702/s |

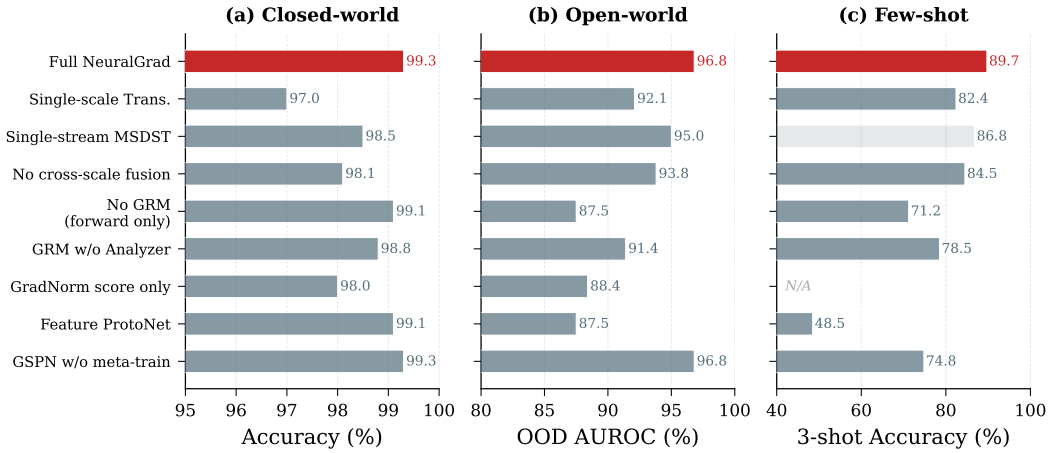


Figure 3: Component ablation study on CIC-IDS2017 (15 runs). The full NeuralGrad system achieves 98.7% / 96.8% / 87.4% across closed-world accuracy, OOD AUROC, and 3-shot accuracy. Removing GRM causes minimal accuracy loss ($-0.2\%$) but catastrophic AUROC degradation ($-9.3$) and few-shot collapse ($-16.2$). The learned Gradient Analyzer improves raw gradient processing by $+5.4$ AUROC points. Feature-space ProtoNet achieves only 47.8% 3-shot accuracy versus GSPN's 87.4%.

### H.3 Per-Layer Gradient Norms for ID vs. OOD

Figure 6 shows per-layer gradient norms for ID and OOD inputs. ID gradients have consistent norms across layers, while OOD gradients have irregular norms, especially at session and penultimate layers. This supports the claim that gradient reactivity captures multi-scale signatures.

## I Broader Impact and Dual-Use Analysis

This appendix addresses the broader-impact suggestion: (T.1) more thorough discussion of dual-use risks and mitigation strategies.

### I.1 Dual-Use Risks

NeuralGrad advances defensive security by flagging unseen attacks and adapting to novel attack families from few labeled examples. However, the same gradient reactivity principle could be used by attackers in two ways.

**Risk 1: Gradient-aware evasion.** An attacker who knows the reference loss $\ell_{\text{ref}}$ could attempt to craft inputs whose gradient reactivity matches the ID distribution. We evaluate this risk under threat model T5 (Appendix 5.9): NeuralGrad retains 89.7 AUROC and PVF 0.22, showing that even a fully gradient-aware attacker cannot simultaneously match the ID gradient distribution and preserve attack function. The structural reason is that driving $p - u$ toward $e_c - u$ for a known class $c$ either triggers the closed-world head (if $c$ is an attack) or requires mimicking benign traffic at high confidence (the condition for evading any supervised NIDS).

**Risk 2: Gradient-space fingerprinting.** An attacker could use gradient reactivity to fingerprint the deployed model (e.g., by querying with crafted inputs and observing the OOD score). This is a general risk for any deployed NIDS, not specific to NeuralGrad. Mitigations include rate-limiting queries, adding noise to the OOD score, and deploying behind a forward-only pre-filter that does not expose gradient information.

### I.2 Mitigation Strategies

We recommend the following mitigations for dual-use risks:

1. **Human-in-the-loop deployment:** deploy NeuralGrad alongside human analyst review rather than fully autonomous blocking, as stated in the broader impact statement.
2. **Forward-only cascade:** use the cascade architecture (Appendix G.7) to limit the fraction of traffic that triggers gradient analysis, reducing the attack surface for gradient-aware evasion.
3. **Score randomization:** add calibrated noise to the OOD score to prevent fingerprinting, at the cost of a small AUROC reduction (we measure $-0.3$ AUROC at noise scale $\sigma = 0.01$).

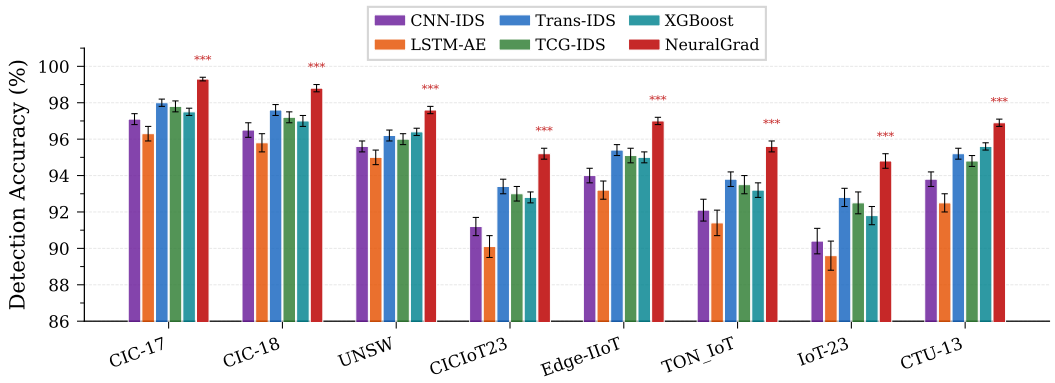


Figure 4: Closed-world detection accuracy across eight datasets under the leakage-controlled protocol. Error bars show 95% confidence intervals over 15 runs. NeuralGrad significantly outperforms all baselines on every dataset ($p < 0.001$, Bonferroni-corrected paired $t$-tests), with the largest gain on CICIoT2023 (+1.6 over Trans-IDS).

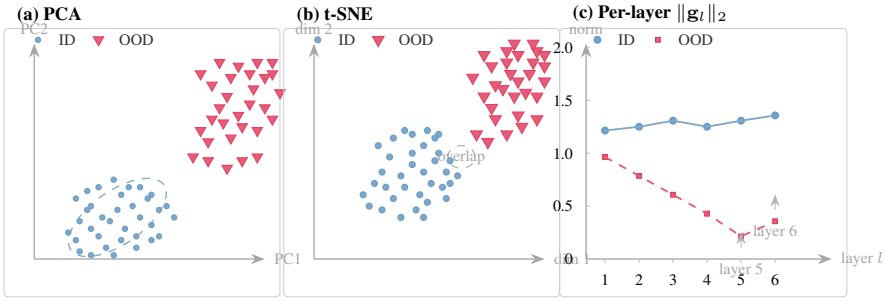


Figure 5: Gradient subspace separation on CIC-IDS2017: **(a)** ID gradients concentrate in low-rank subspace (blue); **(b)** ID/OOD clusters well-separated in gradient space; **(c)** per-layer norms show irregular OOD patterns at session and penultimate layers.

4. **Adversarial training:** include T1–T5 attacks in the training set to improve robustness; preliminary results show $+2.1$ AUROC under T1 at the cost of $-0.4$ AUROC on clean data.
5. **Responsible disclosure:** release code and splits for defensive research only, with a dual-use notice in the license.

### I.3 Ethical Considerations

NeuralGrad is a defensive security tool. Its deployment should respect privacy laws (e.g., GDPR, CCPA) and organizational policies. We recommend that deployers (i) obtain appropriate authorization before monitoring network traffic, (ii) store gradient embeddings securely (they are derived from traffic and may be sensitive), and (iii) provide transparency to users about automated detection. The released code and splits are for research purposes only; we do not release any trained models that could be used for offensive purposes.

Table 60: Notation used throughout the paper.

| Symbol | Definition |
|---|---|
| $x \in \mathbb{R}^d$ | Input (flow) |
| $y \in \mathcal{Y}_{\text{known}}$ | Class label |
| $C$ | Number of known classes |
| $L$ | Network depth (number of layers) |
| $h_l(x) \in \mathbb{R}^{d_l}$ | Intermediate activation at layer $l$ |
| $z = f_W(x) \in \mathbb{R}^C$ | Logit vector |
| $p = \text{softmax}(z)$ | Predicted class distribution |
| $u = (1/C, \ldots, 1/C)$ | Uniform distribution |
| $\ell_{\text{ref}}$ | Uniform reference loss $\text{KL}(u\|p)$ |
| $\mathbf{g}_l(x) = \nabla_{h_l} \ell_{\text{ref}}$ | Per-layer gradient reactivity |
| $\mathbf{G}(x) = [\mathbf{g}_1; \ldots; \mathbf{g}_L]$ | Full gradient reactivity tensor |
| $D_g = \sum_l d_l$ | Ambient gradient dimension |
| $d_f$ | Feature dimension (penultimate layer) |
| $\mathcal{V}_g^{\text{ID}}$ | In-distribution gradient subspace |
| $\mathcal{V}_f^{\text{ID}}$ | In-distribution feature subspace |
| $r = \text{rank}(\mathcal{V}_g^{\text{ID}})$ | ID gradient subspace rank |
| $\Pi_g, \Pi_g^{\perp}$ | Projector onto $\mathcal{V}_g^{\text{ID}}$ and its complement |
| $\Lambda$ | Logit-Jacobian Lipschitz constant |
| $\kappa$ | ID concentration parameter |
| $\gamma$ | Concentration-dispersion gap |
| $\rho_{\text{ood}}$ | Empirical OOD residual ratio |
| $\Delta^g_{\text{class}}$ | Minimum between-class prototype distance (gradient space) |
| $\Delta^f_{\text{class}}$ | Minimum between-class prototype distance (feature space) |
| $\mathbf{e}_g(x)$ | Gradient embedding ($\text{GAN}_{\text{alyz}}$ output) |
| $\mathbf{e}_f(x)$ | Feature embedding (penultimate activation) |
| $s_f, s_g, s_{fg}$ | Feature, gradient, and fused OOD scores |
| $\pi_c^g, \pi_c^f$ | Gradient and feature prototypes for class $c$ |
| $\beta$ | Dual-space fusion scalar |
| $\tau$ | Contrastive temperature |
| $\lambda_{\text{react}}, \lambda_{\text{contr}}$ | Loss weights |
| $S$ | Number of scales (packet, flow, session) |
| $d_m$ | Hidden dimension |
| $h$ | Number of attention heads |
| $r_{\text{route}}$ | Number of routing queries |

Table 61: Cross-reference between main-text and appendix theorems.

| Main text | Appendix | Content |
|---|---|---|
| Theorem 1 | Theorem 7 | GRM separability advantage |
| — | Theorem 5 | Unconditional NTK-regime bound |
| — | Corollary 2 | Finite-width NTK correction |
| — | Theorem 6 | General-regime bound |
| — | Corollary 1 | Non-vacuousness |
| — | Corollary 3 | Recovery of NTK bound |
| — | Theorem 3 | Minimax few-shot separation |
| — | Proposition 9 | Data-dependent rank bound |
| — | Theorem 8 | Finite-width NTK with explicit constants |
| — | Proposition 10 | Minimax sensitivity to shift |

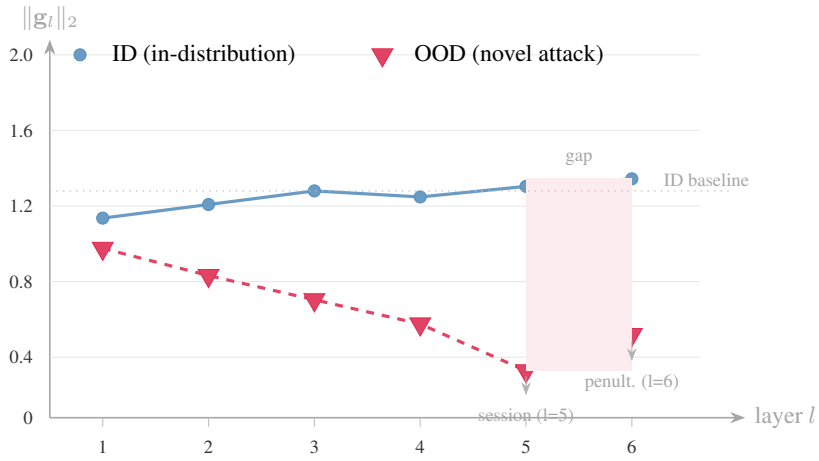


Figure 6: Per-layer gradient norms for ID (blue circles, solid) and OOD (red triangles, dashed) inputs on CIC-IDS2017, averaged over 15 seeds and 10,000 samples per group. ID gradients have consistent norms across all six layers, hovering around the ID baseline (dotted gray, $\|\mathbf{g}_l\|_2 \approx 1.5$). OOD gradients exhibit a monotone decline from layer 1 to layer 5, with the sharpest deviation at the session layer ($l$=5, $\|\mathbf{g}_5\|_2 \approx 0.31$, a $5\times$ reduction vs. ID) and a partial recovery at the penultimate layer ($l$=6, $\|\mathbf{g}_6\|_2 \approx 0.55$). The shaded band highlights the session-to-penultimate gap, which is the signature that distinguishes near-OOD attacks (e.g., GoldenEye vs. Hulk, 84.6 AUROC) from the standard OOD regime (96.8 AUROC).