

Out-of-Distribution Detection via Channelwise Feature Aggregation in Neural Network–Based Receivers

Marko Tuononen^{1,3}, Heikki Penttinen¹, Duy Vu¹, Dani Korpi²,
Vesa Starck¹ and Ville Hautamäki³

¹Nokia Networks, Karakaari 7, 02610 Espoo, Finland

²Nokia Bell Labs, Karakaari 7, 02610 Espoo, Finland

³University of Eastern Finland, P.O. Box 111, 80101 Joensuu, Finland
Corresponding author: marko.1.tuononen@nokia.com

Abstract

Neural network–based radio receivers are expected to play a key role in future wireless systems, making reliable Out-Of-Distribution (OOD) detection essential. We propose a post-hoc, layerwise OOD framework based on channelwise feature aggregation that avoids classwise statistics—critical for multi-label soft-bit outputs with astronomically many classes. Receiver activations exhibit no discrete clusters but a smooth Signal-to-Noise-Ratio (SNR)–aligned manifold, consistent with classical receiver behavior and motivating manifold-aware OOD detection. We evaluate multiple OOD feature types, distance metrics, and methods across layers. Gaussian Mahalanobis with mean activations is the strongest single detector, earlier layers outperform later, and SNR/classifier fusions offer small, inconsistent AUROC gains. High-delay OOD is detected reliably, while high-speed remains challenging.

1 Introduction

Deep neural network–based radio receivers are emerging as a key component of future 6G systems [Hoydis *et al.*, 2021], offering improved performance and reduced design complexity compared to conventional signal-processing pipelines [Farhadi *et al.*, 2025]. However, *neural receivers can be brittle under distribution shift*: when propagation conditions at inference deviate from those seen during training, performance may degrade sharply without any internal indication of failure [Luostari *et al.*, 2025]. As emphasized in recent work on AI safety [Bengio *et al.*, 2024], monitoring and understanding the internal behavior of neural networks is essential for reliable deployment. In wireless systems, which operate under highly variable and unpredictable channel conditions, this requirement is fundamental.

Out-Of-Distribution (OOD) detection is therefore becoming indispensable in wireless machine learning [Liu *et al.*, 2022]. We focus on Type-2 OOD (covariate shift), where the input distribution changes, while the prediction task remains unchanged, as in deployment scenarios with larger delay spreads, higher user velocities, or unforeseen interference conditions. Neural receivers such as DeepRx [Honkala *et al.*,

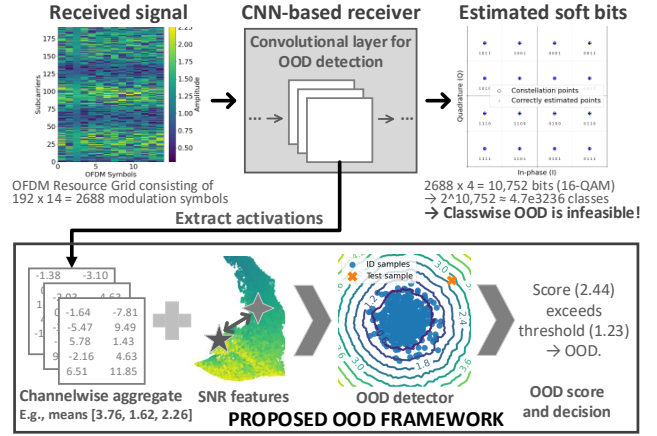


Figure 1: Visual overview of OOD detection for neural receivers. OOD detection operates on aggregated activation features, rather than classwise statistics, forming an SNR-aligned manifold.

2021] produce thousands of soft-bit outputs per sample, forming a *multi-label prediction problem* [Tarekegn *et al.*, 2024]—implying an astronomical output space (Fig. 1, top). This makes classwise OOD detection methods fundamentally ill-posed, as they rely on a limited number of discrete classes.

Practical deployment further constrains OOD detection: the receiver architecture is typically frozen and tightly optimized for latency, power consumption, and hardware compatibility [Tuononen *et al.*, 2025]. Many powerful OOD approaches—such as Bayesian neural networks [Blundell *et al.*, 2015], deep ensembles [Lakshminarayanan *et al.*, 2017], evidential networks [Aguilar *et al.*, 2025], or auxiliary OOD heads [Hendrycks *et al.*, 2019]—require architectural modification, retraining, or additional inference passes, making them unsuitable for production-grade wireless receivers. This motivates post-hoc OOD detection methods that operate solely on internal activations of a frozen model.

In this paper, we propose a post-hoc OOD detection framework for neural network–based radio receivers (Fig. 1, bottom) that avoids classwise modeling and leaves the receiver architecture unchanged. We show that receiver activations do not form discrete clusters but instead lie on a continuous Signal-to-Noise Ratio (SNR)–aligned manifold, a physically plausible structure that motivates manifold-aware OOD

detection. Building on this insight, we develop a layerwise, channelwise feature aggregation framework that supports multiple OOD models and metrics, incorporates SNR-aware conditioning, and enables both intra- and inter-layer scoring. The approach enables both offline debugging and online monitoring, including anomaly detection [Yang *et al.*, 2024], drift awareness [Uzlaner *et al.*, 2025], and network-assisted reporting [Ren *et al.*, 2021], contributing toward reliable and trustworthy neural receivers for future 6G systems.

2 Related Work

According to the manifold hypothesis, high-dimensional data governed by continuous generative factors tends to concentrate near low-dimensional manifolds embedded in the input space [Nilsson *et al.*, 2007; Bengio *et al.*, 2013]. Such continuous structures have been observed in domains ranging from images [Tenenbaum *et al.*, 2000] to biomedical data [Islam *et al.*, 2023], whereas discrete cluster structures are more typical of single-label classification tasks [Bengio *et al.*, 2013].

Understanding internal neural representations is the goal of *mechanistic interpretability*, which studies how networks encode and manipulate information [Räuker *et al.*, 2023; Sharkey *et al.*, 2025]. Prior work spans visualization [Zeiler and Fergus, 2014], concept-based probes [Kim *et al.*, 2018], representation-similarity [Kornblith *et al.*, 2019], and internal circuits or neuron communities [Wang *et al.*, 2023; Lan *et al.*, 2025]. In wireless receivers, DeepRx has been analyzed through representation-level encoding of channel parameters such as SNR [Tuononen *et al.*, 2025], underscoring the practical value of interpretability for deployed systems [Jeffares and van der Schaar, 2025].

OOD detection aims to identify inputs that deviate from the training distribution due to covariate or semantic shift [Yang *et al.*, 2024]. Many post-hoc methods operate on hidden representations, including k -Nearest Neighbors [Cover and Hart, 1967; Papernot and McDaniel, 2018], class-conditional Mahalanobis [Lee *et al.*, 2018; Müller and Hein, 2025], and channel-selection for semantic shifts [Yuan *et al.*, 2024]. Extensions to multi-label settings include energy-based aggregation [Wang *et al.*, 2021], which assumes semantic shift and conditional label independence [Zhang and Taneva-Popova, 2023], and latent-variable models such as β -Variational Auto Encoder (β -VAE) for discrete generative factors [Sundar *et al.*, 2020]. OOD score fusion methods include conformal p-value aggregation [Magesh *et al.*, 2023] and one-class classifiers such as One-Class Support Vector Machine (OC-SVM) [Schölkopf *et al.*, 2001; Bounsiar and Madden, 2014].

In wireless communications, OOD detection has mainly focused on transmitter identification and modulation recognition [Liu *et al.*, 2022]. Related anomaly-detection work includes Deep Convolutional Autoencoder (DCAE)-based detection of physical tampering at Orthogonal Frequency-Division Multiplexing (OFDM) receivers [Dehmollaian *et al.*, 2021], where natural channel variation is excluded. Despite this extensive literature, post-hoc OOD detection for multi-label, Type-2 OOD neural physical-layer receivers—where classwise modeling is infeasible and the architecture must remain frozen—has not been previously addressed.

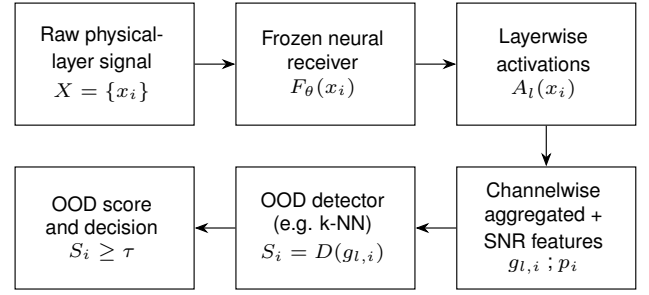


Figure 2: Post-hoc OOD detection operating on a frozen model.

3 Methodology

Let $X \subset \mathbb{C}^{N_f \times N_t}$ denote the space of OFDM resource-grid [3GPP, 2025] inputs consisting of N_f subcarriers and N_t OFDM symbols (e.g., 192×14). For a given modulation order M , the receiver outputs $Y \subset [0, 1]^{N_f \times N_t \times B}$, where $B = \log_2 M$ is the number of bits per symbol; for example, $B = 4$ for 16-Quadrature Amplitude Modulation (16-QAM). We consider a frozen neural receiver F_θ , parameterized by θ , as a mapping $F_\theta : X \rightarrow Y$. During receiver training, data came from the joint distribution $P_{\text{train}}(X, Y)$, while deployment samples follow $P_{\text{test}}(X, Y)$. In Type-2 OOD detection (covariate shift [Quiñonero-Candela *et al.*, 2008]),

$$P_{\text{train}}(Y | X) = P_{\text{test}}(Y | X), \text{ but } P_{\text{train}}(X) \neq P_{\text{test}}(X), \quad (1)$$

as occurs when propagation characteristics—SNR, Doppler spread, multipath profile, interference—differ between training and deployment. Our task is to construct a detector D that maps each received signal to a real-valued OOD score.

Layerwise Activation Extraction. Let the neural receiver contain L convolutional layers. For a chosen layer $\ell \in \{1, \dots, L\}$, the activation tensor is

$$A_\ell(x) \in \mathbb{R}^{C_\ell \times H_\ell \times W_\ell}, \quad (2)$$

where C_ℓ is the number of convolutional channels and (H_ℓ, W_ℓ) are the spatial dimensions (i.e., (N_f, N_t) for DeepRx). For a dataset $X = \{x_i\}_{i=1}^N$, we denote activations

$$A_\ell(X) = \{A_\ell(x_i)\}_{i=1}^N. \quad (3)$$

Channelwise Feature Extraction. Features used for OOD detection are derived from per-channel activation vectors:

$$\mathbf{a}_{\ell,i,c} = \text{vec}(A_\ell(x_i)[c, :, :]) \in \mathbb{R}^{H_\ell W_\ell}, \quad c = 1, \dots, C_\ell. \quad (4)$$

Each feature family applies a normalization scheme appropriate to the statistic being computed, and all resulting per-channel features are concatenated across channels. Thus, for each layer ℓ and input sample x_i , the activation-derived feature vector used for downstream modeling is one of

$$\mathbf{g}_{\ell,i} \in \{\mathbf{m}_{\ell,i}, \mathbf{k}_{\ell,i}, \mathbf{e}_{\ell,i}\}. \quad (5)$$

(i) Mean-activation features. The MaxAbs scaled mean activation for channel c is computed as [Bäuerle *et al.*, 2022]

$$m_{\ell,i,c} = \frac{1}{H_\ell W_\ell} \sum_{q=1}^{H_\ell W_\ell} \frac{\mathbf{a}_{\ell,i,c}[q]}{\max_j |\mathbf{a}_{\ell,j,c}|[q]}, \quad (6)$$

where q indexes the flattened spatial locations. Concatenating across channels gives the mean feature vector

$$\mathbf{m}_{\ell,i} = [m_{\ell,i,1}, \dots, m_{\ell,i,C_\ell}]. \quad (7)$$

(ii) Activation-distribution features. The activation-distribution shape for channel c is computed by first applying Z-score normalization [Han *et al.*, 2012],

$$\tilde{\mathbf{a}}_{\ell,i,c}^{\text{Zscore}} = \frac{\mathbf{a}_{\ell,i,c} - \mu_{\ell,c}}{\sigma_{\ell,c}}, \quad (8)$$

and then computing a one-dimensional Kernel Density Estimate (KDE) [Parzen, 1962],

$$\hat{p}_{\ell,i,c}(t) = \frac{1}{H_\ell W_\ell h_{\ell,c}} \sum_{q=1}^{H_\ell W_\ell} K\left(\frac{t - \tilde{\mathbf{a}}_{\ell,i,c}^{\text{Zscore}}[q]}{h_{\ell,c}}\right), \quad (9)$$

where $\mu_{\ell,c}$ and $\sigma_{\ell,c}$ are computed over all spatial locations and training samples and K is a Gaussian kernel with bandwidth $h_{\ell,c}$. We use Scott's Rule [Scott, 1992] for selecting the bandwidth, providing a standard bias-variance tradeoff:

$$h_{\ell,c} = 1/\sqrt[5]{H_\ell W_\ell}. \quad (10)$$

To obtain fixed-length features, we sample the KDE over a Z-score interval covering $P\%$ of activation mass,

$$[z_{\min}, z_{\max}] = \left[\Phi^{-1}\left(\frac{1-P/100}{2}\right), \Phi^{-1}\left(\frac{1+P/100}{2}\right)\right], \quad (11)$$

where $\Phi^{-1}(\cdot)$ is inverse cumulative distribution function of the standard normal distribution [Bliss, 1934], using R uniformly spaced points,

$$\hat{p}_{\ell,i,c}^{(r)} = \hat{p}_{\ell,i,c}\left(z_{\min} + r \frac{z_{\max} - z_{\min}}{R-1}\right), \quad (12)$$

where $r \in [0, R-1]$. The per-channel KDE vector is

$$\mathbf{k}_{\ell,i,c} = [\hat{p}_{\ell,i,c}^{(0)}, \dots, \hat{p}_{\ell,i,c}^{(R-1)}], \quad (13)$$

and concatenation across channels yields

$$\mathbf{k}_{\ell,i} = [\mathbf{k}_{\ell,i,1}, \dots, \mathbf{k}_{\ell,i,C_\ell}]. \quad (14)$$

(iii) Free-energy features. The absolute energy for channel c is computed as [Liu *et al.*, 2020]

$$e_{\ell,i,c} = -\log \sum_{q=1}^{H_\ell W_\ell} \exp(\mathbf{a}_{\ell,i,c}[q]), \quad (15)$$

and concatenation across channels yields

$$\mathbf{e}_{\ell,i} = [e_{\ell,i,1}, \dots, e_{\ell,i,C_\ell}]. \quad (16)$$

SNR-Based Complementary Features. Let $p_i \in \mathbb{R}$ denote the estimated Signal-to-Noise-Ratio (SNR) of input x_i . SNR estimate p_i can be used directly as an auxiliary feature or through the SNR distance

$$d_{\text{SNR}}(i, j) = |p_i - p_j|. \quad (17)$$

To better separate samples that are close in activation space but correspond to different channel conditions, we incorporate SNR into OOD detectors. In particular, we use conditional k-NN schemes, where neighbor selection is either restricted to samples with similar SNR (hard conditioning) or smoothly weighted by SNR mismatch (soft conditioning), analogous to stratified k-NN [Katila and Tomppo, 2002] but with a continuous auxiliary variable. Full definitions are provided in Supplemental E.

OOD Modeling. Let $\mathbf{g}_{\ell,i}$ denote the feature for layer ℓ and sample i . OOD detectors are fitted on the In-Distribution (ID) training set and produce a scalar OOD score. HDBSCAN and k-NN rely on an explicit distance metric $d(\cdot, \cdot)$, whereas OC-SVM and Mahalanobis induce distances implicitly via a learned decision boundary or covariance structure.

(i) Density-based (HDBSCAN). We cluster ID features using HDBSCAN* [Campello *et al.*, 2013; McInnes and Healy, 2017] and use the cluster-membership strength

$$s_{\ell,i}^{\text{HDBSCAN}} = 1 - \text{membership}(\mathbf{g}_{\ell,i}), \quad (18)$$

since low membership indicates noise and is treated as OOD.

(ii) Neighbourhood-based (k-NN). We train on ID features and use the mean distance to the k -nearest neighbours [Cover and Hart, 1967],

$$s_{\ell,i}^{\text{kNN}} = \frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} d(\mathbf{g}_{\ell,i}, \mathbf{g}_{\ell,j}), \quad (19)$$

where $\mathcal{N}_k(i)$ denotes the k -nearest neighbours in training set for sample i . We do not apply Deep k -Nearest Neighbors [Papernot and McDaniel, 2018], as it relies on class-label agreement, which is undefined for multi-label soft-bit outputs.

(iii) Boundary-based (One-Class SVM). We train on ID features and use the distance to decision boundary [Schölkopf *et al.*, 2001; Bounsiar and Madden, 2014],

$$s_{\ell,i}^{\text{OCSVM}} = -b_\ell(\mathbf{g}_{\ell,i}), \quad (20)$$

where $b_\ell(\mathbf{g}_{\ell,i}) = \sum_j \alpha_{\ell,j} K(\mathbf{g}_{\ell,i}, \mathbf{g}_{\ell,j}) - \rho$ is the learned decision function (positive inside the ID region) with kernel K .

(iv) Covariance-based (Gaussian Mahalanobis). Assuming Gaussian ID features,

$$s_{\ell,i}^{\text{Mahalanobis}} = (\mathbf{g}_{\ell,i} - \mu_\ell)^\top \Sigma_\ell^{-1} (\mathbf{g}_{\ell,i} - \mu_\ell), \quad (21)$$

where (μ_ℓ, Σ_ℓ) are the empirical mean and covariance [Mahalanobis, 1936]. We do not adopt the class-conditional Mahalanobis variants of [Lee *et al.*, 2018; Müller and Hein, 2025], as they rely on per-class distributions, which are ill-defined for multi-label receiver outputs.

Fusion Across Detectors and Layers (Optional). Each layer ℓ yields a set of scores

$$\mathbf{s}_{\ell,i} = [s_{\ell,i}^{\text{kNN}}, s_{\ell,i}^{\text{Mahalanobis}}, s_{\ell,i}^{\text{HDBSCAN}}, s_{\ell,i}^{\text{OCSVM}}, p_i]. \quad (22)$$

Within-layer fusion combines (two or more) detector outputs at a single layer, either by fitting an OC-SVM on scores or by converting the scores to conformal p -values and applying a Benjamini–Hochberg (BH) [Benjamini and Hochberg, 1995] style combination rule [Magesh *et al.*, 2023]. Within-layer fusion leverages classifier diversity.

Across-layer fusion combines layer-level scores (of one or more detectors) either by fitting OC-SVM or applying BH style combining, yielding a continuous OOD score via Supplemental F. Layer fusion leverages across-layer diversity.

OOD Decision. Let S_i denote the final OOD score for sample x_i . A binary decision is obtained by thresholding,

$$x_i \text{ is OOD} \iff S_i \geq \tau, \quad (23)$$

where the threshold τ is chosen to satisfy a desired operating point (e.g., FPR@95%TPR). The continuous score S_i can also be used to generate Detection Error Tradeoff (DET) curves [Martin *et al.*, 1997].

Our framework is post-hoc: the receiver parameters θ are kept frozen. It is not model-agnostic, as feature extraction assumes convolutional layers with channelwise structure, as in ResNet-like neural receivers such as DeepRx [Honkala *et al.*, 2021]. The block diagram in Fig. 2 depicts the full dataflow.

Table 1: Simulation parameters for DeepRx and Out-Of-Distribution (OOD) detector training/testing. In-Distribution (ID) testing mirrors the OOD-training parameters. OOD scenarios: (i) fixed 1000 ns delay spread (very long multipath; Table 7.7.3-1 of [3GPP, 2020]), (ii) fixed 866 Hz Doppler (approximately 100 m/s at 2.6 GHz; high-speed rail). These OOD settings exceed DeepRx’s training range yet remain realistic. OOD testing uses the Clustered Delay Line (CDL)-C model to induce a channel-model mismatch; training uses CDL-A and CDL-E.

Parameter	DeepRx train	Detector train and ID test	Detector OOD test High DS	Detector OOD test High Speed
<i>Uniformly Distributed Channel Parameters</i>				
Signal-to-Noise Ratio (SNR)	-10–25 dB	-10–25 dB	-10–25 dB	-10–25 dB
Doppler Shift	0–330 Hz	0–300 Hz	0–300 Hz	866 Hz
Delay Spread	10–300 ns	10–300 ns	1000 ns	10–300 ns
Channel Models	UMi, UMa	CDL-A, E (train) CDL-C (test)	CDL-C	CDL-C
<i>Other Parameters</i>				
Number of PRBs		16 (192 subcarriers)		
Subcarrier Spacing		30 kHz		
TTI Length		14 OFDM symbols / 0.5 ms		
Modulation Scheme		16-QAM		
RX / TX Antennas		2 / 1		
Pilot Symbols per Slot		1 or 2 OFDM symbols		
Pilot Symbol Distribution		Uniformly distributed, fixed locations		

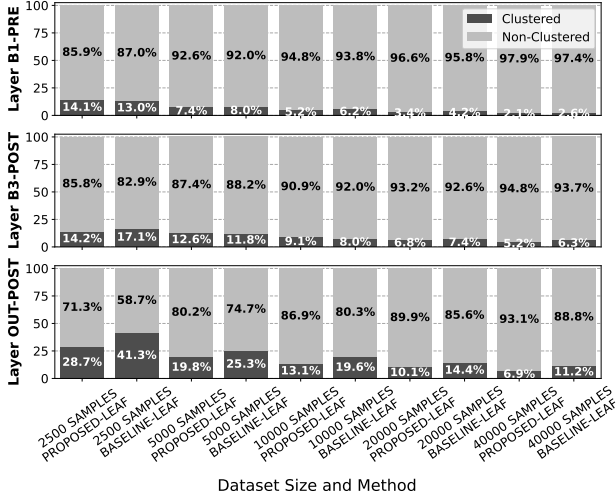


Figure 3: Share of clustered and non-clustered samples versus sample size and aggregation method (minimum cluster size = 5).

4 Cluster and Manifold Analysis

We study whether channelwise activation features from the neural receiver (Supplemental B) form coherent clusters reflecting channel conditions, and how these features are organized on lower-dimensional manifolds. This analysis (i) provides insight into the receiver’s learned internal representations and (ii) informs whether clustering-based or manifold-aware OOD detectors are appropriate.

4.1 Experimental Setup

We evaluate two feature families introduced in Section 3: (i) mean-activation features (Eq. 7) with Euclidean distance [Euclid, 1956], as in [Bäuerle *et al.*, 2022]; and (ii) KDE-

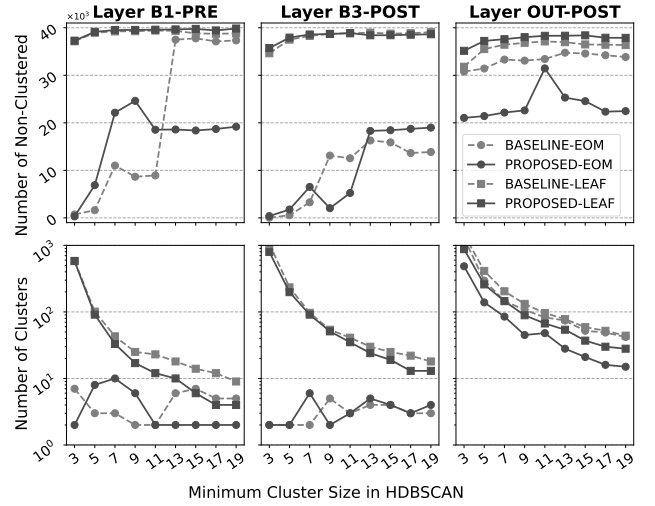


Figure 4: Counts of non-clustered samples and clusters versus minimum cluster size, by aggregation and cluster-selection method.

based activation-distribution features (Eq. 14) with Jensen-Shannon distance [Endres and Schindelin, 2003], representing one of our methodological extensions.

Data are generated with a 5G-compliant PUSCH link-level simulator (NVIDIA Sionna [Hoydis *et al.*, 2023]) using the parameters in Table 1. We reimplemented both the baseline NAP and our extensions in PyTorch [Paszke *et al.*, 2019] using standard scientific Python libraries (Supplementals A.3–A.5). For KDE features, each per-channel distribution is sampled at $R = 10$ Z-scores covering 99% of the standard-normal mass, producing a compact ten-dimensional descriptor per channel.

To explore activation-space structure, we apply HDBSCAN* [Campello *et al.*, 2013; McInnes and Healy, 2017]

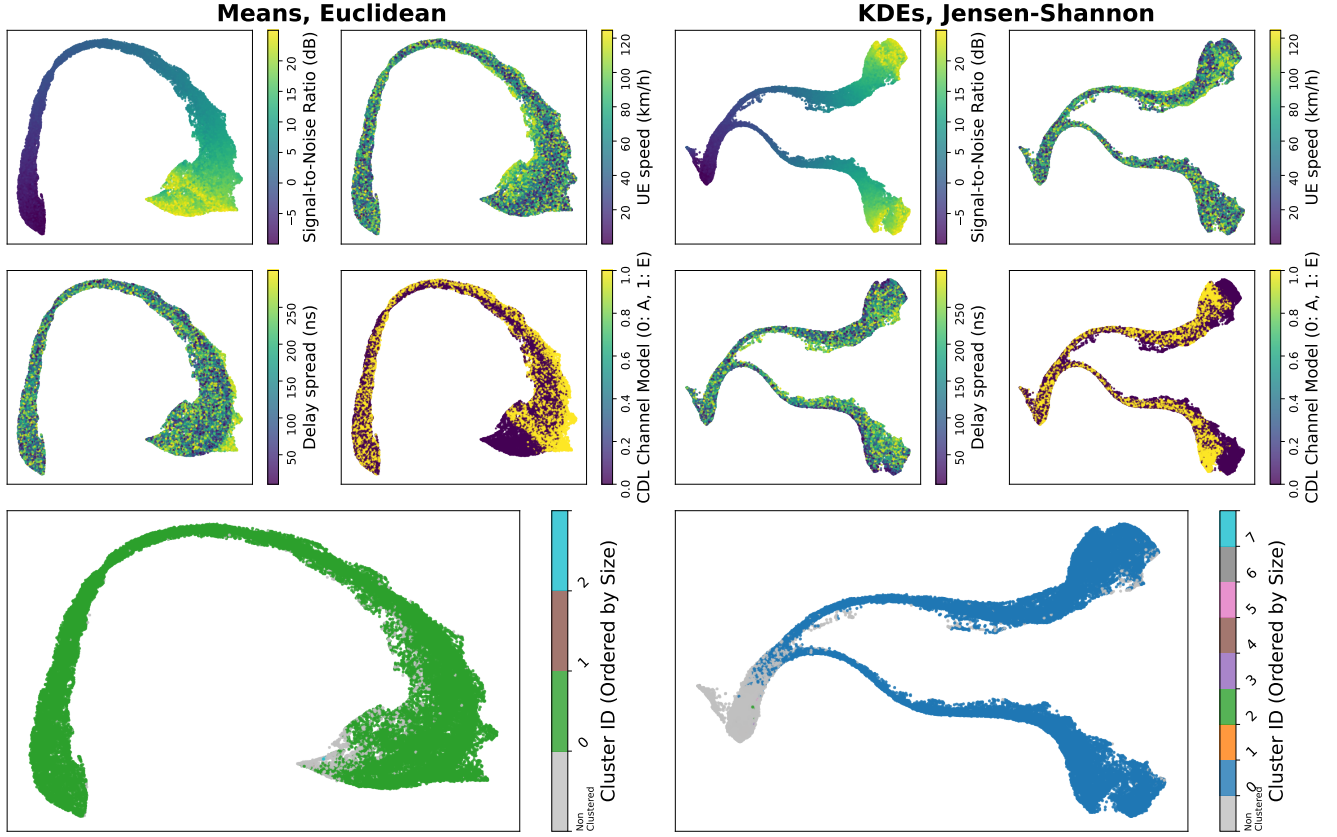


Figure 5: Uniform Manifold Approximation and Projection (UMAP) embeddings of DeepRx activations from layer B1-PRE using the baseline (left) and proposed (right) methodology, colored by channel parameters and partitions with HDBSCAN (EoM, min. cluster size 5).

with both Leaf and Excess-of-Mass (EoM) selection criteria over a sweep of minimum cluster sizes. UMAP [McInnes *et al.*, 2018] is used *only* for visualization, with default Scikit-learn [Pedregosa *et al.*, 2011] settings.

4.2 Results

Absence of discrete, stable clusters. Across layers, feature types, and HDBSCAN configurations, we observe no stable or semantically meaningful clustering structure. Under the Leaf criterion used by [Bäuerle *et al.*, 2022], both mean and KDE-based features yield very high proportions of unclustered samples (often exceeding 90%; Fig. 3). Switching to the Excess-of-Mass (EoM) criterion reduces the number of unclustered samples but produces degenerate solutions when the minimum cluster size is reduced (most samples collapsing into one diffuse cluster; Fig. 4), and internal validation metrics (DBCV, persistence; Supplemental A.7) remain near zero. These findings are consistent with our visual object recognition experiments (Supplemental A.6), suggesting that the lack of cluster structure reflects fundamental properties of activation geometry rather than domain specifics.

Discovery of an SNR-aligned manifold. While clustering fails, a clear low-dimensional manifold emerges when visualized with UMAP (Fig. 5, Supplemental D.1). Activations form a smooth continuum primarily parameterized by SNR, with the channel model contributing only small

secondary variations in the high-SNR regime. This behavior is consistent with classical physical-layer receiver design, where internal processing adapts smoothly as a function of estimated SNR [Goldsmith, 2005]. This observation also aligns with our independent replication using the methodology of [Tuononen *et al.*, 2025], where user speed and delay spread exhibited similarly limited influence (Supplemental C). KDE-based aggregation sometimes reveals two SNR-aligned branches at high SNR that merge at low SNR, suggesting a richer representation than the mean-activation baseline, though the global geometry remains dominated by a single SNR-driven trajectory. This regression-like latent structure is consistent with prior observations that deep regression models tend to produce continuous high-dimensional manifolds rather than discrete clusters [Nilsson *et al.*, 2007; Bengio *et al.*, 2013; Islam *et al.*, 2023].

Implications for OOD detection. The manifold structure has direct consequences for OOD modeling: cluster-based OOD detection (requiring multiple coherent clusters) is unsuitable because ID data do not form compact, well-separated regions. Distance- and distribution-based OOD methods are better aligned with the activation geometry, especially when augmented with SNR information. The clear ordering of activations by SNR directly motivates the SNR-aware and conditional distance metrics introduced in Section 3.

5 Out-Of-Distribution (OOD) Detection

Next we evaluate the proposed OOD detection framework, quantifying the performance of feature–model combinations, the utility of SNR as an auxiliary signal, and the benefits of within- and across-layer fusion.

5.1 Experimental Setup

The experiments use the same dataset as in Section 4 (Table 1). The In-Distribution (ID) set spans a wide range of SNRs, delay spreads, and user speeds. Two OOD scenarios—high delay and high speed—are evaluated, introducing substantial channel mismatch while keeping SNR within the ID range. This prevents trivial SNR-based detection and isolates SNR as an auxiliary signal.

We first establish single-model baselines on the earliest convolutional layer (B1-PRE), motivated by the hypothesis—later confirmed—that earlier layers retain richer information for OOD detection. Using these baselines, we perform two ablations: incorporating SNR to exploit feature diversity, and within-layer fusion to exploit classifier diversity. The same baselines are then evaluated on B3-POST and OUT-POST to enable across-layer fusion.

Feature extraction. We evaluate mean features, KDE features, and free-energy features (Section 3). Free-energy features are computed with temperature $T = 1$. KDE features are as in Section 4. SNR is incorporated for k-NN via (i) raw absolute distance for neighboring points, (ii) hard conditioning, or (iii) soft conditioning (Supplemental E). SNR is incorporated for OC-SVM and Gaussian Mahalanobis by concatenating the activation features and raw SNR values.

Modeling methods and metrics. We evaluate four OOD detector families: density-based (HDBSCAN*, EoM, minimum cluster size 5; Euclidean [Euclid, 1956], cosine [Salton *et al.*, 1975], or Jensen–Shannon distances [Endres and Schindelin, 2003]), neighbourhood-based (k-NN with $K = 5$ using the same distances), boundary-based (OC-SVM with RBF kernel), and covariance-based (Gaussian Mahalanobis with Ledoit–Wolf covariance [Ledoit and Wolf, 2003]).

Classifier fusion. Performed within-layer and across-layer using (i) OC-SVM on concatenated detector scores and raw SNR, or (ii) BH-style conformal p-value fusion (Section 3).

Evaluation metrics. Followed common OOD practice: Area Under Receiver Operating Characteristic (AUROC), False Positive Rate at 95% True Positive Rate (FPR95), and OOD detection latency. Results also visualized as the Detection Error Tradeoff (DET) curve [Martin *et al.*, 1997].

5.2 Results

Single-model baselines on B1-PRE. The best average AUROC across both OOD scenarios selects the following baselines (Table 2): (A) HDBSCAN: mean features + Euclidean distance; (B) k-NN: mean features + cosine distance; (C) OC-SVM: mean features; and (D) Mahalanobis: mean features. Mean features consistently outperform free-energy and KDE features, despite the latter carrying richer distributional information. Mahalanobis achieves the highest AUROC overall, with k-NN close behind; OC-SVM performs clearly worse.

HDBSCAN yields reasonable AUROC in the high-delay scenario (e.g., 0.92) but *always* fails in FPR95 (equal to 1.0), providing no usable operating point. Scenariowise¹, high delay is essentially solved (AUROC > 0.99, FPR95 $\approx$ 2%), whereas high speed remains difficult (AUROC $\approx$ 0.74, FPR95 $\approx$ 79%).

Effect of SNR on B1-PRE. We evaluate SNR integration via conditional k-NN and via concatenation for Mahalanobis and OC-SVM¹. In the high-speed scenario, conditional k-NN yields a small improvement (AUROC 0.7358 vs. 0.7339). In the high-delay scenario, adding SNR slightly degrades AUROC (0.9922 vs. 0.9931). Soft conditioning helps in high speed but hurts in high delay, with hard conditioning showing the opposite trend. Concatenating SNR for OC-SVM and Mahalanobis brings little to no benefit. Overall, SNR yields only marginal AUROC gains (typically fourth decimal place).

Within-layer fusion. We evaluate 27 pairwise and multi-way fusions among the four baselines and raw SNR for each layer¹. Earlier layers are more informative for OOD detection: $\text{AUROC}_{\text{B1-PRE}} > \text{AUROC}_{\text{B3-POST}} > \text{AUROC}_{\text{OUT-POST}}$. OC-SVM applied to detector outputs is the most effective fusion method, outperforming BH-style conformal fusion. Typical best combinations include Mahalanobis + SNR for high delay, while the optimal combination for high speed is layer-dependent. On B1-PRE, for example: high delay–Mahalanobis + SNR increases AUROC from 0.9965 to 0.9966; high speed–k-NN baseline + OC-SVM baseline + SNR increases AUROC from 0.7339 to 0.7550. Overall, within-layer fusion yields modest AUROC gains, primarily in challenging cases such as the OUT-POST layer and high-speed scenarios (Tables 3–4 with OC-SVM fusion).

Across-layer fusion.¹ We extend fusion from within a single layer to jointly combine detector outputs from the B1-PRE, B3-POST, and OUT-POST layers using OC-SVM (Tables 3–4). Across-layer fusion yields complementary AUROC gains, most notably in challenging high-speed scenarios, where both k-NN and Gaussian Mahalanobis benefit.

DET curve summary. Mahalanobis has a steady performance (Figure 6)—especially under high delay scenario—and has the lowest latency (Tables 3–4). When latency is less critical and lower false OOD is prioritized, use conditional k-NN with SNR or within-layer fusion on B1-PRE; for harder cases (e.g., high speed), fuse across layers for added robustness.

6 Conclusions

We introduced a post-hoc OOD detection framework for neural radio receivers and analyzed the geometry of their activation spaces. DeepRx does not form discrete or semantically meaningful activation clusters—most samples remain unclustered even with improved feature extraction and modeling settings—mirroring our visual object recognition results and suggesting that such models inherently lack cluster structure. Instead, activations lie on a low-dimensional manifold dominated by SNR, with other channel parameters contributing only minor variation. This smooth, regression-like geometry aligns with classical physical-layer behavior and offers a natural basis for OOD modeling.

¹ See Supplemental G for the detailed results.

Feature + Distance	HDBSCAN		k-NN		OC-SVM		Mahalanobis	
	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95
Mean + Euclidean	0.7868	1.0000	0.8205	0.5557	0.7694	0.5669	0.8658	0.4384
Mean + Cosine	0.6823	1.0000	0.8635	0.4419				
KDE + Euclidean	0.7806	1.0000	0.8554	0.4327				
KDE + Cosine	0.7855	1.0000	0.8542	0.4409	0.7385	0.6572	0.8605	0.4223
KDE + Jensen-Shannon	0.7806	1.0000	0.8603	0.4222				
Energy + Euclidean	0.7478	1.0000	0.8156	0.5871				
Energy + Cosine	0.7387	1.0000	0.8208	0.5617	0.7432	0.6252	0.8542	0.4587

Table 2: OOD detection on layer B1-PRE averaged over scenarios (mean of five runs; negligible SD). Best feature–distance pairs per metric are **bold**; AUROC-best pairs were chosen as baselines for subsequent experiments. Scenario-wise results provided in the Supplemental G.

Method	Metric	B1-PRE	B3-POST	OUT-POST	Across layers ¹
HDB-SCAN baseline	AUROC	0.9209	0.8370	0.5208	0.9536
	FPR95	1.0000	0.3543	1.0000	0.1483
	Lat. (μ s)	436.40	569.23	263.80	1892.3
k-NN baseline	AUROC	0.9931	0.9174	0.6929	0.9926
	FPR95	0.0218	0.6440	0.8847	0.0312
	Lat. (μ s)	1926.2	1116.2	1417.0	5130.5
OC-SVM baseline	AUROC	0.9394	0.7372	0.5509	0.9198
	FPR95	0.2084	0.8597	0.9486	0.2698
	Lat. (μ s)	2771.5	1722.0	716.51	5115.5
Mahalanobis baseline	AUROC	0.9965	0.9410	0.5677	0.9941
	FPR95	0.0200	0.3327	0.9359	0.0367
	Lat. (μ s)	4.6913	0.8825	0.1678	665.10
Within-layer ¹	AUROC	0.9966	0.9263	0.7753	0.9865
	FPR95	0.0222	0.4924	0.7666	0.0359
	Lat. (μ s)	5448.4	3601.3	956.96	9663.3

Table 3: OOD detection across layers under high delay (mean of five runs; negligible SD); best method–layer per metric are **bold**.

Method	Metric	B1-PRE	B3-POST	OUT-POST	Across layers ¹
HDB-SCAN baseline	AUROC	0.6528	0.6955	0.5096	0.7198
	FPR95	1.0000	0.5966	1.0000	0.9051
	Lat. (μ s)	546.26	309.72	315.09	1790.6
k-NN baseline	AUROC	0.7339	0.7522	0.6620	0.8054
	FPR95	0.8620	0.9138	0.9235	0.7762
	Lat. (μ s)	4068.9	1202.9	1255.9	7170.3
OC-SVM baseline	AUROC	0.5994	0.7387	0.4458	0.7068
	FPR95	0.9253	0.8957	0.9919	0.9232
	Lat. (μ s)	2084.1	1912.5	842.64	4610.1
Mahalanobis baseline	AUROC	0.7351	0.7472	0.5204	0.7941
	FPR95	0.8569	0.8615	0.9465	0.7659
	Lat. (μ s)	2.3000	1.0221	0.1878	639.36
Within-layer ¹	AUROC	0.7550	0.7757	0.7042	0.8073
	FPR95	0.8184	0.9241	0.8742	0.7922
	Lat. (μ s)	6873.0	3785.7	1010.2	11271

Table 4: OOD detection across layers under high speed (mean of five runs; negligible SD); best method–layer per metric are **bold**.

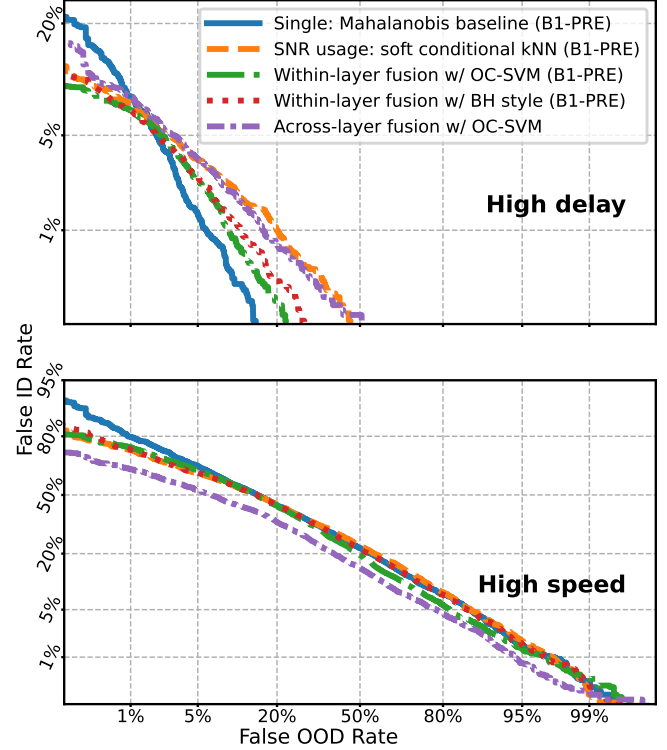


Figure 6: DET curves (single repeat; negligible SD) across scenarios, with selected detectors from Tables 3–4 and Supplemental G.

In OOD experiments, high-delay conditions are detected reliably, whereas high-speed remains challenging. Mahalanobis distance on mean activations is the strongest single detector; OC-SVM provides the most stable fusion; earlier layers outperform later ones; and SNR integration and classifier fusions yield only small, inconsistent AUROC gains—useful mainly when latency is acceptable and extremely low false OOD rate is preferred.

Overall, OOD detectors for neural receivers should be manifold-aware and leverage early-layer representations rather than clustering. Future work includes latent-variable models (e.g., DCAE, β -VAE) to better capture activation structure and improved integration of SNR or other channel-state information. These advances support reliable, trustworthy AI-native communication in 6G and beyond.

Acknowledgments

Marko Tuononen was partially supported by the Nokia Foundation. Ville Hautamäki was partially supported by Jane and Aatos Erkkö Foundation. We acknowledge the creators of the DeepRx framework for their foundational contributions.

References

- [3GPP, 2020] 3GPP. 5G; Study on channel model for frequencies from 0.5 to 100 GHz. Technical Report TR 38.901 V16.1.0, 2020.
- [3GPP, 2025] 3GPP. 5G; NR; Physical channels and modulation. Technical Specification TS 38.211 V19.1.0, 2025.
- [Aguilar *et al.*, 2025] Eduardo Aguilar, Bogdan Raducanu, and Petia Radeva. Multi-label out-of-distribution detection via evidential learning. In *ECCV Workshops*, 2025.
- [Bäuerle *et al.*, 2022] Alex Bäuerle, Daniel Jönsson, and Timo Ropinski. Neural activation patterns: Visual explainability of learned concepts. eprint arXiv:2206.10611 [cs.LG], 2022.
- [Bengio *et al.*, 2013] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, August 2013.
- [Bengio *et al.*, 2024] Yoshua Bengio et al. Managing extreme AI risks amid rapid progress. *Science*, 384(6698):842–845, 2024.
- [Benjamini and Hochberg, 1995] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. R. Stat. Soc., Ser. B (Methodol.)*, 57(1):289–300, 1995.
- [Bliss, 1934] C. I. Bliss. The method of probits. *Science*, 79(2037):38–39, 1934.
- [Blundell *et al.*, 2015] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *ICML*, 2015.
- [Bounsiar and Madden, 2014] Abdenour Bounsiar and Michael G. Madden. One-class support vector machines revisited. In *ICISA*, 2014.
- [Campello *et al.*, 2013] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *PAKDD*, 2013.
- [Cover and Hart, 1967] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE TIT*, 13(1):21–27, 1967.
- [Dehmollaian *et al.*, 2021] Eshagh Dehmollaian, Bernhard Eitzlinger, Núria Ballber Torres, and Andreas Springer. Using channel state information for physical tamper attack detection in OFDM systems: A deep learning approach. *IEEE Wireless Commun. Lett.*, 10(7):1503–1507, 2021.
- [Endres and Schindelin, 2003] D.M. Endres and J.E. Schindelin. A new metric for probability distributions. *IEEE Trans. Inf. Theory*, 49(7):1858–1860, 2003.
- [Euclid, 1956] Euclid. *The Thirteen Books of the Elements*. Dover Publications, 1956. Original work ca. 300 BCE.
- [Farhadi *et al.*, 2025] Hamed Farhadi et al. 6G AI-driven air interface — Hexa-X-II view. *IEEE Commun. Mag.*, 2025.
- [Goldsmith, 2005] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.
- [Han *et al.*, 2012] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. Elsevier, 3 edition, 2012.
- [Hendrycks *et al.*, 2019] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019.
- [Honkala *et al.*, 2021] Mikko Honkala, Dani Korpi, and Janne M. J. Huttunen. DeepRx: Fully convolutional deep learning receiver. *IEEE TWC*, 20(6):3925–3940, 2021.
- [Hoydis *et al.*, 2021] Jakob Hoydis, Fayçal Ait Aoudia, Alvaro Valcarce, and Harish Viswanathan. Toward a 6G AI-native air interface. *IEEE Commun. Mag.*, 59(5):76–81, 2021.
- [Hoydis *et al.*, 2023] Jakob Hoydis et al. Sionna: An open-source library for next-generation physical layer research. eprint arXiv:2203.11854 [cs.IT], 2023.
- [Islam *et al.*, 2023] Md Tauhidul Islam et al. Revealing hidden patterns in deep neural network feature space continuum via manifold learning. *Nature Communications*, 14(1):8506, 2023.
- [Jeffares and van der Schaar, 2025] Alan Jeffares and Michaela van der Schaar. Position: Not all explanations for deep learning phenomena are equally valuable. In *ICML, Position Paper Track*, 2025.
- [Katila and Tomppo, 2002] M. Katila and E. Tomppo. Stratification by ancillary data in multisource forest inventories employing k-nearest-neighbour estimation. *Can. J. For. Res.*, 32(9):1548–1561, 2002.
- [Kim *et al.*, 2018] Been Kim et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *ICML*, 2018.
- [Kornblith *et al.*, 2019] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, 2019.
- [Lakshminarayanan *et al.*, 2017] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.
- [Lan *et al.*, 2025] Guipeng Lan et al. Inner information analysis algorithm for deep neural network based on community. In *ICLR*, 2025.
- [Ledoit and Wolf, 2003] Olivier Ledoit and Michael Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10(5):603–621, 2003.
- [Lee *et al.*, 2018] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A unified framework for detecting out-of-distribution and adversarial samples. In *NeurIPS*, 2018.

- [Liu *et al.*, 2020] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020.
- [Liu *et al.*, 2022] Jinshan Liu, Taiwo Oyedare, and Jung-Min Park. Detecting out-of-distribution data in wireless communications applications of deep learning. *IEEE TWC*, 21(4):2476–2487, 2022.
- [Luostari *et al.*, 2025] Riku Luostari, Dani Korpi, Mikko Honkala, and Janne M. J. Huttunen. Adapting to reality: Over-the-air validation of AI-based receivers trained with simulated channels. In *IEEE WCNC*, 2025.
- [Magesh *et al.*, 2023] Akshayaa Magesh, Venugopal V. Veeravalli, Anirban Roy, and Susmit Jha. Principled out-of-distribution detection via multiple testing. *Journal of Machine Learning Research*, 24(378):1–35, 2023.
- [Mahalanobis, 1936] Prasanta Chandra Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.
- [Martin *et al.*, 1997] Alvin Martin *et al.* The DET curve in assessment of detection task performance. In *Eurospeech*, 1997.
- [McInnes and Healy, 2017] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *IEEE ICDMW*, 2017.
- [McInnes *et al.*, 2018] Leland McInnes, John Healy, and James Melville. Uniform manifold approximation and projection for dimension reduction. eprint arXiv:1802.03426 [stat.ML], 2018.
- [Müller and Hein, 2025] Maximilian Müller and Matthias Hein. Mahalanobis++: Improving OOD detection via feature normalization. In *ICML*, 2025.
- [Nilsson *et al.*, 2007] Jens Nilsson, Fei Sha, and Michael I. Jordan. Regression on manifolds using kernel dimension reduction. In *ICML*, 2007.
- [Papernot and McDaniel, 2018] Nicolas Papernot and Patrick D. McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *CoRR*, abs/1803.04765, 2018.
- [Parzen, 1962] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.
- [Paszke *et al.*, 2019] Adam Paszke *et al.* Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.
- [Pedregosa *et al.*, 2011] F. Pedregosa *et al.* Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.
- [Quiñonero-Candela *et al.*, 2008] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, editors. *Dataset Shift in Machine Learning*. The MIT Press, 2008.
- [Ren *et al.*, 2021] Yuwei Ren, Yu Zhang, Chenxi Hao, and Huilin Xu. Out-of-distribution detection and reporting for machine learning model deployment, 2021. Application No.: US18/258,314; Filed: 2021-02-24; Qualcomm Inc.
- [Räuker *et al.*, 2023] Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent AI: A survey on interpreting the inner structures of deep neural networks. In *IEEE SaTML*, 2023.
- [Salton *et al.*, 1975] Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
- [Schölkopf *et al.*, 2001] Bernhard Schölkopf *et al.* Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
- [Scott, 1992] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 1992.
- [Sharkey *et al.*, 2025] Lee Sharkey *et al.* Open problems in mechanistic interpretability. eprint arXiv:2501.16496 [cs.LG], 2025.
- [Sundar *et al.*, 2020] Vijaya Kumar Sundar *et al.* Out-of-distribution detection in multi-label datasets using latent space of β -vae. In *IEEE SPW*, 2020.
- [Tarekegn *et al.*, 2024] Adane Nega Tarekegn, Mohib Ullah, and Faouzi Alaya Cheikh. Deep learning for multi-label learning: A comprehensive survey. eprint arXiv:2401.16549 [cs.LG], 2024.
- [Tenenbaum *et al.*, 2000] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [Tuononen *et al.*, 2025] Marko Tuononen, Dani Korpi, and Ville Hautamäki. Interpreting deep neural network-based receiver under varying signal-to-noise ratios. In *ICASSP*, 2025.
- [Uzlaner *et al.*, 2025] Nicole Uzlaner, Tomer Raviv, Nir Shlezinger, and Koby Todros. Asynchronous online adaptation via modular drift detection for deep receivers. *IEEE TWC*, 24(5):4454–4468, 2025.
- [Wang *et al.*, 2021] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. Can multi-label classification networks know what they don’t know? In *Advances in Neural Information Processing Systems*, volume 34, pages 29074–29087, 2021.
- [Wang *et al.*, 2023] Kevin Ro Wang *et al.* Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *ICLR*, 2023.
- [Yang *et al.*, 2024] Jingkan Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *Int. J. Comput. Vis.*, 132(12):5635–5662, 2024.
- [Yuan *et al.*, 2024] Yue Yuan *et al.* Discriminability-driven channel selection for out-of-distribution detection. In *IEEE CVPR*, 2024.
- [Zeiler and Fergus, 2014] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[Zhang and Taneva-Popova, 2023] Dell Zhang and Bilyana Taneva-Popova. A theoretical analysis of out-of-distribution detection in multi-label classification. In *ACM ICTIR*, 2023.

Supplemental Material

Contents

A Neural Activation Pattern (NAP) Methodology for Concept Discovery	A-0
A.1 Introduction and Motivation	A-0
A.2 Related Work	A-0
A.3 Baseline Methodology	A-2
A.4 Proposed Methodology	A-3
A.5 Implementation Details	A-3
A.6 Experiments on Visual Object Recognition . .	A-4
A.7 Experiments on Radio Receiver	A-5
B Radio Receiver Model and Data	A-7
B.1 Data Simulation	A-7
B.2 Model Input	A-7
B.3 Model Architecture	A-7
B.4 Model Output and Loss Function	A-7
B.5 Model’s Predictive Performance	A-8
C Neural Unit Channel Parameter Identification (NUCPI) Methodology for Mechanistic Interpretability	A-9
C.1 Introduction	A-9
C.2 Methodology	A-9
C.3 Experiments on Radio Receiver	A-9
D Uniform Manifold Approximation and Projection (UMAP) Methodology for Manifold Analysis	A-16
D.1 Repetition Experiments on Radio Receiver . .	A-16
E Conditional k-NN Methodology for Out-Of-Distribution Detection	A-16
E.1 Hard Conditional Scheme	A-16
E.2 Soft Conditional Scheme	A-16
E.3 Parameter Selection	A-16
F Continuous Conformal P-Values Methodology for Out-Of-Distribution Detection	A-17
G Supplemental Experimental Results on Out-Of-Distribution Detection on Radio Receiver	A-17
G.1 Single-Detector Baselines	A-17
G.2 Effect of SNR-Aware Modeling	A-17
G.3 Within-Layer Fusion	A-17
G.4 Across-Layer Fusion	A-17

A Neural Activation Pattern (NAP) Methodology for Concept Discovery

A.1 Introduction and Motivation

Neural networks are increasingly used in high-stakes applications (e.g., [Jumper *et al.*, 2021])—yet their internal representations remain difficult to interpret ([Molnar, 2022]). This opacity complicates debugging, trust, and accountability, especially in domains where model behavior must align with physical or safety constraints ([Huyen, 2022; Adebayo *et al.*, 2020]). With regulatory frameworks like the EU AI Act mandating transparency in AI systems ([European Commission, 2024; 2019]), interpretable machine learning has become both a technical and societal necessity.

6G is a high-stakes application, projected to reach a \$68.7B global market by 2035 with a CAGR of 76.9% ([MarketsAndMarkets, 2024]), with the first commercial deployments expected by 2030 ([Nokia, 2025]). It promises extreme data rates, ultra-low latency, and dense connectivity, driving the need for advances in spectrum use, ultra-massive MIMO, and adaptive signal processing ([Nokia Bell Labs, 2024]). At the core is the physical layer, which sets the fundamental limits for capacity, efficiency, and reliability. Errors here propagate upward, making its performance critical. Traditional signal processing ([Goldsmith, 2005; Tse and Viswanath, 2005]) is reaching its limits—motivating a shift to data-driven models with better adaptability ([Hoydis *et al.*, 2021]).

We explore this shift in the context of data-driven physical-layer receivers, where emerging 6G architectures are expected to replace hand-crafted pipelines with end-to-end learned models ([Hoydis *et al.*, 2021]). Architectures like DeepRx ([Honkala *et al.*, 2021]) process raw physical-layer radio signals to decode transmitted information under dynamic, noisy channel conditions. Consequently, these models introduce a new challenge: interpretability ([Tuononen *et al.*, 2025]). Physical-layer inputs are non-semantic, combining transmitted data with unpredictable environmental effects (e.g., noise, fading, interference), making most standard interpretability methods ineffective ([Chergui *et al.*, 2025; Fiandrino *et al.*, 2022]).

To address this, we build on the Neural Activation Pattern (NAP) methodology ([Bäuerle *et al.*, 2022]), which clusters full-layer activation distributions to identify distributed, layer-level concepts. We extend NAP with improved normalization, distribution estimation, distance metrics, and varied cluster selection—as outlined in Figure 7—resulting in fewer unclustered samples and more stable clustering, and better out-of-distribution detection accuracy. To interpret the discovered clusters, we compare them with known physical parameters such as Signal-to-Noise Ratio (SNR). While no discrete cluster structure emerged, lower-dimensional embeddings revealed a continuous activation manifold aligned with SNR—suggesting neural receivers implicitly learn representations structured by core physical properties, even without explicit supervision.

A.2 Related Work

Interpreting neural networks has led to methods for analyzing learned representations. This section reviews concept

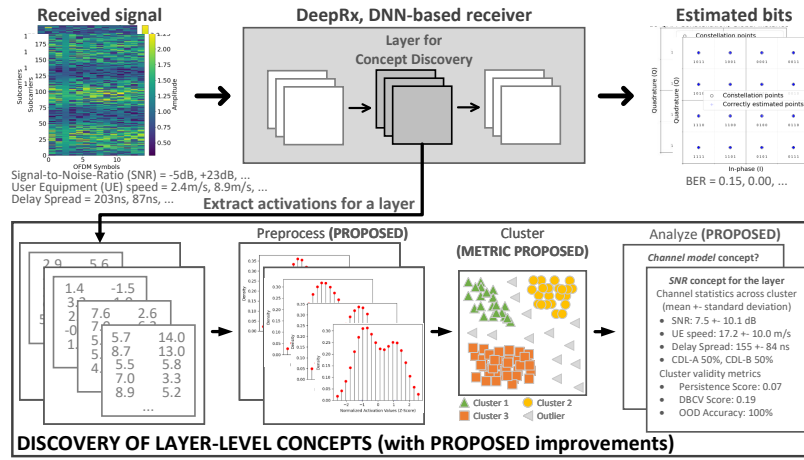


Figure 7: Layer-level concept discovery with proposed improvements highlighted in **PROPOSED**. The top part illustrates the DeepRx model processing received signals to estimate transmitted bits. The bottom part outlines the concept discovery pipeline operating at the layer level.

discovery approaches, including neuron-level and human-interpretable methods, clustering of distributed activation patterns, and clustering evaluation.

Concept Discovery in Neural Networks

Approaches to concept discovery often focus on either individual neurons or broader, human-interpretable concepts but may overlook layer-wide patterns emerging from distributed interactions.

Neuron-centric methods analyze individual neurons to identify features that excite specific units. Techniques include saliency maps ([Simonyan *et al.*, 2014]), which highlight input features relevant to activations; neuron maximization ([Erhan *et al.*, 2009]), which generates inputs that strongly activate specific neurons; activation histograms ([Yosinski *et al.*, 2015]), which summarize neuron behaviors; and input perturbation testing ([Fong and Vedaldi, 2017]), which evaluates the effects of input alterations on activations or outputs. Grad-CAM ([Selvaraju *et al.*, 2017]) produces class-specific visualizations by localizing important regions but focuses on localized activations. Network Dissection ([Bau *et al.*, 2017]) aligns neuron behaviors with predefined semantic categories (e.g., textures, objects) but fails to account for distributed patterns across layers.

Concept-based methods link input features to human-interpretable concepts ([Yeh *et al.*, 2022]). Testing with Concept Activation Vectors (TCAV) ([Kim *et al.*, 2018]) measures how user-defined concepts (e.g., stripes, wheels) influence predictions, while automated approaches ([Ghorbani *et al.*, 2019]) reduce reliance on manual definitions. These methods have been extended to tabular data ([Pendyala and Choi, 2022]), made more efficient ([Schmalwasser *et al.*, 2025]), and improved on a more global level [Zhang *et al.*, 2025], but they still rely on predefined or discovered concept sets and may miss broader layer-wide patterns or interactions.

Modern deep networks often learn distributed and poly-semantic features, where neurons respond to multiple unrelated stimuli or encode several sub-features ([Olah *et al.*, 2017; 2020; Bau *et al.*, 2017; 2020]). This highlights the limitations of focusing on individual neurons or predefined con-

cepts, as individual units may learn similar or no new concepts, or interact with each other to learn a concept ([Kim *et al.*, 2018; Ghorbani *et al.*, 2019; Bäuerle *et al.*, 2022]).

Neural Activation Pattern (NAP) methodology ([Bäuerle *et al.*, 2022]) addresses these limitations by identifying groups of inputs with similar activation profiles across a layer, capturing *layer-level concepts* without retraining the model. It clusters activation patterns to reveal distributed features and visualizes them via the NAP Magnifying Glass, which presents representative images, activation statistics, and metadata filtering for comparison across layers and models, as demonstrated in a visual object recognition task by [Bäuerle *et al.*, 2022]. NAP complements other interpretability methods and supports use cases such as TCAV-based testing or *ante-hoc training*, where interpretability is integrated during model development ([Gupta *et al.*, 2023; Ciravegna, 2025]). By leveraging predefined concepts, NAP has the potential to enhance both interpretability and model performance.

Clustering Methods for Concept Discovery

Clustering is an unsupervised learning task that partitions data such that instances within the same cluster are more similar to each other than to those in different clusters ([Jain and Dubes, 1988]). Selecting an effective clustering method is essential for concept discovery, as it plays a critical role in uncovering patterns within high-dimensional activation spaces.

Hierarchical clustering creates a tree-like partition hierarchy. Methods can be categorized into agglomerative (bottom-up; e.g., BIRCH [Zhang *et al.*, 1996]) and divisive (top-down; e.g., PNN [Virmajoki, 2004]) approaches. Agglomerative methods iteratively merge clusters based on linkage criteria such as single linkage (shortest distance), complete linkage (farthest distance), average linkage (mean distance), or Ward’s method (minimizing variance increase). Divisive methods recursively split clusters and can be monothetic (splitting by a single attribute) or polythetic (multiple attributes).

Partitional clustering divides data into non-overlapping clusters and methods can be categorized based on the frame-

work proposed by [Tuononen, 2006]. This categorization includes optimization-based methods (e.g., K-Means [Steinhaus, 1956; MacQueen, 1967; Lloyd, 1982]) that minimize an objective function such as intra-cluster variance; graph-theoretic methods (e.g., spectral clustering [Ng *et al.*, 2001]) that partition similarity graphs; density-based methods (e.g., DBSCAN [Ester *et al.*, 1996]) that identify dense regions; grid-based methods (e.g., CLIQUE [Agrawal *et al.*, 1998]), which discretize the feature space; and model-based methods (e.g., Gaussian Mixture Models [Dempster *et al.*, 1977], Self-Organizing Maps [Kohonen, 1990]) that fit statistical or neural network models.

HDBSCAN ([Campello *et al.*, 2013]) combines density-based and hierarchical clustering by using a minimum spanning tree for single-linkage agglomeration, effectively handling noise and irregular boundaries—ideal for high-dimensional data like neural activations with unknown cluster counts. Its accelerated variant, **HDBSCAN*** ([McInnes and Healy, 2017]), improves efficiency through optimized neighbor searches and pruning, making it well-suited for large-scale clustering tasks. The NAP framework ([Bäuerle *et al.*, 2022]) adopts this approach to reveal complex activation structures.

Evaluation of Cluster Validity

Evaluating cluster validity is essential in clustering-based concept discovery, as it determines whether the identified groupings reflect meaningful structure in the data. While many validation methods exist ([Hassan *et al.*, 2024]), choosing the right one is nontrivial: the utility of a clustering often depends highly on the specific task and domain. As [Jain, 2010] noted, *clustering is in the eye of the beholder*.

Cluster validation criteria fall into three categories ([Jain and Dubes, 1988]): external (compare to ground truth labels, e.g., Rand index [Rand, 1971], Jaccard index [Jaccard, 1901]); internal (assess intrinsic structure, e.g., Silhouette score [Rousseeuw, 1987], Davies–Bouldin score [Davies and Bouldin, 1979]); and relative (compare multiple clustering results, e.g. [Halkidi *et al.*, 2001]). Key indicators of cluster quality are ([Jain and Dubes, 1988]) *compactness* (minimizing distances within clusters) and *separation* (maximizing distances between clusters). For density-based clustering algorithms, internal validation criteria should favor high-density clusters with low-density separation. Two widely used internal indices for density-based clustering are CDbw ([Halkidi and Vazirgiannis, 2008]), which uses boundary points to assess structure, and DBCV ([Moulavi *et al.*, 2014]), preferred for its more accurate density estimation and boundary-free formulation.

A.3 Baseline Methodology

The Neural Activation Pattern (NAP) methodology ([Bäuerle *et al.*, 2022]) addresses the limitations of focusing on individual neurons or predefined concepts—as described in Supplemental A.2—by identifying groups of inputs with similar activation profiles across a neural network layer. As a model-specific method, it relies on inner model details and provides post-hoc layer-level concept-based interpretability. The key steps in the NAP methodology are as follows:

- **Data Extraction:** Given the layer of interest l for neural network model n and the set of inputs $X = \{x_i \mid i \in [1, \dots, N]\}$ to the model, the activation output $A^l = \{a_i^l \mid i \in [1, \dots, N]\}$ of the sub-network n^l is extracted, denoted as $A^l = n^l(X)$. For convolutional layers, each a_i^l is a collection of matrices (one per convolutional channel). There are no formal requirements on the input X beyond those set by the neural network model. It can be the training set, test set, subsets thereof, or entirely separate data sets.
- **Location Disentanglement:** For convolutional layers, spatial information in activation values can act as a confounding factor ([Bäuerle *et al.*, 2022]). To address this, activations of each convolutional channel are aggregated using statistical measurements, either maximum value (highest activation), minimum and maximum values (range of activations), mean value (average amount of activations), or mean and standard deviation (average amount and spread of activations). The disentangled activations are denoted as $\tilde{A}^l = \{\tilde{a}_i^l \mid i \in [1, \dots, N]\}$. Aggregating activations removes spatial information and reduces computational cost, making the method scalable for larger models.
- **Normalization:** The aggregated activation outputs are normalized by their maximum absolute activation across all inputs X . This *max-abs scaling* is given by Equation (24), where $\tilde{a}_{i,u}^l$ represents the disentangled activations for layer l and convolutional channel u with input data x_i . Channels that can only activate positively are normalized between 0 and 1, while channels with negative activations are normalized between -1 and 1.

$$\hat{a}_{i,u}^l = \frac{\tilde{a}_{i,u}^l}{\max(|\tilde{a}_u^l(X)|)} \quad (24)$$

- **Clustering Analysis:** The aggregated and normalized activation outputs $\hat{A}^l$ are clustered, forming a clustering $C^l = \{c_j^l \mid j \in [1, \dots, M]\}$ with M clusters, using the HDBSCAN* ([McInnes and Healy, 2017]). The clustering process employs the Euclidean distance metric, selects clusters from the leaves of the cluster tree, and uses the minimum cluster size of five. Each cluster $c_j^l \in C^l$ groups together a set of inputs $x_i \in X$ which produce similar activations for layer l . Clusters are mutually exclusive but not collectively exhaustive, meaning that no input x_i can belong to more than one cluster, but some inputs x_i may not belong to any cluster (i.e., they are classified as noise).
- **Post-Clustering Analysis:** HDBSCAN* provides information about the persistence of the clusters. *Persistence score* reflects the stability or strength of a cluster; higher persistence meaning cluster being more well-defined and robust. Clusters are sorted based on their persistence scores, which help identify the most stable and reliable clusters. Clusters are considered to be just “raw mathematical groupings,” while NAPs in general are clusters that reflect features learned by the network. Concepts are interpretable NAPs that align with human understanding.

A.4 Proposed Methodology

We refine the original NAP methodology ([Bäuerle *et al.*, 2022]; detailed in Supplemental A.3) to improve the stability and generalizability of the clustering results. In addition, we propose a principled approach to interpreting and evaluating clustering in the context of neural network-based systems, with a focus on radio receivers. Our refinements are shown in Figure 8 and described below:

Proposed Data Extraction: Follows the same procedure as in the baseline NAP methodology, where we extract activation outputs from a specified layer l of a neural network n for a given input set $X = \{x_i \mid i \in [1, \dots, N]\}$. This yields a set of activations $A^l = \{a_i^l = n^l(x_i)\}$, where each a_i^l contains the output feature maps from layer l . In convolutional layers, these are structured as per-channel activation matrices, used in subsequent processing.

Proposed Normalization: We rescale the activations of each convolutional channel to mean 0 and standard deviation 1 (known as *Z-Score Normalization*), making them more suitable for subsequent distribution estimation by removing scale differences across channels and centering the data:

$$\hat{a}_{i,u}^l = \frac{a_{i,u}^l - \mu_u^l}{\sigma_u^l}, \quad (25)$$

where $\hat{a}_{i,u}^l$ represents the normalized activation, $a_{i,u}^l$ is the original activation, μ_u^l is the mean, and σ_u^l is the standard deviation of the activation output of channel u in layer l across all data samples in X .

Proposed Location Disentanglement: We employ a form of non-parametric density estimation technique, known as *Kernel Density Estimator* (KDE; [Rosenblatt, 1956; Parzen, 1962]), to aggregate the normalized activations per convolutional channel. For KDE, the estimated *Probability Density Function* (PDF) for channel u in layer l is given by

$$\hat{f}_{i,u}^l(a) = \frac{1}{QH} \sum_{q=1}^Q K\left(\frac{a - \hat{a}_{i,u,q}^l}{H}\right), \quad (26)$$

where $K(\cdot)$ is the *kernel function* (e.g., Gaussian), H is a smoothing parameter called the *bandwidth*, and Q is the number of activation values on channel u in layer l . We use *Scott's Rule* ([Scott, 1992]) for selecting the bandwidth, providing a standard bias-variance tradeoff:

$$H = 1/\sqrt[5]{Q}. \quad (27)$$

We sample the estimated PDF (Equation 26) over a Z-score range covering $P\%$ of the normalized activation mass to reduce outlier influence:

$$z_{\min} = \Phi^{-1}\left(\frac{1 - \frac{P}{100}}{2}\right), \quad z_{\max} = \Phi^{-1}\left(1 - \frac{1 - \frac{P}{100}}{2}\right), \quad (28)$$

where $\Phi^{-1}(\cdot)$ is the *Inverse Cumulative Distribution Function* (ICDF) of the standard normal distribution. The range is then discretized into R uniformly spaced points for channel u in layer l to produce a fixed-length representation, as follows:

$$\hat{\mathbf{f}}_{i,u}^l = \left[\hat{f}_{i,u}^l \left(z_{\min} + r \cdot \frac{z_{\max} - z_{\min}}{R - 1} \right) \right]_{r=0}^{R-1}. \quad (29)$$

Proposed Clustering Analysis: We cluster the sampled and concatenated estimated density values

$$\hat{\mathbf{g}}_i^l = \left[\hat{f}_{i,1,1}^l, \hat{f}_{i,1,2}^l, \dots, \hat{f}_{i,1,R}^l, \hat{f}_{i,2,1}^l, \hat{f}_{i,2,2}^l, \dots, \hat{f}_{i,U,R}^l \right]_{u=1}^U, \quad (30)$$

where U is the number of channels in layer l , using HDBSCAN* with the *Jensen-Shannon distance metric* ([Lin, 1991; Endres and Schindelin, 2003]), which is symmetric, bounded, and well-suited for comparing probability distributions such as those estimated by KDE in the previous step. The Jensen-Shannon distance D_{JS}^l between activations for inputs x_i and x_j for all channels in layer l is

$$D_{JS}^l(x_i, x_j) = \frac{1}{\sqrt{2}} \sqrt{\sum_{k=1}^{U \cdot R} \left(\hat{\mathbf{g}}_{i,k}^l \log \frac{2 \cdot \hat{\mathbf{g}}_{i,k}^l}{\hat{\mathbf{g}}_{i,k}^l + \hat{\mathbf{g}}_{j,k}^l} + \hat{\mathbf{g}}_{j,k}^l \log \frac{2 \cdot \hat{\mathbf{g}}_{j,k}^l}{\hat{\mathbf{g}}_{i,k}^l + \hat{\mathbf{g}}_{j,k}^l} \right)} \quad (31)$$

and we use this as a distance metric in clustering.

Proposed Post-Clustering Analysis: We evaluate clustering quality using two internal metrics and one application-specific metric. As internal metrics, we use the *persistence score* ([Campello *et al.*, 2013]), which reflects the stability or robustness of a cluster (with higher values indicating more well-defined clusters), and the *Density-Based Clustering Validation (DBCV) index* ([Moulavi *et al.*, 2014]), which is designed to assess density-based clustering algorithms like HDBSCAN by balancing cluster compactness and separation. As an application-specific metric, we use *Out-of-Distribution (OOD) detection* (see, e.g. [Zhu *et al.*, 2025]), assuming that OOD samples fall outside the learned cluster structure. This allows clustering to act as a proxy for distributional mismatch. In this setting clustering serves not just as an explanation method but as a model debugging tool—similar to how [Adebayo *et al.*, 2020] frame OOD behavior as test-time contamination. To interpret the resulting clusters, we use the channel parameters of the inputs (e.g., SNR) to summarize and explain structure.

A.5 Implementation Details

Although [Bäuerle *et al.*, 2022] mention the availability of their NAP methodology as a Python package, we were unable to locate a public implementation. Consequently, we reimplemented both their baseline approach (Supplemental A.3) and our proposed enhancements (Supplemental A.4) using various Python libraries from which the most essential can be found from Listing 1.

In our methodology, the density is estimated using a Kernel Density Estimator (KDE) with a Gaussian kernel K . To obtain a compact yet informative representation, the estimated probability density function $\hat{f}_{i,u}^l$ (Equation 9) is sampled at $R = 10$ evenly spaced Z-scores (Equation 29) over the interval $[z_{\min}, z_{\max}] = [-2.576, 2.576]$, which contains $P = 99\%$ of the mass of a standard normal distribution (Equation 28). This one-point-per-decade grid (step $\Delta \approx 0.515$) yields a ten-dimensional feature vector per unit—an order-of-magnitude increase in information over a single-mean sum-

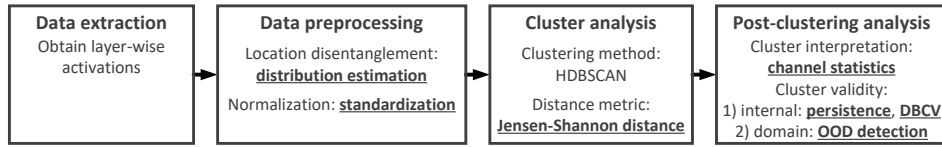


Figure 8: Discovering layer-level concepts by applying NAP methodology with our proposed improvements (underlined): extracting, preprocessing, and clustering activations—followed by interpreting and evaluating the clustering results.

mary in baseline methodology, while remaining compact enough for downstream modeling or tabular presentation.

```

h5py==3.13.0
hdbscan==0.8.40
joblib==1.4.2
matplotlib==3.10.1
numba==0.61.0
numpy==2.1.3
pandas==2.2.3
scikit-learn==1.6.1
scipy==1.15.2
sionna==0.10.0
torch==2.6.0
torchvision==0.21.0
umap-learn==0.5.7
tensorflow==2.6.4
  
```

Listing 1: Most essential Python requirements used in `requirements.txt` format.

A.6 Experiments on Visual Object Recognition

We applied the NAP methodology using the average amount of activations as the baseline, as suggested by [Bäuerle *et al.*, 2022] to be the most suitable choice for extracting a larger number of NAPs. We then compared this baseline with our proposed methodology—outlined in Figure 8 and detailed in Supplemental A.4—and additionally studied the effect of HDBSCAN parameters.

Experimental Setup

We employed a ResNet-50 [He *et al.*, 2016] convolutional neural network pre-trained on the ImageNet-1K dataset—namely IMAGENET1K_V2 weights from TorchVision [Marcel and Rodriguez, 2010]—to perform visual object classification. To align with the experimental protocol of [Bäuerle *et al.*, 2022], we used a randomly selected 1% subset (12,832 images) of the ImageNet ILSVRC2012 training set ([Rusakovsky *et al.*, 2015]). As part of the setup—and following their methodology—we extracted feature activations from the Conv4 and Conv5 stages of the network—corresponding to the `layer3` and `layer4` modules in TorchVision’s implementation—which produce intermediate activations of size $[1024 \times 14 \times 14]$ and $[2048 \times 7 \times 7]$, respectively, when processing 224×224 resolution images.

We used our own PyTorch-based implementation of both the NAP baseline and our proposed methodology; implementation details are provided in Supplemental A.5.

Results

We first evaluated the impact of dataset size using the HDBSCAN parameters recommended by [Bäuerle *et al.*, 2022]—specifically, the Leaves of Tree (Leaf) as the cluster selection

criteria with a minimum cluster size of 5. As shown in Figure 9, the baseline method resulted in 73–95% of samples remaining unclustered. Our proposed methodology did not mitigate this issue. A high proportion of non-clustered activations is problematic, as it hinders the discovery of meaningful layer-level concepts, reducing interpretability. It also undermines confidence in the clustering structure and impairs downstream tasks such as Out-of-Distribution (OOD) detection, which depend on well-formed, dense regions in activation space.

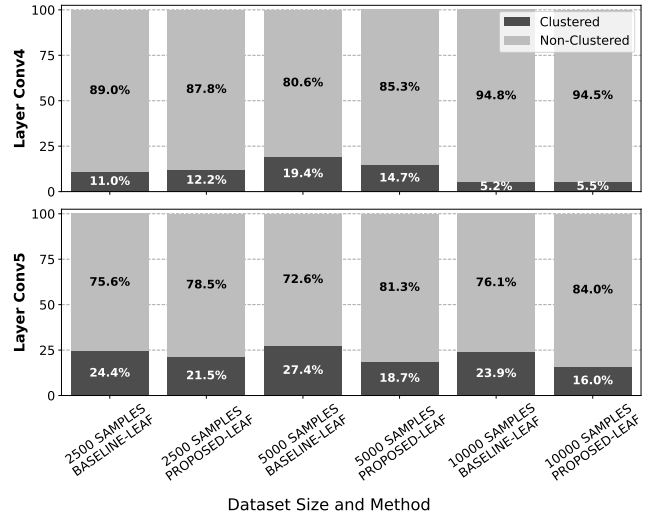


Figure 9: Percentage of clustered and non-clustered samples as a function of sample size and aggregation method with minimum cluster size 5 using ImageNet subset with ResNet-50 model.

Notably, [Bäuerle *et al.*, 2022] do not report any analysis of unclustered or outlier samples, which we found to be a recurring issue in the visual object recognition setting as well. Our results show that increasing dataset size leads to more clusters being discovered—consistent with findings in [Bäuerle *et al.*, 2022]—but the mean persistence score remains relatively stable.

Next, we investigated the effect of HDBSCAN parameters on clustering outcomes. Figure 10 shows that using Excess of Mass (EoM) as the cluster selection criteria consistently reduces the number of unclustered samples across settings. Smaller minimum cluster sizes (e.g., 3) also lead to fewer unclustered samples. With the updated cluster selection criteria (EoM), our methodology generally clusters more samples—except for Conv5 with a minimum cluster size of 5. The number of clusters behaves differently depending on the cluster

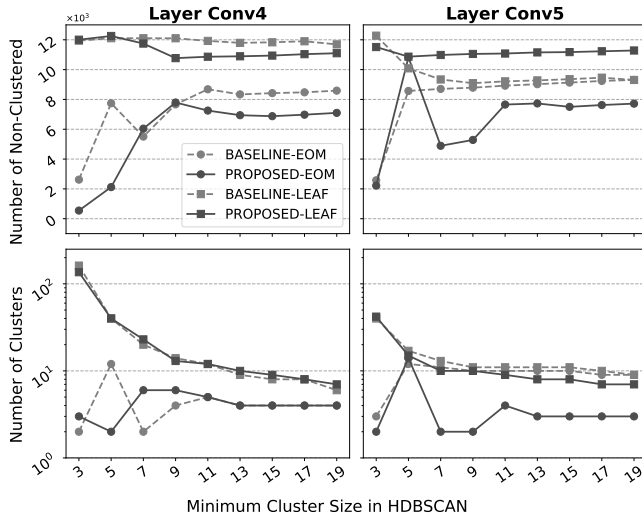


Figure 10: Number of non-clustered samples and clusters by aggregation and cluster selection method as function of minimum cluster size using ImageNet subset with ResNet-50 model.

selection method: with Leaf, the number of clusters decreases as the minimum cluster size increases, likely due to aggressive pruning of small clusters; with EoM, the number remains more stable, as it prioritizes persistence over strict pruning.

We explore the impact of dataset size on visual object recognition using the ResNet50 architecture with pretrained weights and activations from the Conv4 and Conv5 stages of the network, as described in Supplemental A.6. Figures 11 and 12 illustrate how varying the dataset size influences the number of clusters and the mean persistence score across different layers and methodologies.

A.7 Experiments on Radio Receiver

This section provides supplemental results for the radio receiver experiments. The experimental setup follows the main paper (Section 4.1). The central results are reported in the main paper (Section 4.2), while additional analyses are presented here.

Impact of Dataset Size

We study the effect of dataset size on clustering behavior in the DeepRx architecture, using activations from multiple network stages as described in Supplemental B. Figures 13 and 14 show how varying the sample size affects the number of clusters and the mean persistence score across layers and methods.

Cluster Statistics and DBCV Scores

This section reports detailed cluster statistics and DBCV scores for the example shown in Fig. 5. Baseline results are summarized in Tables 5 and 7, while the corresponding results for the proposed method are given in Tables 6 and 7. The consistently low DBCV values indicate weak cluster separability for both methods, supporting the interpretation of a continuous activation manifold rather than discrete clusters.

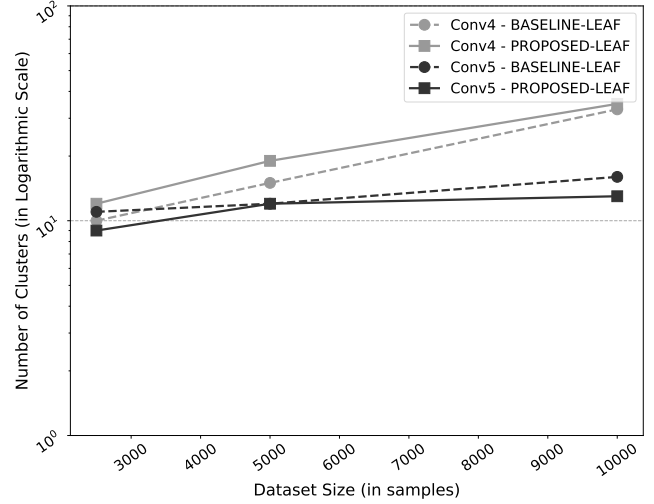


Figure 11: Number of clusters per layer and methodology as function of sample size with minimum cluster size 5 using ImageNet subset with ResNet-50 model.

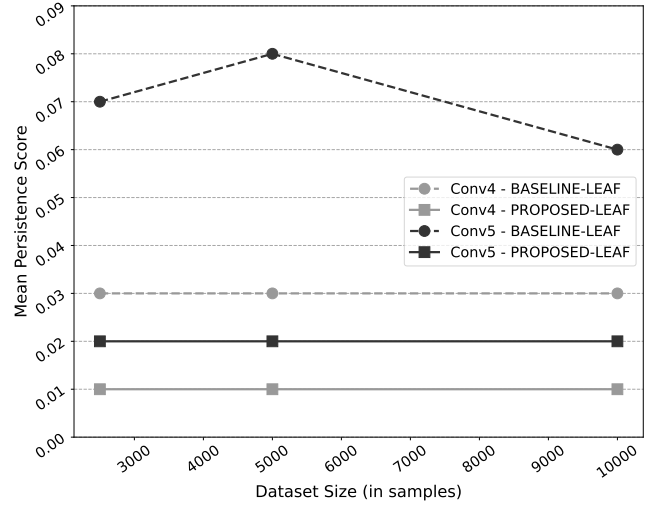


Figure 12: Mean persistence score per layer and methodology as function of sample size with minimum cluster size 5 using ImageNet subset with ResNet-50 model.

Table 5: Cluster statistics for the **baseline** methodology in the example shown in Fig. 5. Values are mean $\pm$ standard deviation.

Cluster	Size	Persist.	SNR [dB]	UE speed [km/h]	Delay [ns]	SER	Dominant CDL
0	7	0.01	7.6 ± 12.0	71.8 ± 31.6	112 ± 72	0.33 ± 0.34	A (100%)
1	38373	0.12	7.5 ± 10.1	62.1 ± 36.0	156 ± 84	0.30 ± 0.31	A/E (49/51)
2	6	0.00	10.1 ± 9.2	58.0 ± 24.5	156 ± 89	0.26 ± 0.23	A (100%)

Table 6: Cluster statistics for the **proposed** methodology in the example shown in Fig. 5. Values are mean $\pm$ standard deviation.

Cluster	Size	Persist.	SNR [dB]	UE speed [km/h]	Delay [ns]	SER	Dominant CDL
0	14	0.00	6.5 ± 11.3	47.2 ± 32.2	145 ± 75	0.32 ± 0.32	A/E (57/43)
1	5	0.00	10.0 ± 10.0	90.2 ± 25.2	222 ± 59	0.26 ± 0.32	E (60%)
2	5	0.00	4.5 ± 9.6	63.3 ± 17.0	198 ± 31	0.38 ± 0.31	A (60%)
3	6	0.00	10.4 ± 10.4	68.9 ± 43.6	119 ± 80	0.27 ± 0.30	E (67%)
4	10	0.00	11.2 ± 8.9	76.1 ± 37.5	157 ± 72	0.19 ± 0.29	E (60%)
5	6	0.01	8.9 ± 7.4	42.4 ± 29.4	186 ± 98	0.23 ± 0.24	A (67%)
6	33063	0.07	7.5 ± 10.1	62.0 ± 36.0	155 ± 84	0.30 ± 0.31	A/E (50/50)
7	6	0.00	4.1 ± 10.5	55.4 ± 27.4	139 ± 70	0.42 ± 0.32	E (67%)

Table 7: DBCV clustering quality scores for the example in Fig. 5.

Method	Overall DBCV	Best cluster	Worst cluster
Baseline	0.055	0.259	0.058
Proposed	0.014	0.187	0.017

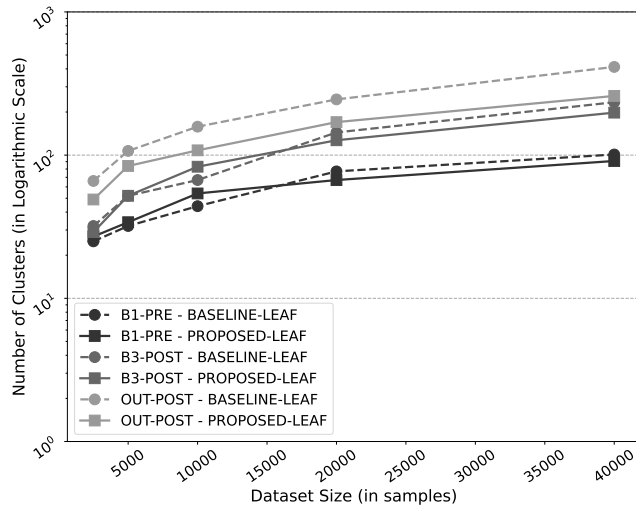


Figure 13: Number of clusters per layer and methodology as function of sample size with minimum cluster size 5 using link-level simulations with DeepRx model.

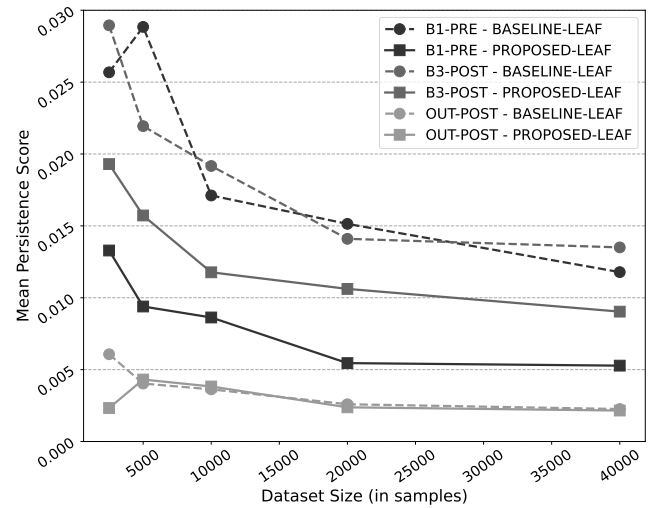


Figure 14: Mean persistence score per layer and methodology as function of sample size with minimum cluster size 5 using link-level simulations with DeepRx model.

B Radio Receiver Model and Data

DeepRx ([Honkala *et al.*, 2021]) is a fully convolutional deep learning model that jointly performs the tasks of channel estimation, equalization, and demapping, as illustrated in Figure 15. Moreover, the model is designed to be compliant with 5G NR specifications, capable of estimating the log-likelihood ratio (LLR), or the uncertainty, of each received bit so that it can be fed into an LDPC decoder. The model supports different demodulation reference signal (DMRS) or pilot configurations included in the 5G NR specifications ([Honkala *et al.*, 2021]).

B.1 Data Simulation

The data are randomly generated using a 5G-compliant physical uplink shared channel (PUSCH) simulator, powered by NVIDIA Sionna ([Hoydis *et al.*, 2023]), and configured with the parameters listed in Table 1. The simulation process is presented in Figure 15 and consists of three main stages:

- **Transmitter:**
 1. Generate random bits.
 2. Encode the bits using a 5G-compliant LDPC encoder.
 3. Map the encoded bits to symbols and distribute them across available physical resource blocks (PRBs) within the transmission time interval (TTI).
 4. Insert pilot symbols, or DMRS, into the specified subcarriers.
 5. Convert the PRBs to an OFDM waveform with 14 OFDM symbols per TTI by performing an inverse Fourier transform (IFFT).
 6. Add a CP to the beginning of each OFDM symbol to mitigate inter-symbol interference.
- **Channel:**
 1. Pass the waveform through a channel model.
 2. Randomly select the maximum Doppler shift and root mean square (RMS) delay spread for each channel realization.
 3. Add noise to the signal.
- **Receiver:**
 1. Receive the signal.
 2. Remove the CP.
 3. Perform FFT on each OFDM symbol to convert them into the frequency domain.
 4. Feed the frequency-domain signal of 14 OFDM symbols into the DeepRx model.
 5. Decode the LLR output using a 5G-compliant LDPC decoder.

B.2 Model Input

To maximize the network’s utilization of the input data, the model is provided with the received signal represented in the frequency domain over the entire transmission time interval (TTI). Denoting $S = 14$ as the number of OFDM symbols, $F = 192$ as the number of subcarriers, and $N_r = 2$ as the number of receive antennas, the input consists of the following three arrays:

1. Received signal after the FFT (data and received pilot symbols), denoted by $\mathbf{Y} \in \mathbb{C}^{S \times F \times N_r}$.
2. Pilot reference symbols within the received signal $\mathbf{Y}$ in both frequency and time, denoted by $\mathbf{X}_p \in \mathbb{C}^{S \times F \times 1}$.
3. Raw channel estimate which is the element-wise product of the received signal $\mathbf{Y}$ and complex conjugate of the transmitted pilot symbols $\mathbf{X}_p$, denoted by $\mathbf{H}_r \in \mathbb{C}^{S \times F \times N_r}$.

B.3 Model Architecture

The architecture of the DeepRx model, based on the work by [Honkala *et al.*, 2021] and [Korpi *et al.*, 2021], is shown in Table 8. It consists of multiple pre-activation ResNet blocks—an extension of the ResNet architecture by [He *et al.*, 2016]—and several convolutional layers. The pre-activation ResNet block, illustrated in Figure 16, is composed of two consecutive groups, each consisting of a batch normalization layer, a Rectified Linear Unit (ReLU) layer, and a depthwise separable convolution layer ([Chollet, 2017]). In each depthwise separable convolution block, the depth multiplier value is set to 1, meaning the number of output channels in the depthwise convolution step is the same as the input. The data flow direction is depicted in Figure 17. Certain architectural details have been omitted as they are not essential to the core contributions of our work.

The layers of interest in the paper are the ResNet blocks B1 through OUT, as illustrated in Figure 17. We focus particularly on the activations after their ReLU layers. Since each ResNet block contains two ReLU layers, we denote them as **PRE** and **POST** accordingly. For example, when referring to the second ReLU layer of ResNet block B3, we use the notation **B3-POST**.

B.4 Model Output and Loss Function

Since DeepRx ([Honkala *et al.*, 2021]) is designed to support multiple Quadrature Amplitude Modulation (QAM) schemes in 5G, including QPSK, 16-QAM, 64-QAM, and 256-QAM, which are hierarchically related (see Figure 18).

The model output is designed so that the same output bits, or log-likelihood ratios (LLRs), correspond to the same part of the constellation space, regardless of the modulation scheme used. For example, for an individual symbol, the first two output bits describe the complex quadrant of the symbol, called the 1st-order point; the next two bits define the quadrant within the previously identified quadrant, called the 2nd-order point. The process continues until all the bits are utilized. The visualization of this hierarchical process is provided in Figure 20.

In practice, the output length for each symbol is set according to the highest supported modulation order. Depending on the QAM scheme, the output is masked accordingly so that unnecessary parts of the output array are not considered. For example, if the highest modulation order is 256-QAM, each symbol in the model output is represented as an array with 8 elements for 8 bits. If a lower modulation order like 16-QAM, which requires 4 binary values, is used, the last 4 bits, or elements, are masked. The masking process is illustrated in Figure 21.

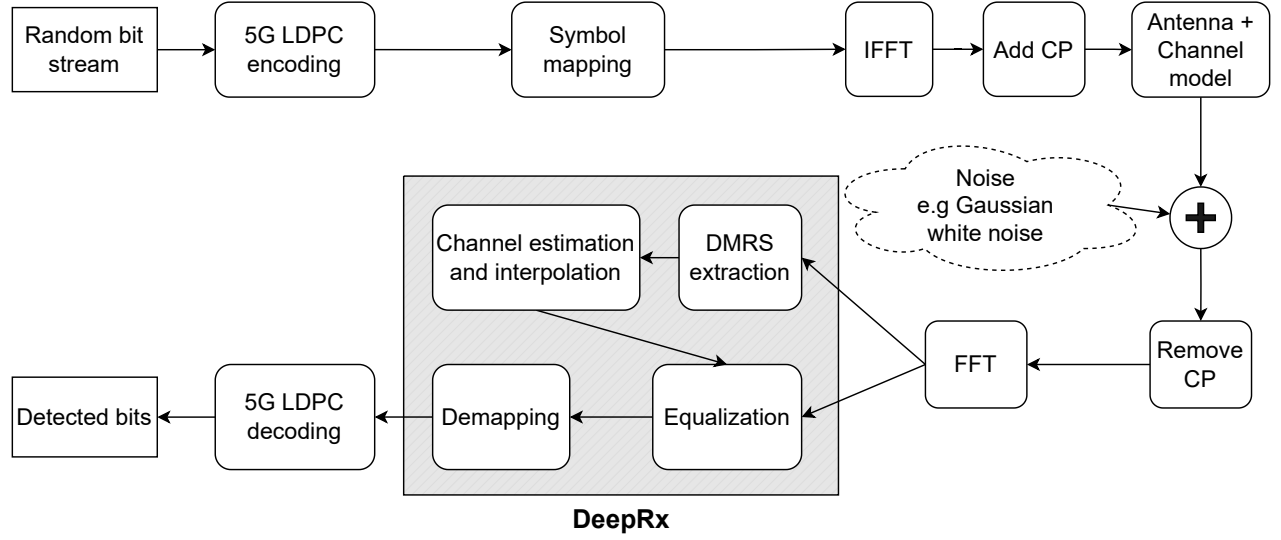


Figure 15: Illustration of the PUSCH simulator (adapted from [Honkala *et al.*, 2021]), where DeepRx is used to replace channel estimation, equalization, and demapping blocks.

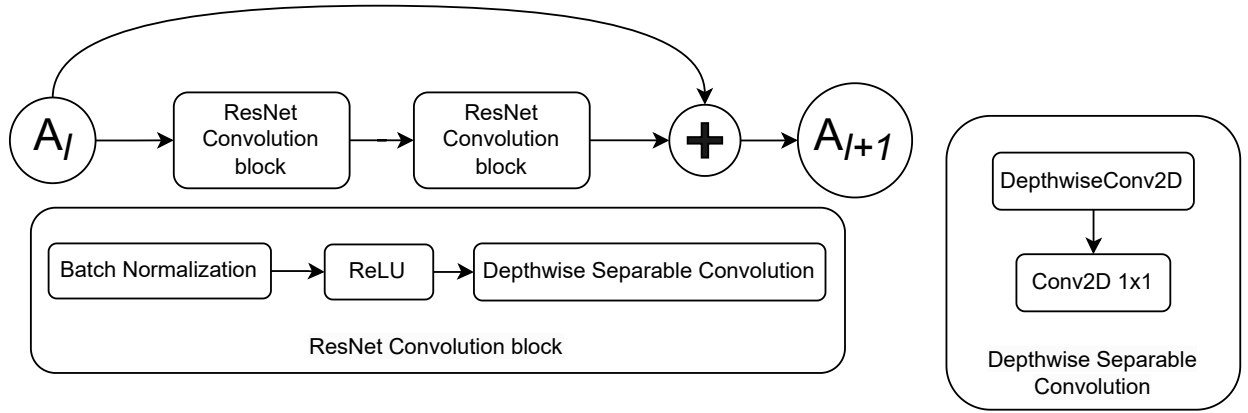


Figure 16: DeepRx's pre-activation ResNet block (adapted from [Honkala *et al.*, 2021]).

Since the bit prediction can be framed as a binary classification problem, and the network output represents the LLR of the transmitted bits, the binary cross-entropy (CE) loss is used as the loss function for the DeepRx model:

$$\text{CE}(\theta) = -\frac{1}{\#DB} \sum_{(i,j) \in \mathcal{D}} \sum_{l=0}^{B-1} \left(b_{ijl} \log(\hat{b}_{ijl}) + (1 - b_{ijl}) \log(1 - \hat{b}_{ijl}) \right), \quad (32)$$

where $\#D$ is the number of resource elements carrying data; B is the number of bits in the constellation used ($B = 4$ for 16-QAM); b_{ijl} is the label bit; and $\hat{b}_{ijl}$ is the probability that bit b_{ijl} equals one, calculated using the sigmoid function:

$$\hat{b}_{ijl} = \text{sigmoid}(L_{ijl}) = \frac{1}{1 + e^{-L_{ijl}}}. \quad (33)$$

B.5 Model's Predictive Performance

The DeepRx model demonstrates competitive performance in terms of uncoded Bit Error Rate (BER) across varying Signal-to-Interference-plus-Noise-Ratio (SINR) levels, as seen in Figure 19. It closely matches the baseline performance with a known channel, particularly at low SINR values. Notably, it outperforms the baseline with one pilot in higher SINR scenarios, which aligns with the findings reported in [Honkala *et al.*, 2021].

Table 8: The DeepRx CNN ResNet architecture, with all convolutional layers having a filter size of (3, 3). The ResNet block is illustrated in Figure 16, and the overall data flow is depicted in Figure 17.

Layer	Type	Number of filters	Dilation size (S, F)	Output Shape
Input 1 $\mathbf{X}_p \in \mathbb{C}$	TX Pilot			(14, 192, 1)
Input 2 $\mathbf{H}_r \in \mathbb{C}$	Raw channel estimate			(14, 192, 2)
Input 3 $\mathbf{Y} \in \mathbb{C}$	RX data			(14, 192, 2)
Pre-DeepRx	Neural preprocessing			(14, 192, 20)
Convolutional layer A0	2D convolution	40	(1,1)	(14, 192, 40)
ResNet block A1	Depthwise separable convolution	40	(1,1)	(14, 192, 40)
ResNet block A2	Depthwise separable convolution	40	(2,3)	(14, 192, 40)
ResNet block A3	Depthwise separable convolution	32	(3,6)	(14, 192, 32)
Convolutional layer B0	2D convolution	64	(1,1)	(14, 192, 64)
ResNet block B1	Depthwise separable convolution	64	(1,1)	(14, 192, 64)
ResNet block B2	Depthwise separable convolution	64	(2,4)	(14, 192, 64)
ResNet block B3	Depthwise separable convolution	32	(3,8)	(14, 192, 32)
ResNet block B4	Depthwise separable convolution	32	(2,4)	(14, 192, 32)
ResNet block B5	Depthwise separable convolution	32	(1,1)	(14, 192, 32)
ResNet block OUT	Depthwise separable convolution	8	(1,1)	(14, 192, 8)
LLR Output $\mathbf{L}$, see Section B.4	Output			(14, 192, 8)

C Neural Unit Channel Parameter Identification (NUCPI) Methodology for Mechanistic Interpretability

C.1 Introduction

We present our replication of the experiments conducted by [Tuononen *et al.*, 2025], employing delay spread (see Appendix C.3) and user equipment (UE) speed (see Appendix C.3) as target parameters. This approach diverges from the original study, which utilized Signal-to-Noise Ratio (SNR) as the channel parameter of interest. Notably, [Tuononen *et al.*, 2025] suggested that their method could be applicable to other channel parameters, which we explore here.

C.2 Methodology

We refer to the methodology proposed in [Tuononen *et al.*, 2025] as *Neural Unit Channel Parameter Identification (NUCPI)*, wherein an **explainer model** is trained to predict a given channel parameter based on intermediate activations from a **performer model**—in our case, the DeepRx model (see Supplemental B). We interpret the predictive performance of the explainer model—measured using Mean Squared Error (MSE), as suggested by [Tuononen *et al.*, 2025]—as an indicator of how much information specific internal components (e.g., layers or channels) of the performer contain about the given channel parameter. This enables insights into the model’s internal behavior.

These insights could potentially be used to identify redundant or over-specialized components, guide model pruning [Han *et al.*, 2016], or adapt architectures to better handle specific channel conditions [Cortes *et al.*, 2017]. While these applications are not experimentally explored in this work, the interpretability provided by NUCPI opens promising directions for future research in model compression and adaptive architecture design.

C.3 Experiments on Radio Receiver

NUCPI Analysis of Delay Spread

We performed NUCPI analysis of delay spread using both layerwise and channelwise (for layer B1-PRE) perspectives. As shown in the layerwise (Figure 22) and channelwise (Figure 23) test losses of the explainer models, predictive performance is consistently poor across all units. Upon closer examination, it is evident that the explainer models are over-fitting: while training loss decreases steadily, evaluation loss plateaus or increases, indicating limited generalization to unseen data [Goodfellow *et al.*, 2016]. This is illustrated in training curves for both the layerwise (B1-PRE in Figure 24; OUT-POST in Figure 25) and channelwise (channels 7 and 12 on B1-PRE, in Figures 26 and 27, respectively) explainer models.

The estimates produced by the explainer models (see Figure 28 for layerwise, and Figures 30 and 29 for channelwise) show that the models tend to predict values near the mean of the delay spread distribution. This pattern occurs both in lay-

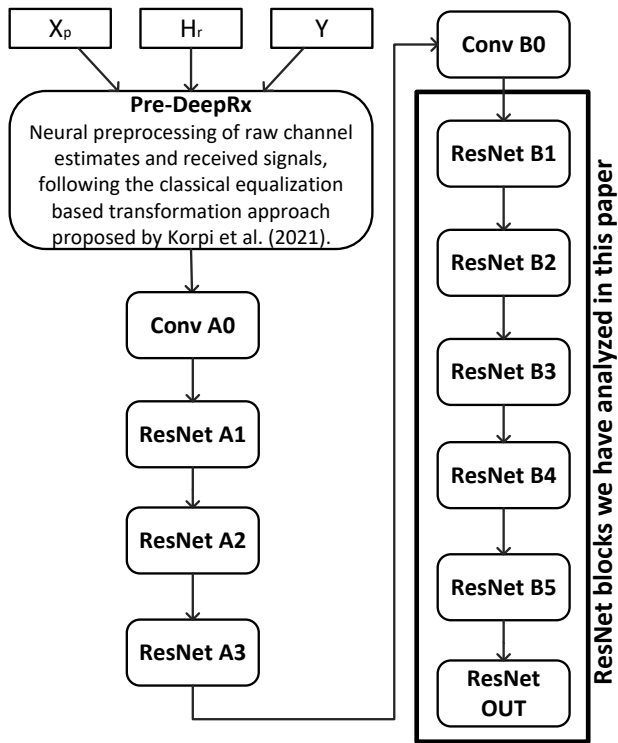


Figure 17: Illustration of the data flow within the DeepRx model architecture.

erwise and channelwise models, suggesting that the models are effectively ignoring input variations and defaulting to the values around the global average. We conclude that the activations of the DeepRx model likely do not contain meaningful information about delay spread, at least not in a directly predictive form.

From a physical layer perspective, this is plausible given our simulation parameters (see Table 1). The bandwidth is $B = 192 \times 30 \text{ kHz} = 5.76 \text{ MHz}$, yielding a time resolution of $\Delta\tau = 1/B \approx 174 \text{ ns}$, and a quantization uncertainty of approximately $\sigma \approx \Delta\tau/\sqrt{12} \approx \pm 50 \text{ ns}$ (standard deviation of a uniform distribution [Earl and Nicholson, 2021, Appendix 15]). As a result, multipath components closer than 174 ns apart (e.g., a 10 ns path) are indistinguishable, and even larger separations are affected by quantization noise. This limitation likely contributes to the lack of delay spread information in DeepRx’s activations. Interestingly, DeepRx still performs well under delay spread conditions, suggesting it may compensate for this effect without needing to explicitly quantify the delay spread (e.g., as RMS).

NUCPI Analysis of User Equipment (UE) Speed

We also applied NUCPI analysis to UE speed, evaluating layer-wise explainer performance for two layers. As with delay spread (see Appendix C.3), the explainer models exhibit consistent overfitting: training losses decrease, but evaluation losses do not improve, as shown in Figures 31 and 32. Based on this behavior, we conclude that DeepRx’s internal activations likely do not encode UE speed in a directly predictive

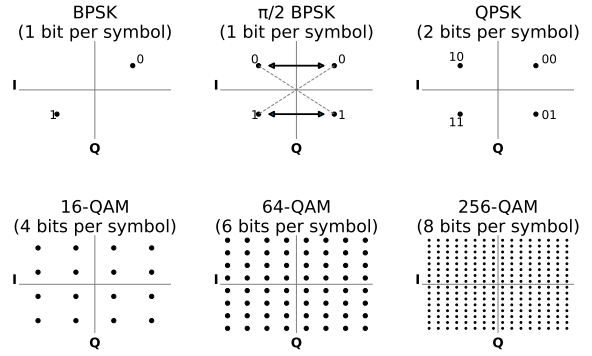


Figure 18: Modulation schemes used in 5G NR, including BPSK, Pi/2 BPSK, QPSK, 16-QAM, 64-QAM, and 256-QAM (redrawn from [Zaidi et al., 2018]).

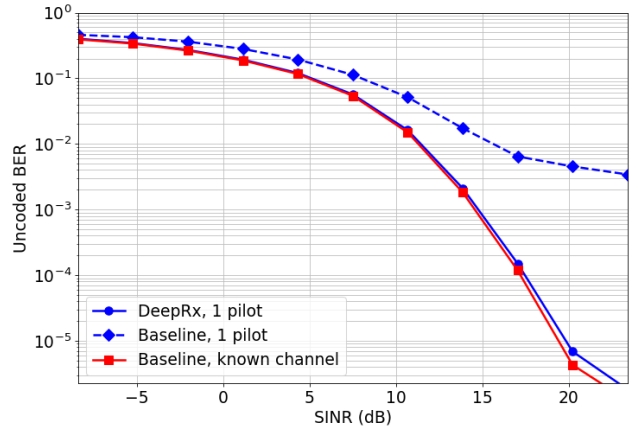


Figure 19: Performance comparison of the DeepRx model against baseline approaches across different SINR levels.

way.

In traditional systems, UE speed—or Doppler spread more technically—is estimated by aggregating over tens of slots ([Yucek et al., 2005; Zhao et al., 2009]). In contrast, DeepRx operates on a single slot without a memory of past slots. Despite this, DeepRx can still handle frequency offset effects caused by user movement (and other factors). A similar explanation as with delay spread may apply: the model may not need to explicitly estimate frequency offset or Doppler spread to effectively compensate for them.

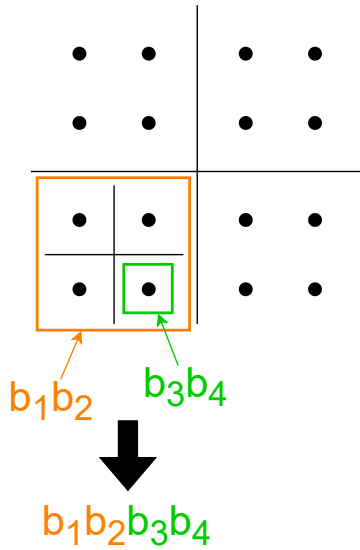


Figure 20: The relationship between 16-QAM and QPSK constellation points (redrawn from [Honkala *et al.*, 2021]). When QPSK is used, only 2 bits b_1 and b_2 are used to define the complex quadrant, but if 16-QAM is used, besides the first two bits, b_3 and b_4 are also used to define the quadrant within the quadrant.

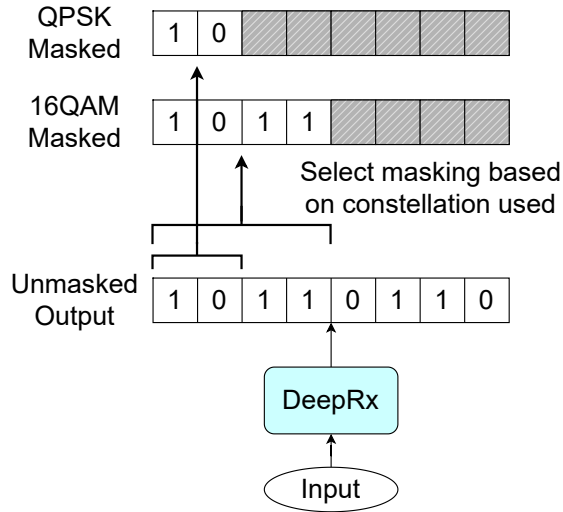


Figure 21: Masking bits based on the modulation, including both QPSK and 16QAM, where masking is selected based on the modulation. The output is masked accordingly so that unnecessary parts of the output array are not considered (redrawn from [Honkala *et al.*, 2021]).



Figure 22: Layerwise test losses for the explainer model on delay spread, showing poor predictive performance across all layers.

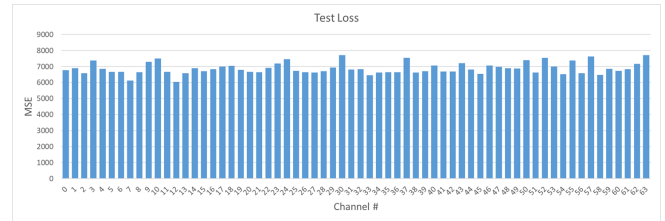


Figure 23: Channelwise test losses for the explainer model on delay spread for layer B1-PRE, showing poor predictive performance across all channels.

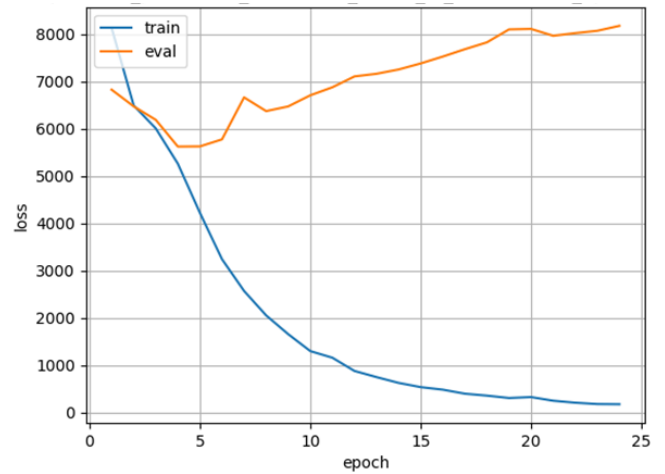


Figure 24: Training curve on delay spread for layer B1-PRE, illustrating overfitting in the explainer model.

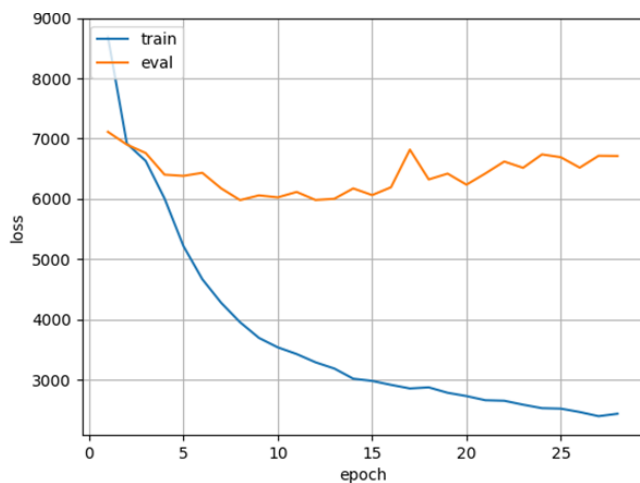


Figure 25: Training curve on delay spread for layer OUT-POST, showing overfitting in the explainer model.

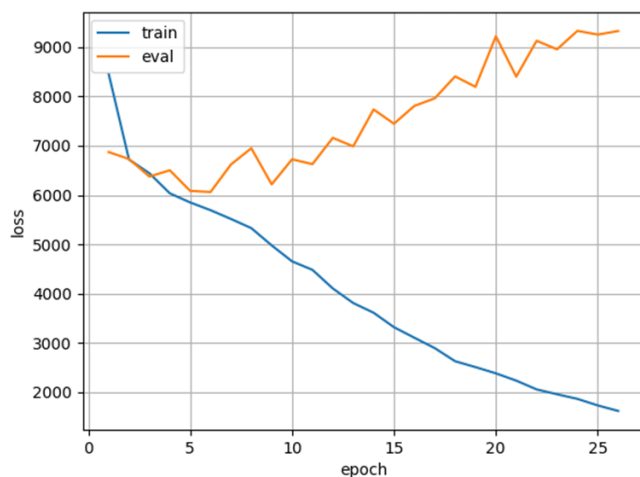


Figure 26: Training curve on delay spread for channel 7 on layer B1-PRE, indicating overfitting in the explainer model.

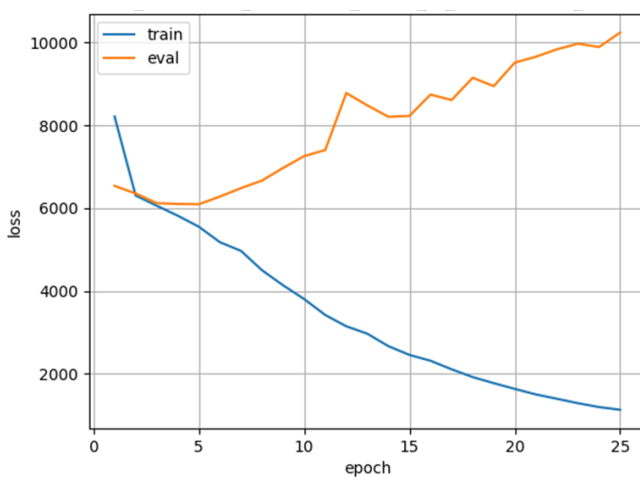


Figure 27: Training curve on delay spread for channel 12 on layer B1-PRE, illustrating overfitting in the explainer model.

Estimates of the layerwise models

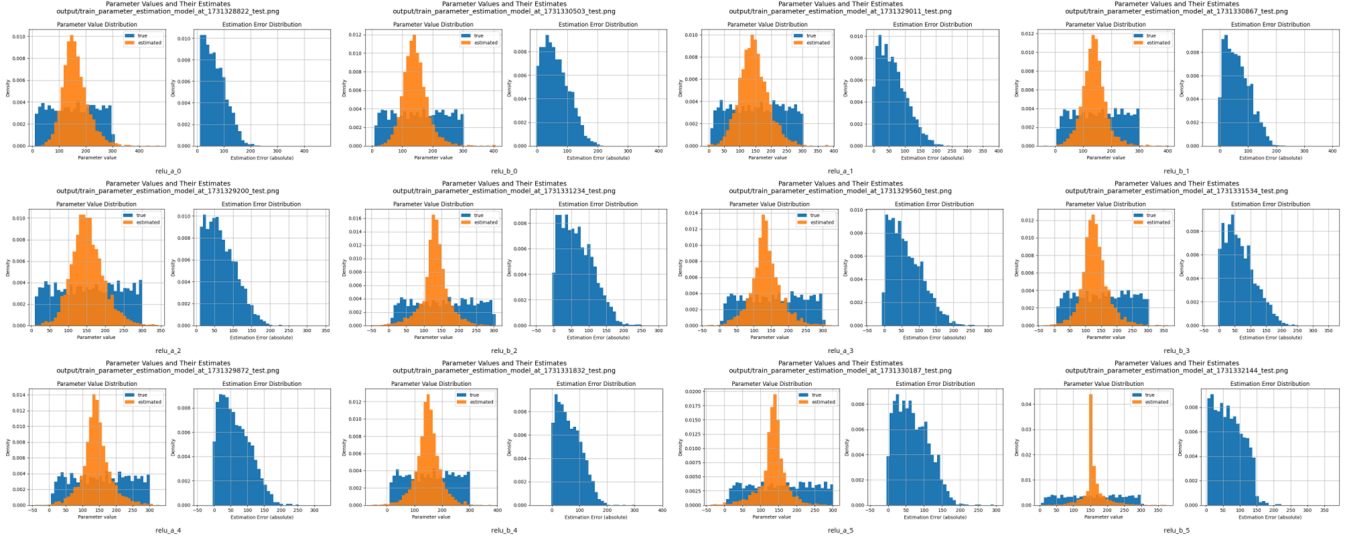


Figure 28: Estimates of delay spread by layerwise explainer models, peaking around the mean of true delay spread parameters.

Estimates of the channelwise models on relu_b_5

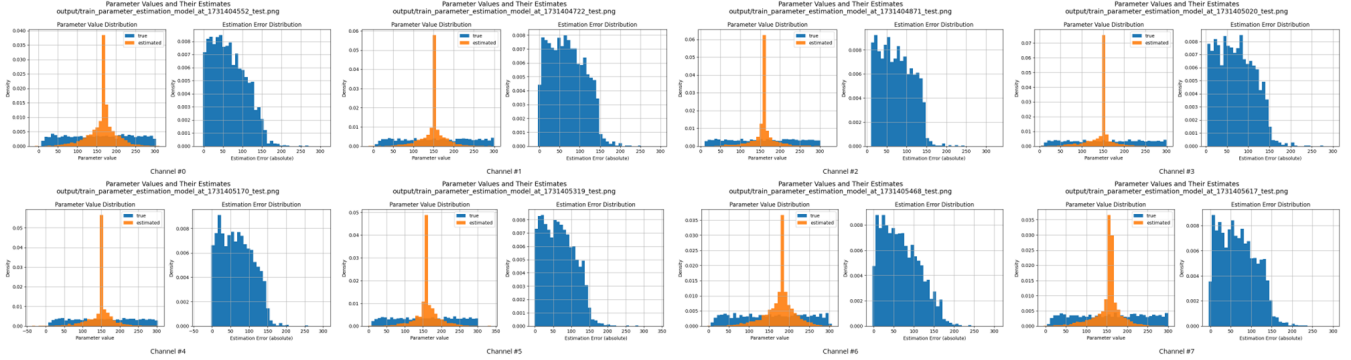


Figure 29: Estimates of delay spread by channelwise explainer models on layer OUT-POST, peaking around the mean of true delay spread parameters.

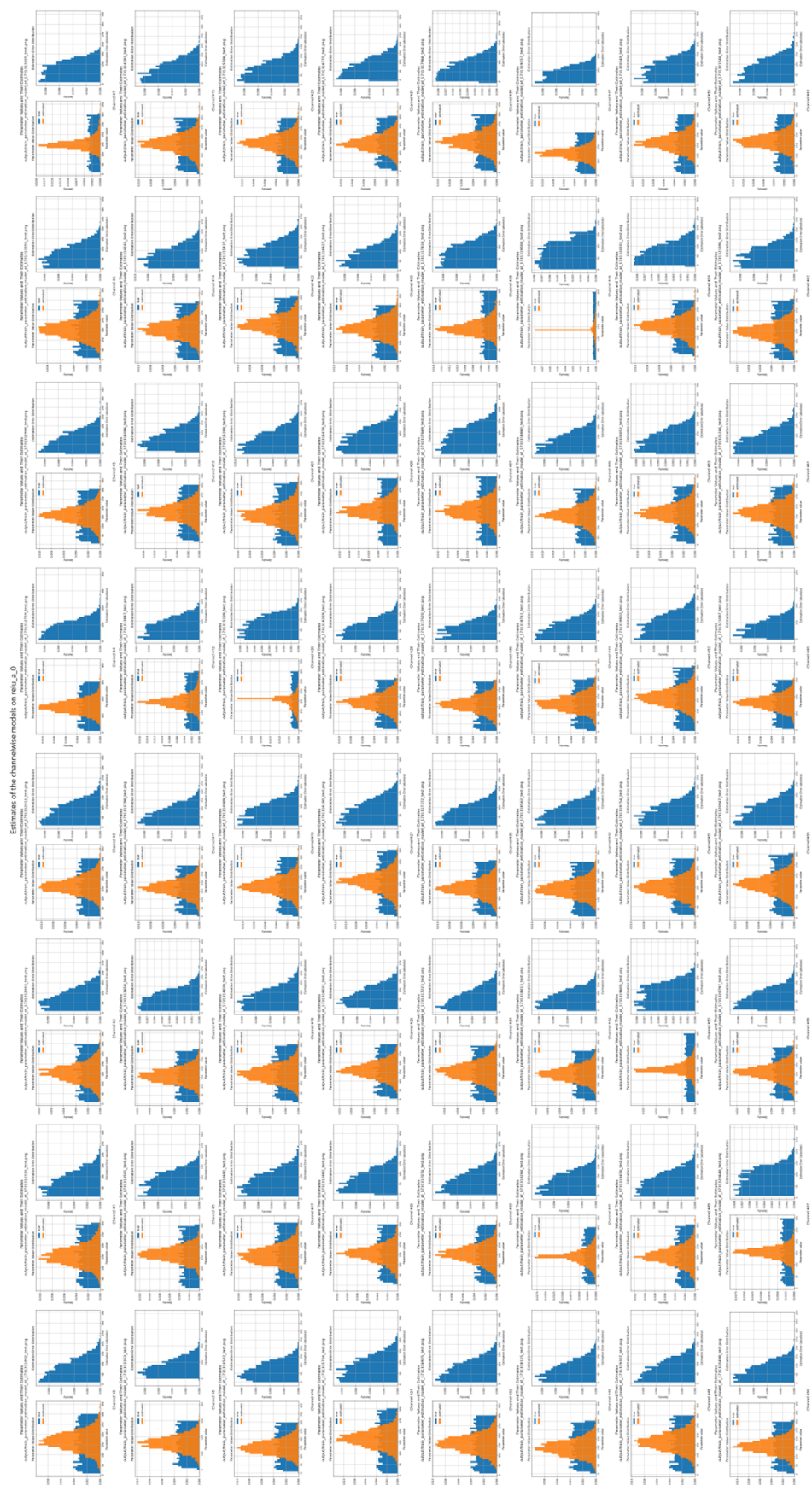


Figure 30: Estimates of delay spread by channelwise explainer models on layer B1-PRE, peaking around the mean of true delay spread parameters.

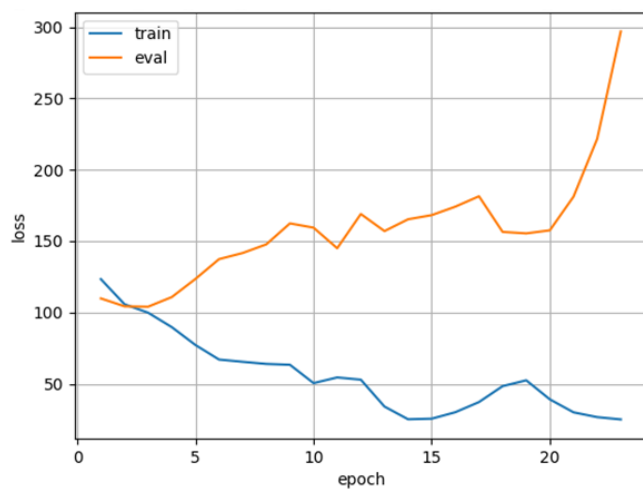


Figure 31: Training curve on UE speed for layer B1-PRE, illustrating overfitting in the explainer model.

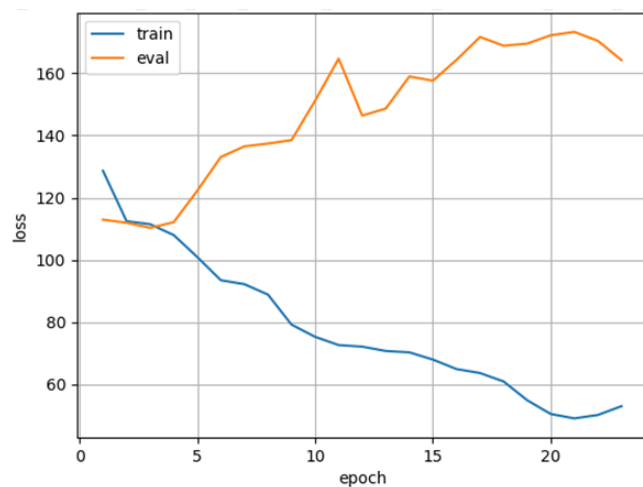


Figure 32: Training curve on UE speed for layer OUT-POST, illustrating overfitting in the explainer model.

D Uniform Manifold Approximation and Projection (UMAP) Methodology for Manifold Analysis

We use Uniform Manifold Approximation and Projection (UMAP) [McInnes *et al.*, 2018] to visualize the structure of high-dimensional activation features in two dimensions. UMAP is a nonlinear, stochastic dimensionality-reduction technique that aims to preserve local neighborhood relationships while maintaining coarse global structure. As a result, local geometry and exact point placement may vary across random initializations, and UMAP embeddings should be interpreted qualitatively rather than metrically. In this work, UMAP is used solely as a visualization tool to reveal large-scale organization in the learned representations; all quantitative results are computed in the original feature space. Repetition experiments assessing the robustness of the structures reported in the main paper (Section 4.2) are provided below.

D.1 Repetition Experiments on Radio Receiver

As noted above, UMAP is a stochastic dimensionality-reduction technique, and its local geometry may vary across random initializations. To assess the robustness of the observed activation structure, we repeat the UMAP embedding five times using different random seeds for DeepRx activations from layer B1-PRE. Figures 33–37 show the resulting embeddings for both the baseline and proposed methodologies, colored by different metadata attributes.

Across all repetitions, the global organization of the embeddings remains stable. In particular, samples consistently form a smooth, low-dimensional manifold primarily aligned with SNR (Fig. 34), while other channel attributes such as delay spread, UE speed, and channel model induce only secondary variations (Figs. 35–37). Although local neighborhoods and fine-scale layout may differ between runs—as expected for UMAP—the dominant SNR-driven structure is preserved. This confirms that the observed manifold is not an artifact of a particular UMAP initialization but reflects a robust property of the learned receiver representations.

E Conditional k-NN Methodology for Out-Of-Distribution Detection

Let each training sample be represented by an activation vector $a_i \in \mathbb{R}^d$ and an associated SNR parameter $p_i \in \mathbb{R}$, and let the test sample and its parameter be (a_*, p_*) . In a standard k -nearest-neighbour (k-NN) method, only distances in activation space are considered, typically

$$d_i = \|a_i - a_*\|, \quad (34)$$

and the k smallest distances determine the neighborhood for test sample in question. However, the relationship between training and test samples may depend on the SNR level. To capture this dependence, we introduce a *conditional distance* in which SNR differences influence the effective neighbour selection.

E.1 Hard Conditional Scheme

In the hard variant, SNR is used as a strict constraint. A training point is considered eligible only if its SNR is sufficiently

close to the test SNR. Let $D > 0$ be the maximum allowed SNR deviation. Define

$$\Delta p_i = |p_i - p_*|. \quad (35)$$

The effective distance is then

$$d_i^{\text{eff}} = \begin{cases} d_i, & \text{if } \Delta p_i \leq D, \\ \infty, & \text{otherwise.} \end{cases} \quad (36)$$

Thus, neighbours are selected *only* from activation-space distances of points whose SNR is within a hard threshold of p_* , and all other points are excluded.

E.2 Soft Conditional Scheme

The hard cutoff can be replaced by a smooth penalty that grows with the SNR difference, allowing all points to contribute while favouring those with more similar SNR. Let $m(\Delta p_i)$ be a monotone increasing penalty function with $m(0) = 1$, and define the effective distance as

$$d_i^{\text{eff}} = m(\Delta p_i) d_i. \quad (37)$$

In these soft schemes, the k neighbours are chosen as the training points with the smallest effective distances d_i^{eff} , enabling a continuous transition between strict conditioning and unrestricted k-NN. Two practical choices for $m(\cdot)$ are power-law and exponential penalty, described below in more detail.

Power-law penalty.

Let D denote the characteristic SNR scale and $F > 1$ the penalty factor at $\Delta p_i = D$. With exponent $p \geq 1$,

$$m(\Delta p_i) = 1 + (F - 1) \left(\frac{\Delta p_i}{D} \right)^p. \quad (38)$$

This yields linear ($p = 1$) or stronger ($p > 1$) growth of the penalty as the SNR difference increases.

Exponential (Gaussian) penalty.

Define a bandwidth $h > 0$ and let

$$m(\Delta p_i) = \exp \left(\frac{1}{2} \left(\frac{\Delta p_i}{h} \right)^2 \right). \quad (39)$$

The bandwidth h may be chosen so that $m(D)$ matches a desired penalty factor F , leading to

$$h = \frac{D}{\sqrt{2 \ln F}}. \quad (40)$$

In this formulation, points with larger SNR mismatch receive exponentially inflated effective distances and thus are gradually deprioritised in the neighbour ranking.

E.3 Parameter Selection

The SNR tolerance D is selected as the 99th percentile of absolute SNR differences observed in the ID training set, reflecting a typical operating range. The penalty factor is fixed to $F = 2.0$ across experiments.

Conditional k-NN is evaluated using mean-activation features with cosine distance, as this combination yields the strongest k-NN baseline performance on average across the two OOD scenarios (Table 2). We adopt the exponential

penalty, as it degrades performance less than the power-law alternative in the high delay scenario, and set the exponent to $p = 1$, which provides the largest gains in the high speed scenario.

Figures 38 and 39 illustrate the sensitivity of OOD detection performance to the penalty exponent under high delay spread and high UE speed, respectively.

F Continuous Conformal P-Values Methodology for Out-Of-Distribution Detection

We briefly recall the Benjamini–Hochberg (BH)-style conformal testing rule of Algorithm 1 in [Magesh *et al.*, 2023]. Let $\{\hat{Q}_{(i)}\}_{i=1}^K$ denote the ordered conformal p-values across K detectors. The algorithm defines

$$m = \max \left\{ i : \hat{Q}_{(i)} \leq \frac{\alpha i}{C(K) K} \right\} \quad (41)$$

and declares a sample to be Out-Of-Distribution (OOD) if

$$m \geq 1. \quad (42)$$

Equivalently, the sample is classified as OOD if there exists an index i such that

$$\hat{Q}_{(i)} \leq \frac{\alpha i}{C(K) K}. \quad (43)$$

Rearranging this condition yields

$$\min_{i=1, \dots, K} \left(\frac{C(K) K}{i} \hat{Q}_{(i)} \right) \leq \alpha. \quad (44)$$

This observation allows us to convert the binary rejection rule of Algorithm 1 into a continuous score. We define the scalar

$$S = \min_{i=1, \dots, K} \left(\frac{C(K) K}{i} \hat{Q}_{(i)} \right), \quad (45)$$

which represents the smallest significance level α at which the sample would be declared OOD under the BH-style conformal test. While this continuous score is not introduced as a separate algorithm in prior work, it follows directly from the decision rule in [Magesh *et al.*, 2023].

The score S provides a calibrated, continuous measure of out-of-distribution confidence and can be used to construct ROC or DET curves, enabling quantitative comparison and fusion with other OOD detectors.

G Supplemental Experimental Results on Out-Of-Distribution Detection on Radio Receiver

This section provides detailed numerical results supporting the Out-Of-Distribution (OOD) detection experiments reported in the main paper (Section 5). We organize results by analysis type—single-detector baselines, SNR-aware modeling, and detector fusion—while reporting performance separately for the three representative layers analyzed in the main text (B1-PRE, B3-POST, and OUT-POST). All results are reported for both high delay and high speed OOD scenarios, as well as their average when appropriate. Unless stated otherwise, results are averaged over five independent runs with different random seeds for DeepRx activations.

G.1 Single-Detector Baselines

We first report performance of individual OOD detectors applied to a single layer without SNR conditioning or fusion. Results for layers B1-PRE, B3-POST, and OUT-POST are reported separately in Tables 9–16.

G.2 Effect of SNR-Aware Modeling

We evaluate the impact of incorporating SNR information into OOD detection using two strategies: (i) conditional k-NN (hard and soft conditioning), and (ii) feature concatenation for OC-SVM and Mahalanobis detectors. Results are reported for all three layers across both OOD scenarios in Tables 17–22. Soft conditional k-NN yields the largest gains under high UE speed on B1-PRE, whereas hard conditioning is more robust to increased delay spread (Figs. 38–39).

G.3 Within-Layer Fusion

We evaluate the fusion of multiple detectors applied to the same layer using OC-SVM and conformal p-value aggregation. All 27 combinations of the four activation-based detectors and raw SNR are evaluated per layer:

- B1-PRE: Tables 24–26;
- B3-POST: Tables 27–29;
- OUT-POST: Tables 30–32;

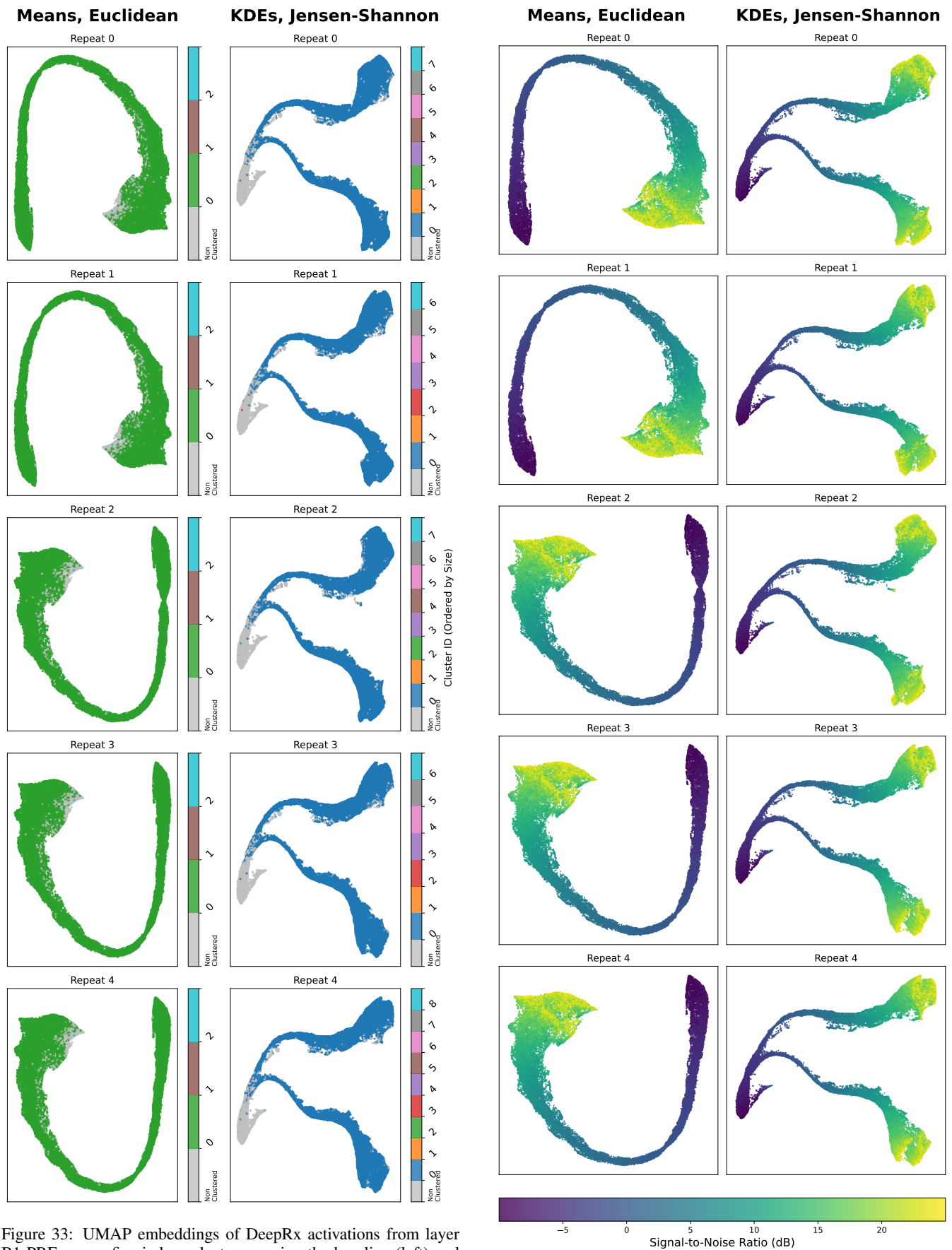
and best-performing fusions compiled in Table 23.

G.4 Across-Layer Fusion

Finally, we evaluate across-layer fusion, where detector outputs from multiple layers are combined to exploit complementary information across network depth. We consider fusing raw detector outputs as well as within-layer fused scores across B1-PRE, B3-POST, OUT-POST, and SNR. All 11 combinations of the scores are evaluated per detector:

- HDBSCAN: Tables 34–36;
- k-NN: Tables 37–39;
- OC-SVM: Tables 40–42;
- Mahalanobis: Tables 43–45;
- Within-layer: Tables 46–48;

and best-performing fusions compiled in Table 33.



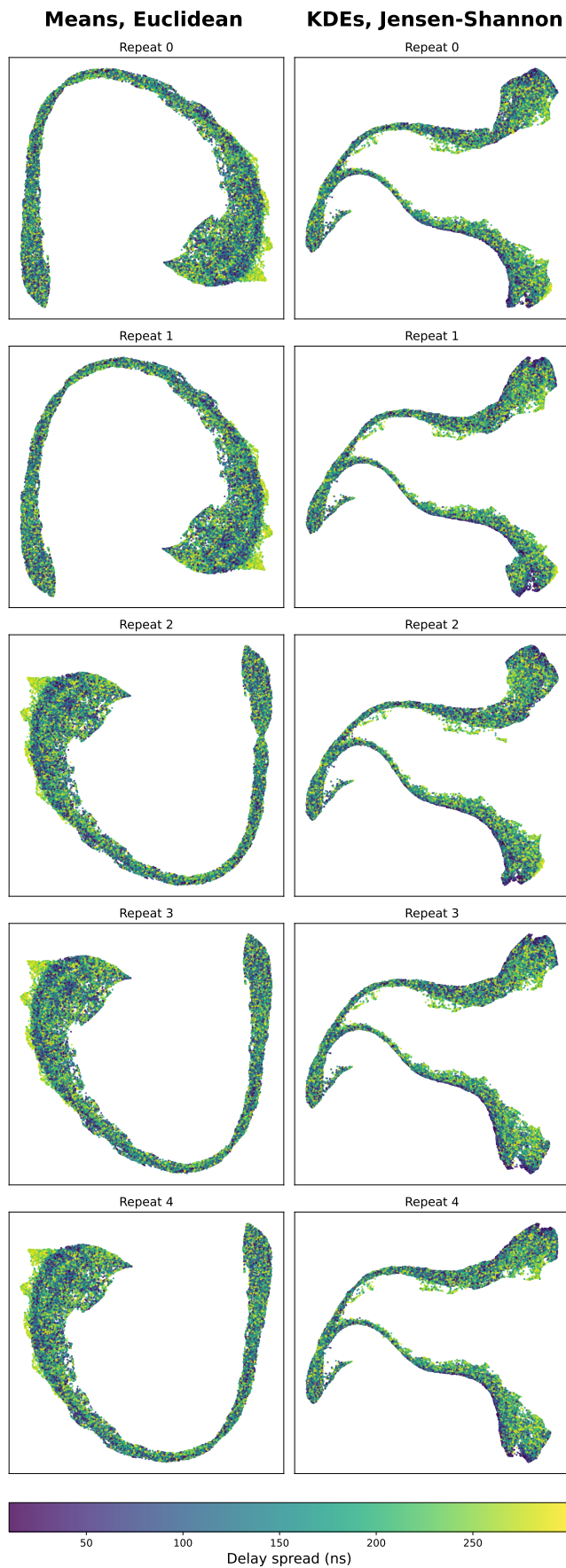


Figure 35: UMAP embeddings of DeepRx activations from layer B1-PRE across five independent runs using the baseline (left) and proposed (right) methodology, **colored by delay spread**. UMAP is used solely for visualization.

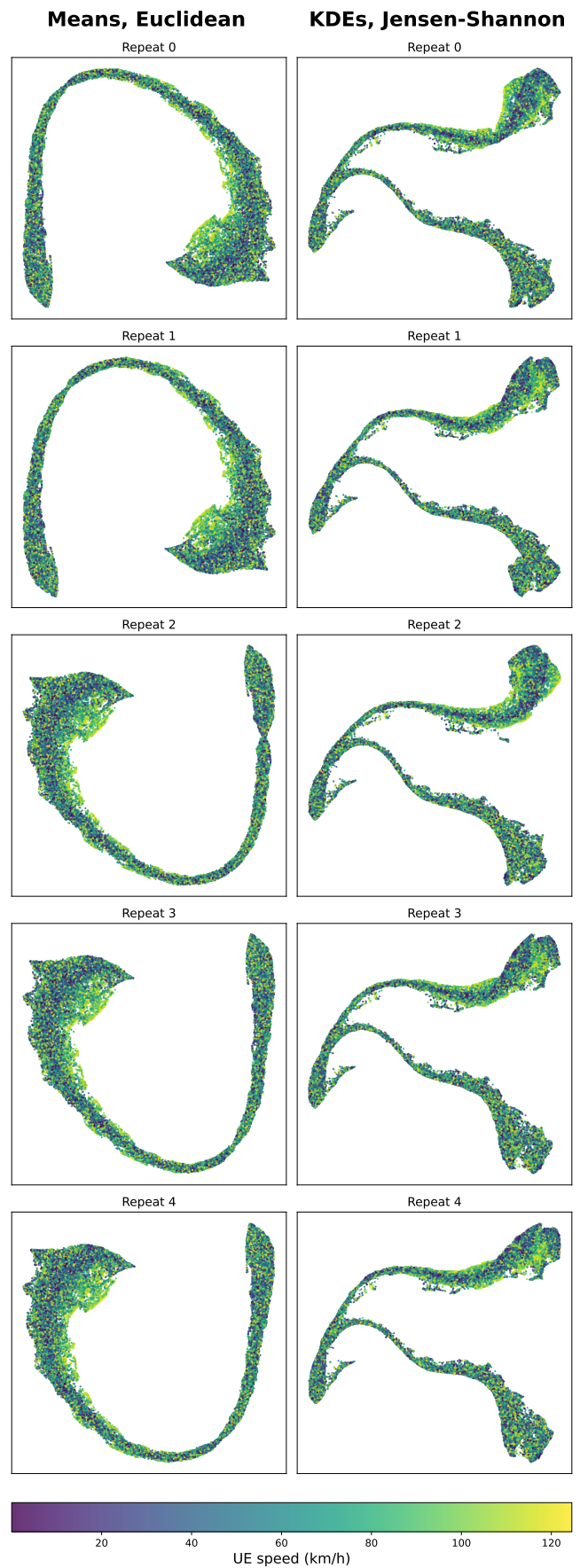


Figure 36: UMAP embeddings of DeepRx activations from layer B1-PRE across five independent runs using the baseline (left) and proposed (right) methodology, **colored by UE speed**. UMAP is used solely for visualization.

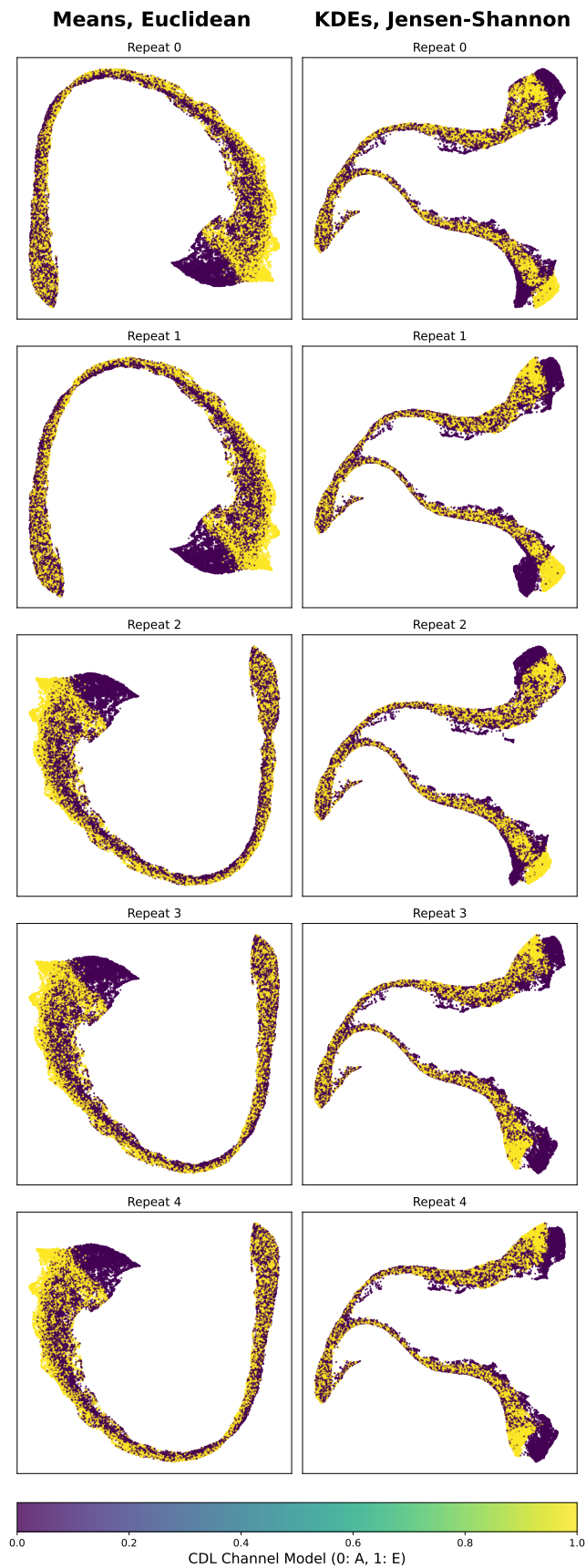


Figure 37: UMAP embeddings of DeepRx activations from layer B1-PRE across five independent runs using the baseline (left) and proposed (right) methodology, **colored by channel model**. UMAP is used solely for visualization.

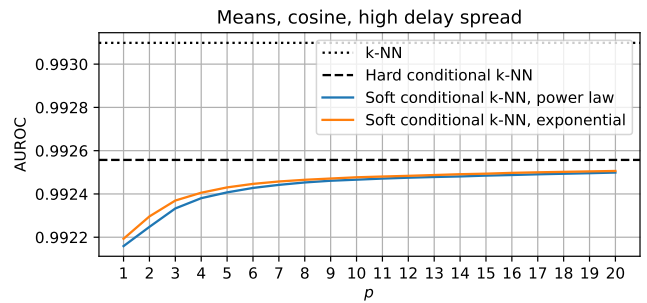


Figure 38: OOD detection performance of k-NN-based detectors under high delay spread on layer B1-PRE as a function of the penalty exponent p in (38) and (39). Mean-activation features and the cosine distance are used.

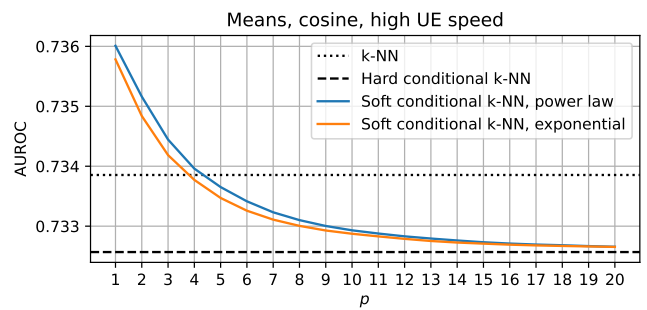


Figure 39: OOD detection performance of k-NN-based detectors under high UE speed on layer B1-PRE as a function of the penalty exponent p in (38) and (39). Mean-activation features and the cosine distance are used.

Feature + Distance	HDBSCAN		k-NN		OC-SVM		Mahalanobis	
	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95
Mean + Euclidean	0.9209	1.0000	0.9741	0.1812	0.9394	0.2084	0.9965	0.0200
Mean + Cosine	0.7360	1.0000	0.9931	0.0218				
KDE + Euclidean	0.9011	1.0000	0.9930	0.0456				
KDE + Cosine	0.8920	1.0000	0.9920	0.0526	0.8628	0.4197	0.9854	0.0582
KDE + Jensen-Shannon	0.8898	1.0000	0.9940	0.0374				
Energy + Euclidean	0.8586	1.0000	0.9622	0.2538				
Energy + Cosine	0.8432	1.0000	0.9675	0.2115	0.8861	0.3286	0.9905	0.0458

Table 9: OOD detection performance under high delay spread on layer B1-PRE. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.055.

Feature + Distance	HDBSCAN		k-NN		OC-SVM		Mahalanobis	
	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95
Mean + Euclidean	0.6528	1.0000	0.6670	0.9301				
Mean + Cosine	0.6285	1.0000	0.7339	0.8620	0.5994	0.9253	0.7351	0.8569
KDE + Euclidean	0.6601	1.0000	0.7178	0.8199				
KDE + Cosine	0.6791	1.0000	0.7164	0.8292	0.6143	0.8947	0.7357	0.7864
KDE + Jensen-Shannon	0.6715	1.0000	0.7266	0.8069				
Energy + Euclidean	0.6369	1.0000	0.6691	0.9203				
Energy + Cosine	0.6342	1.0000	0.6741	0.9120	0.6004	0.9218	0.7179	0.8716

Table 10: OOD detection performance under high UE speed on layer B1-PRE. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.025.

Feature + Distance	HDBSCAN		k-NN		OC-SVM		Mahalanobis	
	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95
Mean + Euclidean	0.8370	0.3543	0.9134	0.4588				
Mean + Cosine	0.8807	0.2073	0.9174	0.6440	0.7372	0.8597	0.9410	0.3327
KDE + Euclidean	0.8812	1.0000	0.9323	0.3689				
KDE + Cosine	0.8896	0.3904	0.9395	0.2397	0.8845	0.5919	0.9389	0.3576
KDE + Jensen-Shannon	0.8591	1.0000	0.9501	0.3023				
Energy + Euclidean	0.7948	0.6573	0.9191	0.3872				
Energy + Cosine	0.8201	0.3305	0.9346	0.3281	0.7569	0.8255	0.9450	0.3299

Table 11: OOD detection performance under high delay spread on layer B3-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.035, except for 0.28 for HDBSCAN Energy + Euclidean.

Feature + Distance	HDBSCAN		k-NN		OC-SVM		Mahalanobis	
	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95
Mean + Euclidean	0.6955	0.5966	0.7317	0.8488	0.7387	0.8957	0.7472	0.8615
Mean + Cosine	0.7249	0.5206	0.7522	0.9138				
KDE + Euclidean	0.6993	1.0000	0.7547	0.8202				
KDE + Cosine	0.7210	0.6114	0.7820	0.7403	0.7344	0.8538	0.7385	0.9170
KDE + Jensen-Shannon	0.7053	1.0000	0.7704	0.7839				
Energy + Euclidean	0.6864	0.7691	0.7590	0.8194				
Energy + Cosine	0.6907	0.5825	0.7693	0.8039	0.7279	0.9121	0.7411	0.8559

Table 12: OOD detection performance under high UE speed on layer B3-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015, except for 0.19 for HDBSCAN Energy + Euclidean.

Feature + Distance	HDBSCAN		k-NN		OC-SVM		Mahalanobis	
	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95
Mean + Euclidean	0.7662	0.4754	0.8225	0.6538	0.7379	0.8777	0.8441	0.5971
Mean + Cosine	0.8028	0.3640	0.8348	0.7789				
KDE + Euclidean	0.7902	1.0000	0.8435	0.5945	0.8094	0.7229	0.8387	0.6373
KDE + Cosine	0.8053	0.5009	0.8608	0.4900				
KDE + Jensen-Shannon	0.7822	1.0000	0.8603	0.5431				
Energy + Euclidean	0.7406	0.7132	0.8391	0.6033	0.7424	0.8688	0.8431	0.5929
Energy + Cosine	0.7554	0.4565	0.8520	0.5660				

Table 13: Average OOD detection performance over high delay spread and UE speed scenarios on layer B3-POST. Mean over 5 independent repeats with random seeds.

Feature + Distance	HDBSCAN		k-NN		OC-SVM		Mahalanobis	
	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95
Mean + Euclidean	0.5208	1.0000	0.5769	0.9495	0.5509	0.9486	0.5677	0.9359
Mean + Cosine	0.6745	1.0000	0.6929	0.8847				
KDE + Euclidean	0.5692	1.0000	0.6397	0.9485	0.4978	0.9746	0.6051	0.9499
KDE + Cosine	0.5574	1.0000	0.6584	0.9484				
KDE + Jensen-Shannon	0.5741	1.0000	0.6698	0.9466				
Energy + Euclidean	0.5195	1.0000	0.5816	0.9495	0.5564	0.9455	0.5697	0.9481
Energy + Cosine	0.5129	1.0000	0.5900	0.9499				

Table 14: OOD detection performance under high delay spread on layer OUT-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015.

Feature + Distance	HDBSCAN		k-NN		OC-SVM		Mahalanobis	
	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95
Mean + Euclidean	0.5096	1.0000	0.5303	0.9565	0.4458	0.9919	0.5204	0.9465
Mean + Cosine	0.6372	1.0000	0.6620	0.9235				
KDE + Euclidean	0.5470	1.0000	0.6039	0.9536	0.4606	0.9813	0.5719	0.9505
KDE + Cosine	0.5418	1.0000	0.6182	0.9546				
KDE + Jensen-Shannon	0.5499	1.0000	0.6187	0.9516				
Energy + Euclidean	0.5086	1.0000	0.5279	0.9570	0.4446	0.9927	0.5147	0.9458
Energy + Cosine	0.5033	1.0000	0.5324	0.9561				

Table 15: OOD detection performance under high UE speed on layer OUT-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015.

Feature + Distance	HDBSCAN		k-NN		OC-SVM		Mahalanobis	
	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95
Mean + Euclidean	0.5152	1.0000	0.5536	0.9530	0.4983	0.9703	0.5441	0.9412
Mean + Cosine	0.6558	1.0000	0.6775	0.9041				
KDE + Euclidean	0.5581	1.0000	0.6218	0.9510	0.4792	0.9779	0.5885	0.9502
KDE + Cosine	0.5496	1.0000	0.6383	0.9515				
KDE + Jensen-Shannon	0.5620	1.0000	0.6443	0.9491				
Energy + Euclidean	0.5140	1.0000	0.5547	0.9532	0.5005	0.9691	0.5422	0.9470
Energy + Cosine	0.5081	1.0000	0.5612	0.9530				

Table 16: Average OOD detection performance over high delay spread and UE speed scenarios on layer OUT-POST. Mean over 5 independent repeats with random seeds.

SNR regime	k-NN				OC-SVM				Mahalanobis			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
Activations only	0.9931	0.0000	0.0218	0.0000	0.9394	0.0000	0.2084	0.0000	0.9965	0.0000	0.0200	0.0000
SNR only	0.7326	-0.2605	0.7897	0.7679	0.5000	-0.4394	1.0000	0.7916	0.5000	-0.4965	0.9589	0.9389
Both combined (hard/concat)	0.9926	-0.0005	0.0240	0.0021	0.9392	-0.0002	0.2096	0.0012	0.9696	-0.0269	0.1484	0.1285
Both combined (soft)	0.9922	-0.0009	0.0278	0.0060								

Table 17: OOD detection performance under high delay spread on layer B1-PRE. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015.

SNR regime	k-NN				OC-SVM				Mahalanobis			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
Activations only	0.7339	-0.0019	0.8620	0.0000	0.5994	-0.0000	0.9253	0.0001	0.7351	0.0000	0.8569	0.0000
SNR only	0.6128	-0.1230	0.9144	0.0524	0.5000	-0.0994	1.0000	0.0748	0.5000	-0.2351	0.9589	0.1020
Both combined (hard/concat)	0.7326	-0.0032	0.8660	0.0040	0.5994	0.0000	0.9252	0.0000	0.6527	-0.0823	0.8987	0.0418
Both combined (soft)	0.7358	0.0000	0.8651	0.0031								

Table 18: OOD detection performance under high UE speed on layer B1-PRE. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015.

SNR regime	k-NN				OC-SVM				Mahalanobis			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
Activations only	0.9174	-0.0042	0.6440	0.0433	0.7372	-0.0006	0.8597	0.0060	0.9410	0.0000	0.3327	0.0000
SNR only	0.6758	-0.2458	0.8233	0.2226	0.5000	-0.2378	1.0000	0.1463	0.5000	-0.4410	0.9589	0.6261
Both combined (hard/concat)	0.9216	0.0000	0.6007	0.0000	0.7378	0.0000	0.8537	0.0000	0.9010	-0.0400	0.3724	0.0396

Table 19: OOD detection performance under high delay spread on layer B3-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015.

SNR regime	k-NN				OC-SVM				Mahalanobis			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
Activations only	0.7522	-0.0089	0.9138	0.0193	0.7387	-0.0000	0.8957	0.0000	0.7472	-0.0004	0.8615	0.0463
SNR only	0.6797	-0.0814	0.8944	0.0000	0.5000	-0.2387	1.0000	0.1043	0.5000	-0.2475	0.9589	0.1437
Both combined (hard/concat)	0.7611	0.0000	0.9015	0.0071	0.7387	0.0000	0.8979	0.0022	0.7475	0.0000	0.8152	0.0000

Table 20: OOD detection performance under high UE speed on layer B3-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015.

SNR regime	k-NN				OC-SVM				Mahalanobis			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
Activations only	0.6929	0.0000	0.8847	0.0000	0.5509	-0.0121	0.9486	0.0362	0.5677	-0.0313	0.9359	0.0501
SNR only	0.5373	-0.1556	0.9309	0.0462	0.5000	-0.0630	1.0000	0.0876	0.5000	-0.0990	0.9589	0.0731
Both combined (hard/concat)	0.6905	-0.0024	0.8952	0.0105	0.5630	0.0000	0.9124	0.0000	0.5990	0.0000	0.8858	0.0000

Table 21: OOD detection performance under high delay spread on layer B6-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015.

Layer	OC-SVM fusion	BH-style fusion
B1-PRE (Table 26)	k-NN, OC-SVM, SNR	k-NN, Mahalanobis
B3-POST (Table 29)	k-NN, OC-SVM, Mahalanobis, SNR	k-NN, OC-SVM, Mahalanobis, SNR
OUT-POST (Table 32)	HDBSCAN, Mahalanobis, SNR	HDBSCAN, k-NN

Table 23: Best-performing within-layer fusions by average AUROC across OOD scenarios. Results are averaged over high-delay and high-speed conditions. All the classifiers are applied to the mean features. The k-NN and HDBSCAN classifiers use the cosine distance and Euclidean distances, respectively.

Baseline classifiers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D, E}	0.9961	-0.0004	0.0122	0.0022	0.9952	-0.0005	0.0265	0.0065
{B, C, D, E}	0.9966	-0.0000	0.0188	0.0088	0.9954	-0.0004	0.0265	0.0065
{A, C, D, E}	0.9945	-0.0020	0.0233	0.0133	0.9937	-0.0020	0.0348	0.0148
{A, B, D, E}	0.9954	-0.0012	0.0100	0.0000	0.9944	-0.0013	0.0237	0.0037
{A, B, C, E}	0.9952	-0.0013	0.0167	0.0067	0.9941	-0.0016	0.0304	0.0104
{A, B, C, D}	0.9959	-0.0006	0.0117	0.0017	0.9956	-0.0002	0.0249	0.0049
{C, D, E}	0.9958	-0.0008	0.0256	0.0157	0.9939	-0.0018	0.0348	0.0148
{B, D, E}	0.9960	-0.0006	0.0167	0.0068	0.9948	-0.0010	0.0237	0.0037
{B, C, E}	0.9960	-0.0006	0.0222	0.0122	0.9944	-0.0013	0.0304	0.0104
{B, C, D}	0.9964	-0.0001	0.0201	0.0101	0.9957	0.0000	0.0249	0.0049
{A, D, E}	0.9947	-0.0019	0.0154	0.0054	0.9945	-0.0012	0.0266	0.0066
{A, C, E}	0.9636	-0.0329	0.1300	0.1200	0.9401	-0.0557	0.1891	0.1691
{A, C, D}	0.9942	-0.0023	0.0284	0.0185	0.9946	-0.0011	0.0302	0.0103
{A, B, E}	0.9918	-0.0048	0.0174	0.0074	0.9913	-0.0045	0.0254	0.0055
{A, B, D}	0.9953	-0.0013	0.0101	0.0002	0.9949	-0.0008	0.0210	0.0010
{A, B, C}	0.9951	-0.0015	0.0172	0.0072	0.9946	-0.0011	0.0273	0.0074
{D, E}	0.9966	0.0000	0.0180	0.0081	0.9952	-0.0005	0.0266	0.0066
{C, E}	0.9363	-0.0602	0.2223	0.2123	0.9192	-0.0765	0.2843	0.2643
{C, D}	0.9960	-0.0005	0.0242	0.0142	0.9949	-0.0008	0.0302	0.0103
{B, E}	0.9929	-0.0036	0.0211	0.0111	0.9922	-0.0036	0.0254	0.0055
{B, D}	0.9959	-0.0007	0.0185	0.0085	0.9953	-0.0005	0.0210	0.0010
{B, C}	0.9958	-0.0007	0.0241	0.0141	0.9949	-0.0008	0.0273	0.0074
{A, E}	0.9406	-0.0559	0.1908	0.1808	0.8874	-0.1083	0.2230	0.2030
{A, D}	0.9945	-0.0020	0.0181	0.0081	0.9957	-0.0000	0.0200	0.0000
{A, C}	0.9505	-0.0460	0.1323	0.1223	0.9514	-0.0444	0.1578	0.1378
{A, B}	0.9912	-0.0054	0.0189	0.0089	0.9920	-0.0037	0.0219	0.0019

Table 24: OOD detection performance under high delay spread on layer B1-PRE. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.005. A: mean, Euclidean, HDBSCAN; B: mean, cosine, k-NN (activations only); C: mean, OC-SVM (activations only); D: mean, Mahalanobis (activations only); E: SNR

Baseline classifiers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D, E}	0.7471	-0.0079	0.8334	0.0150	0.7288	-0.0121	0.8729	0.0166
{B, C, D, E}	0.7525	-0.0025	0.8305	0.0121	0.7330	-0.0079	0.8803	0.0240
{A, C, D, E}	0.7159	-0.0391	0.8995	0.0811	0.7101	-0.0309	1.0000	0.1437
{A, B, D, E}	0.7440	-0.0110	0.8628	0.0444	0.7259	-0.0150	0.8958	0.0395
{A, B, C, E}	0.7503	-0.0047	0.8196	0.0013	0.7241	-0.0168	0.8891	0.0328
{A, B, C, D}	0.7298	-0.0252	0.8664	0.0480	0.7309	-0.0101	0.8680	0.0117
{C, D, E}	0.7225	-0.0325	0.8771	0.0587	0.7132	-0.0278	0.8988	0.0425
{B, D, E}	0.7478	-0.0072	0.8765	0.0581	0.7311	-0.0098	0.9033	0.0470
{B, C, E}	0.7550	0.0000	0.8184	0.0000	0.7275	-0.0135	0.8971	0.0408
{B, C, D}	0.7359	-0.0191	0.8647	0.0463	0.7378	-0.0032	0.8658	0.0095
{A, D, E}	0.7177	-0.0373	0.9003	0.0819	0.7116	-0.0293	1.0000	0.1437
{A, C, E}	0.6793	-0.0757	0.9541	0.1357	0.6658	-0.0751	1.0000	0.1437
{A, C, D}	0.7053	-0.0497	0.9008	0.0825	0.7036	-0.0374	1.0000	0.1437
{A, B, E}	0.7443	-0.0107	0.8555	0.0371	0.7148	-0.0262	0.9375	0.0811
{A, B, D}	0.7275	-0.0275	0.8765	0.0582	0.7311	-0.0098	0.8594	0.0030
{A, B, C}	0.7282	-0.0268	0.8648	0.0464	0.7273	-0.0137	0.8632	0.0069
{D, E}	0.7350	-0.0200	0.8665	0.0481	0.7215	-0.0195	0.9112	0.0549
{C, E}	0.6111	-0.1439	0.9554	0.1370	0.6119	-0.1291	0.9784	0.1220
{C, D}	0.7132	-0.0418	0.8939	0.0755	0.7118	-0.0292	0.8773	0.0210
{B, E}	0.7419	-0.0131	0.8595	0.0411	0.7167	-0.0243	0.9471	0.0908
{B, D}	0.7362	-0.0188	0.8656	0.0473	0.7410	0.0000	0.8563	0.0000
{B, C}	0.7337	-0.0213	0.8641	0.0457	0.7345	-0.0065	0.8607	0.0044
{A, E}	0.6669	-0.0881	0.9170	0.0986	0.6610	-0.0800	1.0000	0.1437
{A, D}	0.7065	-0.0485	0.8982	0.0798	0.7106	-0.0303	1.0000	0.1437
{A, C}	0.6588	-0.0962	0.9269	0.1085	0.6566	-0.0844	1.0000	0.1437
{A, B}	0.7223	-0.0327	0.8719	0.0535	0.7225	-0.0184	1.0000	0.1437

Table 25: OOD detection performance under high UE speed on layer B1-PRE. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015. A: mean, Euclidean, HDBSCAN; B: mean, cosine, k-NN (activations only); C: mean, OC-SVM (activations only); D: mean, Mahalanobis (activations only); E: SNR

Baseline classifiers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D, E}	0.8716	-0.0039	0.4228	0.0046	0.8620	-0.0061	0.4497	0.0111
{B, C, D, E}	0.8745	-0.0010	0.4246	0.0065	0.8642	-0.0039	0.4534	0.0147
{A, C, D, E}	0.8552	-0.0202	0.4614	0.0432	0.8519	-0.0162	0.5174	0.0787
{A, B, D, E}	0.8697	-0.0058	0.4364	0.0182	0.8602	-0.0079	0.4597	0.0211
{A, B, C, E}	0.8728	-0.0027	0.4182	0.0000	0.8591	-0.0090	0.4597	0.0211
{A, B, C, D}	0.8629	-0.0126	0.4390	0.0209	0.8632	-0.0049	0.4465	0.0078
{C, D, E}	0.8591	-0.0164	0.4514	0.0332	0.8536	-0.0145	0.4668	0.0281
{B, D, E}	0.8719	-0.0036	0.4466	0.0284	0.8629	-0.0052	0.4635	0.0248
{B, C, E}	0.8755	0.0000	0.4203	0.0021	0.8609	-0.0072	0.4637	0.0251
{B, C, D}	0.8662	-0.0093	0.4424	0.0242	0.8668	-0.0013	0.4453	0.0067
{A, D, E}	0.8562	-0.0193	0.4578	0.0396	0.8531	-0.0150	0.5133	0.0747
{A, C, E}	0.8215	-0.0540	0.5421	0.1239	0.8030	-0.0652	0.5946	0.1559
{A, C, D}	0.8498	-0.0257	0.4646	0.0465	0.8491	-0.0190	0.5151	0.0765
{A, B, E}	0.8680	-0.0075	0.4365	0.0183	0.8530	-0.0151	0.4814	0.0428
{A, B, D}	0.8614	-0.0141	0.4433	0.0252	0.8630	-0.0051	0.4402	0.0015
{A, B, C}	0.8616	-0.0138	0.4410	0.0228	0.8609	-0.0072	0.4453	0.0066
{D, E}	0.8658	-0.0097	0.4423	0.0241	0.8583	-0.0098	0.4689	0.0303
{C, E}	0.7737	-0.1017	0.5888	0.1707	0.7656	-0.1026	0.6313	0.1927
{C, D}	0.8546	-0.0209	0.4591	0.0409	0.8534	-0.0148	0.4538	0.0151
{B, E}	0.8674	-0.0081	0.4403	0.0221	0.8544	-0.0137	0.4863	0.0476
{B, D}	0.8660	-0.0095	0.4421	0.0239	0.8681	0.0000	0.4386	0.0000
{B, C}	0.8648	-0.0107	0.4441	0.0259	0.8647	-0.0034	0.4440	0.0054
{A, E}	0.8038	-0.0717	0.5539	0.1357	0.7742	-0.0939	0.6115	0.1728
{A, D}	0.8505	-0.0250	0.4581	0.0400	0.8532	-0.0150	0.5100	0.0713
{A, C}	0.8046	-0.0708	0.5296	0.1114	0.8040	-0.0641	0.5789	0.1403
{A, B}	0.8568	-0.0187	0.4454	0.0272	0.8573	-0.0108	0.5109	0.0723

Table 26: Average OOD detection performance over high delay spread and UE speed scenarios on within-layer fusion for layer B1-PRE. Mean over 5 independent repeats with random seeds. A: mean, Euclidean, HDBSCAN; B: mean, cosine, k-NN (activations only); C: mean, OC-SVM (activations only); D: mean, Mahalanobis (activations only); E: SNR

Baseline classifiers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D, E}	0.9232	-0.0104	0.5057	0.1210	0.9296	-0.0043	0.4651	0.1302
{B, C, D, E}	0.9263	-0.0073	0.4924	0.1077	0.9307	-0.0032	0.4651	0.1302
{A, C, D, E}	0.9191	-0.0146	0.4592	0.0744	0.9188	-0.0150	0.4028	0.0679
{A, B, D, E}	0.9162	-0.0175	0.6367	0.2520	0.9263	-0.0076	0.5065	0.1716
{A, B, C, E}	0.9210	-0.0127	0.5433	0.1586	0.9238	-0.0101	0.5273	0.1924
{A, B, C, D}	0.9234	-0.0103	0.5086	0.1239	0.9303	-0.0036	0.4643	0.1294
{C, D, E}	0.9285	-0.0052	0.3925	0.0078	0.9230	-0.0109	0.4017	0.0668
{B, D, E}	0.9203	-0.0133	0.5893	0.2046	0.9277	-0.0062	0.5065	0.1716
{B, C, E}	0.9245	-0.0092	0.5258	0.1411	0.9247	-0.0091	0.5282	0.1933
{B, C, D}	0.9250	-0.0087	0.5021	0.1174	0.9317	-0.0022	0.4643	0.1294
{A, D, E}	0.9223	-0.0114	0.4871	0.1023	0.9253	-0.0085	0.3790	0.0441
{A, C, E}	0.8352	-0.0985	0.8245	0.4398	0.8366	-0.0973	1.0000	0.6651
{A, C, D}	0.9233	-0.0104	0.4099	0.0251	0.9239	-0.0100	0.3731	0.0382
{A, B, E}	0.9145	-0.0192	0.6960	0.3113	0.9164	-0.0175	0.6306	0.2957
{A, B, D}	0.9159	-0.0178	0.6278	0.2430	0.9267	-0.0071	0.5045	0.1696
{A, B, C}	0.9207	-0.0130	0.5459	0.1612	0.9248	-0.0091	0.5285	0.1936
{D, E}	0.9337	0.0000	0.4147	0.0300	0.9310	-0.0028	0.3780	0.0431
{C, E}	0.7127	-0.2209	0.8985	0.5138	0.7210	-0.2128	0.8552	0.5203
{C, D}	0.9292	-0.0044	0.3847	0.0000	0.9290	-0.0048	0.3705	0.0356
{B, E}	0.9190	-0.0147	0.6498	0.2651	0.9178	-0.0160	0.6340	0.2991
{B, D}	0.9174	-0.0162	0.6188	0.2341	0.9284	-0.0055	0.5045	0.1696
{B, C}	0.9227	-0.0110	0.5352	0.1505	0.9263	-0.0076	0.5284	0.1934
{A, E}	0.8467	-0.0870	0.8079	0.4232	0.8380	-0.0958	1.0000	0.6651
{A, D}	0.9215	-0.0122	0.4959	0.1111	0.9339	0.0000	0.3349	0.0000
{A, C}	0.8199	-0.1138	0.8514	0.4667	0.8379	-0.0959	1.0000	0.6651
{A, B}	0.9148	-0.0188	0.6535	0.2688	0.9154	-0.0185	0.6387	0.3038

Table 27: OOD detection performance under high delay spread on layer B3-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.035. A: mean, Euclidean, HDBSCAN; B: mean, cosine, k-NN (activations only); C: mean, OC-SVM (activations only); D: mean, Mahalanobis (activations only); E: SNR

Baseline classifiers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D, E}	0.7707	-0.0050	0.9332	0.0615	0.7746	-0.0012	1.0000	0.1836
{B, C, D, E}	0.7757	0.0000	0.9241	0.0525	0.7757	0.0000	0.8279	0.0115
{A, C, D, E}	0.7550	-0.0207	0.8815	0.0099	0.7591	-0.0167	1.0000	0.1836
{A, B, D, E}	0.7627	-0.0130	0.9623	0.0906	0.7679	-0.0079	1.0000	0.1836
{A, B, C, E}	0.7673	-0.0084	0.9210	0.0494	0.7738	-0.0019	0.8627	0.0463
{A, B, C, D}	0.7612	-0.0145	0.9168	0.0452	0.7662	-0.0096	1.0000	0.1836
{C, D, E}	0.7641	-0.0116	0.8716	0.0000	0.7593	-0.0164	0.8469	0.0305
{B, D, E}	0.7699	-0.0058	0.9515	0.0799	0.7699	-0.0059	0.8164	0.0000
{B, C, E}	0.7735	-0.0022	0.9119	0.0403	0.7744	-0.0013	0.8338	0.0174
{B, C, D}	0.7628	-0.0129	0.9165	0.0449	0.7677	-0.0080	0.8741	0.0577
{A, D, E}	0.7554	-0.0203	0.8857	0.0141	0.7548	-0.0209	1.0000	0.1836
{A, C, E}	0.7351	-0.0406	0.9149	0.0433	0.7469	-0.0288	1.0000	0.1836
{A, C, D}	0.7284	-0.0474	0.9232	0.0516	0.7520	-0.0238	1.0000	0.1836
{A, B, E}	0.7538	-0.0219	0.9374	0.0658	0.7604	-0.0154	1.0000	0.1836
{A, B, D}	0.7518	-0.0239	0.9199	0.0483	0.7562	-0.0196	1.0000	0.1836
{A, B, C}	0.7643	-0.0114	0.9021	0.0305	0.7676	-0.0081	1.0000	0.1836
{D, E}	0.7679	-0.0079	0.8810	0.0094	0.7561	-0.0197	0.8431	0.0267
{C, E}	0.7359	-0.0398	0.8886	0.0170	0.7340	-0.0417	0.8904	0.0740
{C, D}	0.7325	-0.0432	0.9130	0.0414	0.7525	-0.0232	0.8725	0.0561
{B, E}	0.7646	-0.0111	0.8909	0.0193	0.7607	-0.0150	0.8272	0.0108
{B, D}	0.7527	-0.0230	0.9209	0.0493	0.7582	-0.0175	0.8898	0.0734
{B, C}	0.7663	-0.0094	0.9006	0.0290	0.7692	-0.0065	0.8783	0.0619
{A, E}	0.7079	-0.0678	0.9088	0.0372	0.7096	-0.0661	1.0000	0.1836
{A, D}	0.7224	-0.0533	0.9149	0.0433	0.7437	-0.0321	1.0000	0.1836
{A, C}	0.7194	-0.0564	0.9178	0.0462	0.7455	-0.0302	1.0000	0.1836
{A, B}	0.7549	-0.0209	0.8935	0.0219	0.7511	-0.0246	1.0000	0.1836

Table 28: OOD detection performance under high UE speed on layer B3-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.065. A: mean, Euclidean, HDBSCAN; B: mean, cosine, k-NN (activations only); C: mean, OC-SVM (activations only); D: mean, Mahalanobis (activations only); E: SNR

Baseline classifiers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D, E}	0.8470	-0.0041	0.7194	0.0874	0.8521	-0.0011	0.7325	0.1220
{B, C, D, E}	0.8510	0.0000	0.7083	0.0762	0.8532	0.0000	0.6465	0.0359
{A, C, D, E}	0.8370	-0.0140	0.6703	0.0382	0.8390	-0.0143	0.7014	0.0908
{A, B, D, E}	0.8394	-0.0116	0.7995	0.1674	0.8471	-0.0061	0.7532	0.1427
{A, B, C, E}	0.8441	-0.0069	0.7322	0.1001	0.8488	-0.0044	0.6950	0.0844
{A, B, C, D}	0.8423	-0.0087	0.7127	0.0806	0.8482	-0.0050	0.7322	0.1216
{C, D, E}	0.8463	-0.0047	0.6321	0.0000	0.8411	-0.0121	0.6243	0.0137
{B, D, E}	0.8451	-0.0059	0.7704	0.1383	0.8488	-0.0044	0.6614	0.0509
{B, C, E}	0.8490	-0.0020	0.7189	0.0868	0.8496	-0.0036	0.6810	0.0704
{B, C, D}	0.8439	-0.0071	0.7093	0.0773	0.8497	-0.0035	0.6692	0.0586
{A, D, E}	0.8388	-0.0122	0.6864	0.0543	0.8401	-0.0132	0.6895	0.0790
{A, C, E}	0.7852	-0.0659	0.8697	0.2377	0.7918	-0.0615	1.0000	0.3894
{A, C, D}	0.8258	-0.0252	0.6665	0.0345	0.8380	-0.0153	0.6866	0.0760
{A, B, E}	0.8342	-0.0169	0.8167	0.1846	0.8384	-0.0148	0.8153	0.2047
{A, B, D}	0.8338	-0.0172	0.7738	0.1418	0.8415	-0.0118	0.7523	0.1417
{A, B, C}	0.8425	-0.0085	0.7240	0.0920	0.8462	-0.0070	0.7642	0.1537
{D, E}	0.8508	-0.0003	0.6478	0.0158	0.8435	-0.0097	0.6106	0.0000
{C, E}	0.7243	-0.1267	0.8936	0.2615	0.7275	-0.1257	0.8728	0.2622
{C, D}	0.8309	-0.0202	0.6489	0.0168	0.8408	-0.0124	0.6215	0.0109
{B, E}	0.8418	-0.0092	0.7704	0.1383	0.8393	-0.0139	0.7306	0.1200
{B, D}	0.8351	-0.0160	0.7699	0.1378	0.8433	-0.0099	0.6971	0.0866
{B, C}	0.8445	-0.0065	0.7179	0.0859	0.8478	-0.0055	0.7033	0.0928
{A, E}	0.7773	-0.0738	0.8584	0.2263	0.7738	-0.0794	1.0000	0.3894
{A, D}	0.8219	-0.0291	0.7054	0.0733	0.8388	-0.0145	0.6675	0.0569
{A, C}	0.7696	-0.0814	0.8846	0.2526	0.7917	-0.0615	1.0000	0.3894
{A, B}	0.8348	-0.0162	0.7735	0.1414	0.8332	-0.0200	0.8194	0.2088

Table 29: Average OOD detection performance over high delay spread and UE speed scenarios on within-layer fusion for layer B3-POST. Mean over 5 independent repeats with random seeds. A: mean, Euclidean, HDBSCAN; B: mean, cosine, k-NN (activations only); C: mean, OC-SVM (activations only); D: mean, Mahalanobis (activations only); E: SNR

Baseline classifiers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D, E}	0.7268	-0.0485	0.8285	0.0794	0.6425	-0.0514	0.9231	0.0436
{B, C, D, E}	0.7229	-0.0524	0.8776	0.1286	0.6421	-0.0518	0.9299	0.0503
{A, C, D, E}	0.7346	-0.0407	0.8440	0.0949	0.6364	-0.0574	0.8795	0.0000
{A, B, D, E}	0.7578	-0.0175	0.7491	0.0000	0.6583	-0.0355	0.9244	0.0449
{A, B, C, E}	0.6160	-0.1593	0.9805	0.2314	0.6315	-0.0624	0.9213	0.0418
{A, B, C, D}	0.6441	-0.1312	0.8400	0.0910	0.6194	-0.0745	0.9319	0.0523
{C, D, E}	0.7231	-0.0522	0.8976	0.1485	0.6429	-0.0510	0.9363	0.0567
{B, D, E}	0.7601	-0.0152	0.8087	0.0596	0.6603	-0.0335	0.9065	0.0270
{B, C, E}	0.6131	-0.1622	0.9984	0.2493	0.6330	-0.0608	0.9304	0.0509
{B, C, D}	0.6396	-0.1357	0.8421	0.0931	0.6202	-0.0736	0.9316	0.0521
{A, D, E}	0.7753	0.0000	0.7666	0.0175	0.6303	-0.0636	0.9148	0.0353
{A, C, E}	0.6150	-0.1603	0.9751	0.2261	0.6202	-0.0736	0.9042	0.0247
{A, C, D}	0.6363	-0.1390	0.8438	0.0947	0.5709	-0.1229	0.9363	0.0568
{A, B, E}	0.5725	-0.2028	0.9284	0.1793	0.6293	-0.0645	0.9346	0.0551
{A, B, D}	0.6596	-0.1157	0.8620	0.1130	0.6517	-0.0421	0.9265	0.0469
{A, B, C}	0.5643	-0.2110	0.9468	0.1978	0.6147	-0.0791	0.9225	0.0430
{D, E}	0.7628	-0.0125	0.8803	0.1313	0.6408	-0.0530	0.8873	0.0078
{C, E}	0.6209	-0.1544	0.9186	0.1695	0.6277	-0.0661	0.9459	0.0664
{C, D}	0.6269	-0.1484	0.8600	0.1110	0.5643	-0.1295	0.9430	0.0634
{B, E}	0.5737	-0.2015	0.9759	0.2268	0.6304	-0.0635	0.9067	0.0271
{B, D}	0.6487	-0.1266	0.8770	0.1279	0.6517	-0.0421	0.9212	0.0417
{B, C}	0.5764	-0.1989	0.9563	0.2073	0.6147	-0.0791	0.9224	0.0428
{A, E}	0.5222	-0.2531	0.9458	0.1968	0.5000	-0.1938	0.9140	0.0344
{A, D}	0.6511	-0.1242	0.8619	0.1129	0.5795	-0.1144	1.0000	0.1205
{A, C}	0.5542	-0.2211	0.9318	0.1827	0.5556	-0.1382	0.9379	0.0584
{A, B}	0.5838	-0.1915	0.9417	0.1927	0.6938	0.0000	0.8823	0.0028

Table 30: OOD detection performance under high delay spread on layer OUT-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015. A: mean, Euclidean, HDBSCAN; B: mean, cosine, k-NN (activations only); C: mean, OC-SVM (activations only); D: mean, Mahalanobis (activations only); E: SNR

Baseline classifiers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D, E}	0.6593	-0.0449	0.8953	0.0214	0.6006	-0.0607	0.9239	0.0233
{B, C, D, E}	0.6591	-0.0451	0.9224	0.0486	0.6011	-0.0603	0.9197	0.0190
{A, C, D, E}	0.6657	-0.0386	0.8884	0.0145	0.5714	-0.0900	0.9070	0.0064
{A, B, D, E}	0.6920	-0.0123	0.8739	0.0000	0.6066	-0.0547	0.9294	0.0287
{A, B, C, E}	0.6106	-0.0936	0.9777	0.1038	0.6090	-0.0524	0.9182	0.0176
{A, B, C, D}	0.5538	-0.1504	0.9286	0.0547	0.5834	-0.0779	0.9272	0.0266
{C, D, E}	0.6607	-0.0436	0.9233	0.0495	0.5850	-0.0763	0.9633	0.0626
{B, D, E}	0.7009	-0.0033	0.8771	0.0033	0.6096	-0.0518	0.9099	0.0093
{B, C, E}	0.5978	-0.1064	0.9997	0.1258	0.6115	-0.0499	0.9228	0.0222
{B, C, D}	0.5584	-0.1458	0.9299	0.0560	0.5841	-0.0773	0.9283	0.0277
{A, D, E}	0.7042	0.0000	0.8742	0.0004	0.5652	-0.0962	1.0000	0.0994
{A, C, E}	0.5799	-0.1243	0.9754	0.1015	0.5388	-0.1226	0.9385	0.0379
{A, C, D}	0.5412	-0.1630	0.9301	0.0562	0.5040	-0.1574	0.9468	0.0461
{A, B, E}	0.6139	-0.0904	0.9309	0.0570	0.6103	-0.0511	0.9561	0.0554
{A, B, D}	0.5625	-0.1417	0.9316	0.0578	0.6069	-0.0545	0.9318	0.0311
{A, B, C}	0.5322	-0.1720	0.9580	0.0842	0.5938	-0.0676	0.9140	0.0134
{D, E}	0.7013	-0.0029	0.9029	0.0290	0.5806	-0.0807	0.9006	0.0000
{C, E}	0.5685	-0.1358	0.9231	0.0493	0.5513	-0.1101	0.9761	0.0754
{C, D}	0.5459	-0.1583	0.9296	0.0557	0.5104	-0.1509	0.9676	0.0670
{B, E}	0.6091	-0.0951	0.9577	0.0838	0.6132	-0.0481	0.9440	0.0434
{B, D}	0.5658	-0.1385	0.9344	0.0605	0.6069	-0.0544	0.9322	0.0316
{B, C}	0.5348	-0.1694	0.9612	0.0873	0.5939	-0.0674	0.9160	0.0154
{A, E}	0.5291	-0.1751	0.9448	0.0709	0.4970	-0.1643	0.9162	0.0155
{A, D}	0.5377	-0.1665	0.9430	0.0691	0.5231	-0.1383	1.0000	0.0994
{A, C}	0.4945	-0.2098	0.9509	0.0771	0.4501	-0.2113	0.9785	0.0779
{A, B}	0.5922	-0.1121	0.9449	0.0711	0.6614	0.0000	0.9203	0.0197

Table 31: OOD detection performance under high UE speed on layer OUT-POST. Mean over 5 independent repeats with random seeds. All standard deviations less than 0.015. A: mean, Euclidean, HDBSCAN; B: mean, cosine, k-NN (activations only); C: mean, OC-SVM (activations only); D: mean, Mahalanobis (activations only); E: SNR

Baseline classifiers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D, E}	0.6931	-0.0467	0.8619	0.0504	0.6215	-0.0561	0.9235	0.0303
{B, C, D, E}	0.6910	-0.0488	0.9000	0.0886	0.6216	-0.0560	0.9248	0.0315
{A, C, D, E}	0.7001	-0.0396	0.8662	0.0547	0.6039	-0.0737	0.8933	0.0000
{A, B, D, E}	0.7249	-0.0149	0.8115	0.0000	0.6325	-0.0451	0.9269	0.0336
{A, B, C, E}	0.6133	-0.1265	0.9791	0.1676	0.6202	-0.0574	0.9198	0.0265
{A, B, C, D}	0.5989	-0.1408	0.8843	0.0728	0.6014	-0.0762	0.9295	0.0362
{C, D, E}	0.6919	-0.0479	0.9105	0.0990	0.6139	-0.0637	0.9498	0.0565
{B, D, E}	0.7305	-0.0093	0.8429	0.0314	0.6349	-0.0427	0.9082	0.0149
{B, C, E}	0.6055	-0.1343	0.9990	0.1876	0.6223	-0.0553	0.9266	0.0333
{B, C, D}	0.5990	-0.1408	0.8860	0.0746	0.6021	-0.0755	0.9300	0.0367
{A, D, E}	0.7398	0.0000	0.8204	0.0089	0.5977	-0.0799	0.9574	0.0641
{A, C, E}	0.5975	-0.1423	0.9752	0.1638	0.5795	-0.0981	0.9214	0.0281
{A, C, D}	0.5888	-0.1510	0.8869	0.0755	0.5374	-0.1402	0.9415	0.0483
{A, B, E}	0.5932	-0.1466	0.9296	0.1182	0.6198	-0.0578	0.9453	0.0521
{A, B, D}	0.6111	-0.1287	0.8968	0.0854	0.6293	-0.0483	0.9291	0.0358
{A, B, C}	0.5483	-0.1915	0.9524	0.1410	0.6043	-0.0733	0.9183	0.0250
{D, E}	0.7321	-0.0077	0.8916	0.0801	0.6107	-0.0669	0.8940	0.0007
{C, E}	0.5947	-0.1451	0.9209	0.1094	0.5895	-0.0881	0.9610	0.0677
{C, D}	0.5864	-0.1534	0.8948	0.0833	0.5374	-0.1402	0.9553	0.0620
{B, E}	0.5914	-0.1483	0.9668	0.1553	0.6218	-0.0558	0.9253	0.0321
{B, D}	0.6072	-0.1325	0.9057	0.0942	0.6293	-0.0483	0.9267	0.0334
{B, C}	0.5556	-0.1842	0.9588	0.1473	0.6043	-0.0733	0.9192	0.0259
{A, E}	0.5257	-0.2141	0.9453	0.1339	0.4985	-0.1791	0.9151	0.0218
{A, D}	0.5944	-0.1454	0.9025	0.0910	0.5513	-0.1263	1.0000	0.1067
{A, C}	0.5243	-0.2154	0.9414	0.1299	0.5029	-0.1747	0.9582	0.0649
{A, B}	0.5880	-0.1518	0.9433	0.1319	0.6776	0.0000	0.9013	0.0081

Table 32: Average OOD detection performance over high delay spread and UE speed scenarios on within-layer fusion for layer OUT-POST. Mean over 5 independent repeats with random seeds. A: mean, Euclidean, HDBSCAN; B: mean, cosine, k-NN (activations only); C: mean, OC-SVM (activations only); D: mean, Mahalanobis (activations only); E: SNR

Baseline method	OC-SVM fusion	BH-style fusion
HDBSCAN (Table 36)	B1-PRE, B3-POST, OUT-POST, SNR	B1-PRE, B3-POST
k-NN (Table 39)	B1-PRE, B3-POST, OUT-POST, SNR	B1-PRE, B3-POST, OUT-POST, SNR
OC-SVM (Table 42)	B1-PRE, B3-POST, SNR	B1-PRE, B3-POST
Mahalanobis (Table 45)	B1-PRE, B3-POST, OUT-POST, SNR	B1-PRE, B3-POST
Within-layer (Table 48)	B1-PRE, B3-POST, SNR	B1-PRE, B3-POST, OUT-POST

Table 33: Best-performing across-layer fusions by average AUROC across OOD scenarios. Results are averaged over high-delay and high-speed conditions.

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.9536	0.0000	0.1483	0.0203	0.9160	-0.0141	0.2385	0.0430
{B, C, D}	0.8647	-0.0889	0.7688	0.6408	0.8381	-0.0921	1.0000	0.8045
{A, C, D}	0.9493	-0.0043	0.1280	0.0000	0.8873	-0.0428	0.2230	0.0275
{A, B, D}	0.9459	-0.0077	0.2003	0.0723	0.9161	-0.0141	0.2385	0.0430
{A, B, C}	0.9413	-0.0123	0.1731	0.0451	0.9301	0.0000	0.2148	0.0193
{C, D}	0.5222	-0.4314	0.9458	0.8179	0.5000	-0.4301	0.9140	0.7185
{B, D}	0.8467	-0.1069	0.8079	0.6800	0.8380	-0.0921	1.0000	0.8045
{B, C}	0.8471	-0.1065	0.9939	0.8660	0.8369	-0.0932	1.0000	0.8045
{A, D}	0.9406	-0.0130	0.1908	0.0628	0.8874	-0.0427	0.2230	0.0275
{A, C}	0.9258	-0.0278	0.1574	0.0294	0.9208	-0.0094	0.1955	0.0000
{A, B}	0.9312	-0.0224	0.2049	0.0769	0.9301	0.0000	0.2148	0.0193

Table 34: OOD detection performance under high delay spread using HDBSCAN classifiers with means and the Euclidean distance metric, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.7198	-0.0121	0.9051	0.0015	0.7009	-0.0088	1.0000	0.0838
{B, C, D}	0.7319	0.0000	0.9037	0.0000	0.7091	-0.0006	1.0000	0.0838
{A, C, D}	0.6851	-0.0469	0.9175	0.0139	0.6559	-0.0538	1.0000	0.0838
{A, B, D}	0.6986	-0.0334	0.9096	0.0059	0.7035	-0.0062	1.0000	0.0838
{A, B, C}	0.7022	-0.0298	0.9073	0.0036	0.6923	-0.0173	1.0000	0.0838
{C, D}	0.5291	-0.2028	0.9448	0.0411	0.4970	-0.2126	0.9162	0.0000
{B, D}	0.7079	-0.0241	0.9088	0.0051	0.7096	0.0000	1.0000	0.0838
{B, C}	0.7083	-0.0236	0.9939	0.0903	0.6955	-0.0142	1.0000	0.0838
{A, D}	0.6669	-0.0650	0.9170	0.0133	0.6610	-0.0487	1.0000	0.0838
{A, C}	0.6578	-0.0741	0.9277	0.0241	0.6527	-0.0569	1.0000	0.0838
{A, B}	0.6892	-0.0427	0.9344	0.0308	0.6923	-0.0173	1.0000	0.0838

Table 35: OOD detection performance under high UE speed using HDBSCAN classifiers with means and the Euclidean distance metric, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.8367	0.0000	0.5267	0.0039	0.8085	-0.0028	0.6193	0.0215
{B, C, D}	0.7983	-0.0384	0.8362	0.3135	0.7736	-0.0376	1.0000	0.4023
{A, C, D}	0.8172	-0.0195	0.5228	0.0000	0.7716	-0.0396	0.6115	0.0137
{A, B, D}	0.8222	-0.0145	0.5549	0.0322	0.8098	-0.0015	0.6193	0.0215
{A, B, C}	0.8217	-0.0150	0.5402	0.0174	0.8112	0.0000	0.6074	0.0097
{C, D}	0.5257	-0.3111	0.9453	0.4226	0.4985	-0.3127	0.9151	0.3173
{B, D}	0.7773	-0.0594	0.8584	0.3356	0.7738	-0.0374	1.0000	0.4023
{B, C}	0.7777	-0.0590	0.9939	0.4712	0.7662	-0.0450	1.0000	0.4023
{A, D}	0.8038	-0.0329	0.5539	0.0311	0.7742	-0.0370	0.6115	0.0137
{A, C}	0.7918	-0.0449	0.5425	0.0198	0.7868	-0.0245	0.5977	0.0000
{A, B}	0.8102	-0.0265	0.5697	0.0469	0.8112	0.0000	0.6074	0.0097

Table 36: Average OOD detection performance over high delay spread and UE speed scenarios using HDBSCAN classifiers with means and the Euclidean distance metric, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.9926	-0.0023	0.0312	0.0123	0.9924	-0.0024	0.0354	0.0116
{B, C, D}	0.9399	-0.0549	0.4035	0.3846	0.9460	-0.0488	0.3417	0.3179
{A, C, D}	0.9949	-0.0000	0.0189	0.0000	0.9944	-0.0005	0.0263	0.0025
{A, B, D}	0.9902	-0.0047	0.0375	0.0186	0.9893	-0.0056	0.0365	0.0127
{A, B, C}	0.9921	-0.0027	0.0306	0.0117	0.9929	-0.0020	0.0327	0.0089
{C, D}	0.5737	-0.4212	0.9759	0.9570	0.6304	-0.3645	0.9067	0.8829
{B, D}	0.9190	-0.0759	0.6498	0.6309	0.9178	-0.0770	0.6340	0.6102
{B, C}	0.9394	-0.0555	0.4103	0.3914	0.9514	-0.0435	0.3110	0.2872
{A, D}	0.9929	-0.0020	0.0211	0.0022	0.9922	-0.0027	0.0254	0.0016
{A, C}	0.9949	0.0000	0.0197	0.0008	0.9948	0.0000	0.0238	0.0000
{A, B}	0.9891	-0.0058	0.0407	0.0218	0.9900	-0.0049	0.0332	0.0094

Table 37: OOD detection performance under high delay spread using k-NN classifiers with means and the cosine distance metric, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.8054	0.0000	0.7762	0.0020	0.7881	-0.0094	0.8212	0.0541
{B, C, D}	0.7791	-0.0262	0.9074	0.1332	0.7792	-0.0182	0.8349	0.0678
{A, C, D}	0.7512	-0.0542	0.8319	0.0577	0.7285	-0.0690	0.9388	0.1717
{A, B, D}	0.8005	-0.0048	0.8032	0.0291	0.7717	-0.0258	0.8807	0.1136
{A, B, C}	0.7937	-0.0117	0.7742	0.0000	0.7975	0.0000	0.7671	0.0000
{C, D}	0.6091	-0.1962	0.9577	0.1835	0.6132	-0.1842	0.9440	0.1769
{B, D}	0.7646	-0.0408	0.8909	0.1168	0.7607	-0.0368	0.8272	0.0601
{B, C}	0.7751	-0.0303	0.8655	0.0913	0.7911	-0.0063	0.8193	0.0522
{A, D}	0.7419	-0.0635	0.8595	0.0853	0.7167	-0.0808	0.9471	0.1800
{A, C}	0.7397	-0.0657	0.8619	0.0877	0.7454	-0.0521	0.8469	0.0798
{A, B}	0.7830	-0.0224	0.8142	0.0401	0.7793	-0.0181	0.8400	0.0729

Table 38: OOD detection performance under high UE speed using k-NN classifiers with means and the cosine distance metric, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.8990	0.0000	0.4037	0.0013	0.8903	-0.0049	0.4283	0.0284
{B, C, D}	0.8595	-0.0394	0.6555	0.2531	0.8626	-0.0325	0.5883	0.1884
{A, C, D}	0.8730	-0.0259	0.4254	0.0230	0.8614	-0.0338	0.4825	0.0827
{A, B, D}	0.8954	-0.0036	0.4204	0.0180	0.8805	-0.0147	0.4586	0.0587
{A, B, C}	0.8929	-0.0061	0.4024	0.0000	0.8952	0.0000	0.3999	0.0000
{C, D}	0.5914	-0.3075	0.9668	0.5644	0.6218	-0.2734	0.9253	0.5255
{B, D}	0.8418	-0.0572	0.7704	0.3680	0.8393	-0.0559	0.7306	0.3307
{B, C}	0.8573	-0.0417	0.6379	0.2355	0.8712	-0.0239	0.5651	0.1653
{A, D}	0.8674	-0.0316	0.4403	0.0379	0.8544	-0.0408	0.4863	0.0864
{A, C}	0.8673	-0.0317	0.4408	0.0384	0.8701	-0.0251	0.4354	0.0355
{A, B}	0.8861	-0.0129	0.4275	0.0251	0.8847	-0.0105	0.4366	0.0367

Table 39: Average OOD detection performance over high delay spread and UE speed scenarios using k-NN classifiers with means and the cosine distance metric, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.8936	-0.0428	0.3077	0.0855	0.8791	-0.0402	0.3957	0.1114
{B, C, D}	0.7000	-0.2364	0.9575	0.7352	0.6842	-0.2351	0.8673	0.5830
{A, C, D}	0.9036	-0.0328	0.2790	0.0567	0.8954	-0.0239	0.3425	0.0582
{A, B, D}	0.9198	-0.0166	0.2698	0.0475	0.8997	-0.0195	0.3380	0.0537
{A, B, C}	0.8910	-0.0453	0.3035	0.0813	0.8850	-0.0342	0.3716	0.0873
{C, D}	0.6209	-0.3154	0.9186	0.6963	0.6277	-0.2915	0.9459	0.6616
{B, D}	0.7127	-0.2236	0.8985	0.6763	0.7210	-0.1982	0.8552	0.5709
{B, C}	0.6643	-0.2721	0.9385	0.7162	0.6654	-0.2538	0.8514	0.5671
{A, D}	0.9363	0.0000	0.2223	0.0000	0.9192	0.0000	0.2843	0.0000
{A, C}	0.8962	-0.0401	0.2800	0.0578	0.9021	-0.0172	0.3192	0.0349
{A, B}	0.9253	-0.0110	0.2574	0.0352	0.9133	-0.0059	0.3041	0.0197

Table 40: OOD detection performance under high delay spread using OC-SVM classifiers with means, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.7026	-0.0333	0.9517	0.0630	0.7041	-0.0299	0.8831	0.0225
{B, C, D}	0.7217	-0.0142	0.9590	0.0704	0.7169	-0.0171	0.8767	0.0162
{A, C, D}	0.6108	-0.1251	0.9680	0.0794	0.6018	-0.1322	0.9663	0.1058
{A, B, D}	0.7068	-0.0291	0.9232	0.0346	0.7192	-0.0148	0.8810	0.0204
{A, B, C}	0.6702	-0.0656	0.9323	0.0437	0.6865	-0.0475	0.8739	0.0133
{C, D}	0.5685	-0.1674	0.9231	0.0345	0.5513	-0.1828	0.9761	0.1155
{B, D}	0.7359	0.0000	0.8886	0.0000	0.7340	0.0000	0.8904	0.0299
{B, C}	0.6843	-0.0516	0.9413	0.0526	0.6961	-0.0379	0.8839	0.0233
{A, D}	0.6111	-0.1247	0.9554	0.0668	0.6119	-0.1221	0.9784	0.1178
{A, C}	0.5626	-0.1733	0.9470	0.0583	0.5631	-0.1709	0.9286	0.0680
{A, B}	0.6911	-0.0447	0.9276	0.0390	0.7102	-0.0238	0.8606	0.0000

Table 41: OOD detection performance under high UE speed using OC-SVM classifiers with means, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.7981	-0.0152	0.6297	0.0409	0.7916	-0.0202	0.6394	0.0571
{B, C, D}	0.7108	-0.1025	0.9583	0.3694	0.7005	-0.1112	0.8720	0.2897
{A, C, D}	0.7572	-0.0561	0.6235	0.0347	0.7486	-0.0632	0.6544	0.0721
{A, B, D}	0.8133	0.0000	0.5965	0.0077	0.8095	-0.0023	0.6095	0.0272
{A, B, C}	0.7806	-0.0327	0.6179	0.0291	0.7858	-0.0260	0.6227	0.0404
{C, D}	0.5947	-0.2186	0.9209	0.3320	0.5895	-0.2223	0.9610	0.3787
{B, D}	0.7243	-0.0890	0.8936	0.3048	0.7275	-0.0842	0.8728	0.2905
{B, C}	0.6743	-0.1390	0.9399	0.3510	0.6808	-0.1310	0.8676	0.2853
{A, D}	0.7737	-0.0396	0.5888	0.0000	0.7656	-0.0462	0.6313	0.0490
{A, C}	0.7294	-0.0839	0.6135	0.0247	0.7326	-0.0792	0.6239	0.0416
{A, B}	0.8082	-0.0051	0.5925	0.0037	0.8118	0.0000	0.5823	0.0000

Table 42: Average OOD detection performance over high delay spread and UE speed scenarios using OC-SVM classifiers with means, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.9941	-0.0025	0.0367	0.0187	0.9921	-0.0031	0.0452	0.0186
{B, C, D}	0.9389	-0.0576	0.2914	0.2734	0.9116	-0.0836	0.4924	0.4658
{A, C, D}	0.9953	-0.0013	0.0271	0.0091	0.9932	-0.0021	0.0383	0.0117
{A, B, D}	0.9956	-0.0010	0.0245	0.0065	0.9937	-0.0015	0.0339	0.0073
{A, B, C}	0.9933	-0.0033	0.0408	0.0228	0.9931	-0.0021	0.0385	0.0119
{C, D}	0.7628	-0.2337	0.8803	0.8623	0.6408	-0.3544	0.8873	0.8608
{B, D}	0.9337	-0.0629	0.4147	0.3967	0.9310	-0.0642	0.3780	0.3514
{B, C}	0.9267	-0.0699	0.4139	0.3959	0.9156	-0.0796	0.4799	0.4533
{A, D}	0.9966	0.0000	0.0180	0.0000	0.9952	0.0000	0.0266	0.0000
{A, C}	0.9943	-0.0022	0.0332	0.0152	0.9942	-0.0010	0.0315	0.0049
{A, B}	0.9954	-0.0011	0.0249	0.0068	0.9949	-0.0003	0.0275	0.0009

Table 43: OOD detection performance under high delay spread using Mahalanobis classifiers with means, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.7941	0.0000	0.7659	0.0000	0.7454	-0.0235	0.8342	0.0299
{B, C, D}	0.7938	-0.0003	0.7972	0.0312	0.7248	-0.0441	0.8549	0.0506
{A, C, D}	0.7678	-0.0263	0.7798	0.0139	0.7036	-0.0653	0.8547	0.0504
{A, B, D}	0.7863	-0.0078	0.8189	0.0529	0.7687	-0.0002	0.8043	0.0000
{A, B, C}	0.7495	-0.0446	0.8218	0.0559	0.7379	-0.0310	0.8721	0.0678
{C, D}	0.7013	-0.0927	0.9029	0.1370	0.5806	-0.1883	0.9006	0.0963
{B, D}	0.7679	-0.0262	0.8810	0.1150	0.7561	-0.0128	0.8431	0.0388
{B, C}	0.7326	-0.0615	0.8048	0.0388	0.7032	-0.0657	0.8882	0.0839
{A, D}	0.7350	-0.0590	0.8665	0.1006	0.7215	-0.0474	0.9112	0.1069
{A, C}	0.7004	-0.0936	0.8768	0.1109	0.6945	-0.0744	0.8949	0.0906
{A, B}	0.7577	-0.0364	0.8821	0.1162	0.7689	0.0000	0.8280	0.0237

Table 44: OOD detection performance under high UE speed using Mahalanobis classifiers with means, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.8941	0.0000	0.4013	0.0000	0.8687	-0.0132	0.4397	0.0206
{B, C, D}	0.8664	-0.0277	0.5443	0.1430	0.8182	-0.0637	0.6736	0.2546
{A, C, D}	0.8815	-0.0125	0.4035	0.0022	0.8484	-0.0335	0.4465	0.0274
{A, B, D}	0.8909	-0.0031	0.4217	0.0204	0.8812	-0.0007	0.4191	0.0000
{A, B, C}	0.8714	-0.0227	0.4313	0.0300	0.8655	-0.0164	0.4553	0.0362
{C, D}	0.7321	-0.1620	0.8916	0.4903	0.6107	-0.2712	0.8940	0.4749
{B, D}	0.8508	-0.0433	0.6478	0.2465	0.8435	-0.0383	0.6106	0.1915
{B, C}	0.8296	-0.0644	0.6093	0.2080	0.8094	-0.0725	0.6840	0.2649
{A, D}	0.8658	-0.0283	0.4423	0.0410	0.8583	-0.0236	0.4689	0.0498
{A, C}	0.8474	-0.0467	0.4550	0.0537	0.8444	-0.0375	0.4632	0.0442
{A, B}	0.8766	-0.0175	0.4535	0.0522	0.8819	0.0000	0.4277	0.0086

Table 45: Average OOD detection performance over high delay spread and UE speed scenarios using Mahalanobis classifiers with means, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.9635	-0.0235	0.0976	0.0737	0.9929	-0.0020	0.0397	0.0132
{B, C, D}	0.9028	-0.0842	0.4694	0.4455	0.9323	-0.0625	0.4303	0.4038
{A, C, D}	0.9564	-0.0307	0.0851	0.0612	0.9942	-0.0007	0.0322	0.0057
{A, B, D}	0.9865	-0.0006	0.0359	0.0120	0.9935	-0.0014	0.0347	0.0081
{A, B, C}	0.9583	-0.0288	0.0940	0.0701	0.9935	-0.0013	0.0358	0.0093
{C, D}	0.6846	-0.3024	0.9676	0.9437	0.7456	-0.2492	0.8402	0.8137
{B, D}	0.9170	-0.0700	0.5125	0.4886	0.9248	-0.0701	0.5017	0.4752
{B, C}	0.9064	-0.0806	0.3845	0.3606	0.9376	-0.0573	0.3769	0.3504
{A, D}	0.9871	0.0000	0.0239	0.0000	0.9949	0.0000	0.0265	0.0000
{A, C}	0.9550	-0.0321	0.0725	0.0486	0.9948	-0.0000	0.0279	0.0014
{A, B}	0.9798	-0.0072	0.0507	0.0268	0.9942	-0.0007	0.0306	0.0041

Table 46: OOD detection performance under high delay spread using within-layer fused classifiers, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.7958	-0.0116	0.8234	0.0790	0.8108	-0.0073	0.7409	0.0213
{B, C, D}	0.7743	-0.0330	0.9072	0.1628	0.8043	-0.0139	0.7876	0.0681
{A, C, D}	0.7446	-0.0627	0.9159	0.1714	0.7675	-0.0507	0.8048	0.0852
{A, B, D}	0.8073	0.0000	0.7922	0.0478	0.7966	-0.0215	0.7792	0.0597
{A, B, C}	0.7989	-0.0084	0.7444	0.0000	0.8182	0.0000	0.7195	0.0000
{C, D}	0.6141	-0.1932	0.9677	0.2233	0.6807	-0.1375	0.8864	0.1669
{B, D}	0.7777	-0.0296	0.8770	0.1326	0.7757	-0.0425	0.8749	0.1554
{B, C}	0.7919	-0.0154	0.7880	0.0436	0.8133	-0.0049	0.7784	0.0588
{A, D}	0.7430	-0.0643	0.8873	0.1429	0.7353	-0.0828	0.8532	0.1337
{A, C}	0.7531	-0.0542	0.8464	0.1020	0.7791	-0.0391	0.7763	0.0567
{A, B}	0.8037	-0.0036	0.7492	0.0048	0.8060	-0.0122	0.7497	0.0301

Table 47: OOD detection performance under high UE speed using within-layer fused classifiers, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

Layers fused	One-Class SVM				Conformal P-value based approach			
	AUROC	Δ AUROC	FPR95	Δ FPR95	AUROC	Δ AUROC	FPR95	Δ FPR95
{A, B, C, D}	0.8796	-0.0172	0.4605	0.0606	0.9019	-0.0040	0.3903	0.0126
{B, C, D}	0.8386	-0.0583	0.6883	0.2883	0.8683	-0.0375	0.6090	0.2313
{A, C, D}	0.8505	-0.0464	0.5005	0.1005	0.8808	-0.0250	0.4185	0.0408
{A, B, D}	0.8969	0.0000	0.4141	0.0141	0.8951	-0.0108	0.4069	0.0292
{A, B, C}	0.8786	-0.0183	0.4192	0.0193	0.9058	0.0000	0.3777	0.0000
{C, D}	0.6494	-0.2475	0.9676	0.5677	0.7132	-0.1927	0.8633	0.4856
{B, D}	0.8474	-0.0495	0.6948	0.2948	0.8502	-0.0556	0.6883	0.3106
{B, C}	0.8492	-0.0477	0.5863	0.1863	0.8755	-0.0304	0.5776	0.2000
{A, D}	0.8650	-0.0319	0.4556	0.0557	0.8651	-0.0407	0.4399	0.0622
{A, C}	0.8540	-0.0428	0.4595	0.0595	0.8870	-0.0189	0.4021	0.0244
{A, B}	0.8917	-0.0051	0.3999	0.0000	0.9001	-0.0058	0.3902	0.0125

Table 48: Average OOD detection performance over high delay spread and UE speed scenarios using within-layer fused classifiers, fused across layers. A: B1-PRE, B: B3-POST, C: OUT-POST, D: SNR

References

- [Adebayo *et al.*, 2020] Julius Adebayo, Michael Muelly, Ilaria Lliccardi, and Been Kim. Debugging tests for model explanations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS)*, December 2020.
- [Agrawal *et al.*, 1998] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27(2):94–105, June 1998.
- [Bau *et al.*, 2017] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3319–3327, July 2017.
- [Bau *et al.*, 2020] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. In *Proceedings of the National Academy of Sciences*, volume 117, pages 30071–30078, 2020.
- [Bäuerle *et al.*, 2022] Alex Bäuerle, Daniel Jönsson, and Timo Ropinski. Neural activation patterns: Visual explainability of learned concepts. eprint arXiv:2206.10611 [cs.LG], 2022.
- [Campello *et al.*, 2013] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *PAKDD*, 2013.
- [Chergui *et al.*, 2025] Hatim Chergui, Melike Erol-Kantarci, and Christos Verikoukis. *Explainable AI for Communications and Networking: Toward Responsible Automation*. Elsevier Science & Technology, 1st edition, 2025.
- [Chollet, 2017] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.
- [Ciravegna, 2025] Gabriele Ciravegna. *Shall We Stop Using End-to-End Learning? A Short-Survey of Recent Concept-Based Approaches*, pages 245–254. Springer Nature Singapore, 2025.
- [Cortes *et al.*, 2017] Corinna Cortes, Xavier Gonzalvo, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. AdaNet: Adaptive structural learning of artificial neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 874–883, August 2017.
- [Davies and Bouldin, 1979] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):224–227, 1979.
- [Dempster *et al.*, 1977] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- [Earl and Nicholson, 2021] Richard Earl and James Nicholson. *The Concise Oxford Dictionary of Mathematics*. Oxford University Press, 6 edition, 2021.
- [Endres and Schindelin, 2003] D.M. Endres and J.E. Schindelin. A new metric for probability distributions. *IEEE Trans. Inf. Theory*, 49(7):1858–1860, 2003.
- [Erhan *et al.*, 2009] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *Technical Report, Université de Montréal*, 01 2009.
- [Ester *et al.*, 1996] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.
- [European Commission, 2019] European Commission. Ethics guidelines for trustworthy AI, 2019. Accessed: 2025-05-15.
- [European Commission, 2024] European Commission. Proposal for a regulation for laying down harmonised rules on AI, 2024. Accessed: 2025-05-15.
- [Fiandrino *et al.*, 2022] Claudio Fiandrino, Giulia Attanasio, Marco Fiore, and Joerg Widmer. Toward native explainable and robust AI in 6G networks: Current state, challenges and road ahead. *Computer Communications*, 193:47–52, 2022.
- [Fong and Vedaldi, 2017] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457, October 2017.
- [Ghorbani *et al.*, 2019] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, December 2019.
- [Goldsmith, 2005] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.
- [Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [Gupta *et al.*, 2023] Avani Gupta, Saurabh Saini, and P J Narayanan. Concept distillation: Leveraging human-centered explanations for model improvement. In *Advances in Neural Information Processing Systems*, volume 36, pages 63724–63737, December 2023.
- [Halkidi and Vazirgiannis, 2008] Maria Halkidi and Michalis Vazirgiannis. A density-based cluster validity approach using multi-representatives. *Pattern Recognition Letters*, 29(6):773–786, 2008.
- [Halkidi *et al.*, 2001] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of intelligent information systems*, 17:107–145, 2001.

- [Han *et al.*, 2016] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. eprint arXiv:1510.00149 [cs.CV], 2016.
- [Hassan *et al.*, 2024] Bryar A. Hassan, Noor Bahjat Tayfor, Alla A. Hassan, Aram M. Ahmed, Tarik A. Rashid, and Naz N. Abdalla. From A-to-Z review of clustering validation indices. *Neurocomputing*, 601:128198, October 2024.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.
- [Honkala *et al.*, 2021] Mikko Honkala, Dani Korpi, and Janne M. J. Huttunen. DeepRx: Fully convolutional deep learning receiver. *IEEE TWC*, 20(6):3925–3940, 2021.
- [Hoydis *et al.*, 2021] Jakob Hoydis, Fayçal Ait Aoudia, Alvaro Valcarce, and Harish Viswanathan. Toward a 6G AI-native air interface. *IEEE Commun. Mag.*, 59(5):76–81, 2021.
- [Hoydis *et al.*, 2023] Jakob Hoydis *et al.* Sionna: An open-source library for next-generation physical layer research. eprint arXiv:2203.11854 [cs.IT], 2023.
- [Huyen, 2022] Chip Huyen. *Designing Machine Learning Systems*. O’Reilly Media, 2022.
- [Jaccard, 1901] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.
- [Jain and Dubes, 1988] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, USA, 1988.
- [Jain, 2010] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.
- [Jumper *et al.*, 2021] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.
- [Kim *et al.*, 2018] Been Kim *et al.* Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *ICML*, 2018.
- [Kohonen, 1990] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [Korpi *et al.*, 2021] Dani Korpi, Mikko Honkala, Janne M.J. Huttunen, and Vesa Starck. DeepRx MIMO: Convolutional MIMO detection with learned multiplicative transformations. In *IEEE International Conference on Communications (ICC 2021)*, 2021.
- [Lin, 1991] Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.
- [Lloyd, 1982] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [MacQueen, 1967] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
- [Magesh *et al.*, 2023] Akshayaa Magesh, Venugopal V. Veeravalli, Anirban Roy, and Susmit Jha. Principled out-of-distribution detection via multiple testing. *Journal of Machine Learning Research*, 24(378):1–35, 2023.
- [Marcel and Rodriguez, 2010] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of Torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 1485–1488, October 2010.
- [MarketsAndMarkets, 2024] MarketsAndMarkets. 6G market by component, communication infrastructure, end use, region—global forecast to 2035. <https://www.marketsandmarkets.com/Market-Reports/6g-market-213693378.html>, September 2024. Accessed: 2025-05-15.
- [McInnes and Healy, 2017] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *IEEE ICDMW*, 2017.
- [McInnes *et al.*, 2018] Leland McInnes, John Healy, and James Melville. Uniform manifold approximation and projection for dimension reduction. eprint arXiv:1802.03426 [stat.ML], 2018.
- [Molnar, 2022] Christoph Molnar. *Interpretable Machine Learning*. Leanpub, 2nd edition, 2022.
- [Moulavi *et al.*, 2014] Davoud Moulavi, Pablo A Jaskowiak, Ricardo JGB Campello, Arthur Zimek, and Jörg Sander. Density-based clustering validation. In *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM)*, pages 839–847, April 2014.
- [Ng *et al.*, 2001] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, 2001.
- [Nokia Bell Labs, 2024] Nokia Bell Labs. Envisioning a 6G future. https://www.nokia.com/sites/default/files/2024-12/nokia_bell_labs_envisioning_a_6g_future_ebook_en.pdf, December 2024. Accessed: 2025-05-15.
- [Nokia, 2025] Nokia. Nokia’s vision for the 6G era, 2025. Accessed: 2025-05-15.
- [Olah *et al.*, 2017] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017.
- [Olah *et al.*, 2020] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020.
- [Parzen, 1962] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

- [Pendyala and Choi, 2022] Varsha Pendyala and Jihye Choi. Concept-based explanations for tabular data. eprint arXiv:2209.05690 [cs.LG], 2022.
- [Rand, 1971] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.
- [Rosenblatt, 1956] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.
- [Rousseeuw, 1987] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.
- [Russakovsky et al., 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [Schmalwasser et al., 2025] Laines Schmalwasser, Niklas Penzel, Joachim Denzler, and Julia Niebling. Fastcav: Efficient computation of concept activation vectors for explaining deep neural networks. eprint arXiv:2505.17883 [cs.LG], 2025.
- [Scott, 1992] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 1992.
- [Selvaraju et al., 2017] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, October 2017.
- [Simonyan et al., 2014] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations (ICLR)*, April 2014.
- [Steinhaus, 1956] Hugo Steinhaus. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci.*, IV(C1, III):801–804, 1956.
- [Tse and Viswanath, 2005] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
- [Tuononen et al., 2025] Marko Tuononen, Dani Korpi, and Ville Hautamäki. Interpreting deep neural network-based receiver under varying signal-to-noise ratios. In *ICASSP*, 2025.
- [Tuononen, 2006] Marko Tuononen. Paikallishaku klusterointimenetelmänä (Local search as a clustering method). Master’s thesis, University of Joensuu, 2006.
- [Virmajoki, 2004] Olli Virmajoki. *Pairwise Nearest Neighbor Method Revisited*. PhD thesis, University of Joensuu, December 2004.
- [Yeh et al., 2022] Chih-Kuan Yeh, Been Kim, and Pradeep Ravikumar. Human-centered concept explanations for neural networks. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, volume 342, pages 337–352. IOS Press, 2022.
- [Yosinski et al., 2015] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *31st International Conference on Machine Learning (ICML)*, July 2015.
- [Yucek et al., 2005] T. Yucek, R.M.A. Tannious, and H. Arslan. Doppler spread estimation for wireless OFDM systems. In *IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, pages 233–236, 2005.
- [Zaidi et al., 2018] Ali Zaidi, Fredrik Athley, Jonas Medbo, Ulf Gustavsson, Giuseppe Durisi, and Xiaoming Chen. *5G Physical Layer: principles, models and technology components*. Academic Press, 2018.
- [Zhang et al., 1996] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, page 103–114, 1996.
- [Zhang et al., 2025] Hanyu Zhang, Xiting Wang, Chengao Li, Xiang Ao, and Qing He. Controlling large language models through concept activation vectors. eprint arXiv:2501.05764 [cs.CL], 2025.
- [Zhao et al., 2009] Xiaochuan Zhao, Tao Peng, Ming Yang, and Wenbo Wang. Doppler spread estimation by tracking the delay-subspace for OFDM systems in doubly selective fading channels. *IEEE Signal Processing Letters*, 16(3):212–215, 2009.
- [Zhu et al., 2025] Fenghao Zhu, Xinquan Wang, Chen Zhu, Tierui Gong, Zhaohui Yang, Chongwen Huang, Xiaoming Chen, Zhaoyang Zhang, and Mérouane Debbah. Robust deep learning-based physical layer communications: Strategies and approaches. *IEEE Network*, 2025.