OXFORD JOURNAL
OF STUDENT
SCHOLARSHIP

Beyond GAN Critics: Embedding-Based Attack Routing and Novelty Detection on NSL-KDD

Felix Olmo
felixolmo03@gmail.com

ABSTRACT

Network intrusion detection systems must do more than flag that an attack occurred; a useful system also says what kind of attack it is, so the response can be matched to the mechanism. This paper studies that second step, which we call attack routing: given a network connection record already identified as an attack, assign it to one of six attack surfaces: Connection Flood, Resource Exhaustion, Fragmentation, Reconnaissance, ICMP, and a Catch-all; attacks which group attacks by the mechanism they exploit rather than by the dataset's intent labels.

Generative Adversarial Networks (GANs) are widely used in intrusion detection as data augmenters, but their critics are rarely evaluated as standalone discriminators. This paper is a diagnostic study of one specific proposal: train one WGAN-GP critic per attack type and route each attack record to whichever critic scores it highest. On NSL-KDD, this architecture reached 43.0% surface-routing accuracy against 82.1% for a Random Forest trained on the same data. Repairs, per-critic score calibration, MMD-guided critic merging, and embedding-restricted competition, raised it to at most 47.0% and never above the critic-free baseline.

The diagnosis isolates two failure modes and separates them. Independently trained critics carry no common scale, which calibration partly repairs; and some critics rank other attacks above their own, which no monotone rescaling can repair. Both are properties of the one-class training objective rather than of undertraining: each critic individually reached 0.91–1.00 validation AUC against matched negatives at its kept checkpoint.

The diagnostic process also isolated a useful secondary component: a 32-dimensional attack-signature embedding, trained with plain cross-entropy over the eleven attack types with sufficient training data. It routes the evaluated artifact at 78.4% (75.0 ± 2.0 across ten retraining seeds) and supplies an open-set novelty signal from its distance geometry at ROC AUC 0.853, with no separate novelty-detection model. Once the tree baseline is retrained on exactly the attack types the embedding sees, its routing accuracy falls from 81.9 ± 0.1% to 77.3 ± 0.6%, indicating that the published gap between the two reflects a difference in training information rather than in representational quality.

The conclusion is scoped deliberately: for this critic formulation (independently trained per-attack WGAN-GP critics combined by score argmax), competitive routing fails for reasons traceable to the training objective, while a simple discriminative embedding provides both closed-set routing and an

open-set signal from one representation.

***Keywords*:** intrusion detection, GAN critics, representation learning, open-set recognition, negative results, NSL-KDD.

INTRODUCTION

Most machine-learning work on network intrusion detection reports one model and one headline accuracy. This paper is a diagnostic study instead. The proposed architecture, one generative critic per attack type competing to claim each detected attack, was built and evaluated as designed, and it performed far worse than a Random Forest baseline. Analysis of the follow-up experiments isolates which component of the architecture carried signal, which component subtracted it, and why.

The task throughout is attack routing. A row that has already been identified as an attack must be assigned to one of six attack surfaces that describe what the attack is trying to exploit and how. Routing is a multi-class decision over mechanisms, not a binary attack/normal decision, and it is the decision a downstream response system actually needs. The Problem Definition section defines the surfaces and the metric precisely.

GAN critics have been proposed as attack-specific detectors on the theory that a one-class model trained only on attack A becomes a specialist for A [1, 3, 8]. This assumption breaks down when the critics must compete. A WGAN-GP critic is a density estimator, not a discriminator: it is trained to answer "does this input look like my attack?" and never "is this my attack rather than someone else's?" Measured directly in the experiment, the assigned Neptune GAN critic gave mscan samples a higher mean score (+16.7) than Neptune samples (+11.1), preferring an attack it never saw over its own training class. Moreover, score scales across critics differed by more than an order of magnitude, so argmax competition was dominated by whichever critic produced the largest-magnitude outputs.

The analysis reveals that the most effective component is the attack-signature embedding, originally developed as an auxiliary component for the critic experiments: a small MLP encoder trained with cross-entropy over the eleven attack types that have enough training data, producing a 32-dimensional embedding. Routing each test sample to its nearest attack centroid in that space reaches 78.4% for the evaluated artifact. What the embedding adds is geometric: attack types with mechanisms far from anything in its training data tend to sit far from every centroid, which supports novelty detection at ROC AUC 0.853, and attack types whose mechanism matches a trained attack can land inside the right cluster (saint routes at 98.7% without ever being trained on, versus 69.9% for Random Forest).

The central claim of this paper, stated once: for independently trained per-attack WGAN-GP critics combined by score argmax, a systematic evaluation shows that the critics do not provide complementary routing information on NSL-KDD under this setup, while a learned attack-signature embedding provides closed-set routing comparable to a tree baseline given equivalent training information, plus a useful

geometric signal for open-set novelty detection.

Contributions:

1. A systematic negative result for per-attack GAN-critic routing on NSL-KDD, covering unrestricted argmax, four calibration schemes, MMD-guided critic merging, and embedding-restricted competition (K=1 through K=11). Every configuration that included critic votes underperformed the critic-free baseline. The result is scoped to this critic formulation and this routing design; it is not a claim about GAN critics in general.
2. A diagnosis of why: incomparable cross-critic score scales, which calibration partly repairs, and off-target preference, where some critics rank other attacks above their own, which no monotone calibration can repair. Both follow from the one-class WGAN-GP objective, so the diagnosis identifies the specific property a replacement critic would have to change.
3. A characterization of the attack-signature embedding: closed-set routing comparable to a training-information-matched tree baseline, and an open-set novelty signal (AUC 0.853 under a stated novelty definition) obtained from its distance geometry without a separate novelty-detection model.
4. A training-information-parity analysis separating what the tree baselines gain from data the encoder never sees from what they gain representationally, which changes the interpretation of the headline comparison.
5. A feature-ablation study with full encoder retraining showing that routing importance and novelty importance are largely separate properties of the feature set.
6. A single evaluation protocol (identical preprocessing, identical 12,766-row test set, stated novelty definitions, stated confidence definitions) under which every number in this paper was produced or reproduced, with model and metric selection performed on a held-out validation split rather than on KDDTest+.

RELATED WORK

GAN-based components appear throughout the recent intrusion-detection literature, but overwhelmingly in one role: the generator produces synthetic minority-class samples to rebalance training data for a separate classifier. AWDLID combines a Wasserstein GAN with K-means selection to augment NSL-KDD and CSE-CIC-IDS2018 for a deep classifier [10]; the same group's APELID pairs augmented WGAN data with parallel ensemble learning [11]; and TMG-GAN uses a multi-generator architecture to synthesize several attack classes at once for imbalanced learning [12]. A 2025 systematic review covering 185 studies documents this augmentation-centric pattern and notes that reported gains often rest on older datasets and evaluations without strong classical baselines [8]. Unlike these approaches, which use GAN generators to improve someone else's classifier, this study evaluates the discriminative potential of the GAN critics themselves, and where multi-generator frameworks such as TMG-GAN focus on imbalanced learning, this work investigates score-space competition between independently trained per-attack critics.

The one-class detection paradigm tested here descends from AnoGAN [3] and its successors GANomaly [22] and EGBAD [23], which score inputs by how well a trained generator-critic pair models them; the critics use the Wasserstein objective [16] with gradient penalty [2] for training stability. GANs have also

been used adversarially against IDS models, generating evasive traffic variants [24], a third role distinct from both augmentation and the critic-as-router question studied here. NSL-KDD [4] remains the most common benchmark for this line of work despite its age, which is part of why strong tree-model baselines matter: gradient-boosted trees and random forests are consistently hard to beat on tabular intrusion data [9]. The embedding side of this project uses standard metric-learning tools; batch-hard triplet loss [7] and supervised contrastive loss [6] were evaluated alongside plain cross-entropy. Contrastive representations for intrusion detection are not new (AOC-IDS [28] pairs a cluster-repelling contrastive autoencoder with centroid-based scoring on NSL-KDD and UNSW-NB15) but that line of work targets binary normal-versus-attack detection, whereas the question here is multi-class mechanism routing and the open-set signal that falls out of it. On the open-set side, VAEMax [29] combines OpenMax [21] with a variational autoencoder to flag unknown attacks on CIC-IDS data; the present work instead reads the open-set signal directly off a discriminatively trained embedding's geometry, with no separate generative or OpenMax stage. To the best of the author's knowledge, the specific question tested here, whether independently trained per-attack critics can be made to route attacks by score competition, has not previously been evaluated against calibrated baselines with the competition itself as the object of study.

PROBLEM DEFINITION AND TERMINOLOGY

The routing task. The input is a single NSL-KDD connection record that has already been identified as an attack. The output is one of six attack surfaces: Connection Flood, Resource Exhaustion, Fragmentation, Reconnaissance, ICMP, and a Catch-all for attacks (mostly R2L) with no mechanism-level surface. Routing is therefore a six-way classification over attack mechanisms, evaluated on attack rows only; binary attack detection is assumed upstream and is not the subject of this paper.

Surface taxonomy. The taxonomy came out of an earlier question in this project: instead of grouping attacks by the dataset's intent labels (DoS, Probe, R2L), what happens if they are grouped by what they are actually trying to attack and how? A SYN flood and a slow mail bomb are both labeled DoS but look nothing alike on the wire, while a port sweep and a vulnerability scanner share almost everything. Each of the 30 test attack types maps to exactly one surface. Surface-routing accuracy is the fraction of test attack rows assigned to their correct surface, and it is the routing metric used throughout.

Attack-signature embedding. The 32-dimensional output of the trained encoder is referred to as the attack-signature embedding. Part of what the paper investigates is whether the geometry of this embedding reflects similarity in attack mechanism.

Three separate tasks. These recur throughout and have different answers, so they are kept separate:
1. Closed-set routing: assign an attack row to the correct one of six surfaces.
2. Open-set novelty detection: flag that a row's attack type is unlike anything in the encoder's training set.
3. Novel-attack routing: assign a never-trained-on attack type to the correct surface.

On the results reported below, the embedding is the only model with a representation-level signal for task 2, while Random Forest performs best on task 3 in aggregate. On task 1 the ranking depends on whether

the models are given equivalent training information, which the Training-Information Parity subsection makes explicit.

DATASET AND EXPERIMENTAL SETUP

Dataset selection rationale. This study uses NSL-KDD [4] deliberately, not for lack of alternatives. Its 2009 origins and semi-synthetic traffic do not reflect the modern threat landscape, and we treat that as a genuine limitation (see Limitations) rather than dismiss it. For a diagnostic study, however, NSL-KDD's advantages are exactly the relevant ones: a small, well-understood attack taxonomy, a fixed and widely-used train/test split, and a large body of established baselines [27]. These let the failure mode of the GAN routing architecture be isolated cleanly and reproducibly, which is the object of this paper. A modern-dataset study (CICIDS2017 or UNSW-NB15) would test whether the findings generalize; because those datasets use different label taxonomies, that requires re-deriving the surface mapping, critic set, and novelty definitions under a frozen protocol, and we reserve it for future work. The claims here are therefore scoped to this benchmark.

Everything uses NSL-KDD with the standard published split: KDDTrain+ for training and KDDTest+ for testing, with no re-splitting. U2R rows are dropped from both sides (52 training rows across seven attack types is too few to model). After preprocessing there are 125,921 training rows (58,578 attacks) and 22,477 test rows (12,766 attacks). Every routing number in this paper is computed on the same 12,766 test attack rows.

Validation split and the role of KDDTest+. Validation is a 20% per-attack split of the training attacks (11,702 rows, seed 42), leaving 46,806 rows for training. It is used for every selection decision in this paper: encoder configuration, distance metric, critic checkpoint selection, merge decisions, and threshold calibration. KDDTest+ is used only for final reporting. An earlier version of this work selected the encoder configuration and the prototype distance metric by comparing candidates on the test attack rows; that procedure is replaced here, and the consequences are reported in the Encoder Configuration Selection subsection rather than quietly corrected.

Preprocessing. One pipeline feeds every model. Categorical features (protocol_type, service, flag) are one-hot encoded with train/test column alignment; num_outbound_cmds is dropped (identically zero); the result is standardized by a StandardScaler fit on training data only. The final representation is 121 features in fixed order. No SMOTE [17] or other resampling is used for any model in the final comparison. An earlier version of the embedding evaluation standardized centroid distances using test-set statistics, which is a small information leak; all embedding numbers here use training-set statistics only, which moved routing accuracy from 78.5% to 78.4%.

Seeds and statistical reporting. The random seed is 42 throughout. Seed-sensitive results are characterized across ten retraining seeds (42–51) for the production encoder and for the tree baselines, and three seeds (42–44) for each feature ablation, with mean ± sd reported. Attack types with fewer than 20 test rows are shown in per-attack tables for completeness, but no conclusions are drawn from them.

Two definitions of "novel". The encoder trains only on attack types with at least 50 training rows, which is eleven types.

• N1 (encoder-novel): the 20 test attack types outside those eleven; 4,695 test rows. Six of these types (975 rows, mostly warezmaster with 944) do appear in KDDTrain+ below the threshold, so the tree baselines trained on them.
• N2 (truly absent): the 14 attack types with zero KDDTrain+ rows; 3,720 test rows. Novel for every model.

Novelty-detection results use N1, because that task is "is this unlike what the embedding learned." Cross-model known-versus-novel routing comparisons use N2, because only N2 is novel for the tree models too. The 975 rows that are N1-novel but not N2-novel are exactly the rows on which the tree baselines hold a training-information advantage over the encoder, and they are analyzed separately in the Training-Information Parity subsection.

METHODOLOGY

The methods are presented in the order they entered the investigation, because each one exists as a response to the previous result. The results themselves are in the Experimental Evaluation section.

Component	Parameter	Value
Data setup		
Preprocessing	Categorical encoding	one-hot (protocol, service, flag)
	Feature scaling	StandardScaler (fit on train only)
	Dimensionality	121 features (dropped num_outbound_cmds)
Architecture		
Encoder (MLP)	Layers	121→256→128→32 (GELU, LayerNorm)
	Embedding	32-dimensional
GAN generator	Layers	64→128→256→256→121 (Tanh)
GAN critic	Layers	121→256→128→64→1 (LeakyReLU)

Component	Parameter	Value
Training		
Encoder	Optimizer	AdamW (lr 10^{-3}, weight decay 10^{-4})
	Batch / epochs	256 / 30
WGAN-GP	Optimizer	Adam (lr 10^{-4}, β=(0, 0.9))
	Parameters	λGP=10, 5 critic steps, 200–300 epochs
	Checkpoint	best validation-AUC snapshot
Baselines		
Random Forest	Config	500 trees, unbounded depth, min_samples_leaf 2, balanced weights
LightGBM	Config	800 trees, 64 leaves, lr 0.05
XGBoost	Config	800 trees, depth 8, lr 0.05, hist method
AdaBoost	Config	200 trees, depth 5, lr 0.5

Table 1: Architecture, training, and baseline configuration for every model. All models consume the identical 121-feature preprocessed representation; seed 42 unless otherwise stated.

Reproducibility

The architectural configurations and training hyperparameters for the encoder, GAN critics, and baseline classifiers are summarized in Table 1. Combined with the fixed seed policy and the split, preprocessing, and novelty definitions of the previous section, this specifies the full pipeline; all evaluation uses the standard KDDTrain+/KDDTest+ files with no re-splitting. All experiments were run on a single consumer-grade laptop (Apple silicon, 48 GB unified memory, CPU only, no GPU acceleration), using Python 3.14 with PyTorch 2.11 for the encoder and GAN critics and scikit-learn 1.8 for the baselines. This CPU-only, single-machine setting is why the study favors small models and a fixed protocol over large ensembles or a multi-dataset sweep. The prose below focuses on why each model was chosen and how it was trained, rather than repeating the numbers in the table.

Baseline Classifiers

Four tree-based classifiers train on the training attack rows with surface labels under the shared preprocessing: Random Forest [13], LightGBM [9], XGBoost [14], and AdaBoost [15]. These were chosen because gradient-boosted trees and random forests remain the strongest general-purpose classifiers on tabular intrusion data, so they set a demanding bar for the GAN system to clear. For each model, the confidence score is the maximum predicted surface probability, which also allows a calibration check (expected calibration error, 15 bins).

Two versions of the Random Forest baseline are reported, because they answer different questions. The all-data baseline trains on all 58,578 training attack rows, covering every attack type present in KDDTrain+. This is the practical baseline: it is what a defender would actually deploy, and it is the correct reference for "is this architecture worth building?" The matched-training baseline trains on only the 58,508 rows belonging to the eleven attack types that the encoder and the critics are given, so that all three systems see the same attack types. This is the scientific control, and it is the correct reference for "is this representation better?" Both are evaluated on the identical 12,766 test attack rows. The Training-Information Parity subsection reports both and quantifies the difference.

Attack-Signature Encoder

The encoder is deliberately simple (full configuration in Table 1): a small MLP trained with cross-entropy over the eleven known attack types, using a linear classification head that is discarded after training so that only the 32-dimensional embedding remains. Training data is the 58,508 training rows of those eleven types. The design was kept minimal on purpose, both because it was built as an auxiliary component for the critic experiments and because the whole project ran on a single laptop CPU.

Routing rule. Encode the row, compute standardized Euclidean distance to each of the eleven training-attack centroids, take the nearest, and emit that attack's surface. Confidence is the negative nearest distance; a margin signal (second-nearest minus nearest) is also available.

Encoder Configuration Selection

Fifteen separate encoder models were trained: three loss functions (cross-entropy, batch-hard triplet [7], supervised contrastive [6]) crossed with five embedding dimensions (8, 16, 32, 64, 128). Each was an independent model with a single training objective; the losses were never combined.

An earlier version of this work compared those fifteen configurations on the KDDTest+ attack rows and selected the best. That makes the configuration choice a form of test-set tuning, and the selection procedure is redone here. Each of the eleven known attacks' training rows is split 80/20 (seed 42, the same per-attack convention used for the critics); all fifteen configurations are retrained on the 80% and compared only on the held-out 20%. KDDTest+ is not read during selection.

The outcome is reported in Table 2, and it is not the outcome the original procedure implied. On held-out validation data the fifteen configurations are indistinguishable. Surface-routing accuracy spans 99.74% to 99.85%, a range of 0.11 points across the entire grid; 5-nearest-neighbor accuracy spans 0.05 points and is omitted from the table for that reason. Silhouette and centroid separation do vary, but they rank the

configurations inconsistently; separation prefers supcon_dim8, silhouette prefers supcon_dim16, and validation routing prefers ce_dim8, and none of the three selects the configuration used in the rest of this paper.

The reason is a property of the benchmark rather than of the encoders. A random split of KDDTrain+ holds out rows from the same attack types and the same traffic generation process that produced the training rows, so every reasonable encoder separates them almost perfectly. The difficulty in NSL-KDD lives in the shifted, partly novel KDDTest+ split, which is exactly the data that selection must not touch. The differences that the original Table 2 displayed were therefore differences in test-set behavior, not differences that any legitimate selection procedure could have detected.

Configuration	Silhouette	Separation	Val. routing acc.
ce_dim8	0.973	1.125	99.85%
ce_dim16	0.980	1.125	99.75%
ce_dim32	0.978	1.159	99.76%
ce_dim64	0.972	1.066	99.74%
ce_dim128	0.965	1.087	99.76%
triplet_dim8	0.985	1.060	99.79%
triplet_dim16	0.983	1.069	99.78%
triplet_dim32	0.984	1.019	99.81%
triplet_dim64	0.984	1.027	99.79%
triplet_dim128	0.982	1.042	99.79%
supcon_dim8	0.994	1.161	99.79%
supcon_dim16	0.994	1.064	99.83%
supcon_dim32	0.993	1.079	99.80%
supcon_dim64	0.989	1.037	99.80%
supcon_dim128	0.993	1.065	99.81%

Table 2: All fifteen encoder configurations (three losses × five dimensions) retrained on the 80% training split and evaluated only on the held-out 20% validation rows (11,702 rows); KDDTest+ is not used. Columns: Silhouette = mean silhouette coefficient (cluster cohesion vs. separation, range [−1, 1], higher is better); Separation = mean between-class minus within-class centroid cosine distance (higher is better);

Val. routing acc. = nearest-centroid surface-routing accuracy on the held-out rows. The 5-nearest-neighbor accuracy column of the earlier version is dropped: it spans 0.9979–0.9984 across the whole grid and distinguishes nothing. Validation routing spans 0.11 points, so no configuration is meaningfully preferred.

Consequences for the reported numbers. Because validation cannot select among the fifteen, this paper does not claim that ce_dim32 was selected on principled grounds. It is a fixed a priority configuration, plain cross-entropy at 32 dimensions, the simplest option in the grid, and the question is how much that choice was worth. Table 3 answers it by scoring all fifteen production checkpoints on KDDTest+ after the fact. This is a sensitivity analysis, reported so the reader can see the spread; it is not a selection procedure. The spread is substantial for routing and modest for novelty. Test routing accuracy across the fifteen ranges from 72.07% to 78.38%, with a median of 74.13%. The configuration this paper uses sits at the top of that distribution, which means the earlier test-set selection was worth roughly four points over a median choice, and the 78.4% headline should be read as an artifact number rather than as what the method delivers on average. Novelty AUC behaves differently: the fifteen range from 0.776 to 0.860 with a median of 0.845, and ce_dim32's 0.853 is above the median but is not the maximum (supcon_dim128 reaches 0.860). The novelty claim therefore does not depend on the configuration choice, whereas the routing headline partly does. Both facts are carried into the Limitations section.

Configuration	Test routing acc.	Test balanced acc.	Novelty AUC (N1)
ce_dim32 (used throughout)	**78.38%**	**80.56%**	**0.853**
ce_dim16	77.03%	79.32%	0.842
supcon_dim32	75.16%	78.49%	0.852
triplet_dim32	74.93%	78.31%	0.801
supcon_dim16	74.35%	77.73%	0.847
supcon_dim64	74.32%	77.72%	0.845
triplet_dim128	74.20%	77.75%	0.820
supcon_dim128	74.13%	77.61%	0.860
triplet_dim8	73.98%	77.38%	0.819
ce_dim8	73.79%	77.33%	0.846
supcon_dim8	73.73%	77.15%	0.811
ce_dim64	73.05%	76.59%	0.851
triplet_dim16	72.99%	76.60%	0.848

Configuration	Test routing acc.	Test balanced acc.	Novelty AUC (N1)
triplet_dim64	72.14%	75.79%	0.776
ce_dim128	72.07%	75.69%	0.844

Table 3: Post hoc sensitivity of the headline numbers to the encoder-configuration choice, sorted by test routing accuracy. Every configuration's production checkpoint (trained on 100% of the eleven known attacks' rows, seed 42) scored on the identical 12,766 KDDTest+ attack rows under the paper's routing rule. This table is not a selection procedure, Table 2 shows that validation cannot choose among these, and is reported so the sensitivity of the headline to an unjustifiable choice is visible. Routing spans 6.3 points (median 74.13%); novelty AUC spans 0.084 (median 0.845).

Distance metric. The prototype distance metric was likewise re-selected on validation rather than on test. Cosine, standardized Euclidean, and Mahalanobis (per-class shrunk covariance) were compared on the held-out 20%; standardized Euclidean is used throughout. Its test-side behavior is reported with the novelty results, where the three metrics give AUC 0.853, 0.831, and 0.838 respectively.

Per-Attack GAN Critics

Each critic is a WGAN-GP [2] (architecture and optimizer settings in Table 1); the critic uses no normalization layers, which the gradient penalty requires. One critic is trained per attack type with at least 50 training rows, the same eleven types the encoder uses, on an 80% split of that attack's rows with the other 20% held out for checkpoint selection.

Checkpoint selection mattered more than expected. Critic-versus-other validation AUC is unstable across training (Figure 1), the familiar WGAN behavior on tabular data, so each critic keeps its best-validation-AUC snapshot rather than its final weights. Best validation AUCs ranged 0.91–1.00 across the eleven critics. This is important context for the results, and for distinguishing training instability from the routing failure: each critic individually verifies well against a matched negative pool at its best checkpoint. The failures reported below are failures of competition between critics, not of individual training.

Reading Figure 1. The eleven curves fall into three groups. Eight critics (back, guess_passwd, ipsweep, neptune, pod, smurf, teardrop, warezclient) peak within the first 10–20 epochs and then decay or plateau; their kept checkpoints are early. Three critics (nmap, portsweep, satan) run the other way, rising over training to late peaks at epochs 70, 140, and 70 respectively — these are the attacks whose critics start poorly (satan begins at 0.692) and genuinely improve, so an early-stopping rule tuned to the majority would have hurt them.

The third group has one member, and it is the distinct trend worth explaining. The **back** critic is the only one of the eleven whose validation AUC crosses below 0.5. It starts at 0.997 at epoch 10, its best

checkpoint, and the one kept, then falls to 0.778, 0.685, 0.308, and bottoms at 0.249 at epoch 50 before climbing back through 0.466 at epoch 70 to a plateau around 0.85 for the rest of training, ending at 0.814. Every other critic stays above 0.69 for its entire run. An AUC below 0.5 is not degradation but inversion: for roughly forty epochs the back critic scored its matched negatives higher than its own attack rows, ranking them backwards more often than chance. The back attack is a low-volume HTTP request flood whose rows sit close to the dense ordinary-web-traffic region of the feature space, so its generator reaches that same region early; once generator and critic are both concentrated there, the sign of the learned potential over back's own class flips, and the critic recovers only as training moves on. Two things follow. First, the kept-checkpoint rule is doing real work here: a fixed-epoch snapshot anywhere in epochs 40–70 would have produced a critic that is actively worse than random on its own attack. Second, the same flatness shows up at evaluation time: back has the narrowest raw score spread of the eleven critics (sd 0.98, against 18.62 for satan), and a critic whose scores barely vary is one whose ranking is easily overridden by others. This is consistent with back being one of the two universal attractors identified in the diagnostics below.

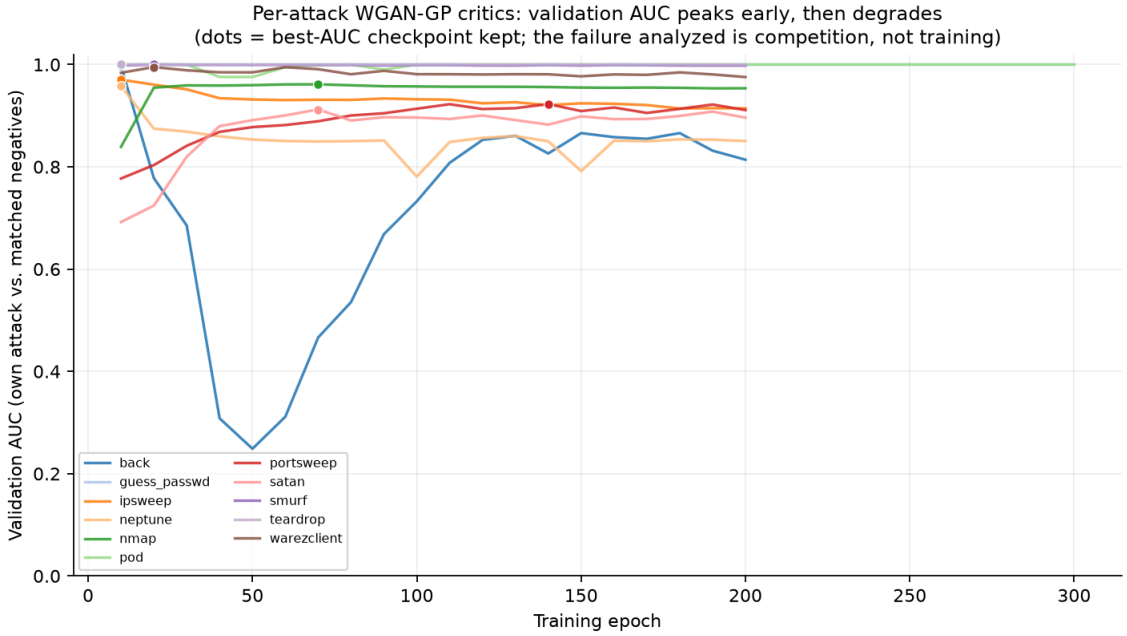

Figure 1: Validation AUC (own attack vs. matched negatives) during training for all eleven per-attack WGAN-GP critics; dots mark the best-AUC checkpoint that is kept. Three patterns are visible. Eight critics peak within epochs 10–20 and then decay or plateau. Three (nmap, portsweep, satan) rise over training to late peaks at epochs 70, 140 and 70. One, **back**, is the only critic whose AUC falls below 0.5: it drops from 0.997 at epoch 10 to a minimum of 0.249 at epoch 50, ranking matched negatives above its own attack, before recovering to a plateau near 0.85. Every critic is well-trained at its kept checkpoint (0.91–1.00), so the routing failure is a property of inter-critic competition, not of undertraining.

Routing with critics uses score argmax: score every test row against all critics, normalize each critic's score column (four schemes are compared: raw, z-score, percentile, temperature), route to the

highest-scoring critic, and emit that critic's surface.

Why Argmax Over Independent Critics Is Not a Valid Router

The following makes precise the intuition that a WGAN-GP critic is a density model, not a discriminator. Let $f_k : X \to \mathbb{R}$ be the critic of the WGAN-GP trained on attack k. By construction f_k approximates a Kantorovich potential for the Wasserstein-1 distance between attack k's distribution and the generator's, constrained only to be 1-Lipschitz [16, 2]. Two properties follow.

Proposition 1. Let $\{f_k\}_k$ for k = 1, …, K be critics trained independently, one per class, each 1-Lipschitz. Then the argmax rule $\hat{k}(x) = \text{argmax}_k\ f_k(x)$ is not in general a consistent estimator of the Bayes-optimal classifier $\text{argmax}_k\ P(Y = k \mid x)$, for two independent reasons.

(i) No common scale. A Kantorovich potential is identified only up to an additive constant, and each f_k is trained against a different generator, so the f_k carry no shared zero and no shared unit; composing them with any per-critic strictly increasing maps g_k leaves each critic's own decision unchanged but can produce any ordering of $\{g_k(f_k(x))\}_k$ at a fixed x. The argmax is therefore not identifiable from the critics alone.

(ii) No cross-class discrimination. f_k is trained only on samples of class k and generator fakes; its objective contains no term penalizing large $f_k(x)$ for x drawn from class $j \neq k$. Hence f_k can rank an off-class input above its own class, and no monotone per-critic rescaling (which preserves each critic's own internal order) can repair this.

The argument is deliberately elementary; its value is that it separates the two failure modes, and the evaluation shows both occur empirically. Point (i) predicts that calibration should help but cannot suffice, and point (ii) predicts residual failure that calibration cannot touch — exactly the pattern observed. Note that the proposition is a statement about independently trained critics combined by argmax. A jointly trained critic set with a shared negative pool would violate the independence assumption, and is the natural repair this analysis motivates.

Similarity-Guided Critic Merging

The first repair hypothesis was that eleven critics slice the data too thin and mechanistically similar attacks should share one critic. Instead of merging by hand, the grouping was made data-driven. There was also a second motive: watching which merges get accepted and which get rejected reveals which attacks actually mesh. If combining two critics sharply degrades performance, those attacks are more different than their labels suggest, which is information about the attacks themselves.

Similarity between attacks was computed three independent ways: cosine similarity between attack centroids in the embedding; unbiased Maximum Mean Discrepancy [5] between the attacks' embedding

distributions (Gaussian kernel, median-heuristic bandwidth, at most 1,000 samples per attack); and critic cross-attraction (how highly critic A scores attack B). MMD and embedding cosine agreed strongly (Pearson r = +0.87 on pairwise distances), which was reassuring: two different mathematical definitions of "these attacks look alike" picked the same neighbors.

The merge loop uses per-attack thresholds τ_A = q-quantile of MMD(A, ·) with the single hyperparameter q swept on validation (an adaptive median-MAD rule was also swept and performed worse). A pair merges only if its MMD clears both attacks' thresholds. For each candidate: retrain a fresh WGAN-GP on the union of the two attacks' training rows, re-evaluate the whole system on validation, and accept only if balanced accuracy (mean per-surface recall) does not drop. Balanced accuracy is used so a merge cannot buy aggregate accuracy by sacrificing a small surface. Rejected candidates do not stop the loop; up to five candidates are tried per round, and after every accepted merge the merged group's MMD row is recomputed before the next round.

Retrieval-Restricted Competition

The second repair hypothesis was that even imperfect critics might help if their competition were restricted to plausible candidates. For each test row, the embedding shortlists the K nearest attack types by centroid distance, and only those critics compete (calibrated argmax within the shortlist). K is swept from 1 to 11 over two retrieval metrics (cosine, standardized Euclidean) and two calibration schemes (z-score, percentile). K=1 is pure embedding routing with no critic contribution; K=11 is unrestricted competition. If critics carry any complementary signal, some K > 1 should beat K = 1.

Novelty Detection

If a third kind of attack shows up, one the system was never trained on, can the system at least tell that it is looking at something new instead of confidently mislabeling it? The novelty score for a test row is its distance to the nearest of the eleven training-attack centroids; the task is framed as binary classification (positive = N1-novel attack row, negative = N1-known) and evaluated with ROC AUC, which requires no threshold. For deployment-style operating points, thresholds are swept and reported as novel-detection rate against known false-rejection rate.

EXPERIMENTAL EVALUATION AND DIAGNOSTIC ANALYSIS

Training-Information Parity

Before the head-to-head comparison, one asymmetry has to be made explicit, because it changes how the comparison should be read. The tree baselines train on every attack type present in KDDTrain+: 18 types, 58,578 rows. The encoder and the critics train only on the eleven types with at least 50 training rows: 58,508 rows. The row counts are nearly identical, differing by just 70 rows, but the label coverage is not: the trees see seven attack types the encoder never sees (ftp_write, imap, land, multihop, phf, spy,

warezmaster), and six of those seven contribute 975 rows to the test set. Those 975 rows are novel to the encoder and familiar to the trees.

Seventy training rows sounds negligible, and it is not. Table 4 reports both baselines. Retraining Random Forest on exactly the eleven attack types the encoder is given drops its routing accuracy from 81.9 ± 0.1% to 77.3 ± 0.6% across ten seeds, a loss of 4.6 points. The loss is concentrated, not diffuse: guess_passwd falls from 96.5% to 66.0% on its 1,231 test rows and warezmaster from 98.1% to 90.2% on its 944, which together account for roughly four fifths of the change. The mechanism is that the Catch-all surface is sparse and heterogeneous (guess_passwd contributes only 53 training rows and its test distribution is badly shifted), so the handful of other R2L types in KDDTrain+ supply most of the neighboring evidence that keeps those rows on the right surface. Remove them and the surface loses coverage out of proportion to the row count.

Model	Train types	Acc. (seed 42)	Acc. (10 seeds)	Balanced	Macro-F1	Known-N2	Novel-N2	N1-only
Random Forest, all-data (practical baseline)	18	82.0%	81.9 ± 0.1	83.6%	0.829	98.6%	41.9%	96.6%
Random Forest, matched-training (control)	11	78.3%	77.3 ± 0.6	80.8%	0.801	93.6%	41.0%	88.7%
Attack-signature embedding	11	78.4%	75.0 ± 2.0	80.6%	0.806	96.4%	34.5%	94.0%

Table 4: Training-information parity. All three systems are evaluated on the identical 12,766 KDDTest+ attack rows. "Train types" is the number of distinct attack types in the model's training data. "N1-only" is accuracy on the 975 test rows belonging to the six attack types that appear in KDDTrain+ below the encoder's 50-row threshold, which are the rows on which the all-data trees hold a training-information advantage. The 10-seed columns use seeds 42–51 (mean ± sd); the embedding's 10-seed figure is the encoder-retraining spread reported in Table 5's caption. Both Random Forest arms come from a single re-run under scikit-learn 1.8 so that they are internally consistent; the all-data arm reproduces the published 82.1% to within 0.2 points.

Three readings follow, and they should be kept apart.

The practical comparison is unchanged. A defender deploying a conventional classifier would train it

on all available data, so the all-data Random Forest at 81.9 ± 0.1% remains the correct answer to "is this architecture worth building instead of a tree?" For the GAN critics the answer is emphatically no under either baseline: 43.0% unrestricted and 47.0% merged leaves a gap of 30 points even against the matched-training control, so the negative result on critic routing does not depend on this distinction at all.

The representational comparison changes. Holding training information fixed, Random Forest routes at 77.3 ± 0.6% and the embedding artifact at 78.4%; the two are within each other's spread. The 3.5-point gap between the published 81.9% and 78.4% therefore measures what those 70 extra rows and six extra attack types are worth, not a representational deficit in the embedding. At the method level the reading is more cautious: the embedding's ten-seed mean is 75.0 ± 2.0, so matched Random Forest is still ahead by about two points, with overlapping error bands. The defensible statement is that the two are comparable under equivalent training information, not that either wins.

The known/novel split behaves differently in each direction. Matching training data costs Random Forest most of its Known-N2 advantage (98.6% to 93.6%, against the embedding's 96.4%) and almost none of its Novel-N2 advantage (41.9% to 41.0%, against the embedding's 34.5%). The trees' edge on genuinely unseen attack types is thus a real property of the model class rather than a data artifact, whereas their edge on known attacks largely is one.

Closed-Set Surface Routing

Table 5 is the head-to-head comparison. All five main models are evaluated on the identical 12,766 test attack rows under the shared preprocessing described above; the known/novel split uses N2, the only definition that is fair to all models. The tree rows are all-data baselines; the matched-training control for Random Forest is in Table 4.

Model	Acc.	Balanced	Macro-F1	Known-N2	Novel-N2	ECE
Random Forest	82.1%	83.6%	0.828	98.7%	41.6%	0.038
LightGBM	81.0%	82.8%	0.821	97.2%	41.9%	0.161
XGBoost	79.0%	81.2%	0.802	95.0%	39.9%	0.173
Embedding (ce_dim32)	78.4%	80.6%	0.806	96.4%	34.5%	—
AdaBoost	74.0%	78.0%	0.775	86.4%	44.0%	0.557
GAN, MMD-merged (8 groups)	47.0%	64.2%	—	—	—	—
GAN, per-attack	43.0%	—	—	—	—	—

Model	Acc.	Balanced	Macro-F1	Known-N2	Novel-N2	ECE
argmax						

Table 5: Final comparison on the identical 12,766-row KDDTest+ attack evaluation set. Metric columns: Acc. = overall (micro) accuracy; Balanced = mean per-surface recall; Macro-F1 = unweighted mean of the per-surface F1 scores; Known-/Novel-N2 = accuracy on known vs. N2-novel rows; ECE = expected calibration error (15 bins), which applies only to the probabilistic models. Known/novel split uses N2 (attack type absent from KDDTrain+). GAN rows are context, produced on the same test rows. Every tree row is a single-seed (seed 42) point estimate; the ten-seed Random Forest figures are in Table 4. The embedding row reports the production seed-42 artifact; across ten retraining seeds the method averages 75.0 ± 2.0, and across the fifteen candidate configurations of Table 3 the median is 74.1%. The tree rows train on 18 attack types and the embedding and GAN rows on 11; Table 4 quantifies that asymmetry.

Table 6 condenses the three central systems onto the three tasks defined earlier. The overall reading is as follows: the all-data Random Forest is the better closed-set router (82.1 vs. 78.4 for the embedding artifact) and the better novel-attack router in aggregate (41.6% vs. 34.5% on the 3,720 N2-novel rows). The closed-set advantage, but not the novel-attack advantage, largely disappears once training information is matched. Coverage behavior is similar at the confident end: at 50% coverage the embedding routes at 99.3% and Random Forest at 99.9%, with Random Forest holding its advantage across the full coverage sweep. Random Forest's probabilities are also well calibrated for the closed-set task (ECE 0.038, best of all models tested).

System	Routing acc.	Routing acc., matched training	Macro-F1	Novelty AUC (N1)
Random Forest	82.1%	77.3 ± 0.6	0.828	0.849
Attack-signature embedding	78.4%	78.4% (already matched)	0.806	0.853 (0.926 ablated)
Per-attack GAN critics	47.0% (merged)	47.0% (already matched)	—	not applicable

Table 6: Three-system summary across the three tasks. Routing acc. is overall accuracy under the all-data tree baseline; the next column repeats it with the tree retrained on only the eleven attack types the other two systems see. Macro-F1 is the unweighted mean per-surface F1; Novelty AUC is threshold-free ROC AUC under the N1 definition, where the Random Forest value is the indirect maximum-probability baseline [20]. The GAN row reports its best variant per column.

GAN Critic Systems

Unrestricted argmax. Scoring every test row against all eleven critics and routing to the z-scored argmax gives 43.0% surface-routing accuracy (36.6% exact attack match). The same rows give Random Forest 82.1%, or 77.3% with matched training data. For context, an earlier phase of this project built a six-critic surface-pooled GAN system that reached 63.7% under a different, Stage-1-gated protocol, so moving from pooled critics to per-attack specialists reduced accuracy rather than improving it. Increasing specialization was not the missing ingredient.

Two observations from the per-attack breakdown motivated the diagnostics. Neptune, the largest class (4,657 rows, 36% of the test set), routed correctly only 31% of the time, with 43% of its rows claimed by the ipsweep critic. And the failures were not random: specific critics (ipsweep, back) absorbed samples from many unrelated attacks, which pointed at the critics rather than the attacks.

Similarity-guided merging. The winning sweep ($q = 0.2$) accepted three merges out of twelve trials: nmap+satan, then smurf joining that group, then back+guess_passwd, taking eleven critics to eight. Validation balanced accuracy rose from 72.8% to 84.8% (11,702 validation rows), and test surface-routing accuracy rose from 43.0% to 47.0%. That is a real improvement, and the accepted merges are sensible (the reconnaissance scanners group together), but the merged system remains 30 points below even the matched-training Random Forest and 17 below the earlier pooled-critic system. The grouping was not the core problem, which raised the next question: is the critic architecture itself the problem?

Diagnostics. Three measurements across all eleven critics on the full test set.

(1) Score scales are incomparable. The satan critic's outputs span +17 to +485; the back critic's span −3 to +11. Across the eleven critics the per-critic standard deviation ranges from 0.98 (back) to 18.62 (satan), a ratio of 19×, and a Levene test rejects equality of variances decisively ($W = 873$, $p < 10^{-300}$; Figure 2). This is not a subtle effect: the score scales genuinely differ by more than an order of magnitude. Under raw argmax the loudest critics win everything: raw routing is 11.9% exact-match. Per-critic z-scoring raises it to 36.6%, percentile normalization to 35.8%, temperature scaling to 18.6%. (These are eleven-way exact-match numbers, the stricter diagnostic; in six-way surface terms, z-scored argmax corresponds to 43.0% and percentile to 45.4%. That is the bridge between the 36.6% and 43.0% figures.) Calibration is a real and large factor, and also clearly not the whole story, because the best calibrated variant is still 32 points behind the matched-training tree on the surface metric.

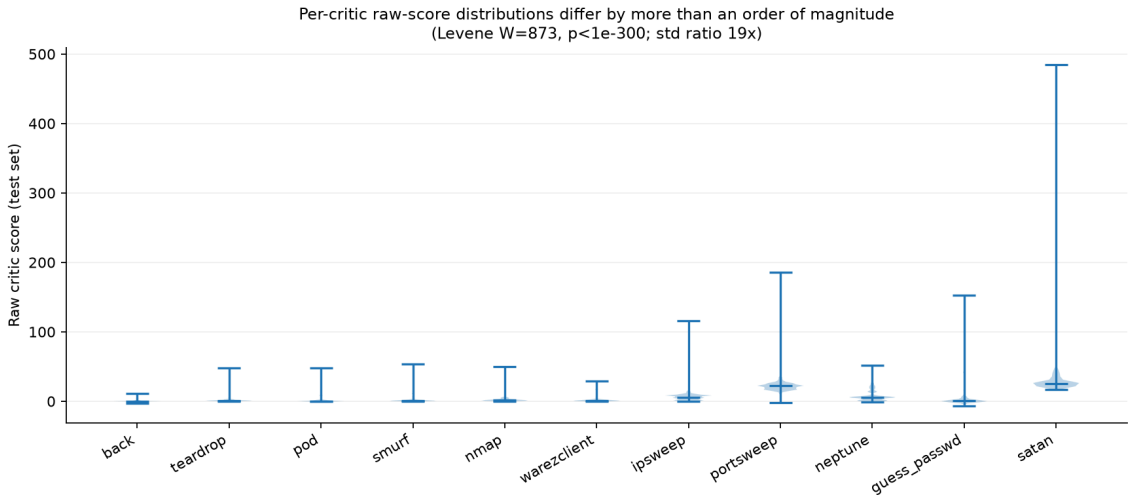

Figure 2: Raw test-set score distributions for the eleven per-attack critics (violin plots, sorted by standard deviation). Scales differ by 19× (satan vs. back); a Levene test rejects variance equality at $p < 10^{-300}$. This is the mechanism behind the argmax failure of Proposition 1.

(2) Some critics prefer other attacks to their own. This is the part no calibration can touch. The Neptune critic scores mscan rows higher than Neptune rows: mean +16.7 vs. +11.1, and the gap is even larger by median (+18.8 vs. +5.9, the Neptune-on-Neptune distribution being right-skewed). The difference is not a small-sample artifact — a Welch t-test gives $t = 23.9$ ($p < 10^{-300}$, n = 996 mscan vs. 4,657 Neptune rows), a Mann–Whitney test agrees ($p < 10^{-300}$), and Cliff's $\delta = +0.52$ indicates a large effect. The satan critic ranks its own attack seventh among test attack types, behind guess_passwd, mscan, and processtable among others. Any monotone per-critic rescaling preserves these orderings, so the argmax stays wrong. This is the empirical content of Proposition 1: independently trained critics estimate unnormalized potentials that are neither on a common scale (point i) nor even correctly ordered on their own class (point ii).

(3) Universal attractors. Under z-scored argmax, the ipsweep and back critics jointly absorb large fractions of Neptune (43%), satan (49%), portsweep (45%), and guess_passwd (60%) samples. The one-class training objective explains this: a critic never sees other attacks during training, so nothing penalizes it for scoring them high, and a critic whose attack shares low-level features with many others becomes a magnet. The back critic's near-constant score scale, visible in its training curve as the inversion described above, is the same property seen from a different angle.

Retrieval-restricted competition. The sweep over $K \in \{1, 2, 3, 4, 5, 7, 11\}$ (Figure 3) settles the question the diagnostics raised. The optimum in all four metric-calibration combinations is K = 1, pure embedding routing with no critic vote, and accuracy falls monotonically as K grows. Under the final leakage-free protocol: 78.4% at K=1, 72.3% at K=2, 55.3% at K=3, 48.6% at K=4, down to 43.0% at K=11. Even a shortlist of two lets a wrong critic outscore the right one on six percent of the test set, and each additional candidate makes it worse. Under every configuration tested, critic votes reduced routing

accuracy.

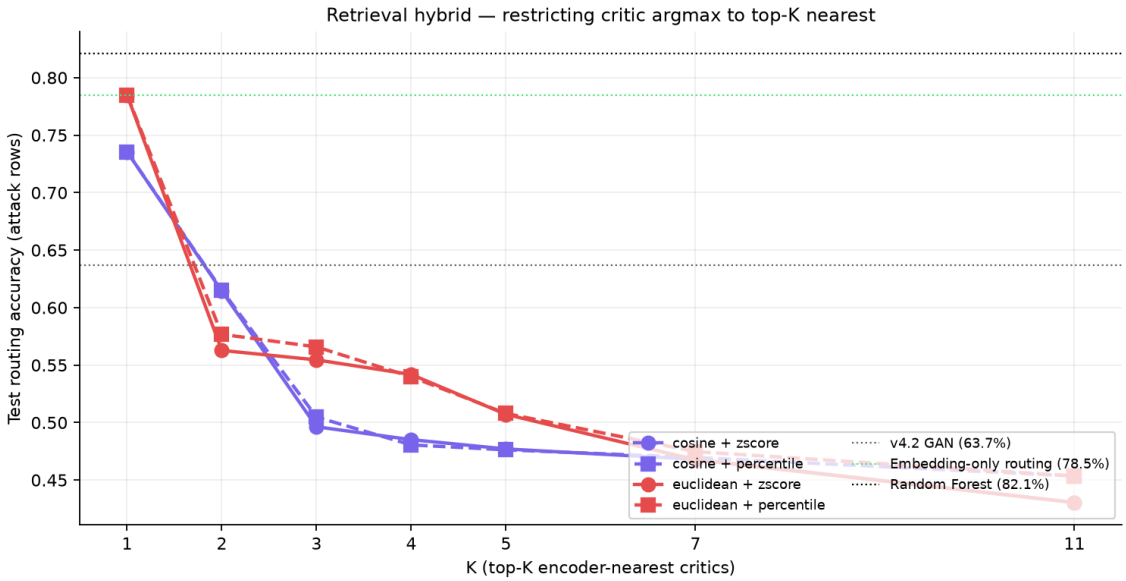

Figure 3: Retrieval-restricted critic competition. Restricting the argmax to the K embedding-nearest critics never beats using no critics at all (K=1). The figure shows the original four-configuration sweep; the sweep was re-run under the final leakage-free standardization with the same shape and conclusion: 78.4% at K=1 declining monotonically to 43.0% at K=11.

Embedding Analysis

Nearest-centroid routing with the production encoder reaches 78.4% surface-routing accuracy, 80.6% balanced accuracy, and macro-F1 0.806 on the full test set. The nearest-centroid classifier (NCC) is used in place of the encoder's trained softmax head deliberately: running the retained softmax head on the same test set gives only 73.9% routing accuracy (77.4% balanced), 4.5 points below NCC. The softmax head is overconfident on NSL-KDD's shifted test split, whereas the centroid distance degrades more gracefully; the two routers agree on 92.6% of rows, so NCC is not a different model, just a more robust decision rule on the same representation. This also gives the embedding its novelty signal, which the softmax head does not. The novelty result itself is not sensitive to the distance metric: on the N1 split the standardized-Euclidean distance used throughout gives AUC 0.853, Mahalanobis (per-class shrunk covariance) 0.838, and cosine 0.831, so the simplest metric is also the best here. Figure 4 visualizes the embedding space itself: well-supported attacks form separated clusters, and several test-only attack types visibly sit either inside a related cluster or away from all of them, which previews the novelty analysis below.

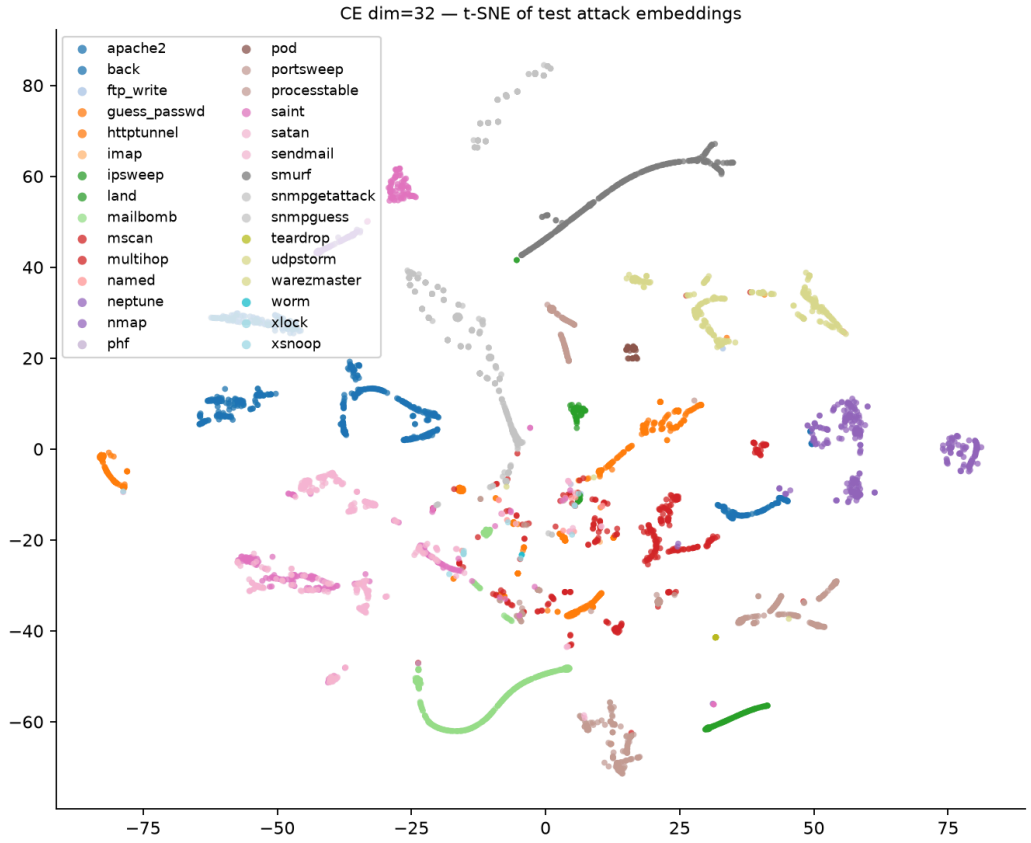

Figure 4: t-SNE projection [18] of the attack-signature embeddings of test attack rows (up to 300 rows per attack type, perplexity 30, seed 42). Cluster structure reflects attack mechanism: the trained attack types form tight clusters, and test-only types either join a mechanistically related cluster or remain isolated.

Table 7 quantifies the same geometry directly: cosine similarities between the eleven training-attack centroids. The matrix is mostly near zero or negative (the classes are well separated), with the notable positive entries matching mechanism relationships (pod and teardrop at +0.42, both fragmentation-adjacent) and the strong negatives separating the large flood cluster from everything else (neptune vs. back at −0.54).

	bck	gpw	ips	nep	nmp	pod	pts	sat	smf	tdp	wcl
bck	1.00	.17	−.12	−.54	.00	.12	−.17	.03	.08	.16	.17
gpw	.17	1.00	−.29	−.10	.00	.11	−.08	−.05	−.00	.10	.18
ips	−.12	−.29	1.00	−.14	.05	−.16	−.05	−.19	.01	−.53	−.06
nep	−.54	−.10	−.14	1.00	−.37	.03	.02	−.48	−.14	−.06	−.09

	bck	gpw	ips	nep	nmp	pod	pts	sat	smf	tdp	wcl
nmp	.00	.00	.05	−.37	1.00	−.16	−.43	.13	.09	.11	.07
pod	.12	.11	−.16	.03	−.16	1.00	−.08	−.17	−.36	.42	−.39
pts	−.17	−.08	−.05	.02	−.43	−.08	1.00	.20	−.47	−.01	.09
sat	.03	−.05	−.19	−.48	.13	−.17	.20	1.00	.16	−.12	−.21
smf	.08	−.00	.01	−.14	.09	−.36	−.47	.16	1.00	−.37	−.05
tdp	.16	.10	−.53	−.06	.11	.42	−.01	−.12	−.37	1.00	.15
wcl	.17	.18	−.06	−.09	.07	−.39	.09	−.21	−.05	.15	1.00

Table 7: Cosine similarity between training-attack centroids in the attack-signature embedding (production encoder). Abbreviations: bck=back, gpw=guess_passwd, ips=ipsweep, nep=neptune, nmp=nmap, pod=pod, pts=portsweep, sat=satan, smf=smurf, tdp=teardrop, wcl=warezclient.

Three results with adequate support are worth pulling out.

• saint (319 rows, never in encoder training): 98.7% correct (Wilson 95% CI [96.8, 99.5]) versus Random Forest's 69.9% (CI [64.7, 74.7]). The intervals are disjoint, so despite the modest support the advantage is not a small-sample artifact. Saint is an extended SATAN scanner, and its rows land inside the satan cluster because the mechanism matches. This is the clearest example of mechanism-level generalization in the project.
• mailbomb (293 rows, absent from all training): 14.3% versus 0.0% for every tree model. Weak in absolute terms, but the only non-zero routing any evaluated system achieved on this attack.
• mscan (996 rows): 41.2% versus 79.6% for Random Forest. The embedding loses substantially here; mscan's rows sit far from the reconnaissance centroids (mean distance 4.36, see below), so the nearest-centroid rule has little to work with.

The embedding also loses on the small-support catch-all attacks (land, xsnoop, sendmail, xlock, named, phf). These are N1-novel but mostly not N2-novel: the tree models trained on those attack types and the embedding has no centroid for them. This is the per-attack face of the training-information asymmetry quantified in Table 4, and it is largely removed when the tree is retrained on matched data: matched Random Forest scores 0.0% on xlock and 0.0% on phf, against 88.9% and 50.0% for the all-data tree. Given their support ($n \leq 17$), these rows affect the aggregate numbers but do not support per-attack conclusions. Table 8 gives the per-attack view against the all-data Random Forest.

Attack	n	Emb.	RF	Attack	n	Emb.	RF
neptune	4657	99.4	98.8	ipsweep	141	98.6	98.6
guess_passwd	1231	81.1	97.8	httptunnel†	133	3.8	5.3
mscan†	996	41.2	79.6	nmap	73	100.0	100.0
warezmaster	944	96.4	98.1	pod	41	100.0	97.6
apache2†	737	67.8	66.4	multihop	18	22.2	44.4
satan	735	100.0	100.0	named†	17	17.6	64.7
processtable†	685	0.4	0.0	sendmail†	14	7.1	85.7
smurf	665	100.0	100.0	teardrop	12	100.0	100.0
back	359	100.0	100.0	xlock†	9	33.3	88.9
snmpguess†	331	0.0	0.0	land	7	0.0	100.0
saint†	319	98.7	69.9	xsnoop†	4	0.0	100.0
mailbomb†	293	14.3	0.0	ftp_write	3	66.7	66.7
snmpgetattack†	178	0.0	0.0	phf	2	0.0	50.0
portsweep	157	98.7	100.0	udpstorm†/worm†/imap	5	0.0	0.0

Table 8: Per-attack surface-routing accuracy, embedding vs. all-data Random Forest, sorted by test support. N2-novel attack types marked †. Rows with n < 20 are included for completeness only; no conclusions are drawn from them.

Open-Set Novelty Detection

Result: AUC 0.853 (4,695 N1-novel vs. 8,071 N1-known rows; production encoder; across ten retraining seeds 0.850 ± 0.013, range 0.833–0.873; across the fifteen candidate configurations of Table 3, median 0.845, range 0.776–0.860). Figure 5 shows the two distance distributions. The geometry is more specific

than "known equals close, novel equals far," and the more specific version is the more interesting one. Novel attack types whose mechanism resembles a trained attack can sit close to the related centroid: saint sits at mean distance 1.40 (near satan) and warezmaster at 2.87 (near warezclient). Novel types whose mechanism is genuinely absent sit far from everything: mscan at 4.36 and processtable at 4.77. So what the embedding appears to encode is mechanism similarity, and distance from the known centroids works as a novelty signal precisely for attacks whose mechanisms are not represented in training.

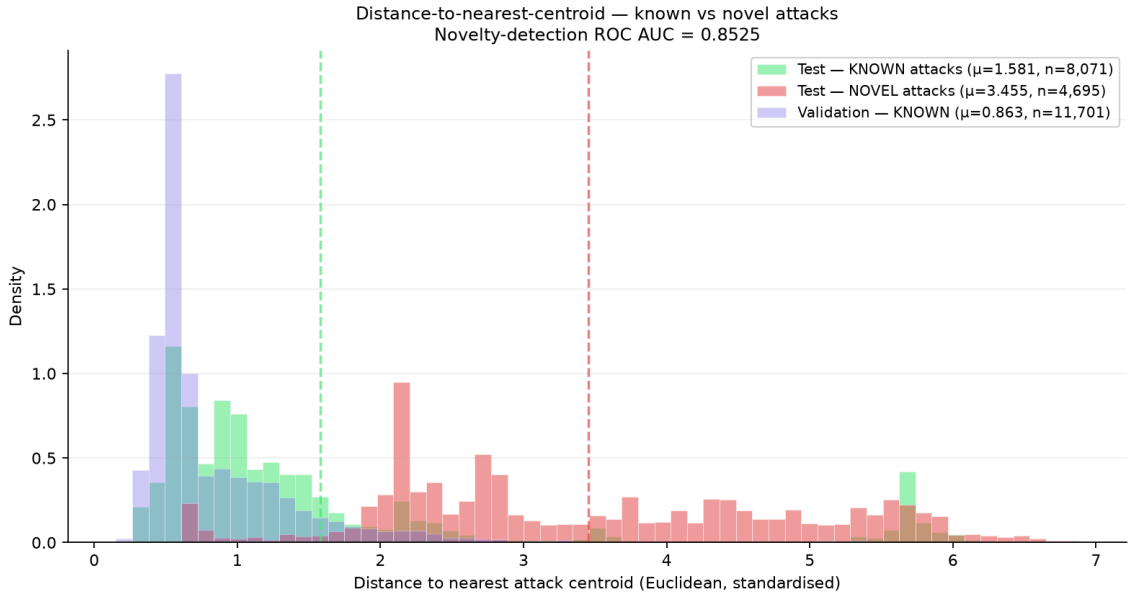

Figure 5: Distance to nearest known-attack centroid, split by whether the test row's attack type was in encoder training (N1 definition). Novel types shift right by roughly two standardized units on average.

Is it novelty or just train/test drift? NSL-KDD's test split is known to be shifted relative to training, so distances could have grown for reasons unrelated to novelty. Decomposing: validation rows of known attacks sit at mean distance 0.86; test rows of the same known attacks at 1.58 (a drift of +0.72); test rows of novel attacks at 3.45 (+2.59). The novel shift is 3.6 times the drift shift. Most of the signal is novelty; the drift component is real and reported.

Operating points. At a threshold detecting 80% of novel rows, 17.7% of known rows are falsely rejected; at 90% detection, 20.6%. Beyond 95% detection the false-rejection cost climbs steeply (33.5%, then 76.0% at 99%).

Comparison with maximum-softmax-probability baselines. The standard baseline for out-of-distribution detection is the maximum softmax probability (MSP): score a row by $1 - \max_k P(k \mid x)$ from a softmax classifier [20]. Two versions apply here. The encoder's own discarded softmax head gives MSP AUC 0.804 on the N1 split; a Random Forest's MSP gives 0.849. The embedding distance (0.853) beats the encoder-MSP by 4.9 points and edges the Random-Forest MSP (Table 9), so against the required

baseline it is the strongest of the three signals, though the margin over RF-MSP is small, and smaller than the spread across encoder configurations, so it should not be read as a decisive advantage. The conclusion is that the embedding's advantage is modest in raw AUC and larger in what the signal exposes. Its advantages are structural: the distance signal is computed in a learned representation whose geometry is interpretable (mechanistically similar attacks are close, as the saint and warezmaster cases show), and the feature ablation below shows how to improve it to 0.926 by removing service-identity features, a design lever the max-probability signal does not expose. Decision uncertainty among six known surfaces and distance from the training manifold are different quantities [19, 21]; on this dataset they perform similarly in aggregate, and distinguishing them empirically would require attacks that are confidently mis-assigned by the trees, which NSL-KDD provides only in the small-support tail.

Novelty signal	ROC AUC (N1)
Embedding distance-to-nearest-centroid	0.853
Random Forest MSP (1 − max prob)	0.849
Encoder softmax-head MSP (1 − max prob)	0.804

Table 9: Novelty detection on the N1 split (4,695 novel vs. 8,071 known test rows), threshold-free ROC AUC. MSP = maximum softmax probability [20].

One caveat the histogram also exposes: guess_passwd, a known attack with only 53 training rows, sits at mean distance 4.41, farther than most novel types. Its train and test distributions genuinely differ. Distance measures unfamiliarity, and unfamiliarity can come from distribution shift within a known class, not only from new classes. This is the same sparsity that makes guess_passwd the biggest loser when the tree baseline is restricted to matched training data.

Feature Ablation

The unit under test is the encoder methodology end-to-end. For each ablation the feature columns are removed from train and test, the encoder is retrained from scratch under the identical protocol, centroids are rebuilt, and the untouched test set is scored for surface-routing accuracy, novelty AUC (N1), and known/novel distance separation. Feature groups follow NSL-KDD's semantic structure: basic/intrinsic, content, time-based traffic, host-based traffic, and the three one-hot families (protocol, service, flag). Group ablations run at three seeds (42–44, mean ± sd); the 121 single-feature ablations run at seed 42. We define the routing importance of a group as ΔAcc, the change in overall surface-routing accuracy when the group is removed and the encoder retrained, and its novelty importance as ΔAUC, the change in N1 novelty ROC AUC under the same removal. A feature group is important for a given objective only if its removal consistently degrades that objective across seeds.

Group removed	#feat	Δrouting (pp)	ΔAUC	Interpretation
time-based traffic	9	−4.5 ± 0.2	−0.060 ± 0.052	backbone; only group both objectives need
basic/intrinsic	6	−2.4 ± 0.7	+0.020 ± 0.025	routing-important (src_bytes, wrong_fragment); mildly masks novelty
host-based traffic	10	−2.2 ± 0.3	+0.015 ± 0.003	routing-important; neutral-to-masking for novelty
service one-hots	70	−1.8 ± 0.5	+0.084 ± 0.012	slight routing help; hurts novelty because novel attacks reuse familiar services
flag one-hots	11	−1.5 ± 0.4	+0.007 ± 0.004	modest routing signal
protocol one-hots	3	−1.5 ± 0.8	+0.027 ± 0.015	modest routing signal
content	12	+0.2 ± 2.2	−0.049 ± 0.019	novelty-only; no routing value, but how the embedding spots unfamiliar R2L behavior

Table 10: Central ablation result: feature importance is objective-dependent. Leave-one-group-out, full encoder retrain per run, 3 seeds, mean ± sd. Columns: #feat = number of features removed; Δrouting (pp) = change in overall routing accuracy in percentage points; ΔAUC = change in N1 novelty ROC AUC. Baseline (all 121 features): routing 75.5% ± 2.1, novelty AUC 0.841 ± 0.008. The features that help route an attack are largely not the features that help recognize it as novel.

The two importances are reported separately (Table 10), and they turn out to be largely different things. We note that three seeds is too few for a powered group-by-metric interaction test; the dissociation is therefore presented as a consistent directional pattern with non-overlapping error bands on the largest

effects, not as a formal significance result.

Three findings from Table 10 and Figure 6.

Time-based traffic features are the backbone. The nine count/rate features (count, srv_count, the serror/rerror rates, the service-diversity rates) are the only group whose removal hurts both objectives at once (−4.5 points routing, −0.060 AUC). This matches the mechanism story: connection tempo and error patterns are what distinguish a flood from a scan from normal traffic.

Routing importance and novelty importance dissociate. Content features (failed logins, root shells, file creations) contribute nothing measurable to routing (+0.2, within noise) but their removal costs −0.049 novelty AUC. They are how the embedding recognizes that R2L-style test rows are unfamiliar. The reverse also occurs: basic and host-traffic features matter for routing but their removal slightly helps novelty separation.

Service one-hots hurt novelty detection. Removing all 70 service indicator columns costs only 1.8 routing points but raises novelty AUC from 0.841 to 0.926. The most defensible interpretation is that novel attacks mostly arrive over services the encoder saw constantly in training (http, telnet, smtp, private), and those service coordinates pull novel rows toward known centroids, masking the mechanism-level unfamiliarity the traffic features would otherwise expose. An encoder intended primarily as a novelty detector should probably not be given raw service identity at all. That is a concrete, testable design rule this ablation produces.

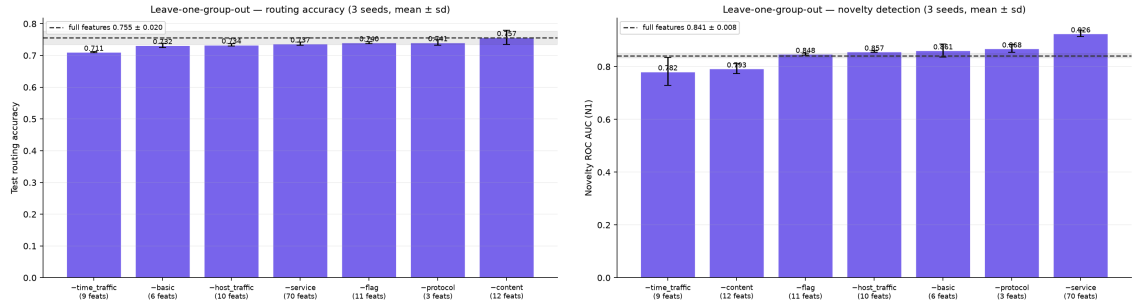

Figure 6: The same seven group ablations scored against the two objectives. Left: routing accuracy; every removal except content hurts. Right: novelty AUC; removing service one-hots improves it to 0.926 while removing content or time-traffic features degrades it. The two rankings disagree, which is the point.

Feature removed	Routing	Δrouting	Novelty AUC	ΔAUC
diff_srv_rate	70.9 ± 0.2	−4.7	0.862 ± 0.019	+0.020
wrong_fragment	72.1 ± 0.4	−3.5	0.883 ± 0.027	+0.042
service_pop_3	72.4 ± 1.6	−3.2	0.910 ± 0.015	+0.068
service_telnet	72.9 ± 1.8	−2.7	0.852 ± 0.014	+0.011

Feature removed	Routing	Δrouting	Novelty AUC	ΔAUC
src_bytes	73.0 ± 0.4	−2.6	0.873 ± 0.014	+0.032
logged_in	73.4 ± 0.4	−2.2	0.825 ± 0.014	−0.016
count	74.3 ± 0.3	−1.2	0.851 ± 0.021	+0.009

Table 11: Multi-seed single-feature ablation (3 runs each). Baseline: routing 75.5 ± 2.1, novelty AUC 0.841 ± 0.008. Only features whose removal effect was re-verified across seeds are shown.

Single-feature results (Table 11). All 121 leave-one-feature-out ablations ran at seed 42. Because removing a column changes the network shape, every such run is effectively a fresh seed draw, so single-seed deltas carry noise comparable to the ±2-point seed spread. Every feature quoted below was therefore re-run at seeds 43–44, with 3-run mean ± sd reported against the 3-seed baseline; features whose effect did not clear the noise are not quoted.

The single most important feature for routing is diff_srv_rate, the fraction of recent connections to different services, whose removal costs 4.7 points with the tightest variance of any ablation (±0.2). This is the core reconnaissance signature: scanners touch many services, floods touch one. wrong_fragment is second (−3.5); it is nearly the sole carrier of the Fragmentation surface. By mean delta, service_pop_3 (−3.2) and service_telnet (−2.7) come next, but both carry roughly four times the run-to-run variance (±1.6–1.8); the third-largest low-variance routing effect is src_bytes (−2.6 ± 0.4), which separates probe-sized from flood-sized traffic. On the novelty side the single feature picture matches the group story: service_pop_3 is the largest individual novelty-harmful feature (AUC +0.068 when removed, consistent with the service-group result), and logged_in is the strongest individual content feature for novelty (−0.016 when removed, in the expected direction but within one sd of noise; the content-group effect of −0.049 across 12 features is the reliable version of this claim). The features whose removal hurts routing most (diff_srv_rate, wrong_fragment, src_bytes) all mildly improve novelty AUC when absent. The routing backbone and the novelty signals live largely in different parts of the representation.

DISCUSISON

The experiments were run in a specific order, and each one existed because of the previous result. The initial hypothesis (H1) was that per-attack critics route better than conventional classifiers; the first full evaluation put them 39 points behind the all-data Random Forest. The natural follow-up (H2) was that the critics were sliced too thin, which motivated the MMD-guided merging; merging helped by 4 points and left a 35-point gap, so specialization was not the core problem. The diagnostics then suggested calibration (H3), which tripled exact-match accuracy and still left the best calibrated variant 37 points behind on the surface metric. The last repair (H4) was restricting critic competition to embedding-similar candidates, and its sweep produced the clearest single result in the project: the optimum was K=1, no critics at all.

What remained (H5) was to characterize the embedding itself, which led to the novelty-detection result and the feature ablation.

Diagnostic evaluation of per-attack GAN critics. A WGAN-GP critic is trained against exactly two kinds of input: real samples of its own attack and generator fakes. Nothing in that objective penalizes scoring a different attack highly. A critic whose attack shares low-level structure with many others (ipsweep, back) becomes a universal attractor, and a critic whose attack sits near a busy region of feature space (neptune, satan) can end up preferring its neighbors to itself. Add independently determined score scales spanning an order of magnitude and argmax competition breaks in two ways at once: calibration repairs the scale problem (11.9% to 36.6% exact-match) but cannot repair off-target preference, because monotone rescaling preserves each critic's internal ranking. The retrieval sweep then settled it: even restricted to two embedding-similar candidates, critic votes only subtracted accuracy. On this dataset and setup, the failure is architectural, a density estimator being asked a discriminative question, rather than a matter of tuning.

It is worth being precise about what this does and does not establish. The failure is a property of the specific combination tested: WGAN-GP critics, trained independently one per attack type with no shared negative pool, combined by per-critic score argmax. Proposition 1 applies to exactly that combination, and the two empirical failure modes are the two the proposition predicts. It does not follow that GAN critics cannot route attacks under a different formulation. A jointly trained critic set that sees other attacks as negatives would break the independence assumption that drives point (ii); a conditional or class-conditioned critic with a shared output scale would attack point (i); a different divergence, a spectrally normalized critic [30], or a diffusion-based density model might behave differently again. None of these were tested here, and the result should not be cited as evidence against them.

GAN-based detection components continue to be proposed in the intrusion-detection literature, usually evaluated in isolation rather than in competition [8]. Each critic here also looked good in isolation (validation AUC 0.91–1.00 against matched negatives). The failure only appears when critics must rank against each other, which is exactly the deployment condition for any per-attack ensemble. That gap between isolated and competitive evaluation is the transferable lesson, and it is independent of the particular critic formulation: any per-class scorer combined by argmax inherits the identifiability problem of point (i) unless its scores are explicitly made commensurable.

Performance analysis of attack-signature embeddings. Against the practical all-data baseline the embedding does not win: 78.4% against 82.1% for closed-set routing, and 34.5% against 41.6% on N2-novel routing. Against the matched-training control the closed-set picture is different, 78.4% for the artifact against 77.3 ± 0.6%, or 75.0 ± 2.0 at the method level so the two are comparable once they are given the same attack types, and the published gap is better read as the value of KDDTrain+'s long tail of rare attack types than as a deficit of the representation. The trees' advantage on genuinely novel attack types survives the control and is real.

The embedding's value is therefore threefold, and none of the three is "higher closed-set accuracy." First, it reaches tree-comparable closed-set routing from a 32-dimensional representation given equivalent training information. Second, it generalizes at the level of mechanism when a novel attack genuinely resembles a trained one (saint, never trained on, routes at 98.7% against 69.9% for the all-data tree, and mailbomb is the only attack any system routes above zero). Third, the novelty signal comes from the representation itself, with no separate novelty-detection model, and its geometry is interpretable and

improvable (the ablation shows how to raise it to 0.926 by withholding service identity). In a deployed pipeline the proposal is complementary: a tree model trained on all available data for closed-set routing, with the embedding distance as an open-set flag for attack types that deserve human review.

Two caveats attach to the embedding numbers and should travel with them. The encoder configuration could not be selected on validation data, because a within-KDDTrain+ split cannot separate the candidates (Table 2); the configuration used here is the best of fifteen on test, and a median choice would have routed about four points lower (Table 3). And the routing number is seed-sensitive at ±2.0 points while the novelty AUC is not. The novelty result is thus the more robust of the two headline claims, surviving both the configuration spread (median 0.845) and the seed spread (0.850 ± 0.013); the routing headline of 78.4% should be read as an artifact figure with roughly 74–75% as the method-level expectation.

The ablation found that feature importance is objective-relative. For a novelty detector built on this kind of embedding, the ablation says: keep the time-based traffic features, keep the content features, and drop raw service identity. None of that is visible from a single aggregate accuracy number, which is the argument for running ablations against each objective separately.

Limitations

Scope of the negative result. The negative result covers per-attack WGAN-GP critics, trained independently with no shared negative pool, combined by score argmax, together with its calibrated, merged, and retrieval-restricted variants, on NSL-KDD under one surface taxonomy. It does not establish that GAN critics are ineffective for attack routing in general. Jointly trained critic ensembles, class-conditional critics, spectrally normalized critics [30], alternative divergences, and diffusion-based density models are untested here, and the diagnosis in fact points to joint training with shared negatives as the specific repair the evidence motivates. Different datasets, GAN objectives, critic architectures, or routing tasks could behave differently.

Configuration selection. A within-KDDTrain+ validation split cannot discriminate among the fifteen encoder configurations (all route at 99.74–99.85% on held-out validation rows), so the configuration used throughout was not selected on held-out data. Reported post hoc, the fifteen span 72.07–78.38% test routing accuracy, and the configuration used sits at the maximum. The routing headline is therefore optimistic by roughly four points relative to a median configuration choice, and both the artifact number and the distribution are reported so the reader can discount accordingly. Novelty AUC is much less affected (ce_dim32 at 0.853 against a median of 0.845). A cleaner design would hold out a second, distributionally shifted validation set for selection, which NSL-KDD does not naturally provide.

One dataset. Everything here is NSL-KDD under one preprocessing pipeline and one surface taxonomy. NSL-KDD was chosen because it remains a benchmark standard for diagnostic work of this kind, which keeps the results comparable to prior GAN-IDS studies, but it does not reflect modern network traffic, and conclusions tied to its specific attack mix may not transfer. The protocol is frozen and ready to rerun on modern datasets such as CICIDS2017 [25] or UNSW-NB15 [26]; dataset choice is a known confounder in this field [27].

Study scope. This work is a diagnostic study whose main result is a scoped negative one, establishing which components of the proposed architecture do not contribute under this setup. That kind of result is valuable because it narrows the search space for later work, but it is not a deployment evaluation, and the

practical claims are limited to the routing and novelty-detection tasks defined here.

Baseline training data. The tree baselines and the embedding do not see the same attack types by default, and the two comparisons answer different questions (Table 4). Both are reported, but only Random Forest was retrained under matched conditions; LightGBM, XGBoost, and AdaBoost are reported in their all-data form only, so the matched-training column of Table 6 should not be extrapolated to them.

Distribution shift. NSL-KDD's test split is shifted relative to training even for known attacks (a measured +0.72 standardized-distance drift), and this partially entangles "new attack type" with "shifted distribution of a known type" in the novelty analysis. The guess_passwd case (a known attack sitting farther out than most novel types) is the clearest example, and the dataset provides no way to fully separate the two causes.

Seed sensitivity. Across ten retraining seeds, encoder routing accuracy is 75.0 ± 2.0 points (range 73.0–79.0), and the production encoder sits near the top of that distribution. Both the artifact number (78.4%) and the method number (75.0 ± 2.0) are reported. The novelty AUC does not share this fragility: its ten-seed mean (0.850 ± 0.013) matches the headline.

Small-support attacks. Eleven test attack types have fewer than 20 rows; per-attack numbers for them are reported for completeness but are not evidence.

Compute budget. Every experiment ran on a single laptop CPU, which shaped several choices: the encoder is deliberately small, the 121 single-feature ablations ran at one seed (any quoted single feature was re-checked across seeds), and candidate critics in the merge loop trained on a reduced epoch budget. These choices should not change the conclusions, but a larger budget would tighten the error bars.

Conclusion

This study investigated a specific proposal: whether per-attack WGAN-GP critics, trained independently and combined by score argmax, can act as competitive discriminators that route network attacks to their underlying attack surfaces. On NSL-KDD they cannot. The architecture reached 43.0% surface-routing accuracy against 82.1% for a Random Forest trained on all available data, or 77.3 ± 0.6% for one restricted to the same attack types the critics were given. Extensive repair attempts, including MMD-guided critic merging, four score-calibration schemes, and embedding-restricted competition across K=1 to K=11, never produced a configuration in which critic votes helped, and the retrieval sweep's optimum was to use no critics at all.

The diagnosis is specific rather than general, and the claim should be read at the same resolution. Two failure modes, both consequences of the one-class training objective, account for the result: independently trained critics carry no common score scale, which per-critic calibration partly repairs, and some critics rank other attacks above their own, which no monotone rescaling can repair. Each critic was individually well-trained (0.91 to 1.00 validation AUC against matched negatives at its kept checkpoint), so the failure belongs to the competition, not to the training. This is a result about this critic formulation and this routing design. It does not show that GAN critics are useless for intrusion detection, and it does not extend to jointly trained critic ensembles with shared negative pools, class-conditional critics, spectrally normalized critics, or non-GAN density models, none of which were tested. Indeed the diagnosis points directly at joint training with shared negatives as the repair the evidence motivates, and that experiment remains open.

The broader and more transferable lesson concerns evaluation rather than architecture: a per-class scorer that looks strong in isolation can fail badly once it must rank against its peers, and any per-class ensemble combined by argmax inherits that identifiability problem unless its scores are explicitly made commensurable. Reporting isolated per-critic AUCs is not evidence of routing performance.

The second contribution is the attack-signature embedding, originally built as an auxiliary component. The 32-dimensional representation routes at 78.4% for the evaluated artifact and 75.0 ± 2.0 across retraining seeds, which is comparable to a Random Forest given the same eleven attack types (77.3 ± 0.6%), and it supplies open-set novelty detection at ROC AUC 0.853 from its own distance geometry, with no separate novelty model. Its embedding space encodes mechanism-level similarity, which lets it generalize to novel attack types that share structure with trained ones (saint, never trained on, routes at 98.7%). Feature ablation further revealed that the features needed for routing and those needed for novelty detection are largely disjoint, yielding a concrete design rule: withhold raw service identity from an encoder intended as a novelty detector, which raises its AUC from 0.841 to 0.926.

Two qualifications travel with those numbers. The encoder configuration could not be chosen on held-out data, because a within-KDDTrain+ validation split cannot separate the candidates; the configuration used is the best of fifteen on test, and a median choice routes about four points lower. And the routing figure is seed-sensitive while the novelty figure is not, so the novelty result is the sturdier of the two.

These findings suggest that, for intrusion-detection systems of this kind, architecturally separating routing from novelty detection is more productive than asking a single generative-discriminative framework to do both. GANs remain useful for data augmentation; as standalone competitive routers in this formulation they would require reformulating the training objective to incorporate cross-class awareness. Future work should test exactly that, a jointly trained critic set with shared negatives, and should port this frozen protocol to a modern dataset, where the surface taxonomy, critic set, and novelty definitions would have to be re-derived before the questions here can be asked again.

Broader Impact and Ethical Considerations

This work is a diagnostic study on a public benchmark and introduces no new attack capability. Its practical message is cautionary: it shows that a GAN-critic routing component which looks strong in isolation can fail badly once deployed in competition, so operators should not treat isolated per-critic AUCs as evidence of routing performance. The novelty signal studied here is a triage aid, not an autonomous decision rule; the operating points reported above make explicit that flagging more novel attacks always costs some false rejection of known traffic, and any deployment must choose that trade-off deliberately rather than treating the detector as a filter. Because the study uses only NSL-KDD, none of its numbers should be read as deployment-ready; the intended use is to inform method design, not to be fielded.

Data and human-subjects statement: NSL-KDD derives from the 1998–99 DARPA/KDD Cup intrusion-detection evaluation traffic (MIT Lincoln Laboratory), generated on a simulated network testbed rather than captured from real users. Its records are connection-level aggregate statistics (protocol, service, byte and packet counts, flags); they contain no packet payloads, personal identifiers, or human-subject data. No new data was collected for this study.

This paper's contribution is defensive: it diagnoses why a GAN-critic routing architecture underperforms

and characterizes an embedding-based alternative; it introduces no new attack technique. The adversarial-evasion literature cited (e.g., IDSGAN [24]) is discussed only as motivating context for why static classifiers can be fragile, not extended or applied here. Intrusion-detection research is nonetheless dual-use in general, and any deployment of the routing or novelty-detection components described here should be evaluated against an adaptive adversary before use, not just against the static NSL-KDD test set.

Reproducibility and Code Availability

All experiments use the public NSL-KDD files with a fixed seed (42) and the preprocessing, splits, and hyperparameters specified above and in Table 1. Model and metric selection is performed on the 20% per-attack validation split of KDDTrain+; KDDTest+ is used only for final reporting and for the explicitly labeled post hoc sensitivity analysis of Table 3. The code and the scripts that regenerate every table and figure are publicly available at https://github.com/felixolmo03/Beyond-GAN-Critics. The repository does not include the NSL-KDD data files or the trained model checkpoints: NSL-KDD is obtained separately from its public distribution, and checkpoints are regenerable from the included pipeline rather than stored in the repository.

References

[1] Goodfellow, Ian, et al. "Generative Adversarial Nets." Advances in Neural Information Processing Systems, vol. 27, 2014, pp. 2672–2680, https://papers.nips.cc/paper_files/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html.
[2] Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. "Improved Training of Wasserstein GANs." Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 5767–5777, https://papers.nips.cc/paper_files/paper/2017/hash/892c3b1c6dccd52936e27cbd0ff683d6-Abstract.html.
[3] Schlegl, Thomas, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery." Information Processing in Medical Imaging, edited by Marc Niethammer et al., Springer, 2017, pp. 146–157, https://doi.org/10.1007/978-3-319-59050-9_12.
[4] Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. "A Detailed Analysis of the KDD Cup 99 Data Set." 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009. IEEE, https://doi.org/10.1109/CISDA.2009.5356528.
[5] Gretton, Arthur, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. "A Kernel Two-Sample Test." Journal of Machine Learning Research, vol. 13, 2012, pp. 723–773, https://jmlr.org/papers/v13/gretton12a.html.
[6] Khosla, Prannay, et al. "Supervised Contrastive Learning." Advances in Neural Information Processing Systems, vol. 33, 2020, https://papers.nips.cc/paper_files/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html.
[7] Hermans, Alexander, Lucas Beyer, and Bastian Leibe. "In Defense of the Triplet Loss for Person Re-Identification." arXiv, 2017, arXiv:1703.07737, https://arxiv.org/abs/1703.07737.
[8] Ndayipfukamiye, Tharcisse, et al. "Adversarial Defense in Cybersecurity: A Systematic Review of

GANs for Threat Detection and Mitigation." arXiv, 24 Sept. 2025, arXiv:2509.20411, https://arxiv.org/abs/2509.20411.
[9] Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 3146–3154, https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.
[10] Vo, Hoang V., et al. "AWDLID: Augmented WGAN and Deep Learning for Improved Intrusion Detection." 2024 1st International Conference on Cryptography and Information Security (VCRIS), 2024, pp. 1–6. IEEE, https://doi.org/10.1109/VCRIS63677.2024.10813392.
[11] Vo, Hoang V., et al. "APELID: Enhancing Real-Time Intrusion Detection with Augmented WGAN and Parallel Ensemble Learning." Computers & Security, vol. 136, 2024, article 103567. Elsevier, https://doi.org/10.1016/j.cose.2023.103567.
[12] Ding, Hongwei, Yu Sun, Nana Huang, Zhidong Shen, and Xiaohui Cui. "TMG-GAN: Generative Adversarial Networks-Based Imbalanced Learning for Network Intrusion Detection." IEEE Transactions on Information Forensics and Security, vol. 19, 2024, pp. 1156–1167. IEEE, https://doi.org/10.1109/TIFS.2023.3331240.
[13] Breiman, Leo. "Random Forests." Machine Learning, vol. 45, no. 1, 2001, pp. 5–32. Springer, https://doi.org/10.1023/A:1010933404324.
[14] Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794. ACM, https://doi.org/10.1145/2939672.2939785.
[15] Freund, Yoav, and Robert E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." Journal of Computer and System Sciences, vol. 55, no. 1, 1997, pp. 119–139. Elsevier, https://doi.org/10.1006/jcss.1997.1504.
[16] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks." Proceedings of the 34th International Conference on Machine Learning, vol. 70, PMLR, 2017, pp. 214–223, https://proceedings.mlr.press/v70/arjovsky17a.html.
[17] Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: Synthetic Minority Over-Sampling Technique." Journal of Artificial Intelligence Research, vol. 16, 2002, pp. 321–357, https://doi.org/10.1613/jair.953.
[18] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing Data Using t-SNE." Journal of Machine Learning Research, vol. 9, 2008, pp. 2579–2605, https://jmlr.org/papers/v9/vandermaaten08a.html.
[19] Scheirer, Walter J., Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. "Toward Open Set Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 7, 2013, pp. 1757–1772. IEEE, https://doi.org/10.1109/TPAMI.2012.256.
[20] Hendrycks, Dan, and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks." International Conference on Learning Representations, 2017. OpenReview, https://openreview.net/forum?id=Hkg4TI9xl.
[21] Bendale, Abhijit, and Terrance E. Boult. "Towards Open Set Deep Networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1563–1572. IEEE, https://doi.org/10.1109/CVPR.2016.173.
[22] Akçay, Samet, Amir Atapour-Abarghouei, and Toby P. Breckon. "GANomaly: Semi-Supervised

Anomaly Detection via Adversarial Training." Computer Vision – ACCV 2018, edited by C. V. Jawahar et al., Springer, 2019, pp. 622–637, https://doi.org/10.1007/978-3-030-20893-6_39.
[23] Zenati, Houssam, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. "Efficient GAN-Based Anomaly Detection." International Conference on Learning Representations Workshop Track, 2018. OpenReview, https://openreview.net/forum?id=BkXADmJDM.
[24] Lin, Zilong, et al. "IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection." Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2022, https://doi.org/10.1007/978-3-031-05981-0_7.
[25] Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A. Ghorbani. "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization." Proceedings of the 4th International Conference on Information Systems Security and Privacy, 2018, pp. 108–116. SCITEPRESS, https://doi.org/10.5220/0006639801080116.
[26] Moustafa, Nour, and Jill Slay. "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)." 2015 Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6. IEEE, https://doi.org/10.1109/MILCIS.2015.7348942.
[27] Ring, Markus, Sarah Wunderlich, Deniz Scheuring, Dieter Landes, and Andreas Hotho. "A Survey of Network-Based Intrusion Detection Data Sets." Computers & Security, vol. 86, 2019, pp. 147–167. Elsevier, https://doi.org/10.1016/j.cose.2019.06.005.
[28] Zhang, Xinchen, Running Zhao, Zhihan Jiang, Zhicong Sun, Yulong Ding, Edith C. H. Ngai, and Shuang-Hua Yang. "AOC-IDS: Autonomous Online Framework with Contrastive Learning for Intrusion Detection." 2024 IEEE Conference on Computer Communications (INFOCOM), 2024, pp. 581–590. IEEE, https://doi.org/10.1109/INFOCOM52122.2024.10621346.
[29] Qiu, Zhiyin, et al. "VAEMax: Open-Set Intrusion Detection Based on OpenMax and Variational Autoencoder." 2024 15th IEEE International Conference on Information and Communication Technology Convergence (ICTC), 2024, pp. 98–105. IEEE, https://doi.org/10.1109/ICTC61510.2024.10601788.
[30] Miyato, Takeru, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. "Spectral Normalization for Generative Adversarial Networks." International Conference on Learning Representations, 2018. OpenReview, https://openreview.net/forum?id=B1QRgziT-.