



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN
INGEGNERIA ELETTRONICA, TELECOMUNICAZIONI E TECNOLOGIE
DELL'INFORMAZIONE

Ciclo 38

Settore Concorsuale: 09/E3 - ELETTRONICA

Settore Scientifico Disciplinare: ING-INF/01 - ELETTRONICA

THEORY AND APPLICATIONS OF PERVASIVE SIGNAL PROCESSING: FROM
SENSING TO TRUSTWORTHY INTELLIGENCE

Presentata da: Livia Manovi

Coordinatore Dottorato

Davide Dardari

Supervisore

Riccardo Rovatti

Co-supervisore

Mauro Mangia

Esame finale anno 2026

Dedicated to the sleepless

who never give up

Abstract

Abstract

Artificial Intelligence (AI) systems are increasingly deployed in safety-critical domains, yet their internal decision-making processes often remain opaque. This thesis presents a systematic approach for analysing the internal geometry of Deep Neural Networks (DNNs) and linking it to quantifiable reliability indicators. Adopting a signal-processing perspective, DNN layers are modelled as affine operators whose learned parameters generate structured, low-rank subspaces. By projecting activations onto coordinates derived directly from these trained parameters and clustering the resulting representations, compact semantic descriptors, termed 'peepholes', are constructed. These descriptors reveal how internal evidence is organised within the network, offering interpretable summaries of its decision-making process. At the single-layer level, structural agreement measures quantify the coherence between the internal activation geometry and the model's output decision. Extending this analysis across depth, distributed evidence maps trace the evolution of semantic support throughout the network hierarchy. A global, layer-wise structural consistency score is then derived from these maps, providing a numerical indicator of the coherence of the model's decision process. The framework is designed to minimise external probing and perturbation. Rather than altering inputs or introducing auxiliary models, it determines whether a decision remains consistent with the structure learned by the network itself. This methodology has been validated on representative convolutional and transformer architectures and examined in the context of satellite onboard anomaly detection, where confidence assessment must coexist with strict computational budgets and operational constraints. Deviations from nominal internal structure align with reliability-related outcomes across classification errors, distributional shifts, and adversarial perturbations. Since the analysis relies solely on forward activations and parameter-induced projections, it has limited computational requirements compared to methods based on repeated perturbations, stochastic sampling, auxiliary retraining or memory-intensive reference comparisons. This makes the framework well-suited to embedded and safety-critical scenarios. This work advances trustworthy AI by grounding confidence and robustness in learned network geometry.

Contents

Abstract	i
Introduction	3
I Foundations	9
1 From Signals to Structure	11
Introduction	11
1.1 Signals as Measurable Variation	11
1.2 Normality and the Covariance Operator	11
1.2.1 Principal Component Analysis	12
1.3 From Eigen-Decomposition to SVD	13
1.4 Low-Rank Structure and Optimal Approximation	17
1.5 Summary and Outlook	18
2 From Structure to Learning	19
Introduction	19
2.1 Structure Discovery in Machine Learning	19
2.2 The Curse of Dimensionality	20
2.3 Learning as Structure Estimation	21
2.4 Deep Neural Networks as Structured Operators	22
2.4.1 Layer Composition and Nonlinear Operators	23
2.4.2 Classification and Natural Confidence	24
2.4.3 Dimensional Hierarchy and the Curse Revisited	27
2.4.4 Extracting Structure from High Dimension	28
2.4.5 Affine Formulation and Layer Composition	29
2.4.6 From Structure to Interpretability	30

2.5	Summary and Outlook	32
II	The <i>Peephole</i> Framework	35
	Introducing <i>Peepholes</i>	37
	Related Work	39
	Identified Gap and Design Perspective	44
3	Single-Layer Peephole Extraction and Cluster-Based Accordance	47
	Introduction	47
3.1	Dimensionality Reduction via the Layer’s Affine Operator	48
3.2	Clustering of Corevectors	49
3.2.1	Rank Selection	51
3.3	Softmin Mapping	52
3.4	Clustering-Network Accordance (CNA)	53
3.5	CNA with High-Confidence (CNA-HC)	54
3.6	Offline and Online Computations	55
4	Distributed Peepholes and Layer-wise Consistency	59
	Introduction	59
4.1	The Peephole Extraction Block	60
4.1.1	Corevector Extraction	61
4.1.2	Low-Level Feature Extraction	62
4.1.3	Feature Mixing and Semantic Descriptor	62
4.2	Distributed Peepholes Across Layers	63
4.3	Layer-wise Evidence Maps, Conceptograms and Classification-Maps	64
4.4	Prototype Evidence Maps	66
4.5	Layer-wise Structural Consistency as a Confidence Signal	66
4.5.1	Global Similarity Between Sample and Prototype	67
4.5.2	Interpretation of the Consistency Score	68
III	Empirical Evaluation and Interpretations	71
	Introduction	73

5	Motivation and Background	75
5.1	Background Concepts	75
5.1.1	Confidence and Uncertainty	76
5.1.2	Out-of-Distribution Detection	76
5.1.3	Evaluation metrics	77
5.1.4	Existing approaches	77
5.1.5	Geometric perspective adopted in this work	77
5.2	From Distributional Shifts to Adversarial Perturbations	78
5.2.1	Adversarial Attacks and Robustness	78
5.3	Computational Efficiency	80
5.4	Positioning with Respect to the State of the Art	80
5.4.1	Summary of Evaluation Dimensions	81
6	Single-Layer Structural Agreement	83
6.1	Experimental Setup	83
6.1.1	Tuning the Peephole Dimensionality	84
6.1.2	CNA: Network-Clustering Accordance	85
6.2	CNA with Confidence Estimation	88
6.3	Discussion and Structural Interpretation	89
6.3.1	CNA as structural sanity check	89
6.3.2	Structural Behaviour under Progressive Corruption	90
6.3.3	CNA-HC as internal support intensity	92
6.3.4	Implications for trust and reliability	93
6.3.5	From empirical observation to structural principle	94
7	Layer-wise Analysis	97
7.1	Experimental Setup	97
7.1.1	Tuning Considerations	98
7.2	Tracing Decision Dynamics through Layer-wise Evidence Maps (CIFAR10)	99
7.3	Scaling Evidence Maps to Fine-Grained Classification (CIFAR100) . .	104
7.3.1	Nominal Case: Stable and Coherent Trajectory	104
7.3.2	Semantic Structure in VGG16 Classification Maps	104
7.3.3	Near-OOD Case: Output Stable, Internal Trajectory Degraded .	106
7.3.4	Adversarial Example: Imperceptible Change, Internal Collapse .	106
7.4	Quantitative Assessment of Model Trustworthiness	110
7.5	Confidence Estimation	110
7.6	Out-of-Distribution Detection	112

7.6.1	Near-Out-of-Distribution Behaviour	112
7.6.2	Far-Out-of-Distribution Detection	113
7.7	Adversarial Robustness	114
7.8	Computational Complexity	116
7.8.1	Offline Computational Requirements	116
7.8.2	Online Computational Requirements	117
7.8.3	Comparison with Baselines	118
7.8.4	Architectural Considerations	119
7.8.5	Final Remark on Efficiency	119
8	Applications in Constrained and Safety-Critical Domains: Autonomy in Space	121
8.1	From Thresholds to Learned Normality	122
8.1.1	Design Constraints for Onboard AI Systems	123
8.1.2	From Threshold Monitoring to Structural Assessment in Space Systems	123
8.1.3	Multivariate Structural Representation of Subsystem Behaviour	125
8.1.4	Model Families for Telemetry-Based Anomaly Detection	126
8.1.5	Empirical Findings Across Telemetry Scenarios	126
8.1.6	Autoencoders as Structural Models of Nominal Behaviour	127
8.2	Deployability on Space-Qualified Hardware	128
8.3	Toward Trustworthy Autonomy: Runtime Confidence	129
8.3.1	Boolean Abstraction and Operational Logic	129
8.3.2	Output-Level Confidence	131
8.3.3	Latent-Space Confidence	132
8.3.4	Deep Feature-Space Confidence via Peepholes	132
8.4	Interpretability for Telemetry Diagnostics via Peepholes	133
8.4.1	Experimental Setup and Peephole Construction	134
8.4.2	Semantic Interpretation of Peepholes	135
8.5	Discussion	137
IV	Synthesis and Future Outlook	139
9	Synthesis and Outlook	141
	Introduction	141
9.1	A Design Philosophy for Trustworthy AI	142
9.2	From Geometry to Operational Signals	142

9.3	Structural Diagnostics as an Engineering Instrument	143
9.4	From Model-Centric AI to AI Systems Engineering	144
9.5	Implications for Trustworthy Autonomy	145
9.6	Future Outlooks	145
	Conclusion	146
V	Appendix	149
A	Experimental and Implementation Details	151
A.1	Models	151
A.2	Dataset Overview	153
A.3	Layers Overview	153
A.4	Hyper-parameters and Tuning	154
A.4.1	Adversarial Generation Parameters	157
A.5	Additional Results on Confidence and Robustness	158
A.6	Software and Computational Environment	159
	Bibliography	161
	Acknowledgements	171

Prologue

It all begins with a signal: a fragment of the world translated into numbers. In signal processing, the task was clear: acquire, condition and interpret these fragments in order to describe the state of something real, be it a structure, a circuit or a phenomenon. We learnt how to listen to systems, filter out the noise and design chains of transduction that convert vibrations, voltages or photons into meaningful patterns. It was never just mathematics; it was a form of attention. Over time, the objects of that attention changed. Signals became data; patterns became models. The same mathematical processes — decomposition, projection and inference — that were once used to detect faults in aircraft wings or power lines now shape how we train neural networks and extract features from images. The domain expanded from the physical to the abstract. And as systems became more autonomous, another question arose: who monitors the model itself? This thesis emerged from that shift in focus. Initially, it followed the logic of health monitoring: defining what normality looks like, extracting its structure and detecting when a departure occurs. Then, the focus shifted to internal processes. The anomalies of interest were no longer mechanical deviations, but rather conceptual ones: the internal states of a learning system that were drifting away from the manifold that the system had created for itself. Could a model’s activations be treated as its vital signs? Could we observe the physiological traces of its reasoning? This intuition has guided the development of the methods presented in the following chapters. Rather than interrogating the model from the outside using common practices such as gradients or perturbations, we examine its internal transformations as signals. Carefully chosen projections and statistical summaries reveal how understanding emerges and evolves across depth, as well as how it fractures under stress. The *peephole*, the main character of this journey, is a fitting metaphor for this stance. Like the small aperture in a door, a peephole does not offer a complete view, but reveals enough to recognise coherence or detect contradiction. The mathematical apertures opened in this thesis manifest through low-rank descriptors, empirical posteriors and layerwise summaries, but they serve a similar purpose: they allow us to inspect how a neural network reasons and

whether its reasoning remains aligned with the structure it has learned.

No system exists in isolation. Every intelligent mechanism is bound by its physical infrastructure, defined by sensor fidelity, memory capacity, computational budgets, and rigid temporal deadlines. Consequently, true trustworthiness must account for these constraints. While resource-intensive explanations may suffice for cloud environments, they are often untenable for onboard satellite systems or power-constrained, safety-critical embedded hardware. The methods presented here bridge this gap, providing reasoning that is observable, compact, and certifiable without compromising operational integrity.

The reflections offered in this prologue are intended to serve as a compass. They introduce the perspective from which the technical developments of this thesis were conceived, namely viewing learning systems as signal processes whose internal transformations can be observed, summarised and relied upon. The thesis formalises this view by progressively moving from foundational tools to a complete framework for interpretable and dependable intelligence.

Introduction

From Signals to Meaning in the Age of Pervasive AI

Artificial Intelligence (AI) systems are increasingly embedded in operational environments where their decisions carry technical, economic, and societal consequences. No longer confined to research prototypes or cloud-based analytics, AI systems are integrated into infrastructures, industrial processes, scientific instruments, and autonomous platforms. In such contexts, performance alone is insufficient. What is required is *dependable operation*.

Recent regulatory and institutional initiatives reflect this shift. Frameworks such as the European AI Act [European Union \(2024\)](#); [High-Level Expert Group on AI \(2019\)](#) and the National Institute of Standards and Technology (NIST) AI Risk Management Framework [Tabassi \(2023\)](#) articulate requirements that extend beyond accuracy, calling for reliability, robustness, transparency, accountability, and demonstrable safety. Contemporary surveys on Trustworthy AI [Li et al. \(2023\)](#) similarly emphasize that next-generation AI systems must be not only powerful, but auditable, stable under perturbation, and aligned with operational constraints.

The central challenge is therefore no longer whether AI systems can perform well under nominal conditions, but whether they can be trusted to operate reliably when conditions deviate from the ideal.

In this context, and consistent with established frameworks and recent surveys, trustworthiness is understood not merely as a normative aspiration, but as the capacity of an AI system to exhibit reliable and robust behaviour that can be meaningfully assessed and monitored under operational constraints.

Achieving dependable operation requires more than high predictive accuracy. It requires understanding how and why decisions are produced, particularly when systems operate under uncertainty or outside nominal conditions. Yet modern Deep Neural Networks (DNNs), despite their remarkable performance across domains, remain inher-

ently difficult to interpret. The composition of millions of parameters through layered nonlinear transformations produces internal representations whose semantics are not directly observable.

This opacity introduces a fundamental tension: while performance improves through architectural complexity, interpretability often diminishes. In operational environments where decisions must be audited, validated, or justified, this tension becomes especially problematic.

In response to this challenge, the field of eXplainable AI (XAI) has emerged [Lipton \(2018\)](#) with the aim of developing methods that render learned models more transparent and understandable. A wide range of techniques has been proposed, including gradient-based saliency analyses, perturbation-based explanations, and surrogate models that approximate local decision boundaries.

While these approaches frequently provide valuable insight, many operate *post hoc*: they analyze model behavior externally, without directly interrogating the internal organization of learned representations. As noted in recent critiques of explainability research [Longo et al. \(2024\)](#), post hoc explanations may generate plausible narratives without guaranteeing structural faithfulness to the model’s actual reasoning process.

This observation motivates a complementary direction. Rather than approximating decisions from the outside, this thesis investigates whether interpretability can be grounded in the intrinsic geometric structure of the learned operators themselves.

Adopting such a structural perspective requires a precise characterization of what is meant by “internal organization.” In this thesis, AI systems are interpreted as *structured signal-processing chains*. Rather than treating a network as a monolithic function approximator, we examine it as a composition of operators acting on representations that can be analyzed geometrically.

Each layer of a neural architecture implements an affine transformation followed by a nonlinearity, progressively reshaping the geometry of the input space. Activations and latent representations can be interpreted as structured signals whose variation encodes learned regularities. The associated weight matrices and bias vectors define the operators that transform these signals across layers. From this viewpoint, a deep network can be understood as a cascade of transformations that redistribute energy and correlation across depth.

The singular value structure of these operators provides a principled lens through which this redistribution can be examined. Dominant modes capture the primary directions along which information is organized, while lower-energy components reflect redundancy or noise. Normality, in this context, corresponds to coherent geometric

trajectories within the learned representation manifold, whereas deviation manifests as structural misalignment across layers.

This operator-based interpretation enables a transition from viewing neural networks as opaque function approximators to analyzing them as layered geometric systems in which learned operators reshape structured signals across depth and whose internal coherence can, in principle, be quantified.

The objective of this thesis is not merely to produce explanations, but to enable *operational understanding* of when and why a model’s internal reasoning aligns with, or diverges from, its output.

Building on the perspective introduced above, the proposed framework performs a layer-wise structural analysis of neural representations. Compact low-rank descriptors are extracted from intermediate activations, namely the internal feature representations produced at successive layers of the network, capturing the dominant geometric organization of each layer. By relating these structural signatures to the model’s predicted decisions, internal coherence is operationalized as a measurable quantity.

When representations evolve consistently across depth and remain aligned with the final output, the system exhibits structural agreement. Conversely, when this alignment deteriorates, internal inconsistency becomes observable, even in cases where output-level confidence remains high.

From this analysis, quantitative trust indicators are derived directly from the geometry of learned operators. These indicators do not require retraining, architectural modification, or external surrogate explanations; they arise from intrinsic structural properties of the network itself.

In this sense, the framework acts as a form of structural instrumentation, exposing measurable signals of internal reasoning stability that can be monitored alongside conventional predictive outputs.

The structural instrumentation described above motivates a broader question: can internal geometric coherence relate to properties traditionally associated with dependability? Borrowing terminology from dependable computing [Avizienis et al. \(2004\)](#), where dependability encompasses attributes such as reliability, safety, and robustness, this thesis examines whether analogous properties can be meaningfully investigated in the context of learned AI systems.

Rather than treating trustworthiness as a purely normative requirement, the analysis explores whether measurable structural agreement within a network’s representations aligns with operationally relevant behaviours. In particular, potential connections are investigated between internal coherence and several dependability-related dimensions:

- **Reliability.** Whether consistent layer-wise trajectories correlate with predictive correctness under nominal conditions.
- **Robustness.** Whether internal coherence degrades in a systematic manner under distributional shift or adversarial perturbations.
- **Stability.** Whether the evolution of representations across depth reveals sensitivity to small input variations or structural collapse.
- **Operational compatibility.** Whether geometry-derived indicators can be computed without modifying the learned model and can support monitoring, auditability, and traceability in constrained or regulated environments.

These aspects are not assumed a priori. They are examined empirically throughout the thesis to assess whether internal structural consistency can function as a practical signal of dependable behaviour in AI systems.

Under this perspective, trustworthiness is not treated as an abstract label, but as a property that may emerge from measurable characteristics of the system’s internal organization.

The relevance of structure-based instrumentation becomes particularly evident in constrained and safety-critical domains. Autonomous space systems, for example, increasingly incorporate onboard learning components operating under strict energy, memory, and computational constraints [Barretta et al. \(2025\)](#). As learning-based modules transition from experimental demonstrations to operational subsystems, the reliability of their outputs becomes a matter of system-level safety.

In such contexts, automated decisions cannot be accepted solely on the basis of predictive performance. Mechanisms are required to assess whether a model’s output can be trusted in a given instance, particularly when operating outside nominal conditions. Confidence estimation and interpretability tools therefore play a central role in enabling responsible deployment.

Many post-hoc explainability techniques are well suited for offline analysis or ground-based validation, where repeated perturbations, sampling procedures, or surrogate modeling can be performed without stringent resource limitations. However, when inference must occur onboard under tight computational budgets, such approaches may become impractical.

In these environments, any confidence or interpretability mechanism must respect additional constraints. It should remain computationally lightweight, be intrinsically grounded in the model’s internal structure rather than rely on extensive external

sampling, and be compatible with monitoring and verification processes in systems where architectural modification or retraining is limited. These considerations further motivate the structural approach developed in this thesis.

The framework developed here has been validated in both benchmark image classification tasks and telemetry-based anomaly detection scenarios for spacecraft autonomy. These case studies illustrate how structural coherence analysis can support dependable operation without imposing substantial additional computational or architectural burden.

The preceding discussion leads to the central research questions of this thesis:

- Can internal geometric coherence serve as a faithful proxy for decision reliability in deep learning systems?
- Can such coherence be quantified in a computationally viable and architecture-agnostic manner?
- Can geometry-derived confidence indicators remain interpretable and informative under operational constraints and distributional stress?

These questions guide the theoretical development, methodological design, and empirical validation presented throughout the work.

The remainder of this thesis develops these questions progressively.

Part I (Foundations) establishes the mathematical framework underlying the proposed approach. Concepts from structured signal analysis, covariance operators, singular value decomposition, and low-rank approximation are introduced as tools for extracting geometric structure from high-dimensional representations.

Part II (The Peephole-Based Framework) formalizes the proposed methodology. It introduces compact representation descriptors, referred to as *peepholes*, and develops the mechanisms through which internal geometric coherence is quantified and translated into confidence indicators.

Part III (Empirical Evaluation and Interpretations) investigates the behaviour of the proposed indicators across benchmark classification tasks and telemetry-based anomaly detection scenarios. Through these studies, the relationship between internal geometric coherence and observable system behaviour is examined under both nominal and non-nominal conditions.

Part IV (Synthesis and Future Outlook) situates the findings within the broader discourse on trustworthy and dependable AI systems, discussing implications for operational deployment and safety-critical autonomy.

Together, these contributions articulate a unified perspective: AI systems can be analyzed and instrumented through their internal geometry. By transforming latent structural organization into measurable indicators, the thesis advances a framework in which interpretability, confidence estimation, and dependability are grounded in the intrinsic properties of learned representations.

I

Foundations

1

From Signals to Structure

Introduction

This chapter establishes the foundations for interpreting data as structured signals. We begin by formalizing what a *signal* is for the purposes of this discussion, move toward its statistical and geometric description through covariance analysis, and arrive at the Singular Value Decomposition (SVD) as a universal tool to represent and extract structure. This reasoning provides the mathematical bridge that will later connect signal representations to learning systems.

1.1 Signals as Measurable Variation

A signal is a measurable manifestation of a phenomenon. It carries information through *variation*: changes in amplitude, time, frequency, or other observable dimensions. Whether it is a voltage trace, a temperature field, or an activation map within a neural network, a signal can be formalized as a vector $\mathbf{x} \in \mathbb{R}^d$ lying in some feature space $\mathcal{X}$.

The collection of N such samples defines a dataset

$$\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N, \quad \mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d. \quad (1.1)$$

Our goal is to describe *structure*, i.e., the organized, reproducible part of variation within $\mathcal{D}$, separating it from stochastic or non-informative fluctuations. In signal processing, identifying structure amounts to finding coordinate systems where the essential information is compact and interpretable.

1.2 Normality and the Covariance Operator

Regularity in data manifests as correlations between its components. To capture these dependencies, we introduce the *covariance operator*, which expresses the second-order structure of a dataset.

For $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ with each sample $\mathbf{x}_i \in \mathbb{R}^d$, the empirical covariance is

$$\mathbf{C} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top, \quad \bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i. \quad (1.2)$$

Hence, $\mathbf{C} \in \mathbb{R}^{d \times d}$ is a symmetric and positive semi-definite operator that measures pairwise co-variations among the signal components.

The mean vector $\bar{\mathbf{x}}$ summarizes the *first-order statistics* of the data, that is, its average position in the feature space. In contrast, the covariance operator describes how individual components of the signal vary *together* around that mean. It is therefore a *second-order statistic*: while the mean measures overall level, the covariance measures the extent and orientation of variability. Each entry C_{ij} quantifies how variations along dimensions i and j coincide on average. Geometrically, $\mathbf{C}$ encodes the shape of the data cloud: a purely diagonal covariance indicates independent features, whereas strong off-diagonal terms reveal structured dependencies.

The eigen-decomposition of $\mathbf{C}$,

$$\mathbf{C} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^\top, \quad (1.3)$$

where $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_d]$ is an orthogonal matrix whose columns are the eigenvectors of $\mathbf{C}$, and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_d)$ is the diagonal matrix that collects the corresponding eigenvalues ordered such that $\lambda_1 \geq \lambda_2 \geq \dots \geq 0$, reveals a set of orthogonal directions forming a natural coordinate system for the data.

Geometrically, the covariance operator induces an ellipsoidal region aligned with $\mathbf{Q}$, whose semi-axes are proportional to $\sqrt{\lambda_i}$. The eigenvectors define the principal directions of variability, while the eigenvalues determine the extent of dispersion along each direction. As illustrated in Fig. 1.1, this ellipsoid represents a region of normal concentration for the data, so that samples lying outside it can be interpreted as deviations from the expected second-order structure.

This characterization defines the *geometry of normality*, namely the expected range and orientation of variation of the phenomenon under observation.

1.2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical method that formalizes the exploration of this geometry. It seeks a linear basis that captures the largest possible variance in the data through successive orthogonal projections. Practically, it diagonalizes the covariance operator $\mathbf{C}$: the first principal component aligns with $\mathbf{q}_1$, the direction of

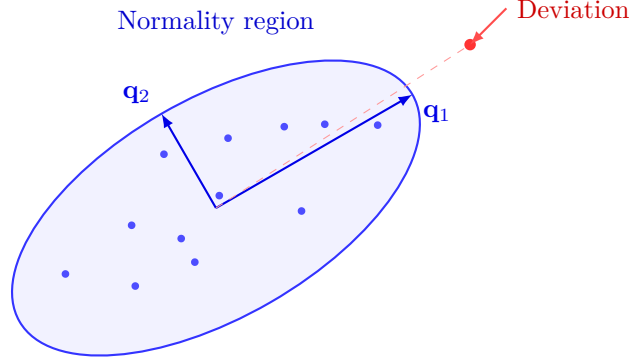


Figure 1.1. Covariance geometry: the data distribution (blue) is approximated by an ellipsoid whose principal axes correspond to the eigenvectors of $\mathbf{C}$. An outlier lies outside the region of normality.

maximum variance; the second aligns with $\mathbf{q}_2$, orthogonal to the first, and so on.

By retaining only the first k eigenvectors of $\mathbf{C}$, we obtain the projection matrix $\mathbf{Q}_{(:,1:k)} \in \mathbb{R}^{d \times k}$, which defines the k -dimensional principal subspace. Each sample $\mathbf{x}$ is represented by its projected coordinates

$$\mathbf{z} = \mathbf{Q}_{(:,1:k)}^\top (\mathbf{x} - \bar{\mathbf{x}}), \quad (1.4)$$

yielding a compact vector $\mathbf{z} \in \mathbb{R}^k$ that preserves the dominant modes of variability in $\mathcal{D}$.

This geometric interpretation is illustrated in Fig. 1.2, where samples in $\mathbb{R}^3$ are decomposed along the orthonormal directions $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ and projected onto the principal subspace spanned by the leading components.

PCA thus provides a parsimonious model of *normality*: most of the dataset’s variance concentrates in a low-dimensional subspace, while residual directions carry negligible or noisy information. From a geometric perspective, it identifies the axes of maximal energy; from an algebraic one, it factorizes $\mathbf{C}$ into its eigenvectors and eigenvalues.

This connection is crucial for what follows. The eigen-decomposition that supports Principal Component Analysis (PCA) applies only to symmetric, positive semi-definite matrices such as $\mathbf{C}$.

1.3 From Eigen-Decomposition to SVD

The eigen-decomposition introduced for the covariance operator applies only to *symmetric*, positive semi-definite matrices. However, most operators encountered in signal processing, such as linear filters, regression maps, or the weight matrices of neural

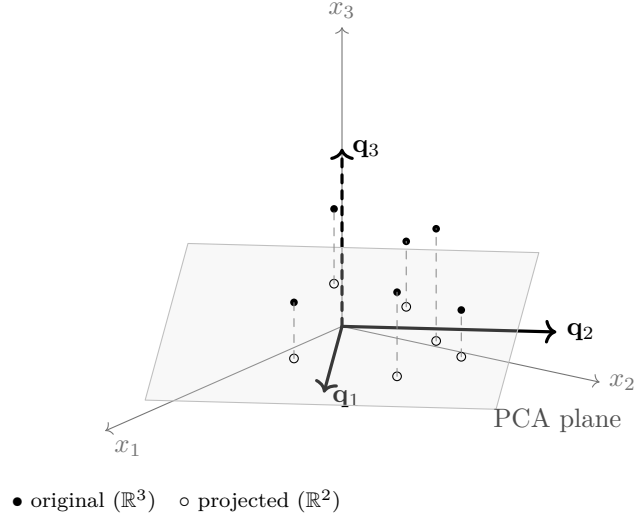


Figure 1.2. Illustration of PCA: samples in $\mathbb{R}^3$ are decomposed along the orthonormal directions $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$. The PCA projection retains the components on the plane spanned by $\mathbf{q}_1$ and $\mathbf{q}_2$, while discarding the residual component along $\mathbf{q}_3$.

networks, are generally *rectangular* and not symmetric. To extend the same geometric interpretation of “structure as orthogonal directions and magnitudes” to arbitrary operators, we resort to the *SVD*.

For any matrix

$$\mathbf{A} \in \mathbb{R}^{m \times n}, \quad (1.5)$$

there exist orthonormal matrices

$$\mathbf{U} \in \mathbb{R}^{m \times m}, \quad \mathbf{V} \in \mathbb{R}^{n \times n}, \quad (1.6)$$

and a rectangular diagonal matrix

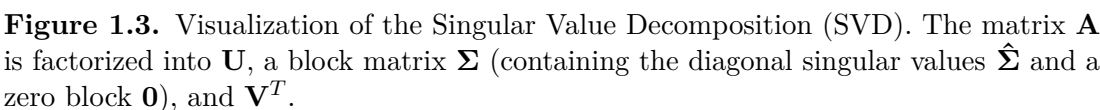
$$\mathbf{\Sigma} \in \mathbb{R}^{m \times n}, \quad (1.7)$$

whose diagonal entries $\sigma_1 \geq \sigma_2 \geq \dots \geq 0$ are called the *singular values* of $\mathbf{A}$, such that

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top. \quad (1.8)$$

The structural organization of this factorization, including the rectangular shape of $\mathbf{\Sigma}$ and its block-diagonal form, is summarized in Fig. 1.3.

If $\mathbf{A}$ has rank r , then only the first r singular values are non-zero. The decomposition


$$\mathbf{A} \approx \mathbf{U}_r \Sigma_r \mathbf{V}_r^\top, \quad (1.9)$$

with $\mathbf{U}_r \in \mathbb{R}^{m \times r}$, $\mathbf{\Sigma}_r \in \mathbb{R}^{r \times r}$, and $\mathbf{V}_r \in \mathbb{R}^{n \times r}$. The columns of $\mathbf{U}_r$ and $\mathbf{V}_r$ represent orthonormal bases for the output and input subspaces associated with the dominant directions of the transformation.

$$\mathbf{V}^\top \quad (\text{rotation of input}), \quad \mathbf{\Sigma} \quad (\text{axis scaling}), \quad \mathbf{U} \quad (\text{rotation of output}). \quad (1.10)$$

Connection to covariance analysis. When $\mathbf{A}$ is chosen as the centered data matrix

15

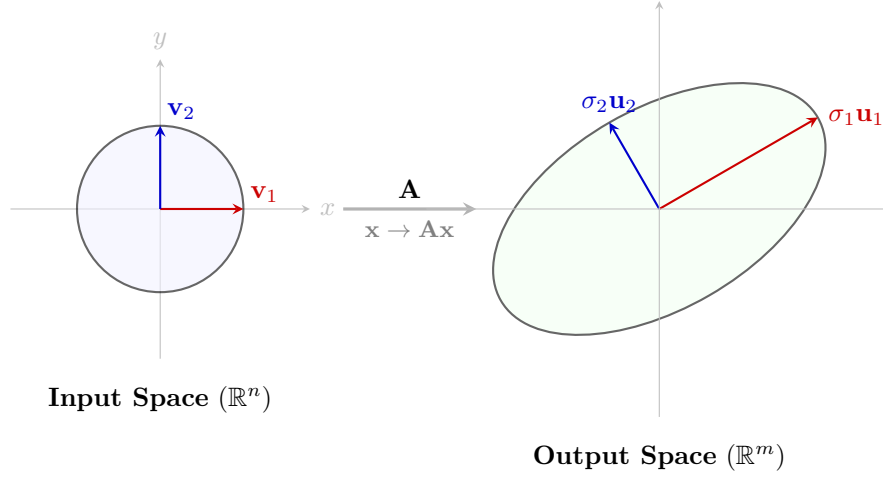


Figure 1.4. Geometric view of the SVD. The linear operator $\mathbf{A}$ maps the unit circle (input space) to a hyperellipse (output space). The singular values σ_i represent the lengths of the semi-axes of this ellipse, while the vectors $\mathbf{u}_i$ define their direction.

the empirical covariance can be written as

$$\mathbf{C} = \frac{1}{N} \mathbf{X} \mathbf{X}^\top. \quad (1.12)$$

Performing the SVD of $\mathbf{X}$,

$$\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top, \quad (1.13)$$

yields

$$\mathbf{C} = \frac{1}{N} \mathbf{U} \mathbf{\Sigma}^2 \mathbf{U}^\top. \quad (1.14)$$

Hence, the columns of $\mathbf{U}$ coincide with the eigenvectors of $\mathbf{C}$, and the squared singular values σ_i^2/N correspond to its eigenvalues λ_i . This shows that PCA can be seen as a particular case of SVD applied to the data matrix itself.

Interpretation. The SVD provides a universal algebraic lens for studying structure in linear transformations. While the covariance operator describes variability within data, the SVD describes how any transformation redistributes that variability between orthogonal input and output directions. In both cases, the underlying principle is the same: structure is revealed through orthogonality and energy concentration along a few dominant modes.

1.4 Low-Rank Structure and Optimal Approximation

A key feature of the SVD is that its truncation after the first k singular values yields the best possible rank- k approximation of $\mathbf{A}$ under any orthogonally (or unitarily) invariant norm, as established by the classical Eckart–Young–Mirsky theorem [Eckart & Young \(1936\)](#); [Mirsky \(1960\)](#); [Golub & Van Loan \(2013\)](#). Formally,

$$\mathbf{A}_k = \arg \min_{\text{rank}(\mathbf{B})=k} \|\mathbf{A} - \mathbf{B}\|_F, \quad \|\mathbf{A} - \mathbf{A}_k\|_F^2 = \sum_{i=k+1}^r \sigma_i^2, \quad (1.15)$$

where $\mathbf{A}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ is obtained by retaining the k dominant singular triplets.

This well-known result establishes that the truncated SVD provides the optimal rank- k approximation of $\mathbf{A}$ under any orthogonally invariant norm. The interpretation of such approximation as preserving the *essential structure* of $\mathbf{A}$, however, depends on how information is distributed across its singular spectrum.

When the singular values exhibit concentration, that is, when a subset of dominant modes carries a substantial fraction of the operator’s energy, the associated singular vectors span subspaces that effectively capture the principal directions along which the transformation acts. In these regimes, low-rank truncation isolates the dominant geometric structure of the operator while discarding components that contribute marginally in a global energy sense.

Conversely, when the singular spectrum is nearly flat, as may occur in unstructured noise-like transformations or uniformly distributed variance, no clear separation between dominant and residual modes exists. In such cases, low-rank approximation remains algebraically optimal but does not imply the presence of a strongly concentrated structural subspace. Moreover, in certain tasks (e.g., rare-event or anomaly modeling), informative variability may reside in low-energy directions and therefore requires careful interpretation beyond pure variance criteria.

Throughout this thesis, the notion of “structure as low rank” is therefore understood conditionally: low-rank projections serve as meaningful structural summaries when the learned operator exhibits spectral concentration, allowing dominant subspaces to act as faithful proxies of its intrinsic organization.

The perspective of analysing transformations through the geometry of their dominant subspaces will reappear in the following chapters, where it is applied directly to the learned affine operators of deep networks. In doing so, the SVD serves not merely as a dimensionality reduction tool, but as a canonical decomposition rooted in the network’s parameters themselves. It thus provides an intrinsic coordinate system through which

the organization of internal representations can be examined, bridging classical data geometry and the structured analysis of learned transformations.

1.5 Summary and Outlook

Signals are measurable expressions of phenomena, and their structure emerges from correlations that can be revealed by the covariance operator. PCA captures these correlations by projecting data onto principal axes of variance. The SVD extends this reasoning beyond covariance matrices to arbitrary linear transformations, providing a canonical geometric description of how information is distributed across orthogonal modes.

This operator-centered perspective forms the conceptual backbone of the thesis. In the following chapter, learning systems will be interpreted as compositions of data-driven affine operators, whose internal geometry can be examined through the same principles. Structure thus becomes not only a property of observed signals, but also of the learned transformations that encode and organize them.

2

From Structure to Learning

Introduction

The previous chapter framed signals as structured variability and showed how covariance, PCA, and the SVD expose geometry through principal axes and low-rank structure. We now move from *describing* structure to *constructing* it algorithmically: learning systems estimate operators that compress high-dimensional data into task-relevant subspaces, mitigating the curse of dimensionality. Deep networks realize this process as cascades of affine maps interleaved with non-linearities, a design that we will analyse layer-wise through their singular structure to obtain compact, interpretable representations.

Learning systems can be interpreted as data-driven mechanisms that *discover and exploit structure*. Where classical signal analysis relies on explicit operators, modern machine learning and neural networks learn such operators directly from examples. This chapter formalizes this perspective and establishes the bridge between linear structure modeling and the representation geometry underlying deep networks.

2.1 Structure Discovery in Machine Learning

As the complexity of data and tasks increases, so does the machinery designed to model them. Early analytical models in signal processing were *white boxes*: their parameters and functional forms were explicitly derived from physical or statistical principles, and their behaviour could be inspected and explained. Linear filters, regression models, and state-space representations belong to this family.

Machine learning generalizes this paradigm. Instead of prescribing the structure of the model, we *learn* it from examples by solving an optimization problem of the form

$$\min_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y}),$$

where $\boldsymbol{\theta}$ collects the model's parameters, $f_{\boldsymbol{\theta}}$ denotes its mapping from input to output, and $\mathcal{L}$ is a loss function measuring discrepancy from desired behaviour. This formulation covers a wide spectrum of models, from linear regression to neural networks, and defines learning as a search for parameters that minimize prediction error or maximize likelihood.

As models grow more flexible to handle nonlinearity, hierarchical features, and high-dimensional data, the explicit relationship between parameters and behaviour becomes harder to trace. Simple models such as linear regression remain transparent: their parameters directly map to the influence of each input. Kernel methods and decision trees are often considered *grey boxes*: interpretable in principle, though complex in practice. DNNs, with millions of parameters arranged in non-linear cascades, push this continuum toward the *black-box* regime: highly expressive, yet largely opaque. Understanding what each layer encodes and how it contributes to the final decision becomes challenging.

This opacity motivates the field of XAI, which seeks to restore visibility into the inner workings of learned models. If learning is a process that discovers structure, then a structured interpretation must exist, only hidden behind the layers' composition. The question becomes not whether such structure exists, but how to reveal it.

In this context, we can reinterpret a Neural Network (NN) as a composition of operators, each transforming information in a specific way. The weight matrices $\mathbf{W}$ and biases $\mathbf{b}$ define affine mappings, while the non-linearities introduce distortions that enhance expressivity but obscure direct interpretability. By studying the internal transformations of these operators, we aim to recover traces of *structure* that can shed light on the decision process itself.

2.2 The Curse of Dimensionality

While structure can be defined in any feature space, most real-world data inhabit spaces of extremely high nominal dimension. Let $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, with $\mathbf{x}_i \in \mathbb{R}^d$. As d grows, three fundamental effects appear:

1. **Sparsity.** The volume of a unit hypercube grows exponentially with d , so data points occupy only a vanishingly small portion of it. Even a large number of samples N becomes insufficient to densely populate the space.
2. **Distance concentration.** In high dimension, the ratio between the nearest and farthest neighbor distances tends to one, making proximity measures unreliable and reducing the discriminative power of distance-based algorithms.
3. **Statistical fragility.** Estimating moments, densities, or correlations in large d requires an exponentially increasing amount of data. Without structural assumptions, inference becomes ill-posed.

These phenomena collectively constitute the *curse of dimensionality*. They make both representation and inference challenging: data become sparse, similarity loses meaning, and statistical estimates grow unreliable. To counter this, we search for mappings that retain the informative variation of $\mathcal{D}$ while discarding redundant degrees of freedom. Formally, we seek a transformation

$$\mathbf{z} = \mathbf{T}(\mathbf{x}), \quad \mathbf{T} : \mathbb{R}^d \rightarrow \mathbb{R}^k, \quad k \ll d, \quad (2.1)$$

such that $\mathbf{z}$ preserves the essential geometry of $\mathcal{D}$. Linear projections such as PCA and more general nonlinear embeddings form the mathematical foundation of dimensionality reduction. From an operator viewpoint, these transformations estimate the subspace where normality and structure concentrate, enabling subsequent learning algorithms to operate efficiently and robustly.

However, in modern scenarios, $\mathbf{T}$ is often unknown and must itself be *learned* from data. This leads naturally to the formulation of learning as a process of *structure estimation*.

2.3 Learning as Structure Estimation

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, learning seeks a parametric mapping

$$f_{\boldsymbol{\theta}} : \mathbb{R}^d \rightarrow \mathbb{R}^m \quad (2.2)$$

that minimizes an expected loss

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y})], \quad (2.3)$$

where $\boldsymbol{\theta}$ denotes the trainable parameters and ℓ quantifies the discrepancy between predictions and targets. In this sense, learning can be viewed as a process of *estimating structure*: the model parameters define an operator that best reproduces the statistical dependencies within $\mathcal{D}$.

Supervised and unsupervised learning. Depending on the availability of target values, two main paradigms arise:

- **Supervised learning.** The dataset contains pairs $\mathcal{D}^{\text{train}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}$, where each input $\mathbf{x}_i$ is associated with a known output or label $\mathbf{y}_i$. The objective is to learn a function $f_{\boldsymbol{\theta}}$ that generalizes this input–output relationship. Classification and regression tasks belong to this family. Good generalization critically depends on

the diversity, balance, and quality of $\mathcal{D}^{\text{train}}$, since the model discovers structure only within the statistical manifold spanned by the training data.

- **Unsupervised learning.** The dataset $\mathcal{D}^{\text{train}} = \{\mathbf{x}_i\}$ consists of unlabeled samples. Here, learning aims to infer the latent regularities that define *normality*—for instance, by estimating clusters, manifolds, or generative models that capture the intrinsic structure of the data distribution. Dimensionality reduction, autoencoding, and density modeling are typical examples.

In both cases, the goal is to extract and represent the underlying structure that best explains the observations. Choosing appropriate data and regularization is fundamental: learning is a data-driven task, and the ability to discover meaningful structure depends on how well $\mathcal{D}^{\text{train}}$ reflects the intended operational domain.

Learning, generalization, and robustness. Once the model has been trained on $\mathcal{D}^{\text{train}}$, it is evaluated on new data $\mathcal{D}^{\text{test}}$ to assess how well the learned structure generalizes. Poorly chosen or unbalanced training data lead to fragile representations, while diverse and representative data promote robust structure extraction. From this perspective, a well-trained model approximates the operator that maps inputs to outputs along the manifold of normality, precisely the notion introduced earlier for signals.

2.4 Deep Neural Networks as Structured Operators

Deep Neural Networks (DNNs) can be interpreted as hierarchical compositions of operators that progressively transform data representations across layers. Within the framework developed so far, where structure and geometry are expressed through linear mappings and low-rank decompositions, they represent the most general and expressive class of learned operators. Rather than prescribing structure analytically, they *learn* it from examples, encoding statistical regularities of the data into a sequence of affine and nonlinear transformations. This section formalizes that view and establishes the link between network architecture and the operator-based perspective introduced in the previous chapters. We first describe the mathematical form of a deep network and its elementary components (Sec. 2.4.1), then connect it to the probabilistic formulation of classification and confidence estimation (Sec. 2.4.2). Finally, we discuss how the internal dimensional hierarchy and low-rank structure of layers are related to interpretability and the extraction of compact, meaningful representations. This forms the basis of the *peephole*-based analyses that are developed later in this thesis.

2.4.1 Layer Composition and Nonlinear Operators

A *feedforward network* refers to an architecture in which information flows unidirectionally from the inputs to the outputs through a finite sequence of transformations, with no recurrent or feedback connections. This distinguishes it from recurrent or dynamical models, which maintain state across time steps, and from energy-based formulations, in which outputs influence earlier activations. In a feedforward system, each layer receives input only from the previous layer and passes output only to the next layer. This enables clear operator composition, mapping inputs to outputs in a single forward pass.

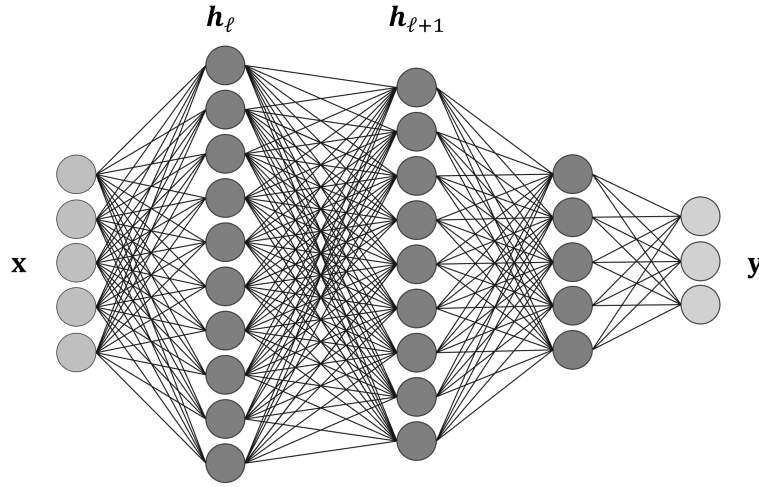


Figure 2.1. High-level schematic of a generic feedforward neural network with hidden layers.

Formally, a feedforward network with L layers (Fig. 2.1) implements a hierarchical composition of parametric transformations

$$f_{\theta}(\mathbf{x}) = f^{(L)} \circ \dots \circ f^{(1)}(\mathbf{x}), \quad (2.4)$$

where $\mathbf{h}^{(0)} = \mathbf{x}$ is the input and $\mathbf{h}^{(L)} = f_{\theta}(\mathbf{x})$ is the output.

Each layer applies an affine transformation followed by a nonlinearity:

$$\mathbf{h}_{\ell+1} = f_{\ell}(\mathbf{W}_{\ell}\mathbf{h}_{\ell} + \mathbf{b}_{\ell}), \quad \ell = 0, \dots, L-1, \quad (2.5)$$

where:

- $\mathbf{h}_{\ell} \in \mathbb{R}^{d_{\ell}}$ is the activation vector, input to layer ℓ ;
- $\mathbf{W}_{\ell} \in \mathbb{R}^{d_{\ell+1} \times d_{\ell}}$ and $\mathbf{b}_{\ell} \in \mathbb{R}^{d_{\ell+1}}$ are the trainable weight matrix and bias;

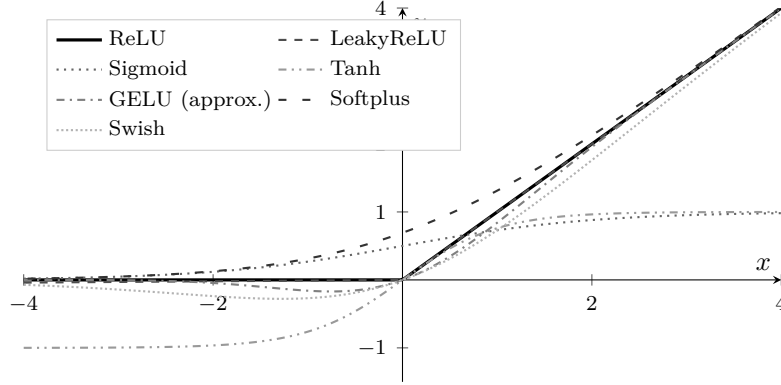


Figure 2.2. Common activation functions applied after an affine transform.

- $f_\ell : \mathbb{R}^{d_{\ell+1}} \rightarrow \mathbb{R}^{d_{\ell+1}}$ is an element-wise or channel-wise nonlinearity.

Thus $\mathbf{h}_{\ell+1} \in \mathbb{R}^{d_{\ell+1}}$ and the mapping $f_\ell(\cdot)$ defines the operator $f_\ell : \mathbb{R}^{d_\ell} \rightarrow \mathbb{R}^{d_{\ell+1}}$.

From a signal-processing perspective, each layer acts as a *linear filter bank*, captured by its affine operator which projects the feature vector $\mathbf{h}_\ell$ onto $d_{\ell+1}$ response channels, while f_ℓ introduces local nonlinear distortions that enrich expressivity and enable hierarchical abstraction.

The cascade of such operators progressively transforms the representation geometry and re-encodes structure at increasingly abstract levels of representation.

In the remainder of this section, we interpret these layers as structured mappings between feature spaces, characterized by their singular spectra and activation geometry. This provides a bridge between classical operator analysis and modern learning systems, allowing us to study how deep architectures internalize structure and how their internal consistency relates to interpretability.

2.4.2 Classification and Natural Confidence

When a network is trained for classification, its final layer transforms the last hidden representation $\mathbf{h}_{L-1}$ into a vector of *logits*

$$\mathbf{z} = \mathbf{W}_L \mathbf{h}_{L-1} + \mathbf{b}_L, \quad (2.6)$$

where

$$\mathbf{h}_{L-1} \in \mathbb{R}^{d_{L-1}}, \quad \mathbf{W}_L \in \mathbb{R}^{C \times d_{L-1}}, \quad \mathbf{b}_L \in \mathbb{R}^C,$$

and C is the number of classes. The logits $\mathbf{z} = [z_1, \dots, z_C]^\top$ represent unnormalized evidence for each class before probability normalization.

A *softmax* activation converts logits into class posteriors:

$$p(y = c \mid \mathbf{x}) = \frac{\exp(z_c)}{\sum_{j=1}^C \exp(z_j)}, \quad (2.7)$$

producing a categorical distribution $\mathbf{p}(\mathbf{x}) \in [0, 1]^C$ satisfying $\sum_c p(y = c \mid \mathbf{x}) = 1$.

The predicted label and its associated confidence are given by:

$$\hat{y}(\mathbf{x}) = \arg \max_c p(y = c \mid \mathbf{x}), \quad p_{\max}(\mathbf{x}) = \max_c p(y = c \mid \mathbf{x}). \quad (2.8)$$

The scalar $p_{\max}$, known as the *maximum softmax probability* (MSP) [Hendrycks & Gimpel \(2017\)](#), expresses how decisively the model favors the selected class over the alternatives. It provides a direct measure of the model’s self-assessed certainty for a given input and is widely used as a baseline confidence estimator.

Training typically minimizes the cross-entropy loss

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^N \log p(y_i \mid \mathbf{x}_i), \quad (2.9)$$

which encourages the network to assign high posterior probability to the correct class. However, high softmax confidence does not necessarily imply correctness, as it reflects the network’s internal competition among classes rather than the true likelihood of being right.

From Prediction to Uncertainty

Interpreting $p_{\max}$ as a measure of reliability implicitly assumes that a high posterior probability corresponds to a high chance of correctness. In practice, deep networks often display strong *overconfidence*: they may assign large $p_{\max}$ values to incorrect or, generally speaking, corrupted samples. This disconnect motivates the formal distinction between confidence and *uncertainty*.

Uncertainty can be broken down into two complementary components:

- **Aleatoric uncertainty**, which reflects intrinsic ambiguity or noise in the data and is irreducible even with infinite observations.
- **Epistemic uncertainty**, which arises from incomplete knowledge of model parameters and is reducible as training data or model capacity increase.

Following the taxonomy introduced by the authors [Kendall & Gal \(2017\)](#) and later formalized by [Hüllermeier & Waegeman \(2021\)](#), *aleatoric* uncertainty reflects stochastic

variability inherent in the observations and cannot be reduced even with more data, whereas *epistemic* uncertainty captures ignorance about the true model parameters and can in principle be reduced as training data increase.

In Bayesian formulations, these two forms of uncertainty correspond respectively to noise in the likelihood and variability in the posterior over model parameters. Let the predictive distribution be denoted by $p(y \mid \mathbf{x}, \mathcal{D})$, conditioned on the training dataset $\mathcal{D}$. For Bayesian models, uncertainty corresponds to the entropy of this distribution,

$$\mathcal{H}(y \mid \mathbf{x}, \mathcal{D}) = - \sum_{c=1}^C p(y = c \mid \mathbf{x}, \mathcal{D}) \log p(y = c \mid \mathbf{x}, \mathcal{D}), \quad (2.10)$$

which combines both aleatoric and epistemic contributions. In deterministic networks, this quantity collapses to the entropy of the softmax output $\mathbf{p}(\mathbf{x})$ from Eq. (2.7), often referred to as the *predictive entropy*.

A perfectly *calibrated* classifier would satisfy

$$\mathbb{P}[\hat{y}(\mathbf{x}) = y \mid p_{\max}(\mathbf{x}) = p] = p, \quad \forall p \in [0, 1], \quad (2.11)$$

meaning that predictions made with confidence p are correct in roughly $p\%$ of the cases. While training under cross-entropy encourages well-calibrated probabilities, the hierarchical nonlinearity of modern networks frequently distorts this correlation between confidence and accuracy, motivating *post-hoc estimators* that correct or reinterpret the softmax outputs. To quantify such deviations, metrics such as the *Expected Calibration Error* (ECE) Guo et al. (2017) are commonly employed and will be discussed later in Sec. 7.5. These shortcomings motivate alternative confidence estimators, including calibration methods like DOC Granese et al. (2021) and representation-based measures such as REL-U Dadalto et al. (2024), which attempt to better align probability outputs with empirical reliability.

Limitations of Output-Based Confidence

Although the MSP (Eq. 2.8) provides a convenient scalar indicator of belief, it carries no information about the internal process that generated a decision. Two samples with identical softmax confidence may correspond to entirely different internal representations, one coherent and one erratic, yielding the same outward certainty. Moreover, deep networks tend to remain highly confident even on inputs far from the training distribution, or on adversarially perturbed samples that lead to incorrect predictions. These limitations motivate the search for confidence measures that can incorporate

information about the network’s internal organization and reasoning stability, a direction further explored in the interpretability framework developed in the following chapters.

2.4.3 Dimensional Hierarchy and the Curse Revisited

The architecture of a deep network defines a *dimensional hierarchy* governed by the sequence of feature dimensions

$$d_0 \rightarrow d_1 \rightarrow \cdots \rightarrow d_L,$$

where d_0 is the input dimension and d_L corresponds to the output space. Each layer ℓ thus operates on a feature space $\mathcal{H}_\ell$ whose dimensionality depends on the layer type. For fully connected layers, $\mathcal{H}_\ell = \mathbb{R}^{d_\ell}$ with d_ℓ equal to the number of units. For convolutional or attention-based layers, the activation $\mathbf{h}_\ell$ is a tensor $\mathbf{h}_\ell \in \mathbb{R}^{C_\ell \times H_\ell \times W_\ell}$ comprising C_ℓ feature channels over a spatial grid of height H_ℓ and width W_ℓ . In such cases we denote the equivalent unrolled dimension as

$$d_\ell = C_\ell H_\ell W_\ell, \tag{2.12}$$

so that $\mathbf{h}_\ell$ can be represented as a vector in $\mathbb{R}^{d_\ell}$ when analyzing its linear transformations. This unified notation allows us to treat all layers types as mappings $f_\ell : \mathcal{H}_\ell \rightarrow \mathcal{H}_{\ell+1}$ with well-defined operator dimensions.

In classical signal analysis, high-dimensional domains suffer from the *curse of dimensionality*: data become sparse, distances lose discriminative meaning, and statistical estimation demands exponentially more samples. Deep architectures address this indirectly by factorizing the global mapping into a cascade of local transformations, each of limited dimension and receptive scope. Early layers act as low-level filters extracting generic primitives, while deeper ones synthesize these into abstract, task-specific representations. This progressive re-encoding concentrates variability along a reduced set of dominant directions, effectively lowering the intrinsic dimensionality of intermediate representations even when their ambient dimension remains large. In this sense, deep architectures do not eliminate the curse of dimensionality, but redistribute it by organizing information into structured subspaces.

However, the same hierarchical expressivity that enables generalization also reintroduces the curse at the parameter level. The number of free parameters scales roughly as

$$\#\boldsymbol{\theta} \approx \sum_{\ell=0}^{L-1} d_{\ell+1} d_\ell, \tag{2.13}$$

which grows combinatorially with depth and width. This rapid growth increases the expressive capacity of the network: the hypothesis class becomes sufficiently rich to approximate highly complex functions, including spurious patterns present in finite datasets. In regimes where the number of parameters greatly exceeds the amount of available supervision, the model can interpolate training data without necessarily uncovering the underlying generative structure. Such overparameterization may lead to memorization effects and to internal representations that are highly entangled, making it difficult to discern whether learned transformations reflect meaningful structure or incidental correlations. From a statistical learning perspective, this corresponds to operating in a regime where model capacity exceeds intrinsic data complexity, shifting the burden of generalization from explicit constraints to implicit regularization mechanisms.

Understanding how coherent structure nonetheless emerges within this vast parameter space therefore requires analyzing the geometry of the learned transformations $\mathbf{W}_\ell$ and their induced activations $\mathbf{h}_\ell$.

From an operator standpoint, each $\mathbf{W}_\ell$ maps a subspace of $\mathbb{R}^{d_\ell}$ into $\mathbb{R}^{d_{\ell+1}}$, redistributing energy and correlation among feature dimensions. Studying the spectral properties of these operators, such as their singular value distributions—provides insight into how information is compressed, amplified, or discarded through depth. This observation motivates the low-rank analyses developed in the following sections, where the notion of *structure extraction* is applied directly to neural operators.

2.4.4 Extracting Structure from High Dimension

The high dimensionality of neural activations is both a strength and a limitation. It enables the encoding of rich, distributed representations, but it also obscures how information is internally organized. Each layer transforms its input feature space $\mathcal{H}_\ell$ into the next one $\mathcal{H}_{\ell+1}$ through the linear operator $\mathbf{W}_\ell$ followed by a nonlinearity, as defined in Eq. (2.5). Even a single matrix $\mathbf{W}_\ell \in \mathbb{R}^{d_{\ell+1} \times d_\ell}$ can contain millions of parameters, making direct interpretation impractical.

A practical approach is to reduce these operators to their dominant subspaces. By applying the reasoning introduced for signal structure in Section 1.3, we can decompose each weight matrix $\mathbf{W}_\ell$ through its singular value spectrum:

$$\mathbf{W}_\ell = \mathbf{U}_\ell \mathbf{\Sigma}_\ell \mathbf{V}_\ell^\top, \quad (2.14)$$

where $\mathbf{U}_\ell \in \mathbb{R}^{d_{\ell+1} \times d_{\ell+1}}$ and $\mathbf{V}_\ell \in \mathbb{R}^{d_\ell \times d_\ell}$ are orthogonal, and $\mathbf{\Sigma}_\ell \in \mathbb{R}^{d_{\ell+1} \times d_\ell}$ is diagonal with non-negative singular values, we can examine how energy and correlation are

distributed among the learned transformation modes.

The leading singular vectors identify the dominant linear subspaces through which most of the energy of $\mathbf{W}_\ell$ is transmitted from $\mathbf{h}_\ell$ to $\mathbf{h}_{\ell+1}$. Truncating to the k largest singular values,

$$\mathbf{W}_\ell^{(k)} = \mathbf{U}_\ell^{(k)} \boldsymbol{\Sigma}_\ell^{(k)} \mathbf{V}_\ell^{(k)\top}, \quad k \ll \min(d_\ell, d_{\ell+1}), \quad (2.15)$$

yields the best rank- k approximation of $\mathbf{W}_\ell$ in the Frobenius norm, as discussed in Sec. 1.4. Importantly, this optimality holds at the level of the linear operator itself. In deep networks, however, each layer is embedded within non-linear activations, normalization mechanisms, and compositional interactions. As a consequence, the dominant singular directions of $\mathbf{W}_\ell$ do not necessarily coincide with the functionally most relevant directions of the overall mapping implemented by the network.

Nevertheless, the singular spectrum provides a principled linear proxy for assessing how information is distributed across transformation modes. Even when non-linear effects reshape activations, spectral concentration remains indicative of redundancy, compression, and structural bias within the learned operator. The low-rank form thus offers a diagnostic lens on the layer’s effective structure rather than a claim of optimal functional computation.

For convolutional layers, the same reasoning applies once the convolution is expressed in matrix form. Each kernel of size $K_h \times K_w$ with C_ℓ input and $C_{\ell+1}$ output channels can be unrolled into an equivalent Toeplitz matrix $\mathbf{W}_\ell^{(\text{conv})} \in \mathbb{R}^{d_{\ell+1} \times d_\ell}$, whose rows correspond to spatial receptive fields and columns to flattened input patches. Although this expansion is seldom performed explicitly in practice, it provides a conceptual bridge between convolutional and fully connected mappings, allowing their singular structures to be studied under a unified algebraic lens.

Analyzing the singular spectra $\boldsymbol{\Sigma}_\ell$ across layers offers insight into how information is compressed or amplified through depth. Sharp spectral decay suggests redundancy and strong feature correlation, while flatter spectra indicate diverse, decorrelated activations. This spectral perspective underlies the notion of *structure extraction*, forming the mathematical foundation for the low-rank analyses and layer-wise interpretability tools developed in the following sections.

2.4.5 Affine Formulation and Layer Composition

The transformations described so far can be compactly expressed in *affine form*, which explicitly combines weights and biases of some layer ℓ within a single operator. Introducing

the augmented activation vector

$$\bar{\mathbf{h}}_\ell = \begin{bmatrix} \mathbf{h}_\ell \\ 1 \end{bmatrix} \in \mathbb{R}^{d_\ell+1}, \quad (2.16)$$

and defining the augmented parameter matrix

$$\mathbf{A}_\ell = \begin{bmatrix} \mathbf{W}_\ell & \mathbf{b}_\ell \end{bmatrix} \in \mathbb{R}^{d_{\ell+1} \times (d_\ell+1)}, \quad (2.17)$$

the layer mapping in Eq. (2.5) can be rewritten as

$$\mathbf{h}_{\ell+1} = f_\ell(\mathbf{A}_\ell \bar{\mathbf{h}}_\ell). \quad (2.18)$$

This notation emphasizes that each layer applies a linear transformation in an augmented space followed by a nonlinearity acting element-wise on the result. The matrix $\mathbf{A}_\ell$ encapsulates both the geometric transformation of the feature space and the translation induced by the bias term. From a geometric standpoint, $\mathbf{A}_\ell$ maps affine subspaces of $\mathbb{R}^{d_\ell}$ onto affine subspaces of $\mathbb{R}^{d_{\ell+1}}$, allowing the overall network to be viewed as a composition of such affine mappings interleaved with nonlinear distortions.

The global forward mapping of the network can then be expressed recursively as

$$\mathbf{h}_L = (f_{L-1} \circ \mathcal{A}_{L-1}) \circ \cdots \circ (f_0 \circ \mathcal{A}_0)(\mathbf{x}), \quad (2.19)$$

where $\mathcal{A}_\ell(\cdot) = \mathbf{A}_\ell[\cdot; 1]$ denotes the affine transformation of layer ℓ . The successive application of these operators forms a nested sequence of affine projections and nonlinear expansions that progressively reshape the input manifold into a decision-oriented representation.

In this formulation, both $\mathbf{A}_\ell$ and its submatrix $\mathbf{W}_\ell$ can be analyzed through their singular value spectra, as in Eq. (2.14). This enables the study of how energy, correlation, and structure evolve not only through linear transformations but also through the affine composition as a whole. In subsequent sections, this operator-based view will serve as the foundation for layer-wise structural analysis, providing a bridge between classical linear algebra and modern deep learning interpretability.

2.4.6 From Structure to Interpretability

The formulation above allows a deep network to be viewed as a chain of affine operators, each acting on its own feature space and deforming the geometry inherited from the previous one. From the standpoint of structure analysis, this perspective reinterprets

learning as the progressive estimation of a sequence of subspaces that capture the dominant variations of the data in increasingly abstract coordinates.

The singular value decomposition of each operator $\mathbf{A}_\ell$,

$$\mathbf{A}_\ell = \mathbf{U}_\ell \mathbf{\Sigma}_\ell \mathbf{V}_\ell^\top, \tag{2.20}$$

reveals its principal transformation directions: the columns of $\mathbf{V}_\ell$ span orthogonal bases of input variation, those of $\mathbf{U}_\ell$ span the corresponding output subspaces, and the singular values in $\mathbf{\Sigma}_\ell$ quantify their relative importance. Truncating to the k largest singular values yields the best rank- k approximation of $\mathbf{A}_\ell$ in the Frobenius norm, retaining the transformations that convey most of the information flow through layer ℓ .

The evolution of these subspaces across depth describes how information is compressed, rotated, and re-expressed within the network. Early layers emphasize local correlations or low-level features, while deeper ones align with semantic directions associated with the task. Understanding how these representations evolve offers a path to reintroduce transparency into architectures that are otherwise treated as black boxes [Lipton \(2018\)](#); [Doshi-Velez & Kim \(2017\)](#).

The progression from statistical *normality* to learned structure thus becomes evident: where covariance analysis identifies the dominant modes of natural variability, deep networks internalize analogous subspaces through their weight operators. Each layer estimates and re-expresses what constitutes “normal” within its own feature domain, allowing the model to generalize from observed examples to new inputs that share the same structural manifold.

From an operator perspective, interpretability stems from exposing and quantifying the structure learnt by the network, rather than merely visualizing activations. By analyzing the principal components of the transformations $\mathbf{A}_\ell$ and by tracking their evolution across layers, we obtain a geometric description of the model’s internal reasoning process. This view resonates with the interpretability principles outlined in [Montavon et al. \(2018\)](#), bridging classical signal analysis, where operators reveal structure through their eigenmodes, with modern deep learning, where operators are learned but remain analyzable through the same algebraic principles. The next chapters will build upon this foundation. First, we introduce a compact representation of layer activations known as *peepholes*, through which local reasoning patterns can be observed and compared. Then, this idea will be extended into a full layer-wise framework linking internal structure to confidence, robustness, and trust in deep models.

2.5 Summary and Outlook

This chapter traced the evolution from the notion of a signal as a structured manifestation of variability to the modern view of a learning system as a hierarchy of operators that discover and internalize such structure from data.

We began by formalizing how regularity and correlation define *normality* through the covariance operator, whose eigen-decomposition exposes the intrinsic geometry of data. The SVD extended this reasoning to general transformations, providing a universal language to describe structure in terms of orthogonal modes of energy and variation. Learning, in this context, can be interpreted not as arbitrary function fitting, but as the progressive alignment of representations with dominant structural directions embedded in data and operators.

By expressing neural networks as cascades of affine operators interleaved with non-linearities, we established a bridge between classical signal operators and neural architectures. The dimensional hierarchy introduced by depth and width enables hierarchical feature abstraction, redistributing variability across structured subspaces rather than eliminating dimensional complexity. At the same time, parameter growth expands hypothesis capacity, making structural analysis necessary to distinguish meaningful organization from incidental complexity.

From an operator standpoint, the singular structure of learned transformations provides an intrinsic coordinate system for examining internal representations. Although optimality guarantees apply strictly to linear approximations of isolated operators, deep networks are composed of non-linear and interacting transformations. Consequently, dominant singular directions do not necessarily coincide with the functionally most relevant features of the global mapping.

Instead, the singular spectrum serves as a *structural probe* of the network’s internal geometry. It offers a principled diagnostic lens on how information is concentrated, amplified, or suppressed across depth. In this sense, interpretability emerges not as post-hoc explanation, but as the recovery and quantification of parameter-induced structure embedded within learned operators.

Crucially, the low-rank perspective developed in this chapter is not introduced as a claim of optimal functional computation. Rather, it provides an intrinsic coordinate system derived directly from the learned parameters. Projecting activations onto these dominant singular subspaces yields compact representations that remain anchored to the operator itself, rather than to external dimensionality-reduction heuristics. This parameter-induced projection establishes the methodological foundation for the

structural analyses that follow.

Rather than treating deep models as opaque mappings, we adopt a system-level perspective in which data, parameters, architecture, and transformations jointly define a structured computational process. Within this view, explainability becomes an exercise in structural analysis: making the hidden geometry of internal reasoning explicit and measurable [Lipton \(2018\)](#); [Doshi-Velez & Kim \(2017\)](#); [Montavon et al. \(2018\)](#).

The next chapter builds upon this foundation by introducing *peepholes*, compact, parameter-induced low-rank descriptors of layer activations derived from these operator-defined subspaces. Through peepholes, we will investigate how internal representations evolve across depth, how structural patterns can be compared and visualized, and how their consistency can serve as a quantitative proxy for reliability, confidence, and robustness in learned systems.

II

The *Peephole* Framework

Introducing *Peepholes*

From Layer Structure to Trust

The previous discussion presented learning systems as structured signal processors, i.e. models that extract and organise latent patterns rather than simply establishing input-output associations. It also suggested that the geometry encoded in their internal activations carries a form of implicit, distributed, yet systematic meaning that can be expressed, analysed and ultimately measured. Drawing on these principles, this chapter introduces the *Peephole* framework: a methodology designed to reveal this hidden geometry and link it to a quantifiable notion of trust.

The idea arose from an attempt to understand *where* information becomes meaningful within a deep model. Rather than probing the network from the outside, as is done in classical *post-hoc* methods based on saliency or perturbation, we look inward. Each layer is assumed to encode part of the reasoning process. By examining these fragments, we *see* how the network forms, refines and certifies its decisions.

The term *peephole* is used in its literal sense, referring to a small aperture in a door. A peephole never reveals the full scene beyond it, yet it discloses enough detail to discern whether what lies outside appears familiar, coherent, or suspicious. Following this analogy, the framework constructs *semantic apertures* within a deep network: low-dimensional interpretable summaries of layer activations do not fully reconstruct the internal state, but retain the features necessary to judge whether current reasoning remains compatible with the model’s learned notion of normality.

Although this approach is considered part of the broader category of post-hoc interpretability methods, its operational logic differs significantly from the mainstream. Gradient-based saliency maps highlight which pixels influence a decision, whereas perturbation-based approaches mask or modify the input in order to determine which regions are important. Although these techniques provide valuable qualitative evidence, they remain tied to pointwise explanations and do not provide an objective measure of

how closely a decision aligns with the model’s internal statistical structure. Peepholes take a contrasting trajectory: they analyse how internal representations *evolve*, cluster and organise semantically during normal operation. The resulting descriptors are quantitative objects that can be compared and aggregated to assess the coherence of the underlying reasoning.

This insight was first operationalised by modelling each layer as a structured operator. Through a Singular Value Decomposition (SVD) of its augmented affine map, the dominant axes of variation were isolated and collected into an intrinsic coordinate system, the *corevectors*. When a new sample is processed, its activation is projected onto this basis to produce a *peephole* vector: a compact yet faithful representation of the local geometry of the decision.

Clustering these corevectors across samples reveals how the network organises its internal evidence. For each layer, the empirical distribution of cluster memberships defines a mapping between activation modes and the network’s own output behaviour, yielding an *empirical posterior* that expresses how often each activation pattern supports a given prediction. This construction leads to the two measures introduced in the following chapter.

The first, Clustering-Network Accordance (CNA), quantifies how well the internal evidence of a layer aligns with the model’s predicted class. It is a structural agreement score: when the peephole of a layer falls into a region of the activation space that the model typically associates with the predicted class, the CNA score is high.

The second, Clustering-Network Accordance with High-Confidence (CNA-HC), isolates the subset of decisions whose internal support is not only aligned but also *consistent* with the model’s normal behaviour. It measures how sharply and unambiguously the empirical posterior supports the prediction. In this sense, CNA-HC acts as a *certification* criterion: it distinguishes decisions that are merely correct from those whose underlying activation geometry conforms to the model’s characteristic operating pattern. Decisions with low CNA-HC indicate diffuse or contradictory internal evidence, even when the output classification is correct.

As the analysis expanded to include multiple layers, a richer picture emerged. The early layers capture statistical regularities, the intermediate layers build abstract concepts and the later layers consolidate discriminative geometry. The proposed framework integrates these perspectives by aggregating layer-wise agreement to produce a global confidence score and by introducing *conceptograms*: visual maps that trace the evolution of semantic evidence through depth. Moving from a single viewpoint to multiple ones transforms isolated local snapshots into a coherent, distributed description of reasoning,

thus providing a structured basis for interpreting, comparing, and certifying decisions.

To guide the reader through the development of the proposed methodology, this chapter is organised into three interconnected components. We begin by positioning the work within the broader landscape of interpretability and post-hoc analysis, reviewing the main families of existing methods and outlining the limitations that motivate a structurally grounded alternative. We then retrace the conceptual path that led to the *Peephole* framework, starting from the single-layer analysis and its surrogate measures of internal coherence, and progressing toward the multi-layer formulation that underpins the notion of layer-wise consistency. This theoretical construction lays the foundation for the subsequent part of the thesis, where the approach is evaluated experimentally and its quantitative and qualitative implications are examined across architectures, datasets, and reliability scenarios. Together, these sections provide both the conceptual background and the methodological tools required to understand how internal structure, interpretability, and trust can be jointly analysed within deep learning systems.

Related Work

The study of interpretability in deep neural networks has evolved into a broad, multidisciplinary field. The approaches proposed in the literature vary widely in terms of their scope, methodological assumptions, and epistemic goals. Despite this diversity, most methods can be categorised within a small number of conceptual taxonomies, which clarify their relationships with each other and their position within the landscape of modern explainability research.

Taxonomies and Notions of Interpretability

Interpretability has been defined and categorised in many ways across the literature, reflecting the diversity of goals, users and assumptions involved in explaining machine learning models. Several foundational surveys propose taxonomies that clarify these dimensions.

[Lipton \(2018\)](#) distinguishes between *simulatability*, i.e. models small enough to be mentally executed, *decomposability*, which refers to components that can be individually interpreted and *algorithmic transparency*, that is learning procedures that are themselves understandable. The authors also separate *intrinsic* interpretability from *post-hoc* explanations. [Doshi-Velez & Kim \(2017\)](#) emphasise the need for a scientific framework for interpretability, noting that explanations serve different stakeholders—developers, end-users and regulators—who require clarity at very different levels of abstraction.

Murdoch et al. (2019) and Molnar (2020) offer broader overviews integrating classical statistical models and modern deep learning, while the work of Barredo Arrieta et al. (2020) explicitly links interpretability to responsible and trustworthy AI, highlighting the role of explanations in verification, auditing and risk mitigation. Moreover, Samek et al. (2021) provide a unifying perspective on explainability methods, organising them into attribution-based, concept-based and model-based families and discussing their theoretical foundations and practical limitations.

Across these taxonomies, interpretability methods are commonly classified along three axes:

Local vs. Global. *Local* explanations describe why a specific prediction was made for a particular input sample, usually by attributing importance to input features. *Global* explanations aim instead to characterise the overall behaviour of the model, such as its latent structure, feature hierarchy or decision boundaries.

Post-hoc vs. Intrinsic. *Post-hoc* methods analyse a trained model without altering its parameters, often through external probes, visualisations or surrogate approximations. *Intrinsic* approaches incorporate interpretability into the model architecture or training process, for instance through disentangled representations, attention mechanisms or symbolic components.

Model-agnostic vs. Model-specific. *Model-agnostic* methods operate independently of architecture and require only query access to the model. *Model-specific* methods exploit architectural details such as gradients, feature maps or layer connectivity.

The proposed framework is post-hoc and model-specific: it uses the trained parameters of the network to analyse the geometry of its internal representations. However, with respect to locality, it occupies a hybrid position: it provides *local, per-sample descriptors* that collectively reveal a *global structural organisation* across layers.

This conceptual framing informs the subsequent review of existing methods and situates the proposed methods within the landscape of post-hoc explainability. The principal families of interpretability techniques will be revisited through this lens hereafter.

Feature Attribution and Saliency Methods

The earliest and most widely adopted interpretability tools are feature attribution methods, which aim to quantify the contribution of input dimensions to a specific prediction.

Many techniques of this kind derive explanations from the gradients computed during *backpropagation*, the standard procedure used to compute parameter derivatives in feedforward networks as those described in Sec. 2.4.

Classical gradient-based saliency maps [Simonyan et al. \(2014\)](#) highlight pixels to which the output score is locally most sensitive. Using the chain rule through the layer-wise maps in Eq. (2.19), backpropagation recursively propagates local Jacobians starting from the output layer and large values of the computed gradients are interpreted as input dimensions to which the model is more sensitive. Variants such as Guided Backpropagation [Springenberg et al. \(2015\)](#) modify the backward pass at nonlinearities, suppressing negative contributions to obtain sharper and more localized maps.

Methods such as Grad-CAM [Selvaraju et al. \(2017\)](#) compute gradients with respect to convolutional feature maps, aggregates them into class-specific weights, and projects them back to the input space as coarse heatmaps. Integrated Gradients [Sundararajan et al. \(2017\)](#) is a technique that replace the plain gradient with a path integral from a baseline input to the target input. This aims to distribute the total change in prediction along the path in a way that satisfies certain axioms. SmoothGrad [Smilkov et al. \(2017\)](#) instead averages saliency maps over noisy versions of the input to reduce visual noise and improve robustness.

These approaches tend to be *post-hoc*, *local*, i.e., they return one explanation per input, and sensitive to choices of baselines, input scaling, or activation saturation. Their outputs typically take the form of heatmaps over the input space.

A parallel line of work replaces raw gradients with *relevance redistribution*, offering a distinct attribution principle. Through a technique called Layer-wise Relevance Propagation (LRP) [Bach et al. \(2015\)](#); [Montavon et al. \(2017\)](#), and its subsequent refinements [Montavon et al. \(2019\)](#); [Samek et al. \(2021\)](#), a relevance score is propagated backward through the layers of a Deep Neural Network (DNN) according to layer-specific redistribution rules, producing pixel-level heatmaps that reflect how the internal activations of a network contribute to the final output. Unlike gradient-based methods, which examine local sensitivities, LRP decomposes the prediction score itself and propagates it backwards layer by layer using relevance conservation rules. Each neuron receives a portion of the relevance proportional to its contribution to the activation of the next layer, yielding heatmaps that should reflect how evidence flows through the network. LRP thereby addresses some of the instability of plain gradients, but remains a *local*, *input-specific* explanation that can vary considerably depending on architectural choices and on how relevance is redistributed through nonlinearities. In fact, different parameterisations of functionally equivalent models may produce different explanation

maps, an issue extensively discussed in Montavon et al. (2019). Later extensions explored improved stability, robustness to noise, and concept-level decompositions Samek et al. (2021).

In all these cases, whether based on gradients, activation weighting or relevance redistribution, the explanation ultimately takes the form of a *heatmap of the input*. Such heatmaps are visually intuitive, especially in image classification, but their quantitative meaning is limited: two different parameter settings in LRP or two different baselines in Integrated Gradients may yield markedly different explanations for the same input. This sensitivity to methodological choices raises a fundamental concern: if the explanation depends more on the mechanism generating it than on the model itself, it becomes unclear whether these methods truly capture the internal reasoning of the network.

The objective of the *Peephole* framework is precisely to avoid this ambiguity. Rather than relying on externally imposed heuristics, it derives explanations from the network’s *internal geometry*, i.e., from how activations naturally organise in the feature spaces learned during training. The resulting descriptors are stable, low-dimensional summaries with a clear interpretation and, crucially, form the basis for quantitative measures of internal coherence and trust.

Perturbation-based and Model-agnostic Explanations

A complementary line of work uses perturbations of the input to infer the influence of each feature on the output. LIME Ribeiro et al. (2016) fits sparse linear surrogates around individual predictions by sampling perturbed instances and learning a local regression model that approximates the classifier in that neighbourhood. SHAP Lundberg & Lee (2017) provides a unified framework grounded in Shapley values, attributing contributions to features based on cooperative game theory. These methods approximate the decision function locally by querying the model under a large number of perturbations. They do not require access to gradients or internal representations, making them broadly applicable. Nevertheless, their explanations are heavily dependent on perturbation schemes, masking strategies, and sampling distributions, and they may require thousands of queries to achieve stability.

Therefore, these approaches are *local* and largely *model-agnostic*: they treat the network as a black box queried through evaluations. They provide flexible attributions and are applicable beyond deep learning, but they require large numbers of forward passes, depend on the perturbation distribution, and crucially remain confined to the input space. They do not characterise how internal feature spaces are organised or how the network’s *reasoning manifold* behaves under normal operation.

Concept-Based and Representation-Oriented Approaches

Other methods seek to interpret a network by analysing its internal activations. TCAV [Kim et al. \(2018\)](#) introduces Concept Activation Vectors to quantify how sensitive a prediction is to human-defined concepts, learned from concept vs. non-concept examples. ACE [Ghorbani et al. \(2019\)](#) automates part of this process by clustering image segments into candidate concepts before applying TCAV. Network Dissection [Bau et al. \(2017\)](#) takes a unit-centric perspective, measuring the alignment between individual neurons and semantic attributes extracted from dense pixel-level annotations. Although these techniques provide semantically meaningful insights, they require external concept sets, curated datasets, or additional supervision.

Representation-based approaches, including PCA, ICA, clustering of activations, or manifold-geometry analysis, highlight structural regularities inside the network [Olah et al. \(2017\)](#); [Maennel et al. \(2020\)](#). However, they typically stop at qualitative inspection and do not offer numerical measures relating internal structure to confidence or trust.

These methods provide a more *global* view of what internal units or directions mean in human terms, but they require external concept definitions or annotated datasets. They are therefore constrained by the availability and quality of semantic labels and do not directly quantify whether a given decision is *well supported* by the network’s internal statistics.

Intrinsic Explanations and Self-explaining Models

A different strategy is to embed the explanation mechanism into the model architecture. Self-Explaining Neural Networks [Alvarez-Melis & Jaakkola \(2018\)](#) learn a decomposition of predictions into interpretable basis concepts and associated relevance scores, effectively coupling prediction and explanation. Attention mechanisms [Bahdanau et al. \(2015\)](#) have also been interpreted as providing intrinsic explanations by highlighting where the model “looks”. However, several works [Jain & Wallace \(2019\)](#) have shown that attention weights, by themselves, do not guarantee faithful explanations.

These intrinsic models are valuable when one can design and train a system from scratch under interpretability constraints. In many practical scenarios, however, one must analyse large pre-existing architectures (e.g. vision transformers or high-capacity CNNs) without modifying their training pipeline. In such cases, post-hoc methods remain necessary.

Representation Geometry and Structural Analysis

A separate body of work studies the geometry of learned representations using tools from linear algebra and statistics. SVCCA [Raghu et al. \(2017\)](#), CCA-based analyses [Morcos et al. \(2018\)](#) and related similarity metrics [Kornblith et al. \(2019\)](#) compare the subspaces spanned by different layers or different networks, revealing convergence phenomena and shared structure. The Neural Collapse phenomenon [Papayan et al. \(2020\)](#) characterises a highly structured regime in which class means and last-layer weights converge to a simplex configuration. At the architectural level, [Sedghi et al. \(2018\)](#) study the spectrum of convolutional layers, exploiting their Toeplitz structure to analyse singular values at scale.

These works treat activations and weights as geometric objects, providing a *global* understanding of how representations are organised. However, they do not typically bridge this analysis to a per-sample, post-hoc notion of confidence or trustworthiness. They answer “how does the model represent?” but not “for this specific input, is the decision supported by the internal structure the model has learned?”.

Identified Gap and Design Perspective

Despite significant progress in post-hoc explainability, a tension remains between interpretative richness, computational feasibility, and structural faithfulness. Many attribution-based methods provide visually compelling input heatmaps, yet they rely on gradient approximations or perturbation schemes whose quantitative faithfulness is often debated and sensitive to methodological choices. Uncertainty and perturbation-based approaches introduce repeated forward passes or stochastic sampling, increasing computational overhead and limiting applicability in time-critical or resource-constrained environments. Concept-based and representation-oriented analyses reveal global structure, but frequently depend on external semantic annotations or dimensionality-reduction criteria not intrinsically tied to the learned operators.

In system contexts where decisions must be both interpretable and operationally reliable, a different perspective becomes relevant. Rather than asking only which input features influenced an output, one may ask: can we trace how a decision emerges from the internal geometry learned during training? Can we determine whether a prediction is supported by the same structural patterns that the model has historically associated with normal behaviour? And can such assessment be obtained without repeated perturbations or expensive auxiliary computation?

From this standpoint, the design objective is not to generate visually persuasive

explanations, but to construct a parameter-induced structural descriptor that:

1. derives its coordinate system directly from the learned operators;
2. monitors per-sample internal representation dynamics;
3. yields numerically verifiable measures of internal coherence;
4. operates with minimal additional computational overhead.

This orientation is particularly motivated by scenarios in which interpretability must coexist with strict computational budgets and timely response requirements, such as embedded or mission-critical AI systems.

The framework introduced in this work addresses this perspective by constructing low-rank, operator-induced projections, termed *peepholes*, that expose internal decision geometry while remaining anchored to the network’s learned transformations. Rather than approximating the full non-linear function, peepholes act as structural probes of internal geometry, enabling both interpretative inspection and operational, quantitative assessment within a unified coordinate system.

To the best of our knowledge, no existing post-hoc interpretability method simultaneously satisfies these structural, computational, and parameter-induced constraints within a unified framework.

3

Single-Layer Peephole Extraction and Cluster-Based Accordance

Introduction

This chapter presents the first formulation of the *peephole extraction mechanism*, originally introduced to probe the internal decision geometry of a deep network at a *single* layer, typically near the classification head [Manovi et al. \(2024a\)](#). The central objective is to map layer activations into a compact, operator-induced low-dimensional subspace in which class-dependent structure becomes geometrically explicit. Such a representation allows the network’s predictions to be directly compared with the internal activation patterns that historically support them, thereby establishing a measurable notion of structural accordance.

In its original formulation, the projected vector obtained from the layer’s affine operator was referred to as a *peephole*. In subsequent developments, terminology was refined: the raw projection is now termed a *corevector*, while *peephole* denotes the fully processed semantic descriptor. For clarity and consistency, this chapter adopts the updated terminology throughout.

Building upon the foundations established in the previous part of the discussion, we describe below a pipeline consisting of three main steps:

- (i) dimensionality reduction via SVD of the affine operator associated to a DNN layer,
- (ii) clustering of low-dimensional representations into activation modes,
- (iii) mapping into a normalized distance-based descriptor.

These ingredients allow the construction of two complementary metrics: *Clustering–Network Accordance* (CNA), which measures agreement between the output of a deep neural model and the proposed auxiliary mechanism, and *CNA with High-Confidence* (CNA-HC) estimation, which assesses the numerical support that internal representations provide for a network’s predicted class.

3.1 Dimensionality Reduction via the Layer’s Affine Operator

We focus on the case of a trained feedforward network as the ones modeled in Sec. 2.4.1. Consider layer ℓ receiving input activation $\mathbf{h}_\ell \in \mathbb{R}^{n_\ell}$ and producing

$$\mathbf{h}_{\ell+1} = f_\ell(\mathbf{W}_\ell \mathbf{h}_\ell + \mathbf{b}_\ell), \quad \mathbf{W}_\ell \in \mathbb{R}^{m_\ell \times n_\ell}, \mathbf{b}_\ell \in \mathbb{R}^{m_\ell}. \quad (3.1)$$

We rewrite the affine map in homogeneous form:

$$\mathbf{A}_\ell = [\mathbf{W}_\ell \mid \mathbf{b}_\ell] \in \mathbb{R}^{m_\ell \times (n_\ell + 1)}, \quad \bar{\mathbf{h}}_\ell = \begin{bmatrix} \mathbf{h}_\ell \\ 1 \end{bmatrix}. \quad (3.2)$$

Recalling what described in Sec. 2.4.4, the SVD of $\mathbf{A}_\ell$,

$$\mathbf{A}_\ell = \mathbf{P}_\ell \mathbf{\Sigma}_\ell \mathbf{Q}_\ell^\top, \quad (3.3)$$

reveals the dominant directions of variation in the linearized mapping.

Retaining the first κ_ℓ right singular vectors yields the *corevector*

$$\mathbf{v}_\ell = \mathbf{Q}_\ell^{(\kappa_\ell)\top} \bar{\mathbf{h}}_{\ell-1} \in \mathbb{R}^{\kappa_\ell}, \quad (3.4)$$

a compact representation capturing the main degrees of freedom of the layer.

Projecting activations onto a low-dimensional subspace provides a compressed version of how the layer transforms its inputs. Obtaining them through SVD ensures the best possible low-rank approximation (Sec. 1.4). However, the primary goal is to understand how these representations *organise*. To assess whether the projected activations exhibit structure, that is, whether they concentrate around characteristic modes, separate into distinct regions, or share common geometric patterns, we require an additional, principled step.

A central design principle of the framework is to keep the analysis as *objective* and *transparent* as possible. As argued in the foundational eXplainable AI (XAI) literature (e.g. [Doshi-Velez & Kim \(2017\)](#); [Lipton \(2018\)](#); [Barredo Arrieta et al. \(2020\)](#)), post-hoc interpretability should avoid introducing new opaque components. An explanation cannot credibly rely on mechanisms more complex or less interpretable than the model it seeks to analyse. For this reason, we deliberately refrain from introducing learned modules, auxiliary networks, or non-linear embeddings. Instead, we use strictly algebraic and classical unsupervised tools, ensuring that every transformation is both human-

inspectable and mathematically well-understood.

Clustering naturally satisfies these requirements. It is a model-agnostic, unsupervised method that identifies the dominant regularities emerging from the data itself. When applied to the corevectors, clustering reveals how the layer internally groups activations, exposing its operational “modes” without any supervision, parametric training, or architectural assumptions. This provides a structural description of the representation that mirrors the network’s own organisation rather than imposing external semantics.

Moreover, clustering aligns with the broader rationale of explainability: to discover *concise patterns* underlying high-dimensional behavior. As emphasised in [Molnar \(2020\)](#); [Montavon et al. \(2018\)](#), interpretability improves when explanations highlight stable regions or distinct behaviours rather than isolated local gradients. Cluster-based regions thus offer a natural way to convert raw activations into discrete, stable units of interpretation.

Finally, clustering prepares the ground for the empirical posterior construction and the subsequent analyses. Its output defines the layer-specific internal modes against which the network’s own decisions will be compared. The next subsection formalises this step.

3.2 Clustering of Corevectors

Let $\{\mathbf{v}_\ell^{(i)}\}$ be the set of corevectors extracted from training samples at layer ℓ , each standardized to zero mean and unit variance. The distribution is partitioned into K_ℓ clusters

$$\mathcal{C}_\ell = \{C_{1,\ell}, \dots, C_{K_\ell,\ell}\}, \quad (3.5)$$

where historically K_ℓ was set equal to the number of classes L in the dataset.

To uncover potentially distinct activation modes, we considered three complementary clustering families, each encoding a different assumption on cluster geometry and, consequently, a different notion of similarity between corevectors.

(a) K-Means. K-Means is one of the most widely adopted clustering algorithms and dates back to the seminal works of [MacQueen \(1967\)](#) and [Lloyd \(1982\)](#). In its classical formulation, the method implicitly assumes that clusters have *isotropic* geometry. It partitions data by minimising the average squared *Euclidean* distance between samples and their assigned centroid. The metric,

$$d_E(\mathbf{v}_a, \mathbf{v}_b) = \|\mathbf{v}_a - \mathbf{v}_b\|_2, \quad (3.6)$$

treats all directions of the feature space uniformly, leading to *spherical Voronoi regions* centered at the centroids $\boldsymbol{\mu}_{c,\ell}$. Two samples are therefore considered similar if they lie within neighbourhoods of comparable radius in all directions. This geometry makes K-Means an informative baseline for assessing whether internal representations exhibit “shape-agnostic” distributions dominated by uniform variance across dimensions.

(b) Agglomerative (Ward) Clustering. Hierarchical agglomerative clustering offers a contrasting paradigm. Instead of partitioning the data all at once, clusters are constructed *bottom-up*. Ward’s method, in particular, merges the pair of clusters whose union yields the smallest increase in the total within-cluster variance, a criterion established in the classical work of [Murtagh & Legendre \(2014\)](#). This produces *compact and well-separated* groups, reflecting a preference for clusters that remain statistically coherent as they grow. While new samples are still assigned to the nearest centroid during inference, the geometry learned during training differs markedly from that of K-Means, as it favours partitions tuned to local variance structure rather than assuming isotropic neighbourhoods.

(c) Gaussian Mixture Models (GMMs). A Gaussian Mixture Model provides a fully probabilistic perspective on clustering and has been extensively studied in the statistical signal processing literature, notably in the tutorial by [Reynolds \(2009\)](#). Each cluster is modelled as a Gaussian component characterised by a mean $\boldsymbol{\mu}_{k,\ell}$ and a covariance matrix $\boldsymbol{\Sigma}_{k,\ell}$, which captures both the variability and the correlations among features. Similarity is measured using the *Mahalanobis distance*,

$$d_M(\mathbf{v}_a, \boldsymbol{\mu}_{k,\ell}) = \sqrt{(\mathbf{v}_a - \boldsymbol{\mu}_{k,\ell})^\top \boldsymbol{\Sigma}_{k,\ell}^{-1} (\mathbf{v}_a - \boldsymbol{\mu}_{k,\ell})}, \quad (3.7)$$

which naturally adapts to *anisotropic* cluster shapes. Dimensions along which the data exhibit high variance contribute less to the distance, while tightly constrained directions contribute more. As a result, GMMs can capture elongated, skewed, or correlated activation modes that would be artificially split, or entirely missed, by methods assuming spherical geometry.

Taken together, these three clustering approaches offer different perspectives on the organisation of core vector activations. K-means clustering probes whether representations distribute themselves around spherical basins, Ward’s method tests whether compactness and variance minimisation better characterise their structure, and GMMs uncover anisotropic, covariance-driven manifolds. This diversity of geometric assumptions is essential for determining whether the internal representations of each layer are

predominantly isotropic, compact or directionally structured. This is an early empirical indicator of the types of features encoded and the degree of redundancy or anisotropy present in the learned representations.

3.2.1 Rank Selection

A central step in the construction of peepholes is the choice of the retained rank κ_ℓ used to compress the activation matrix of each layer. While the Eckart–Young theorem mentioned in Sec. 1.4 guarantees that the truncated SVD provides the best low-rank approximation in a norm sense, it does not prescribe how many singular directions should be preserved. Selecting κ_ℓ therefore requires a principled criterion that reflects the semantic structure carried by the activations while avoiding unnecessary dimensionality.

In the exploratory stage of this work, where the objective was to characterise whether the internal representations near the classification head encode class-relevant structure, we adopted a supervised diagnostic based on the *Adjusted Rand Index* (ARI). The rationale is simple: if a low-dimensional projection preserves the geometric organisation induced by the trained model, then the clustering of its corevectors should remain consistent with the dataset’s class structure. ARI provides a quantitative way to measure this consistency.

The ARI, introduced in [Hubert & Arabie \(1985\)](#), compares two partitions of the same dataset while correcting for agreements that would occur by chance. Let $\mathcal{U} = \{U_1, \dots, U_R\}$ denote the ground-truth classes and $\mathcal{V} = \{V_1, \dots, V_S\}$ the clusters obtained from corevectors at a given rank. For each pair (i, j) , let $n_{ij} = |U_i \cap V_j|$ be the number of samples shared by the two subsets, with marginals

$$a_i = \sum_j n_{ij}, \quad b_j = \sum_i n_{ij}, \quad N = \sum_{i,j} n_{ij}. \quad (3.8)$$

The Adjusted Rand Index is then defined as

$$\text{ARI} = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \frac{(\sum_i \binom{a_i}{2}) (\sum_j \binom{b_j}{2})}{\binom{N}{2}}}{\frac{1}{2} \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \frac{(\sum_i \binom{a_i}{2}) (\sum_j \binom{b_j}{2})}{\binom{N}{2}}}, \quad (3.9)$$

with the standard properties:

- ARI = 1 indicates perfect agreement between the two partitions;

- $\text{ARI} \approx 0$ corresponds to chance-level similarity;
- $\text{ARI} < 0$ indicates worse-than-chance agreement.

The ARI was therefore used *offline* to select κ_ℓ by identifying the smallest rank that retained the class-consistent structure present in the original activations. Once ranks were fixed, all subsequent components of the framework, including clustering and inference, operate independently of any ground-truth supervision. In this sense, ARI served purely as an analytical tool to calibrate the amount of information that each peephole should preserve, without influencing the framework at run time.

3.3 Softmin Mapping

Once the dimensionality κ_ℓ is fixed and each activation has been projected onto its corevector representation, the next step is to associate the new sample with the activation modes uncovered during clustering. For a given layer, this requires comparing the incoming corevector with the cluster prototypes and translating these comparisons into a compact, probabilistic descriptor.

For a new input with corevector $\mathbf{v}_\ell$, distances to the prototypes are collected as

$$d_{c,\ell} = d(\mathbf{v}_\ell, \boldsymbol{\mu}_{c,\ell}), \quad c = 1, \dots, K_\ell. \quad (3.10)$$

where $d(\cdot, \cdot)$ denotes the metric induced by the selected clustering model. Regardless of the geometry, we collect these values into a distance vector $\mathbf{d}_\ell$.

To obtain a descriptor that is both normalised and sensitive to relative distances, we apply a *softmin* transformation:

$$s_{c,\ell} = \frac{\exp(-d_{c,\ell})}{\sum_{j=1}^{K_\ell} \exp(-d_{j,\ell})}, \quad \mathbf{s}_\ell \in [0, 1]^{K_\ell}. \quad (3.11)$$

The intuition behind this choice parallels the role of the *softmax* function used at the network’s output. While softmax converts unnormalised logits into a probability distribution over classes, the softmin converts distances into a probability-like distribution over activation modes: small distances receive higher weight, large distances receive lower weight, and the result is always a vector in $[0, 1]^{K_\ell}$ whose entries sum to one. In this sense, $\mathbf{s}_\ell$ can be interpreted as a probabilistic expression of how strongly the incoming corevector aligns with each of the learned feature patterns.

In the early formulation of the framework, this vector $\mathbf{s}_\ell$ constituted the semantic descriptor - or *peephole* - of layer ℓ : a compact summary of the sample’s position within

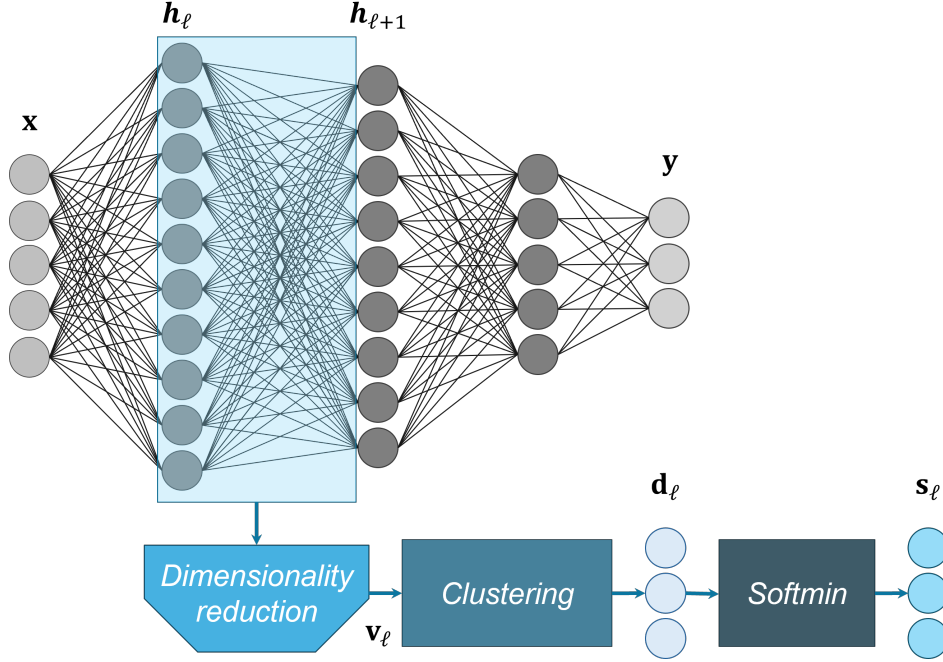


Figure 3.1. Single-layer peephole extraction pipeline. The activation of layer ℓ is projected onto a low-rank basis, clustered into K_ℓ activation modes, and mapped via softmin into a normalized descriptor $\mathbf{s}_\ell$.

the layer’s internal geometry.

Fig. 3.1 summarises the full sequence of operations, from the original high-dimensional activation of a DNN to the final descriptor, highlighting how dimensionality reduction, clustering and softmin mapping are integrated into a coherent pipeline.

3.4 Clustering-Network Accordance (CNA)

Having established a way to map activations onto a set of interpretable activation modes through the softmin scores $\mathbf{s}_\ell$, we can now ask a natural question: *to what extent does the network’s decision align with the internal geometry revealed by clustering?* If clustering captures coherent activation patterns, and the network uses those patterns consistently, the two should agree. CNA formalises this intuition.

Let the network assign to input $\mathbf{x}$ the label

$$\hat{y} = \arg \max_c y_c(\mathbf{x}), \quad (3.12)$$

where $y_c(\mathbf{x})$ is the softmax posterior defined in Eq. 2.7.

Independently, the clustering surrogate produces its own prediction by identifying the activation mode closest to the corevector:

$$\hat{c} = \arg \min_k d_{k,\ell} = \arg \max_k s_{k,\ell}. \quad (3.13)$$

CNA checks whether the model’s decision is consistent with the structure encoded in its intermediate activations:

$$\text{CNA}(\mathbf{x}) = \begin{cases} 1, & \hat{y} = \hat{c}, \\ 0, & \text{otherwise.} \end{cases} \quad (3.14)$$

A dataset-level CNA score is therefore the fraction of samples for which the two decision mechanisms coincide. It provides a first sanity check: if the network frequently disagrees with the geometry of its own activations, its decision-making process is unstable or internally inconsistent. While intentionally simple, this binary measure anticipates the richer notion of trustworthiness introduced next.

3.5 CNA with High-Confidence (CNA-HC)

CNA identifies agreement, but it does not quantify *how strongly* the internal representation supports the network’s decision. To address this, we introduce a refined measure that exploits the softmin mapping itself. Recall that $s_{k,\ell}$ expresses how closely the corevector aligns with the k -th activation mode. If the network predicts class $\hat{y}$, a natural indicator of *internal support* is the score associated with the cluster corresponding to that class.

For input $\mathbf{x}$, let the network’s predicted class be again $\hat{y}$. The *high-confidence score* is defined as

$$\gamma_\ell(\mathbf{x}) = s_{\hat{y},\ell}. \quad (3.15)$$

This value, lying in $[0, 1]$, quantifies how close the activation at layer ℓ lies to the region typically associated with the predicted class. Large values correspond to corevectors positioned near the “prototype” activation mode for that class, whereas small values signal that the class decision is not supported by the geometric structure of the layer. To obtain a decision-level indicator of trust, we introduce a user-defined threshold $\tau \in (0, 1)$ and declare:

$$\gamma_\ell(\mathbf{x}) \geq \tau. \quad (3.16)$$

The value of τ represents a *trust criterion*. It identifies those samples whose internal

representations are geometrically coherent with the network’s predicted class.

CNA-HC therefore complements CNA. CNA asks “*Does the network follow its internal geometry?*” On the other hand, CNA-HC asks “*And if so, how strongly is this geometry supporting the decision?*”

Together, they aim to provide an interpretable, layer-wise perspective on the stability, coherence, and reliability of a model’s decision process.

3.6 Offline and Online Computations

The single-layer peephole mechanism relies on a small number of components, most of which are computed once *offline* after the network has been trained. This decomposition highlights a key advantage of the approach: the computationally intensive operations are performed only once, while the inference-time cost remains negligible.

Offline stage. For each layer ℓ , the following quantities are precomputed:

- the truncated SVD basis $\mathbf{Q}_\ell^{(\kappa_\ell)}$ that defines the corevector projection;
- the cluster prototypes extracted from the collection of corevectors (centroids for K-Means or Ward, and both centroids and covariances for GMMs);
- the rank κ_ℓ selected via the ARI-based criterion discussed in Section 3.2.1;
- the mapping matrix Π_ℓ , which associates each cluster with the distribution of dataset labels observed within that cluster.

Fig. 3.2 highlights how these quantities are used *at inference time* to relate a new activation to the activation modes identified offline.

The matrix $\Pi_\ell \in [0, 1]^{K_\ell \times L}$ is defined as:

$$\Pi_\ell(c, l) = \frac{|\{\mathbf{x} \in \mathcal{D}_{\text{train}} : \text{corevector of } \mathbf{x} \text{ falls in cluster } c \text{ and } y(\mathbf{x}) = l\}|}{|\{\mathbf{x} \in \mathcal{D}_{\text{train}} : \text{corevector of } \mathbf{x} \text{ falls in cluster } c\}|}. \quad (3.17)$$

Therefore, Π_ℓ measures the empirical probability that a training sample in cluster c belongs to class l . Thus, if the corevector of a sample falls into cluster c , the row $\Pi_\ell(c, \cdot)$ provides a distribution over labels describing the semantic tendencies of that cluster.

Although this quantity is not yet used directly for inference, it establishes the first semantic bridge between unsupervised activation modes and the supervised label space. In Chap. 4 this object will be formalised as an *empirical posterior* and will form the basis for cross-layer or depth-wise semantic tracking.

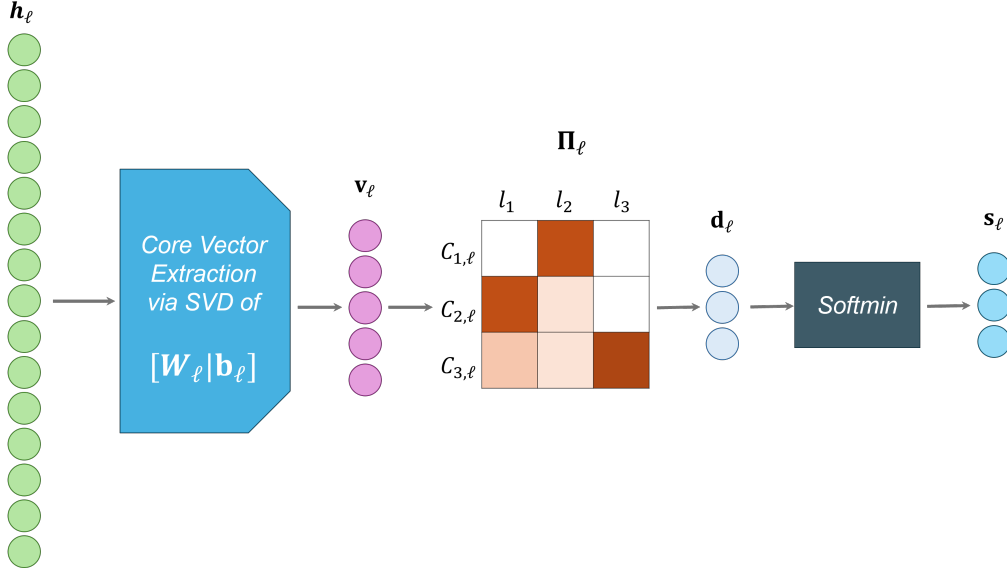


Figure 3.2. Schematic of the peephole mechanism at layer ℓ . The activation is projected into the corevector space via the truncated SVD of $[W_\ell | b_\ell]$. Distances to cluster prototypes are computed and normalised through a softmin operator, producing a probability-like descriptor that supports CNA and CNA-HC. The matrix Π_ℓ encodes the empirical association between clusters and labels.

Online stage (inference). Once these quantities are stored, the peephole computation at test time requires only:

- i. one projection onto the corevector basis,
- ii. one distance computation to all cluster prototypes,
- iii. one softmin normalisation to obtain the probability-like descriptor $\mathbf{s}_\ell$,
- iv. the extraction of $\gamma_\ell(\mathbf{x})$ for CNA-HC, when required.

No gradients, backpropagation, or retraining are involved. The computational footprint is dominated by vector–matrix products and distance evaluations, making the peephole mechanism lightweight and also potentially compatible with resource-constrained environments.

Summary

The mechanism presented in this chapter constitutes the first operational instantiation of the *peephole-based framework*. Starting from the high-dimensional internal activations of a deep network, we constructed a compact, cluster-based latent geometry via

dimensionality reduction, unsupervised clustering, and softmax normalisation. This geometry provides interpretable activation modes and a structured descriptor $\mathbf{s}_\ell$, as well as quantitative measures of agreement and trust (CNA and CNA-HC), which reflect the relationship between the network’s decision and the organisation of its intermediate representations.

These insights raise the question of how these geometric structures evolve across depth and whether we can track their semantic progression layer by layer. The next chapter extends the framework to a multi-layer setting and introduces concepts that reveal a richer view of the model’s internal decision-making process.

4

Distributed Peepholes and Layer-wise Consistency

Introduction

The single-layer analysis developed in the previous chapter demonstrated that structured activation geometry can be derived from deep networks: after projecting each layer’s activations into a compact subspace and clustering the resulting *corevectors*, coherent activation modes emerge. When the considered layer is one close to the final classification *head* of the model, these modes supply a natural surrogate for the network’s decisions, yielding the CNA and CNA-HC metrics, and allow us to probe whether the classifier behaves consistently with the internal structure it has learned. However, these metrics remain inherently local, reflecting the structural behaviour of a single depth rather than the distributed reasoning process of the full network.

Yet, a modern network is a cascade of transformations, not a single operator. A local analysis focused on one layer can reveal *how* a model behaves at that specific depth, but it cannot describe the global formation of evidence across the hierarchy. In particular, structural agreement observed at a late layer may conceal unstable or contradictory intermediate representations, a phenomenon that cannot be detected by single-layer inspection alone. If trust is to be associated with the coherence of the reasoning process, we must look beyond an isolated layer and examine how internal support evolves throughout the entire computational chain. Trust, in this perspective, cannot be reduced to the alignment of a single representation with a predicted label; it must reflect the stability and coherence of structural evidence throughout the hierarchy.

This chapter extends the peephole framework to the multi-layer setting. Rather than treating each layer as an isolated diagnostic point, the analysis now considers the ordered sequence of internal projections as a structured trajectory of evidence accumulation.

The resulting construction yields a distributed structural object that records the evolution of internal evidence across depth for a given input. This object, referred to as a layer-wise *evidence map*, provides a structured trace of how semantic support accumulates throughout the hierarchy. Rather than offering a single-depth snapshot, it captures the ordered trajectory of activation patterns that collectively shape the model’s

decision.

From this distributed representation, both interpretative visual summaries and quantitative measures of *layer-wise structural consistency* are derived. Together, these tools enable a global assessment of how coherently the network’s internal geometry supports its final prediction.

The discussion proceeds by formalising the peephole extraction block, explaining how it operates independently at each layer, and by defining the *evidence map*, a matrix collecting all peepholes for a given sample. We then introduce *prototype evidence maps*, which summarise the typical internal behaviour of each class. These constructions prepare the ground for evaluating structural *consistency across depth* through aggregation of layer-wise accordance. Together, they provide a distributed view of reasoning that transforms local geometric probes into a coherent, hierarchical account of internal decision formation.

4.1 The Peephole Extraction Block

To make the internal reasoning of a deep model intelligible, we introduce a *peephole extraction* (PE) block: a self-contained module that converts the activation of a chosen layer into a compact, semantically aligned descriptor. The objective is to expose, for each layer, *how* its internal representation relates to concepts of interest and *how strongly* it supports or contradicts a given prediction.

The PE block operates by distilling the activation into a geometric summary and assigning it to a representative behavioural pattern. This pattern is then translated into a concept-space or class-space signal. In practice, this involves three sequential transformations that expand the mechanism presented in the previous chapter:

1. *Corevector extraction*: the raw activation is projected onto a low-dimensional, information-preserving subspace, capturing the dominant directions induced by the layer’s parameters;
2. *Clustering*: corevectors are grouped to identify typical activation regimes for that layer, forming a set of prototypical behaviours;
3. *Feature mixing*: each new activation is mapped to the empirical class-distribution associated with its cluster, producing a vector that quantifies the alignment between that layer’s behaviour and the output classes.

The resulting descriptor acts as a *semantic peephole* into the layer: a compact and interpretable view of the information the network is using at that point in the processing

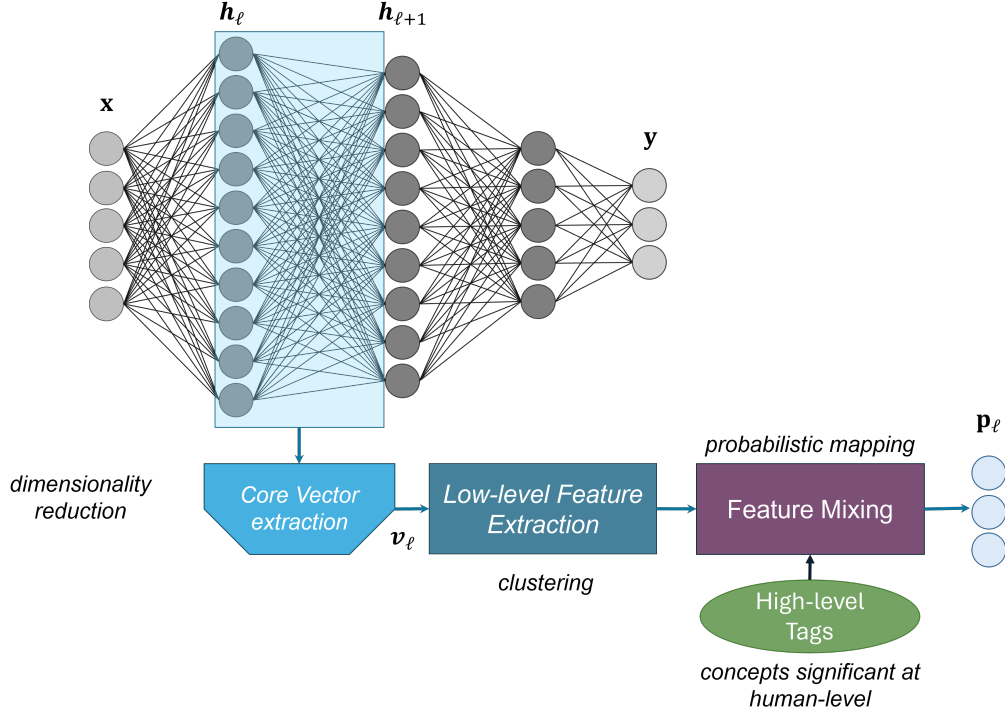


Figure 4.1. The peephole extraction block at layer ℓ . The activation is projected via SVD into a corevector, assigned to a cluster, and transformed via the empirical posterior Π_ℓ into a compact descriptor $\mathbf{p}_\ell$.

chain. Fig. 4.1 illustrates the flow. For clarity, the full PE block is visualised in Fig. 4.2.

4.1.1 Corevector Extraction

The first stage of the peephole extraction block reuses the linear-algebraic machinery introduced in Chap. 3. For a given layer ℓ , the activation $\mathbf{h}_\ell$ is processed through the augmented affine operator defined in Eq. (2.17) and projected onto the dominant right-singular subspace obtained in Sec. 3.1.

The resulting vector,

$$\mathbf{v}_\ell \in \mathbb{R}^{\kappa_\ell},$$

is the *corevector* of layer ℓ : a low-dimensional coordinate of the activation within the affine subspace that concentrates the layer’s principal degrees of freedom. As discussed previously, this subspace is identified once offline through the truncated SVD of the layer operator, and κ_ℓ is the retained rank.

The corevector thus serves as the linearised representation upon which the subsequent clustering and semantic mixing stages operate.

4.1.2 Low-Level Feature Extraction

The collection of corevectors $\{\mathbf{v}_\ell^{(i)}\}$ extracted from the training set is grouped into activation regimes through clustering:

$$\mathcal{C}_\ell = \{C_{1,\ell}, \dots, C_{K_\ell,\ell}\},$$

where K_ℓ is selected independently of the number of available labels or semantic categories. This design choice reflects the fact that the peephole extraction block aims to characterise the *intrinsic activation modes* of layer ℓ , rather than to replicate any externally imposed taxonomy.

Each clustering method (be it K-Means, Ward, or GMM discussed in Sec. 3.2) induces a membership vector $\mathbf{m}_\ell(\mathbf{v}_\ell) \in \mathbb{R}^{K_\ell}$, whose components satisfy

$$m_{c,\ell}(\mathbf{v}_\ell) = \begin{cases} 1, & \text{if } \mathbf{v}_\ell \in C_{c,\ell}, \\ 0, & \text{otherwise,} \end{cases}$$

for hard assignments, or equal to the Gaussian responsibility in the case of GMMs. The vector $\mathbf{m}_\ell(\mathbf{v}_\ell)$ thus identifies the activation mode expressed at that depth.

Unlike the single-layer construction of Chapter 3, no constraint forces K_ℓ to match the number of dataset classes. Rather, we are interested here in a more general concept of semantic tags. This decoupling allows each layer to be represented at the granularity that best captures its activation geometry, increasing expressiveness, stability, and generalisation across layers.

4.1.3 Feature Mixing and Semantic Descriptor

To relate activation modes to interpretable concepts, we consider a set of abstract *high-level tags* indexed by $t \in \{1, \dots, T\}$. These tags may coincide with dataset classes in classification tasks, but the construction is intentionally more general: tags may represent attributes, behavioural categories, or broader semantic notions relevant to the analysis.

From the training set, the empirical association between clusters and tags is estimated in the matrix

$$\mathbf{\Pi}_\ell(c, t) = \Pr(\text{tag} = t \mid C_{c,\ell}), \quad 1 \leq c \leq K_\ell, \quad 1 \leq t \leq T.$$

Each row of $\mathbf{\Pi}_\ell$ describes how the c -th activation mode tends to co-occur with each

high-level tag across training samples.

Given a new activation at layer ℓ , the membership vector $\mathbf{m}_\ell(\mathbf{v}_\ell)$ is mapped through $\mathbf{\Pi}_\ell$ to produce the *peephole descriptor*

$$\mathbf{p}_\ell = \mathbf{\Pi}_\ell \mathbf{m}_\ell(\mathbf{v}_\ell) \in \mathbb{R}^T, \quad (4.1)$$

which provides a compact, tag-aligned summary of the behaviour expressed at layer ℓ .

The vector $\mathbf{p}_\ell$ can be viewed as a semantic fingerprint of the layer’s instantaneous activity: it reflects which high-level tags are supported by the current activation mode, how strongly, and in what proportions. Because $\mathbf{p}_\ell$ is obtained by combining geometric structure (corevectors), statistical regularities (clusters), and semantic associations (empirical posteriors), it serves as an interpretable proxy of the information passing through layer ℓ .

We conclude this section with a remark on single *vs* multi-layer analysis. In the single-layer setting of Chapter 3, the number of clusters was chosen to coincide with the number of tags, enforcing a one-to-one alignment between activation modes and semantic categories. The multi-layer construction generalises this principle: by allowing K_ℓ and T to differ at each layer, the framework adapts to the natural activation structure of the network. This flexibility increases robustness, avoids degeneracies due to rank limitations, and enables comparisons across layers where the geometry of the activation space evolves with depth.

4.2 Distributed Peepholes Across Layers

Attaching a PE block to multiple layers yields a distributed, depth-resolved representation of the network’s internal processing. For a model with L analysed layers, we obtain the collection

$$\{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{L-1}\},$$

where each peephole $\mathbf{p}_\ell$ summarises the high-level semantic evidence available at depth ℓ .

Fig. 4.3 shows how PE blocks are inserted after selected layers in a feedforward architecture. Each peephole captures a distinct stage of the internal reasoning pipeline: early layers reflect low-level patterns and generic concepts, intermediate layers encode composition and structural relations, and deeper layers accumulate task-specific evidence. Tracking the peepholes as they move across the layers reveals whether the semantic evidence is amplified, stabilised, redistributed or contradicted as the input propagates.

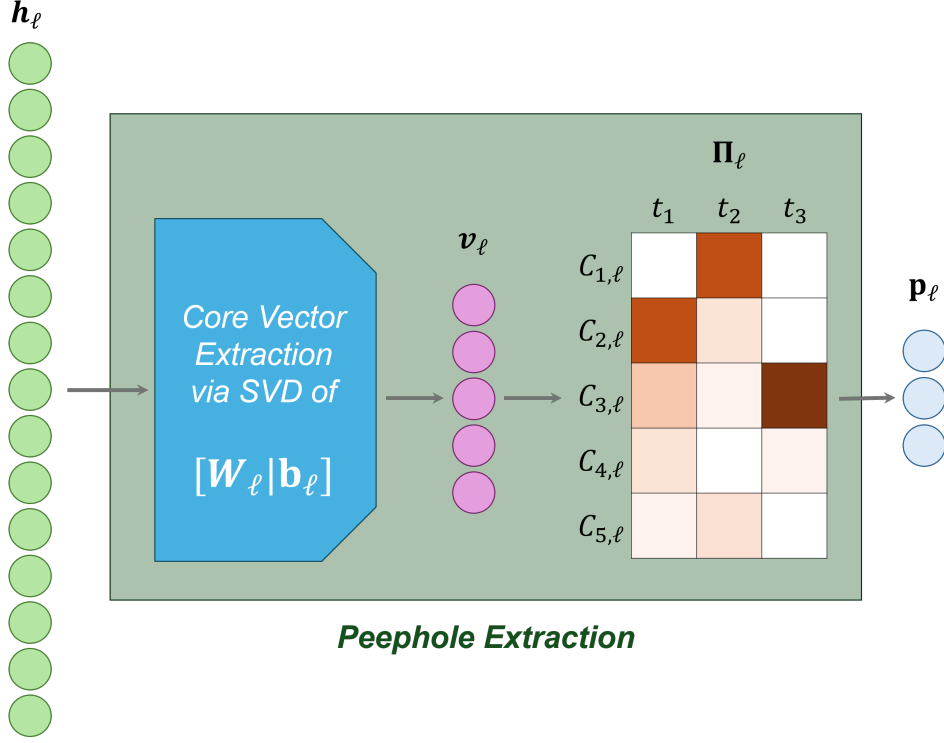


Figure 4.2. Detailed view of the peephole extraction (PE) block at layer ℓ . The activation is projected onto the layer’s dominant right-singular subspace to obtain the corevector $\mathbf{v}_\ell$, which is then assigned to one of the activation modes $\{C_{c,\ell}\}$. The empirical posterior matrix Π_ℓ maps this cluster identity to a distribution over high-level semantic tags, producing the peephole descriptor $\mathbf{p}_\ell$.

4.3 Layer-wise Evidence Maps, Conceptograms and Classification-Maps

Once the peephole descriptors $\{\mathbf{p}_\ell\}$ of all layers are computed for a given input $\mathbf{x}$, they can be arranged into a matrix of the form

$$\mathbf{P}(\mathbf{x}) = [\mathbf{p}_0(\mathbf{x}), \mathbf{p}_1(\mathbf{x}), \dots, \mathbf{p}_{L-1}(\mathbf{x})] \in \mathbb{R}^{T \times L}, \quad (4.2)$$

which we refer to as a *layer-wise evidence map*. Each column ℓ describes the tag distribution inferred at layer ℓ , while the T rows correspond to the tracked high-level tags.

The layer-wise evidence map admits different interpretations depending on the nature of the tracked tags. When the rows correspond to broader or human-level semantic concepts (e.g., “fur”, “wheel”, “face”, “edge pattern” in standard image classification

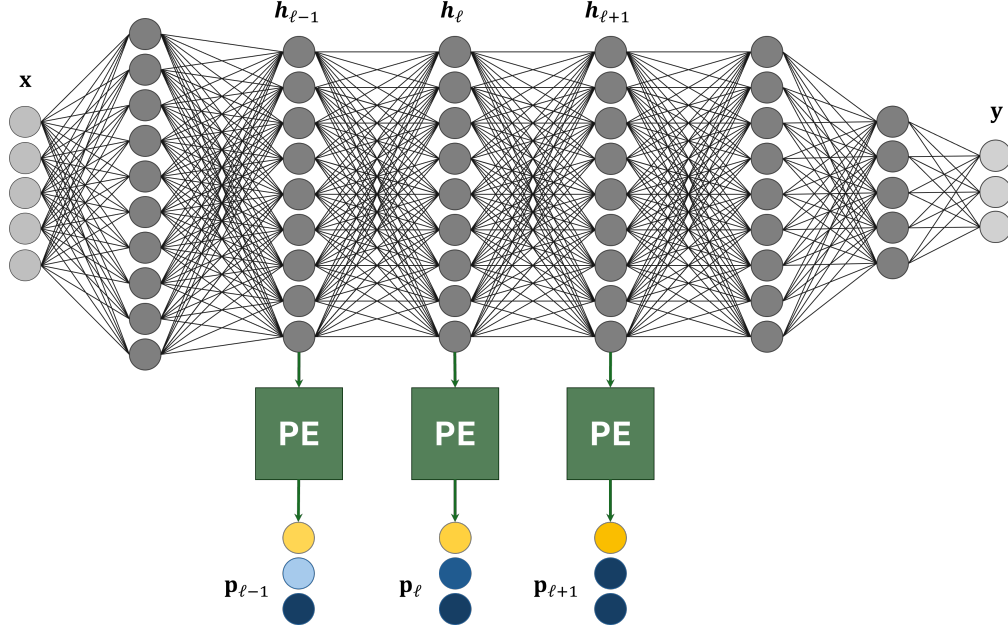


Figure 4.3. Distributed peepholes across selected layers. Each PE block outputs a descriptor $\mathbf{p}_{\ell}$ that provides a layer-wise account of the semantic information expressed at depth ℓ .

benchmarks), we refer to the matrix as a *conceptogram*. When the rows correspond to dataset classes, the matrix reduces to a *classification-map*. The two objects are structurally identical; only the interpretation of rows differs.

- **Conceptogram:** an interpretation of the evidence map where rows correspond to semantic concepts. Each row tracks how a concept is expressed layer by layer. This representation is particularly useful when semantic information is shared across classes or when investigating emergent internal taxonomies.
- **Classification-map:** a special case of the evidence map where rows correspond to dataset classes. Each row measures how strongly a class signal “echoes” within the network while processing a single input. This form is especially useful when analysing class-level reasoning, decision consistency, or superclass structures.

Thus, conceptograms provide the most general view, while classification-maps can be regarded as their task-specific instantiation.

4.4 Prototype Evidence Maps

To capture the “typical” internal trajectory associated with a given tag (or class), we construct *prototype evidence maps*. Let $\mathcal{D}_t^{\text{proto}}$ be the subset of training samples for which tag t is correctly expressed by the model and for which the prediction is deemed reliable under a standard confidence measure. The prototype map is defined as

$$\bar{\mathbf{P}}_t = \frac{1}{|\mathcal{D}_t^{\text{proto}}|} \sum_{\mathbf{x} \in \mathcal{D}_t^{\text{proto}}} \mathbf{P}(\mathbf{x}), \quad (4.3)$$

followed by a columnwise normalisation ensuring that each layer yields a valid distribution over the tracked tags.

Prototype maps highlight statistical regularities in the layer-wise flow of semantic evidence: they reveal which concepts consistently emerge at each depth, how early the network forms discriminative structures, and where semantic ambiguities tend to collapse or resolve.

4.5 Layer-wise Structural Consistency as a Confidence Signal

The map $\mathbf{P}(\mathbf{x})$ provides a structured, layer-wise account of the semantic evidence expressed by the network while processing $\mathbf{x}$. When the tracked tags correspond to dataset classes, $\mathbf{P}(\mathbf{x})$ takes the form of a classification-map; when they represent more general semantic features, it becomes a conceptogram. In either case, the matrix encodes the internal trajectory followed by the input as it moves through the model’s depth.

For each tag t , we have constructed in Section 4.4 a prototype map $\bar{\mathbf{P}}_t$ that summarises the typical, high-confidence behaviour associated with that tag. Comparing $\mathbf{P}(\mathbf{x})$ with the prototype corresponding to the tag predicted for $\mathbf{x}$ allows us to assess whether the internal reasoning of the network is consistent with the expected pattern.

This raises a natural question: *can such agreement be quantified?* If the depthwise trajectory of $\mathbf{x}$ aligns with the characteristic prototype of its predicted tag, then the network is behaving in a stable and predictable manner. Conversely, if several layers diverge from the prototype, the reasoning pathway may be atypical, unstable, or poorly supported by learned semantics.

These considerations motivate a *global similarity measure* between $\mathbf{P}(\mathbf{x})$ and the corresponding prototype $\bar{\mathbf{P}}_{t(\mathbf{x})}$, yielding a numerical confidence score that captures *layer-wise consistency*. This score serves as an interpretable indicator of trustworthiness,

grounded in the network’s own internal behaviour rather than its output alone.

4.5.1 Global Similarity Between Sample and Prototype

For a new input $\mathbf{x}$, let $t(\mathbf{x})$ denote the tag assigned by the network, for instance, the predicted class in classification settings. To assess whether the internal trajectory of $\mathbf{x}$ is consistent with typical behaviour for tag $t(\mathbf{x})$, we compare its full semantic map $\mathbf{P}(\mathbf{x})$ with the corresponding prototype $\bar{\mathbf{P}}_{t(\mathbf{x})}$ introduced in Section 4.4.

A global measure of alignment between the two matrices is obtained through the (normalised) Frobenius inner product:

$$s(\mathbf{x}) = \frac{\langle \mathbf{P}(\mathbf{x}), \bar{\mathbf{P}}_{t(\mathbf{x})} \rangle_F}{\|\mathbf{P}(\mathbf{x})\|_F \|\bar{\mathbf{P}}_{t(\mathbf{x})}\|_F}. \quad (4.4)$$

This quantity is a cosine similarity computed over the entire depth–tag matrix, rather than on a single layer or descriptor. Since both matrices contain non-negative entries, the score satisfies $s(\mathbf{x}) \in [0, 1]$.

Cosine similarity is particularly well suited for this setting. First, it is intrinsically scale invariant: the magnitude of $\mathbf{P}(\mathbf{x})$ and $\bar{\mathbf{P}}_{t(\mathbf{x})}$ depends on implementation details (e.g., layer depth, spread of the empirical posterior, or the number of active tags), whereas the quantity of interest is the *structural pattern* of activations across layers and tags. By normalising both matrices, $s(\mathbf{x})$ focuses on their directional agreement, i.e., on whether the semantic trajectory of $\mathbf{x}$ evolves through depth in a way that is coherent with the prototype, independently of absolute activation magnitude.

From a signal-processing perspective, the normalised Frobenius inner product in Eq. (4.4) can be interpreted as the multidimensional analogue of the classical matched-filter correlation statistic used in detection theory [Kay \(1998\)](#); [Trees \(2001\)](#). In its canonical formulation, a matched filter evaluates the cross-correlation between an observed signal and a known template, yielding the decision statistic that maximizes the signal-to-noise ratio under additive noise. When normalised, the resulting coefficient measures alignment independently of amplitude scaling.

In this view, $\mathbf{P}(\mathbf{x})$ plays the role of the observed signal and $\bar{\mathbf{P}}_{t(\mathbf{x})}$ acts as the template; their cosine similarity therefore provides a geometrically grounded statistic that quantifies structural alignment between the observed semantic trajectory and the canonical layer-wise pattern learned from reliable examples.

Cosine-based similarity measures are commonly adopted in representation analysis and prototype-based explainability to compare feature embeddings or layer-wise activations (e.g., [Kornblith et al. \(2019\)](#); [Papernot & McDaniel \(2018\)](#)). In those

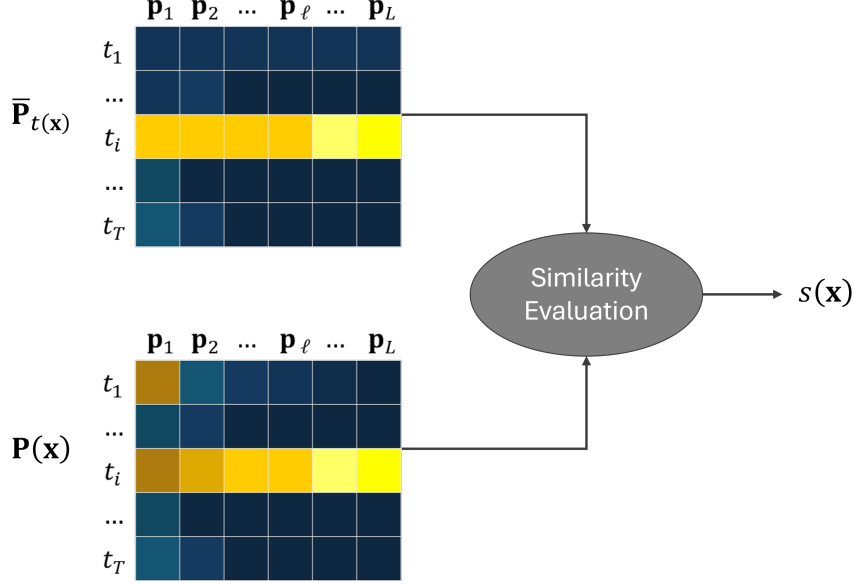


Figure 4.4. Schematic illustration of the similarity evaluation defined in Eq. (4.4). The classification-map $\mathbf{P}(\mathbf{x})$ extracted for a given input is compared to the prototype map $\bar{\mathbf{P}}_{t(\mathbf{x})}$ associated with its predicted tag t_i , which we suppose here shows strong linear evidence across depth. Each matrix encodes the evolution of semantic evidence across layers (columns) and tags (rows). The normalized Frobenius inner product measures their global alignment, yielding a scalar confidence score that reflects the structural coherence between the observed reasoning trajectory and the canonical pattern learned from reliable samples.

settings, similarity is typically evaluated at the level of a single representation space or individual layer. In contrast, the present formulation operates on structured depth-tag matrices, capturing *global* trajectory alignment across the entire network rather than local embedding agreement.

Fig. 4.4 provides a schematic representation of this comparison mechanism. The classification-map extracted for a given input is contrasted with the prototype map associated with its predicted tag. The similarity is evaluated globally across the entire depth-tag matrix, rather than locally at a single layer, thereby capturing the coherence of the full semantic trajectory.

4.5.2 Interpretation of the Consistency Score

The similarity defined in Eq. (4.4) provides a global measure of layer-wise consistency. It condenses the full semantic trajectory of $\mathbf{x}$ into a scalar quantity that reflects how closely the network’s internal processing aligns with the prototype associated with the

predicted tag.

Unlike output-based confidence indicators, which depend solely on the final representation, the proposed score evaluates the coherence of the entire depth-wise reasoning chain. Because it is computed over the full depth-tag matrix, the resulting value captures structural agreement across layers rather than agreement at a single representational stage.

High values of $s(\mathbf{x})$ indicate that the distribution of semantic evidence throughout the network depth follows a characteristic pattern previously observed in reliable, high-confidence training samples of the predicted class. In this regime, the prediction is not only sharp at the output level, but also internally supported by a coherent evolution of intermediate representations.

Conversely, low similarity values reveal a mismatch between the observed layer-wise evolution and the canonical class-specific pattern. Importantly, such divergence may occur even when the softmax output remains confident. A sample may produce a peaked output distribution while following an atypical or unstable internal pathway, resulting in a reduced $s(\mathbf{x})$ value. This situation may arise in visually plausible but uncommon in-distribution examples, inputs located near the boundary of the training distribution, or adversarially manipulated samples.

Symmetrically, moderate output confidence does not necessarily imply structural unreliability. When the internal trajectory remains aligned with the prototype despite output ambiguity, the score may remain comparatively high, indicating benign uncertainty rather than failure of the learned representation.

Overall, the consistency score should be interpreted as a model-intrinsic reliability indicator: it evaluates whether the predicted tag is supported by a characteristic chain of internal evidence. By complementing output-based measures with a structural assessment of the decision pathway, it enriches confidence estimation with an interpretable geometric criterion.

Summary

This chapter extended the peephole framework from a single-layer diagnostic tool to a distributed mechanism capable of tracking how semantic information evolves throughout the depth of a network. By assembling layer-wise descriptors into conceptograms or classification-maps and comparing them against prototypical trajectories, we introduced a structurally grounded confidence score rooted in the internal dynamics of the model rather than in its output alone.

The resulting measure provides a principled perspective on how consistently a model reasons when processing an input, capturing depth-wise structural alignment across intermediate representations.

In the following chapter, we present qualitative and quantitative results, analysing the empirical behaviour of the proposed consistency score under diverse operational conditions.

III

Empirical Evaluation and Interpretations

Introduction

Emergent Structures and Trustworthy Behaviour

The previous part introduced the *Peephole* framework and its extension to the multi-layer setting, showing how a network’s internal geometry can be analysed to recover structured evidence about its own decisions. We now turn from construction to observation: from defining how the framework operates to examining what it reveals in controlled experimental settings.

The experiments presented in this part are not intended as an exhaustive survey of modern AI systems. Rather, they are structured as controlled contrasts designed to stress-test structural coherence under different architectural biases and operational regimes. The selected architectures serve as structurally distinct testbeds that allow us to examine how internal consistency manifests across operator families and task contexts.

We revisit the guiding questions of the thesis: How does a model organise its internal representational space when learning to discriminate? How does this organisation change when it errs, encounters unfamiliar data, or processes perturbed inputs? And to what extent can structural agreement measures such as CNA and CNA-HC, together with layer-wise structural consistency, anticipate or explain these behaviours?

The first group of results focuses on controlled visual recognition tasks, using convolutional and transformer architectures as complementary structural testbeds. The analysis proceeds progressively. We begin with a single-layer study on a convolutional network, where CNA and CNA-HC are introduced as measures of local structural agreement between internal evidence and predicted class. This setting isolates how coherence at a specific depth relates to prediction reliability.

We then extend the analysis to the multi-layer setting, constructing layer-wise evidence maps that trace how semantic support evolves across depth. In classification scenarios, these evidence maps take the form of *classification-maps*, providing a structured representation of how class-related signals form, stabilise, or fragment as

information propagates through the hierarchy. This distributed perspective is evaluated across both convolutional and transformer architectures.

Finally, we examine whether aggregated layer-wise structural consistency translates into measurable improvements in confidence estimation, robustness, and out-of-distribution awareness, thereby connecting internal geometry with externally observable reliability.

The subsequent sections extend this reasoning to more demanding and safety-critical contexts. In the domain of anomaly detection for spacecraft and on-board autonomy, the same structural methodology serves a different purpose: ensuring that decisions made under strict resource constraints remain interpretable and assessable. Here, internal structural agreement is not merely descriptive but operational, linking interpretability with dependability. In high-risk environments, autonomy requires mechanisms capable of verifying whether system behaviour remains consistent with its nominal internal geometry.

Together, these studies form the empirical core of the thesis. They demonstrate that the geometry uncovered by the peephole-based framework is not an abstract construct, but a measurable structural substrate of model behaviour. The observed patterns across tasks and architectures substantiate the central idea that learning normality and detecting deviation are closely intertwined. Interpretability and reliability, often treated as separate objectives, emerge instead as correlated expressions of internal structural coherence.

The chapters that follow present these findings progressively, from classification experiments to exploratory studies on autonomy. They illustrate how the framework scales across operator families, adapts to different domains, and exposes the connection between analysing internal representations and assessing the trustworthiness of model decisions.

5

Motivation and Background

The following discussion is organised around two complementary perspectives:

- **Qualitative insights**, obtained from single-layer structural measures (CNA/CNA-HC) and from the layer-wise evidence maps introduced in Chap. 4. These analyses illustrate how internal evidence forms, stabilises, or breaks down across depth, offering an interpretive view of the model’s reasoning trajectory.
- **Quantitative evaluations**, which assess whether internal structural agreement and layer-wise consistency translate into measurable improvements in confidence estimation, out-of-distribution (OOD) awareness, adversarial robustness, and computational feasibility. These results position the framework with respect to established baselines.

To maintain clarity and avoid redundancy, the *background concepts*, *evaluation dimensions*, and the *experimental setup* are consolidated in this chapter. This shared foundation supports the analyses carried out in the subsequent sections, which focus respectively on:

1. **CNA and CNA-HC** (single-layer structural coherence in convolutional architectures),
2. **Layer-wise evidence maps and classification-maps** (distributed structural analysis across depth),
3. **Global structural consistency as a confidence estimator.**

5.1 Background Concepts

Several complementary notions describe different facets of a model’s *trustworthiness*. This section recalls the definitions and evaluation metrics relevant to the quantitative assessment that follows.

5.1.1 Confidence and Uncertainty

The notion of *confidence* formalized in Sec. 2.4.2 quantifies how decisively a model assigns a sample to a class. For a classifier producing posteriors $p(y|x)$ via softmax, the natural confidence is the maximum posterior probability, $\text{MSP}(x) = \max_c p(y = c | x)$, known as the *Maximum Softmax Probability* (MSP) Hendrycks & Gimpel (2017). While MSP offers a convenient scalar score, it is purely output-based and therefore insensitive to internal inconsistencies in the reasoning process.

As discussed in Sec. 2.4.2, confidence can be interpreted as the complement of uncertainty, which comprises *aleatoric* (data-related) and *epistemic* (model-related) components Kendall & Gal (2017); Hüllermeier & Waegeman (2021). Post-hoc estimators such as DOC Granese et al. (2021) and REL-U Dadalto et al. (2024) approximate epistemic uncertainty directly from the network outputs. DOC derives a confidence measure from the *dispersion* of the posterior distribution, refining softmax probabilities through temperature scaling and a small gradient-based adjustment of the input to accentuate differences between correct and incorrect predictions. REL-U, instead, learns a class-wise transformation that maps softmax responses to a “reliability space” calibrated on validation statistics, where distances from known reliable regions serve as uncertainty estimates. Both methods operate solely at the output level and therefore ignore possible inconsistencies within the internal representations of the network.

In the experiments that follow, these output-based baselines are compared to the proposed score, which links confidence to the *structural consistency* of layer-wise representations, measuring how stable the internal reasoning manifold remains under distributional or adversarial shifts.

5.1.2 Out-of-Distribution Detection

As already discussed in Sec. 2.4.2, a trained model is expected to operate reliably only within the statistical manifold spanned by its training data. *Out-of-distribution (OOD) detection* assesses the model’s ability to recognize when an input sample does not belong to this manifold and therefore lies outside its domain of competence. Accurate OOD detection is crucial for deploying neural networks in open-world scenarios, where the test distribution can differ from training conditions Hendrycks & Gimpel (2017); Lee et al. (2018a).

OOD inputs may arise from multiple sources:

- **Corrupted or shifted variants of the training set**, where synthetic degradations (noise, blur, contrast, pixelation) emulate common real-world perturbations;

- **Semantically different datasets**, such as SVHN [Netzer et al. \(2011\)](#) or PLACES365 [Zhou et al. \(2017\)](#), used as “far-OOD” sources whose content is unrelated to the target classes.

5.1.3 Evaluation metrics

OOD detection reduces to a binary discrimination problem between in-distribution (ID) and OOD samples based on a scalar confidence score $s(x)$. Two complementary metrics are typically reported:

- the **Area Under the Receiver Operating Characteristic (AUROC)**, defined as

$$\text{AUROC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(u)) du, \quad (5.1)$$

where TPR and FPR denote the true- and false-positive rates;

- the **Area Under the Precision–Recall Curve (AUPR)**, which measures detection quality under class-imbalance conditions [Davis & Goadrich \(2006\)](#).

Lower-level summaries such as the false-positive rate at 95% true-positive rate (FPR@95) are also reported for consistency with previous studies [Hendrycks & Gimpel \(2017\)](#); [Lee et al. \(2018b\)](#).

5.1.4 Existing approaches

Classical post-hoc methods differ mainly in the space where $s(\mathbf{x})$ is defined. Softmax-based scores, e.g. the MSP baseline [Hendrycks & Gimpel \(2017\)](#), evaluate confidence directly from the output probabilities. Feature-based methods such as the Mahalanobis Distance (MD) [Lee et al. \(2018b\)](#) estimate per-layer Gaussian statistics of training activations and compute the minimum Mahalanobis distance of a test sample to these distributions. Such detectors achieve high OOD accuracy but require supervised access to negative validation data and remain task-specific.

5.1.5 Geometric perspective adopted in this work

The proposed framework interprets OOD detection through a structural lens: instead of measuring deviations in the output or feature space, it evaluates the *layerwise geometric alignment* between an input’s internal representation and the reference prototypes (*proto-signatures*) derived from normal data. A low layer-wise structural consistency score signals a departure from the stable internal reasoning manifold and therefore acts

as an unsupervised indicator of OOD behaviour. The following subsection extends this analysis to deliberate, worst-case perturbations, introducing the notion of adversarial robustness.

Details on dataset splits, layer selection, and score aggregation used for OOD evaluation are provided in Appendix A.2.

5.2 From Distributional Shifts to Adversarial Perturbations

While Out-Of-Distribution (OOD) inputs arise from natural or uncontrolled deviations between training and test data, *adversarial perturbations* represent their deliberate, worst-case counterpart. Instead of sampling from a new distribution, an attacker constructs inputs that remain within the original data manifold but are specifically engineered to mislead the classifier. These inputs are visually indistinguishable from legitimate samples yet elicit incorrect predictions with high confidence, revealing how fragile the decision boundaries of deep networks can be.

Studying adversarial robustness thus complements OOD detection. The former examines how models behave when subjected to intentional, fine-grained manipulations of valid samples, while the latter probes how they respond to naturally occurring shifts or unseen distributions. Both phenomena challenge the notion of model trustworthiness, although from different angles. They also provide complementary testbeds for evaluating the proposed confidence score.

5.2.1 Adversarial Attacks and Robustness

As anticipated, *adversarial attacks* expose vulnerabilities in a network’s decision boundaries. They aim to find a small perturbation δ such that

$$x' = x + \delta, \quad \hat{y}(x') \neq \hat{y}(x), \quad \|\delta\|_p \leq \epsilon, \quad (5.2)$$

where x is a correctly classified input, x' the adversarial counterpart, $p \in \{2, \infty\}$ the norm used to constrain the perturbation magnitude, and ϵ the allowed perturbation budget.

Despite δ being imperceptibly small, the modified sample x' can lead to a confident misclassification.

Several well-established first-order attacks are considered in this work. They are summarized below, following the same notation as in the experimental setup.

FGSM (Fast Gradient Sign Method) [Goodfellow et al. \(2015\)](#). FGSM computes a single-step perturbation along the sign of the gradient of the loss $\mathcal{L}(f_\theta(x), y)$ with respect to the input:

$$x' = x + \epsilon \operatorname{sign}(\nabla_x \mathcal{L}(f_\theta(x), y)). \quad (5.3)$$

Here f_θ denotes the classifier, y the true label, and ϵ controls the perturbation amplitude. FGSM is efficient and exposes the local linear vulnerability of a model’s loss surface.

Basic Iterative Method (BIM) [Kurakin et al. \(2018\)](#). BIM refines FGSM through multiple small steps of size α :

$$x^{(t+1)} = \operatorname{clip}_{x,\epsilon} \left(x^{(t)} + \alpha \operatorname{sign} \left(\nabla_{x^{(t)}} \mathcal{L}(f_\theta(x^{(t)}), y) \right) \right), \quad x^{(0)} = x. \quad (5.4)$$

The operator $\operatorname{clip}_{x,\epsilon}(\cdot)$ projects each intermediate sample back into the ℓ_∞ ball of radius ϵ centered at the original input. BIM therefore accumulates small directional updates until the constraint is reached.

Projected Gradient Descent (PGD) [Madry et al. \(2017\)](#). PGD generalizes BIM by randomizing the initialization and maintaining projection within the feasible set at each step:

$$x^{(0)} = x + \operatorname{Uniform}[-\epsilon, \epsilon], \quad x^{(t+1)} = \Pi_{\mathbb{B}_\infty(x,\epsilon)} \left(x^{(t)} + \alpha \operatorname{sign} \left(\nabla_{x^{(t)}} \mathcal{L}(f_\theta(x^{(t)}), y) \right) \right). \quad (5.5)$$

$\Pi_{\mathbb{B}_\infty(x,\epsilon)}$ denotes projection onto the ℓ_∞ -norm ball. PGD is often considered the strongest first-order adversary and forms the basis for adversarial training benchmarks.

Carlini-Wagner (CW) attack [Carlini & Wagner \(2017\)](#). CW formulates adversarial generation as an explicit optimization problem seeking the smallest perturbation that changes the model’s decision:

$$\min_{\delta} \|\delta\|_p + c f(x + \delta), \quad \text{s.t. } x + \delta \in [0, 1]^d. \quad (5.6)$$

The term $f(x + \delta)$ enforces misclassification, c balances distortion and attack success, and d is the input dimensionality. CW typically yields near-minimal perturbations and serves as a benchmark for optimization-based attacks.

DeepFool (DF) [Moosavi-Dezfooli et al. \(2016\)](#). DeepFool approximates each decision boundary locally as a hyperplane and iteratively computes the minimal ℓ_2 perturbation required to cross it:

$$x^{(t+1)} = x^{(t)} + \frac{|f_k(x^{(t)}) - f_{\hat{y}(x^{(t)})}(x^{(t)})|}{\|\nabla f_k(x^{(t)}) - \nabla f_{\hat{y}(x^{(t)})}(x^{(t)})\|_2^2} (\nabla f_k(x^{(t)}) - \nabla f_{\hat{y}(x^{(t)})}(x^{(t)})), \quad (5.7)$$

where $f_k(x)$ is the logit score for class $k \neq \hat{y}(x)$. The iterations stop once $x^{(t)}$ crosses the nearest boundary. DeepFool produces almost imperceptible perturbations, providing a lower bound on the robustness margin.

Each of these attacks explores a distinct region of the vulnerability spectrum: FGSM reveals immediate gradient sensitivity, BIM and PGD expose iterative exploitation of local nonlinearity, CW identifies precise boundary-crossing perturbations, and DeepFool estimates the smallest feasible distortion. All adversarial examples are generated on the trained models, with perturbation budgets $\|\delta\|_\infty \leq \epsilon$ or $\|\delta\|_2 \leq \epsilon$ depending on the method, to assess the behaviour of the proposed confidence metric under progressively stronger adversarial pressure.

These attacks differ in how they alter the model’s *logits* $z(x)$: BIM and PGD tend to induce *overconfident* errors, while CW and DF lead to uncertain or low-confidence predictions. As a consequence, output-based confidence scores (e.g. MSP, DOC, REL-U) often fail to detect the first type but succeed on the latter, whereas the proposed representation-based structural consistency score remains effective in both cases by inspecting the coherence of internal reasoning across layers.

5.3 Computational Efficiency

The practical applicability of post-hoc confidence estimators also depends on their computational footprint. For each method, we evaluate the average time overhead per sample, defined as the additional computation introduced beyond the original forward pass. The proposed pipeline, relying on a matrix multiplication and clustering step per layer, remains lightweight compared to typical uncertainty-estimation techniques or adversarial detectors.

5.4 Positioning with Respect to the State of the Art

Existing post-hoc approaches in the literature typically focus on one specific aspect of model reliability. Methods such as DOC [Granese et al. \(2021\)](#) and REL-U [Dadalto et al.](#)

(2024) operate directly on the softmax outputs and are mainly designed for confidence estimation and, to a limited extent, OOD detection. Conversely, activation-based methods such as Mahalanobis Distance (MD) Lee et al. (2018a) excel at OOD and adversarial attack detection but cannot generalize to confidence estimation due to their supervised nature and dependence on negative validation sets. Finally, adversarial-specific detectors such as Feature Squeezing (FS) Xu et al. (2018) rely on heavy image pre-processing steps and remain confined to vision tasks.

From a computational standpoint, these methods differ substantially in cost. Softmax-based scores Hendrycks & Gimpel (2017) are inexpensive but limited to output statistics. Calibration-oriented techniques may require temperature scaling or gradient backpropagation. Activation-based detectors such as Mahalanobis Distance introduce additional overhead due to per-layer feature modeling and, in some cases, supervised validation sets.

By contrast, the proposed layer-wise structural consistency score requires a single offline decomposition (via SVD and clustering), while inference-time computation involves only low-dimensional projections and similarity operations. This results in a lightweight overhead relative to many feature-based or perturbation-based approaches, making the method suitable for scenarios where computational resources are constrained.

In summary, the proposed structural consistency framework provides a unified geometric criterion through which confidence estimation, OOD awareness, and adversarial robustness can be analysed. By grounding these assessments in the internal geometry of pretrained networks, the method complements output-based and feature-based post-hoc approaches while maintaining computational feasibility.

5.4.1 Summary of Evaluation Dimensions

The empirical assessment developed in the following chapters spans four complementary axes, corresponding to progressively richer notions of reliability:

1. **Confidence discrimination:** ability to distinguish correct from incorrect predictions based on internal structural measures (single-layer analysis via CNA/CNA-HC);
2. **Confidence estimation:** comparison of structural scores with output-based baselines in predicting reliability;
3. **Out-of-distribution detection:** identification of samples drawn from unseen or shifted domains;

4. **Adversarial robustness:** resilience of structural consistency under imperceptible worst-case perturbations;
5. **Computational efficiency:** runtime overhead and scalability of the proposed structural analysis.

While the single-layer experiments primarily establish the interpretative and discriminative validity of structural coherence, the multi-layer formulation enables systematic comparison against established baselines across distributional and adversarial settings.

6

Single-Layer Structural Agreement

This chapter collects the results presented in [Manovi et al. \(2024b\)](#) and expands them with further evidence to understand how the mathematical framework laid down earlier in the discussion translates in practice. Starting from hyperparameter tuning considerations, first we understand how the *accordance* between network under analysis and the designed peephole-based method manifests (CNA method from Sec. 3.4). Finally, we introduce corruptions on the input and compare the reaction observed on the DNN output with the confidence score adopted in the CNA-HC setting presented in Sec. 3.5.

6.1 Experimental Setup

As reference architecture we consider a DNN belonging to the Visual Geometry Group (VGG), namely VGG8B. This network shares the main building blocks with its relative VGG16, whose architecture is shown in Fig. A.1 and which is used in the following chapter to present the results with layer-wise analysis, although it is relatively shallow compared to it. This type of architecture is of interest as it can be used as stress-test: it is composed of both convolutional and fully connected layers and produces highly dimensional intermediate activations. Moreover, it is versatile to dataset adoption and, in general, is a reference model for benchmarking emerging techniques in the field of XAI.

Details on the size of the parameters that characterize these networks are reported in Appendix Tables A.1 and A.2 for VGG8B and VGG16, respectively. These tables serve as references to understand the entity of the performed dimensionality reduction.

We train the network considered here on the reference benchmarks FASHION-MNIST and CIFAR10. Information on these datasets, as input dimension and sizes of the training, validation and test splits, can be found in Appendix Table A.4.

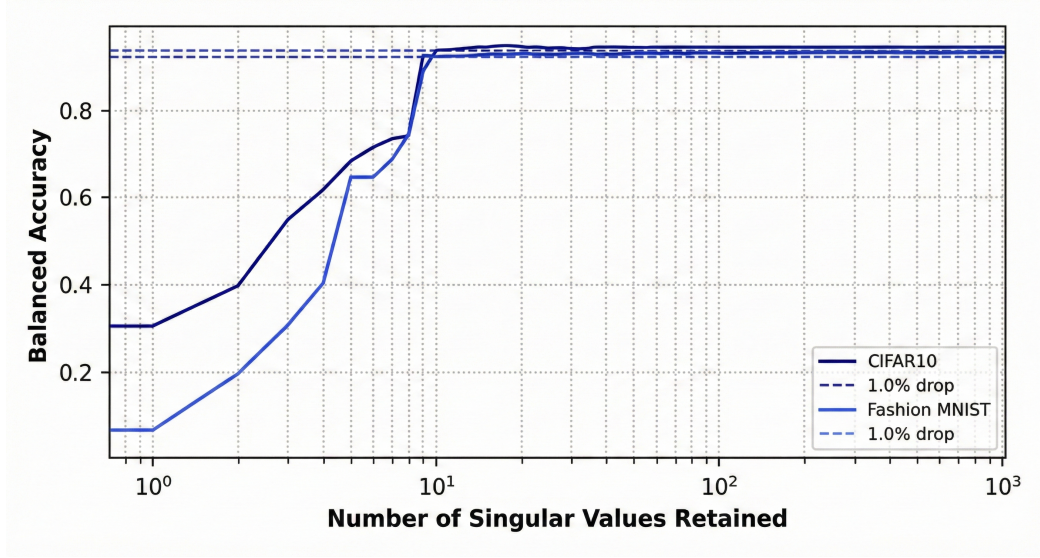


Figure 6.1. Balanced accuracy as a function of the number of singular values retained in the penultimate dense layer of VGG8b. Accuracy quickly converges and remains within 0.1% of the original value once $\kappa \approx 9$ components are kept, matching the ARI saturation point. Dashed lines indicate a 1% accuracy drop threshold.

6.1.1 Tuning the Peephole Dimensionality

The peephole dimensionality κ determines how many singular components are retained from the SVD of the layer weights and biases. This parameter governs both the expressiveness of the reduced-dimensionality representation and the computational complexity of the subsequent clustering.

We tune κ by evaluating two complementary criteria:

1. **Goodness of fit to class structure:** measured by the Adjusted Rand Index (ARI) between cluster assignments and ground-truth labels.
2. **Preservation of model accuracy:** measured by the balanced accuracy of the original DNN when the layer is replaced with its rank- κ approximation.

The consistency between Fig. 6.1 and Figs. 6.2- 6.3 demonstrates a strong correspondence between clustering goodness (ARI) and performance retention. In both CIFAR10 and FASHION-MNIST, the knee of the ARI curve aligns with the inflection point where balanced accuracy stabilizes. Keeping only the top $\kappa = 9$ singular vectors (out of 1024) preserves the discriminative structure of the network while drastically reducing dimensionality.

Substituting the original layer weights with their SVD-based approximation of rank $\kappa = 9$ led to less than 0.1% accuracy loss on both datasets [Manovi et al. \(2024a\)](#). This result provides quantitative evidence that the latent geometry captured by the first few singular components is both sufficient and semantically meaningful.

It also supports the interpretability-performance coexistence that motivates the peep-hole framework: the same compact representation that optimizes clustering alignment ($\text{ARI} \geq 0.95$) also preserves the network’s predictive quality.

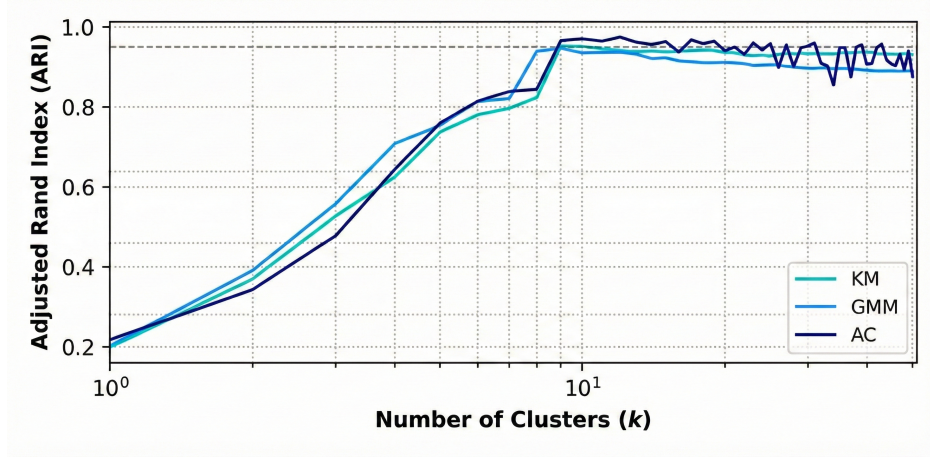


Figure 6.2. ARI as a function of the retained SVD dimensionality κ on FASHION-MNIST for KM, GMM, and AC. The index increases sharply for small κ and stabilizes around $\kappa \approx 9$ -10, indicating that a low-rank subspace already captures the class structure encoded in the penultimate layer.

6.1.2 CNA: Network-Clustering Accordance

The CNA framework evaluates the agreement between the decisions of the original network and those of its peephole-based classifier, acting as a proxy that is meant to certify the validity of the output based on the observation of model internals.

To analyse this structural agreement in relation to predictive correctness, each test sample is assigned to one of four regimes, as summarised in Table 6.1.

Although the PP/PN/NP/NN notation resembles a classical confusion taxonomy, its interpretation here is fundamentally different. CNA is not introduced as a surrogate classifier competing with the DNN, but as a *structural consistency probe*.

The four cases disentangle predictive correctness from internal geometric coherence:

- **PP** (Correct + Agreement): the network predicts the correct label and its internal geometry supports that decision. This represents structurally grounded reasoning.

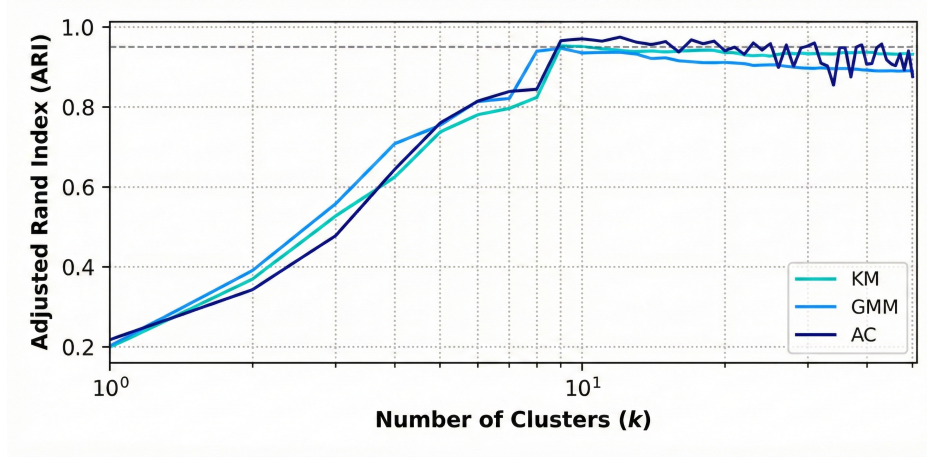


Figure 6.3. ARI as a function of the retained SVD dimensionality κ on CIFAR-10 for KM, GMM, and AC. The saturation point around $\kappa \approx 9-10$ aligns with the intrinsic class cardinality and with the dimensionality required to preserve balanced accuracy (cf. Fig. 6.1).

Table 6.1. Decomposition of CNA outcomes separating predictive correctness and structural agreement.

Code	Network w.r.t. Ground Truth	Structural Agreement (CNA)
PP	Correct	Agreement
PN	Correct	Disagreement
NP	Incorrect	Agreement
NN	Incorrect	Disagreement

- **PN** (Correct + Disagreement): the network is accurate, yet the layer geometry does not strongly support the prediction. These cases may correspond to borderline or locally unstable representations.
- **NP** (Incorrect + Agreement): the network makes an error, but its internal geometry is self-consistent. This suggests that the model has learned a coherent yet semantically misaligned representation, possibly reflecting class overlap or dataset bias.
- **NN** (Incorrect + Disagreement): both predictive accuracy and structural support fail, indicating instability at both decision and representation levels.

This decomposition therefore separates correctness from structural coherence. High CNA does not guarantee correctness; it guarantees internal consistency. This distinction

is crucial in safety-critical contexts, where coherent but wrong behaviour must be distinguished from incoherent decision-making.

The relative frequencies of these cases express the level of *accordance* between the network and the peephole-based proxy. High proportions of **PP** and **NN** indicate that both systems tend to make the same type of decision, either right or wrong, suggesting that the proxy captures the model’s reasoning manifold faithfully. Conversely, large **PN/NP** components reveal disagreement and possible uncertainty, or a limited capability of the model itself to distinguish between inputs sharing very similar features.

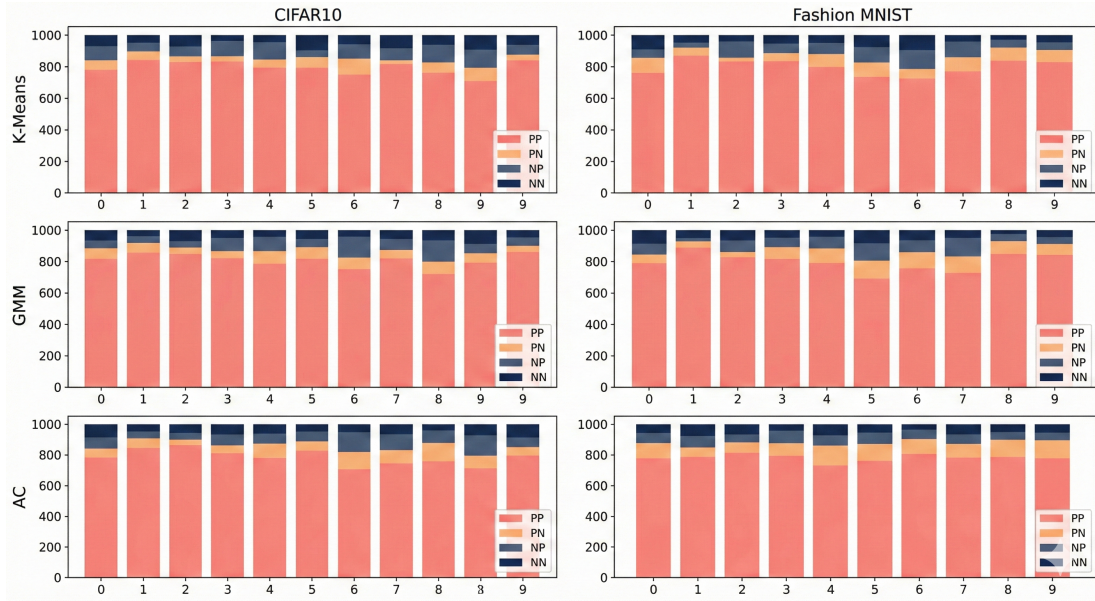


Figure 6.4. Distribution of CNA outcomes on CIFAR10 (left) and FASHION-MNIST (right) across clustering strategies. Samples are decomposed into the four regimes defined in Table 6.1. The predominance of agreement cases indicates that the peephole-based proxy captures the internal geometric organisation of the network, while disagreement highlights structurally ambiguous regions.

As illustrated in Fig. 6.4, the majority of samples fall in the PP regime for both datasets, indicating that correct predictions are typically supported by the internal layer geometry. Disagreement cases (PN and NP) concentrate around semantically overlapping classes (e.g., cat/dog in CIFAR-10 or shirt/top in Fashion-MNIST), where the learned representation itself exhibits reduced separability.

Among the clustering strategies, GMM achieves the highest structural alignment, suggesting that modelling covariance through the Mahalanobis metric better captures the intrinsic geometry of the activation manifold.

6.2 CNA with Confidence Estimation

To further probe the sensitivity and robustness of the peephole-based models, we analyze how both they and the DNN respond to smooth transitions between classes. Each test input is obtained by linearly combining two samples $\mathbf{x}'$ and $\mathbf{x}''$ belonging to distinct classes:

$$\mathbf{x} = (1 - \vartheta)\mathbf{x}' + \vartheta\mathbf{x}'', \quad \vartheta \in [0, 1]. \quad (6.1)$$

The parameter ϑ controls the degree of overlap between the two images, as illustrated by the examples in the bottom panels of each figure. Note that the samples have been chosen in the worst case selecting class or cluster prototypes (meaning examples closer to the relative centroid) with centroids close to each other in terms of Mahalanobis distance (Eq. 3.7).

For each mixed input $\mathbf{x}$, we compute the probability of belonging to either class for both the original VGG8B and the models (KM, GMM, AC) that cluster peepholes. For the DNN, this is the softmax output for the respective classes. For our auxiliary models, the probability is obtained by applying a softmax to the negative distances, Euclidean or Mahalanobis depending on the method, between the peephole of $\mathbf{x}$ and all cluster centroids:

$$p_c(\mathbf{x}) = \frac{\exp(-d_c(\mathbf{x}))}{\sum_{c'} \exp(-d_{c'}(\mathbf{x}))}. \quad (6.2)$$

This operation ensures that nearer cluster centroids yield higher assignment probabilities. Plots report the probability of the two classes as a function of ϑ , providing a measure of how each model perceives gradual semantic changes.

Importantly, the selected class pairs in the following examples do not represent arbitrary transitions. They are deliberately chosen among classes whose centroids lie close to each other in the corevector space, as measured by the Mahalanobis distance (Eq. 3.7). These constitute structurally challenging cases, where the internal geometry of the network already exhibits partial overlap.

The transition analysis presented in Sec. 6.3.2 below should therefore be interpreted as a stress test of structural sensitivity: if CNA-HC can detect ambiguity under these conditions, its response in more separated class regimes is expected to be even more pronounced.

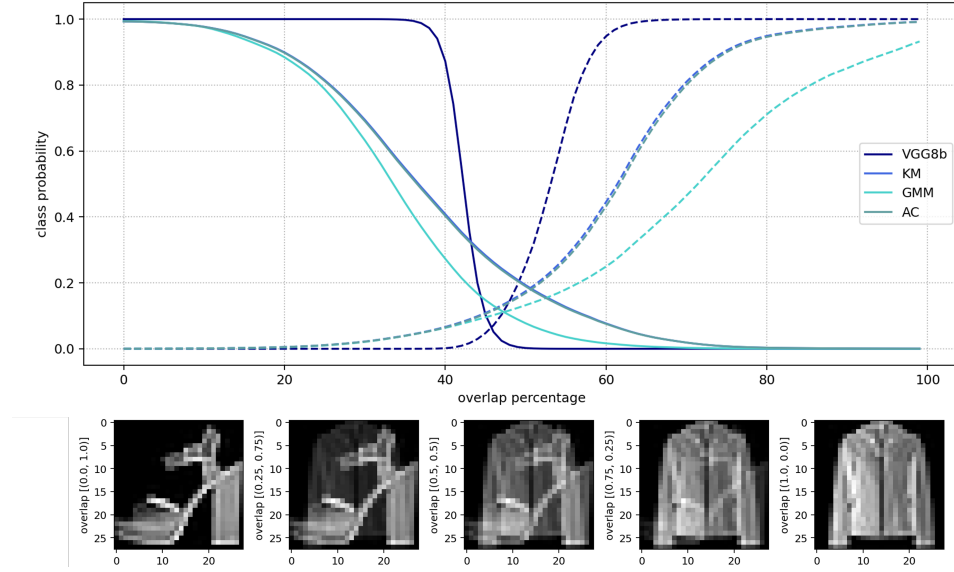


Figure 6.5. Fashion-MNIST – Transition *sandal* (class 5) $\rightarrow$ *coat* (class 4). Solid lines correspond to the first class and dashed lines to the second. VGG8b reacts sharply to the input interpolation, changing decision only when the overlap is already substantial ($\vartheta \simeq 0.4\text{--}0.5$), whereas the peephole-based proxies display smoother probability profiles, indicating earlier anticipation of the class change and better awareness of uncertainty.

6.3 Discussion and Structural Interpretation

The results obtained in this chapter provide the first empirical validation of the central hypothesis introduced in Chap. 2: that internal geometric structure is not an incidental by-product of training, but a measurable substrate of decision-making.

6.3.1 CNA as structural sanity check

The accordance analysis shows that, even when reduced to a single low-rank peephole extracted from the penultimate layer, the designed auxiliary mechanism reproduces the network’s behaviour with high fidelity (Fig. 6.4). The predominance of **PP** and **NN** cases indicates that the network’s output decisions are largely consistent with the clustering geometry of its own activations.

This is not a trivial observation. CNA does not rely on logits or output calibration; it relies solely on the internal activation manifold uncovered through SVD and clustering. The high agreement therefore implies that the discriminative boundaries learned by the DNN are already encoded in a compact geometric subspace. In other words, the

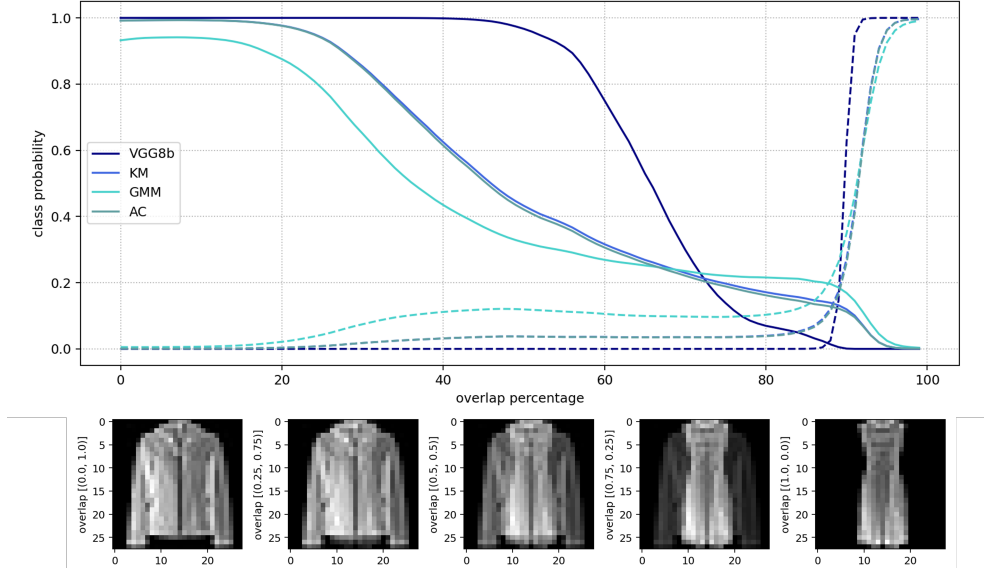


Figure 6.6. Fashion-MNIST – Transition *coat* (class 4) → *dress* (class 3). The smoother evolution of cluster-based probabilities compared to the abrupt DNN switch demonstrates that the peephole-based models identify ambiguous regions earlier and react proportionally to gradual changes in input semantics.

model’s reasoning is not arbitrarily distributed across dimensions: it concentrates along a few dominant structural modes.

Disagreement cases (**PN/NP**) concentrate in semantically ambiguous classes (e.g., cat/dog, shirt/top), which suggests that CNA is sensitive to regions where the learned representation itself is less sharply separated. This reinforces the interpretation of CNA as a probe of structural coherence rather than mere predictive coincidence.

6.3.2 Structural Behaviour under Progressive Corruption

The transition experiments provide a controlled way to observe how internal geometry reacts to gradual semantic deformation. Conceptually, one may imagine drawing a horizontal line corresponding to a minimum acceptable level of structural support, interpreted as a trust threshold on the CNA-HC score. The most sensitive method is therefore the one whose score decreases as soon as the input departs from its nominal prototype.

Across all tested class pairs (Figs. 6.5–6.11), the DNN exhibits a crisp, almost step-like boundary: the predicted class switches only when the interpolation clearly favours the new class. This behaviour reflects the discriminative nature of the final classifier.

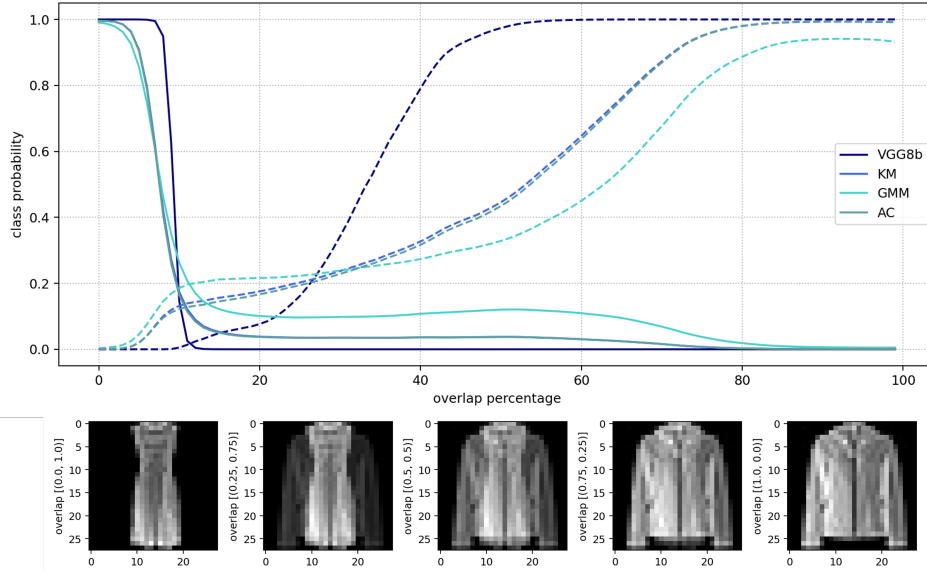


Figure 6.7. Fashion-MNIST – Transition *dress* (class 3) → *coat* (class 4). As the overlap grows, the DNN output remains almost binary, while clustering methods exhibit intermediate probabilities, signalling increased epistemic uncertainty.

In contrast, the peephole-based models (KM, GMM, AC) generate smooth probability transitions. For example, in the sandal–coat and coat–dress transitions (Figs. 6.5, 6.6), the CNA-HC score begins to decrease well before the DNN changes its predicted label, revealing early geometric awareness of ambiguity. Similarly, in visually dissimilar pairs such as bag–ankle boot (Fig. 6.8), a gradual crossing of probabilities reflects continuous deformation in feature space rather than abrupt decision collapse. The CIFAR-10 ship–deer example (Fig. 6.11) confirms that this phenomenon is not dataset-specific.

This smoothness is not cosmetic: it indicates that distances in the corevector space vary continuously with semantic deformation. The softmax mapping preserves this geometric continuity, allowing CNA-HC to express intermediate uncertainty regions that the DNN output layer suppresses.

Among the clustering strategies, GMM consistently provides the most stable and well-calibrated transitions, likely due to its probabilistic modelling of cluster covariance via the Mahalanobis metric. AC exhibits greater variance in intermediate regions, while KM remains sensitive but less statistically structured.

These results confirm that the peephole-based proxy does more than approximate predictive accuracy. It exposes how strongly the internal representation supports a decision and how that support degrades under controlled perturbation. This structural perspective will naturally extend to the multi-layer setting, where distributed peepholes

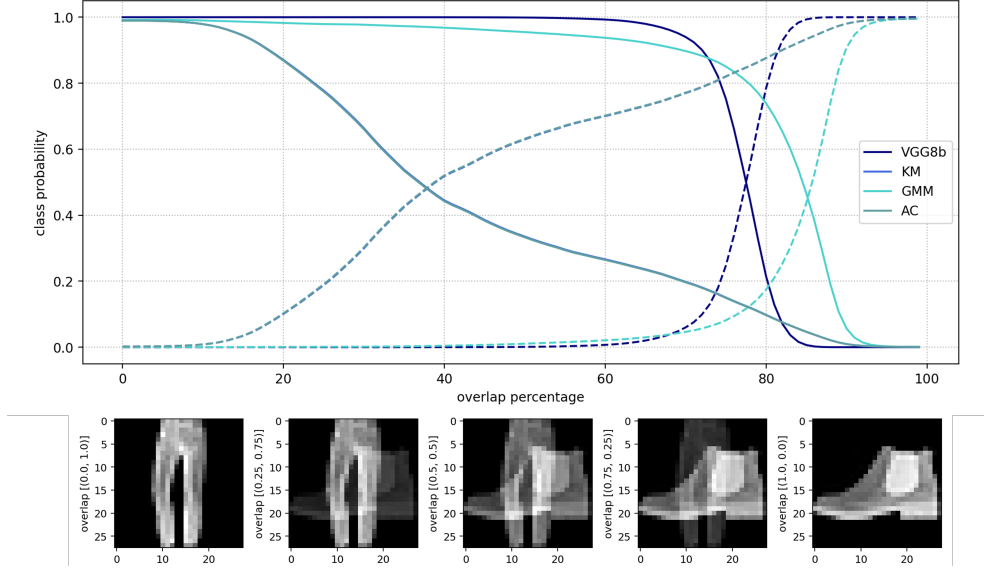


Figure 6.8. Fashion-MNIST – Transition *bag* (class 8) → *ankle boot* (class 9). The peephole-based methods again provide a smoother transition, revealing uncertainty intervals where features of both classes coexist.

trace the evolution of semantic support across depth.

6.3.3 CNA-HC as internal support intensity

While CNA establishes alignment, CNA-HC quantifies its strength. The visual transition experiments (Figs. 6.5–6.11) reveal a consistent and non-trivial phenomenon: the peephole-based probabilities evolve smoothly as the input interpolates between classes, whereas the DNN softmax exhibits abrupt, nearly binary switches.

This contrast has structural implications.

The DNN output layer reflects the final discriminative boundary. It does not express gradual internal ambiguity; it collapses uncertainty into a categorical decision. In contrast, CNA-HC operates in the corevector space, where distances to activation prototypes vary continuously. The softmax transformation preserves this geometry, producing probability-like scores that track the progressive deformation of the input in feature space.

Across all class transitions and datasets, three recurrent patterns emerge:

1. *Earlier sensitivity.* The CNA-HC score begins to decrease before the DNN changes its predicted label. This anticipatory behaviour indicates that the internal geometry detects ambiguity earlier than the final classifier boundary.
2. *Continuous uncertainty representation.* Instead of switching abruptly, the

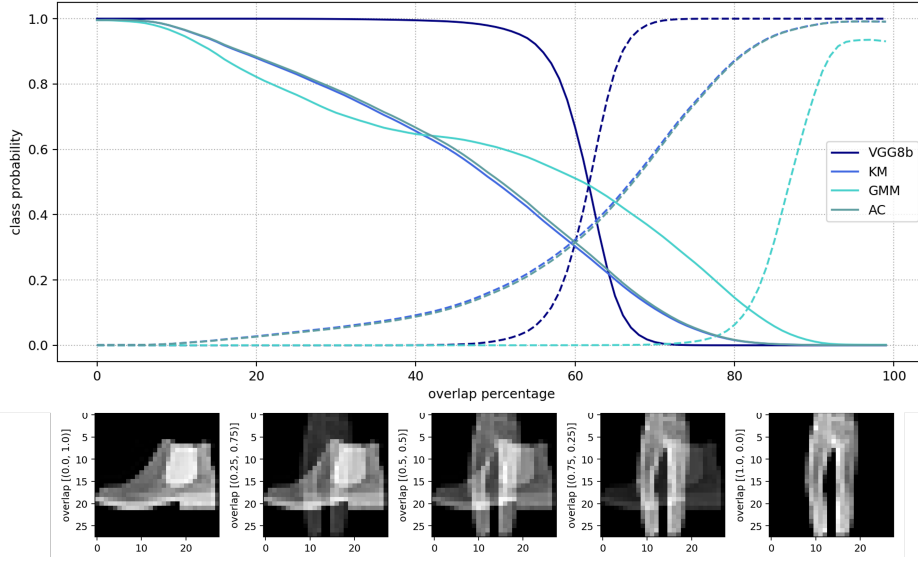


Figure 6.9. Fashion-MNIST – Transition *ankle boot* (class 9) $\rightarrow$ *trouser* (class 1). Across overlapping ratios, GMM shows the most stable behaviour, while AC displays larger variance in the intermediate range.

probabilities generated from peepholes form smooth crossings, revealing an intermediate region where both class prototypes exert comparable attraction. This region corresponds to genuine semantic overlap in the learned manifold.

3. *Method-dependent stability.* GMM consistently provides the most stable and calibrated transitions, reflecting its probabilistic modelling of cluster covariance. This confirms that the latent geometry is not merely partitioned, but statistically structured.

6.3.4 Implications for trust and reliability

Taken together, these findings elevate CNA-HC beyond a descriptive tool. It acts as a geometric confidence estimator grounded in internal structure rather than output entropy.

CNA asks: *Does the network follow its own geometry?*

CNA-HC asks: *How strongly does that geometry support the decision?*

The visual evidence demonstrates that internal structural support degrades smoothly under semantic corruption, even when the output prediction remains unchanged. This property is crucial in safety-critical scenarios: it enables early detection of borderline cases without requiring label supervision or adversarial generation.

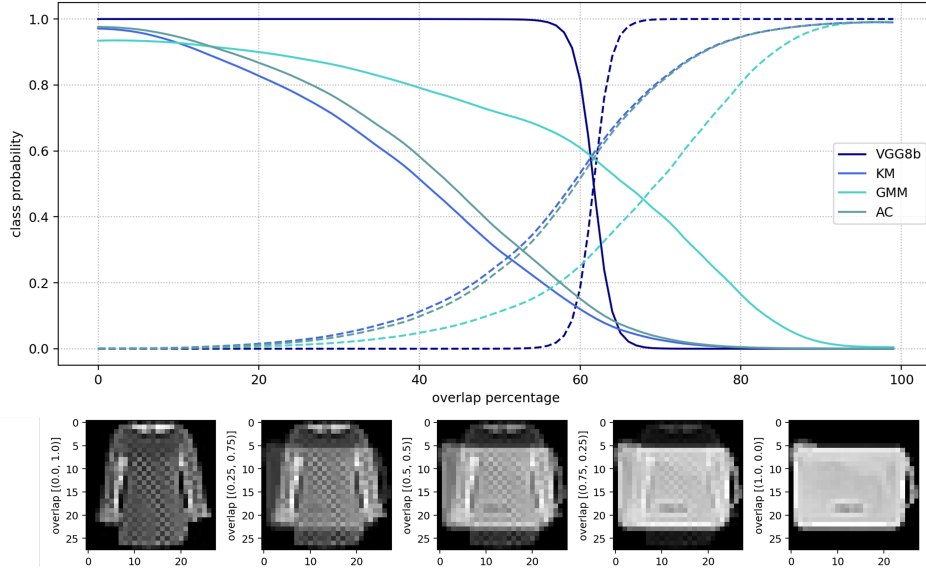


Figure 6.10. Fashion-MNIST – Transition *shirt* (class 6) $\rightarrow$ *bag* (class 8). Even in cross-category transitions, the probabilities generated by the designed proxy evolve gradually and symmetrically, whereas the DNN response remains nearly binary. This behaviour reflects the smoother manifold geometry induced by the SVD compression and clustering.

6.3.5 From empirical observation to structural principle

These single-layer results substantiate a broader thesis claim: performance and interpretability are not competing objectives. The same low-rank structure that preserves accuracy (Fig. 6.1) also reveals the semantic organisation of the representation (Figs. 6.2–6.3) and supports geometrically grounded confidence estimation.

The internal geometry uncovered by peepholes is therefore not an analytical artefact. It is a measurable trace of how the network organises evidence. Agreement, sharpness, and stability of this structure translate directly into observable reliability.

In this sense, CNA and CNA-HC constitute the first operational instantiation of trust as structural coherence. The following chapters extend this idea across depth, showing how distributed peepholes and layerwise consistency generalise this principle to full-network analysis.

Summary

The analyses presented in this chapter establish a first operational notion of structural agreement at the single-layer level. They show that a compact geometric proxy extracted

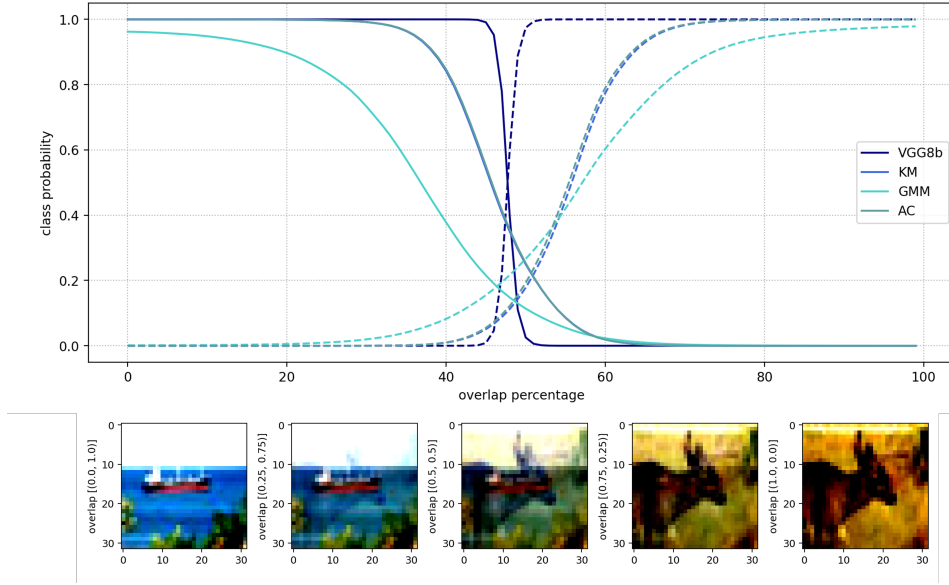


Figure 6.11. CIFAR-10 – Transition *ship* (class 8) $\rightarrow$ *deer* (class 4). The analysis extends beyond Fashion-MNIST: similar trends emerge in natural images, confirming that the peephole-based models exhibit earlier awareness of feature blending and potential ambiguity.

from the penultimate representation captures essential aspects of the network’s internal reasoning manifold.

CNA quantifies whether the network’s decision aligns with its own learned activation geometry, while CNA-HC measures the intensity of that structural support. Together, these results demonstrate that internal coherence is observable and measurable without relying on output calibration or external perturbations.

However, decision formation in deep architectures is inherently distributed across depth. Limiting the analysis to a single layer provides only a partial view of how semantic evidence accumulates, transforms, and stabilises before reaching the output.

The next chapter extends this perspective *from local structural agreement to distributed evidence tracing*, examining how multiple peepholes across depth reveal the evolution of semantic support and enable a global assessment of layer-wise structural consistency.

7

Layer-wise Analysis

The previous chapter established structural agreement at a single-layer level, showing that a compact geometric representation extracted from the penultimate layer already captures essential aspects of the network’s reasoning manifold.

Decision formation in deep architectures, however, is inherently distributed across depth. Internal evidence does not arise at a single stage but progressively concentrates, transforms, and stabilises throughout the hierarchy.

In this chapter, we extend the analysis *from local structural agreement to distributed evidence tracking*. By extracting peepholes at multiple layers, we construct layer-wise evidence maps that reveal how semantic support evolves from early feature extraction to final decision layers.

This shift enables two complementary perspectives. First, we provide a visual interpretation of internal reasoning dynamics through *evidence maps*, which trace the flow of semantic activation across depth. Second, we derive a *global layer-wise structural consistency score* that quantifies how coherently the internal representations support the final prediction.

These analyses combine visual interpretability and quantitative trust assessment, progressing from local structural sanity checks to a comprehensive evaluation of structural coherence across the entire network.

7.1 Experimental Setup

Compared to the settings mentioned in Sec.6.1, we increase the complexity of both the considered architectures and datasets. Hereafter we analyse results obtained with VGG16 (Fig. A.1, Table A.2) trained on both CIFAR10, for a preliminary introduction to the method, and CIFAR100. We extend the analysis to a non-convolutional and transformer-based architecture, namely ViT-B16, of which we report the sizes of the related parameters in Table A.3.

Comparison of VGG16 and ViT-B16. To investigate the impact of architectural inductive biases on learned representations, we compare two fundamentally different paradigms for visual processing: the convolutional network VGG16 and the transformer-

based model ViT-BASE16 [Dosovitskiy et al. \(2021\)](#). The VGG family, built upon deep stacks of 3×3 convolutions and spatial pooling, embodies strong local priors such as translation invariance and hierarchical feature composition. These priors enforce a spatially localized processing scheme that progressively abstracts information from edges and textures toward object-level concepts. Conversely, the Vision Transformer discards convolutional filters in favor of a *patch-wise tokenization* mechanism. Each 16×16 pixel patch is linearly embedded into a vector, augmented with positional encodings, and processed through a stack of *self-attention* and *feed-forward* layers. This design enables global contextual modeling, i.e. any image region can directly interact with any other, while reducing built-in locality assumptions. By contrasting these two architectures, we aim to examine how structural bias and global attention differently affect feature geometry, layer activation distributions, and the interpretability patterns revealed by our peephole analysis.

7.1.1 Tuning Considerations

In the single-layer setting of Chap. 6, the selected layer was located close to the classification head. In that case, it was reasonable to assume that the reduced-dimensional subspace could reflect the dataset class structure directly.

In the multi-layer setting, this assumption no longer holds. Earlier layers encode progressively more generic and entangled features, and no explicit ground-truth partition is available to evaluate clustering quality. Tuning must therefore rely on indirect criteria reflecting structural usefulness.

Rejection-aware criterion. A first tuning strategy adopts a selective-acceptance perspective, in the spirit of [Zhang et al. \(2023\)](#).

The acceptance decision is driven by a scalar peephole score computed per sample and per layer. Given the cluster-assignment posterior distribution $P(y \mid \text{cluster}_\ell(x))$, we consider two alternative confidence proxies:

$$s_\ell^{\max}(x) = \max_y P(y \mid \text{cluster}_\ell(x)), \quad (7.1)$$

which measures the dominance of the most likely tag, and

$$s_\ell^{\text{ent}}(x) = - \sum_y P(y \mid \text{cluster}_\ell(x)) \log P(y \mid \text{cluster}_\ell(x)), \quad (7.2)$$

which quantifies the dispersion of the cluster posterior via Shannon entropy. The former promotes sharply peaked structural assignments, while the latter favours low-

uncertainty representations.

For each layer ℓ and candidate configuration (κ, N_C) , we compute either of the above scores and sweep a threshold τ .

Predictions are accepted if

$$s_\ell(x) \geq \tau, \quad (7.3)$$

and rejected otherwise.

We retain configurations that:

1. improve the base DNN accuracy on the accepted set by a predefined margin, and
2. minimise the rejection rate among all satisfying configurations.

This criterion favours layer representations that separate confident from ambiguous cases with minimal sample discard. Configurations selected under these criterion for VGG16 on CIFAR10 are reported in Table A.7.

Top- k structural support criterion. An alternative strategy evaluates whether the correct class appears among the top- k most structurally supported components of the peephole at layer ℓ .

The purpose is not to construct a competing classifier, but to verify that meaningful semantic evidence for the correct class is present within the dominant modes of the layer representation.

At early depths, where representations capture generic visual primitives, a broader top- k tolerance is acceptable. Closer to the classification head, where representations become more class-specific, the criterion is progressively tightened.

Even when rows correspond to dataset classes (classification-map case), the peephole reflects shared structural features rather than discrete labels. The top- k criterion therefore measures whether correct semantic support emerges and concentrates consistently across depth.

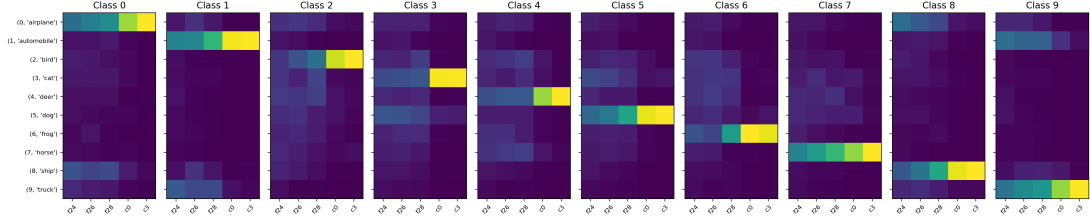
Configurations selected under these criteria for VGG16 and ViT-B16 on CIFAR100 are reported in Tables A.8 and A.9.

7.2 Tracing Decision Dynamics through Layer-wise Evidence Maps (CIFAR10)

In the present setting, the distributed evidence maps introduced in Sec. 4.3 take the specific form of *classification-maps*, where each row corresponds to a dataset class.

7.2 TRACING DECISION DYNAMICS THROUGH LAYER-WISE EVIDENCE MAPS (CIFAR10)

(a) Correct prediction (MSP > 0.99)



(b) Misclassified samples (MSP > 0.99)

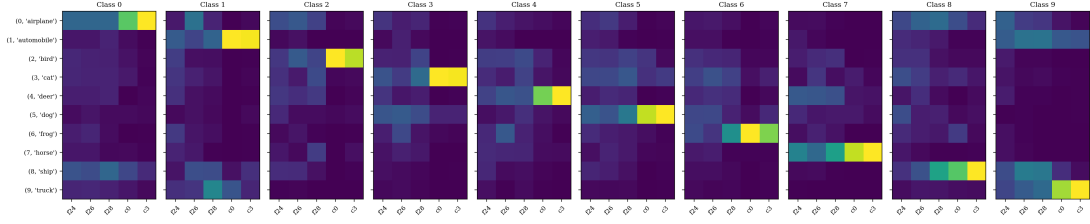


Figure 7.1. Classification maps at *very high* confidence: proto-class maps across selected peepholes for (a) a correct prediction and (b) a confident misclassification. Brighter intensities denote stronger class support.

For this introductory case, we selected the layers Conv5_1, Conv5_2, Conv5_3, FC1, and FC2 in Table A.2. The tuning configurations, obtained with the rejection-aware criterion of Sec. 7.1.1 and listed in Table A.7, highlight the compression ratio between the original activation space and the corresponding corevector representation.

To illustrate the interpretive potential of evidence maps, we compare six representative cases extracted from the CIFAR10 test set. Each case shows the proto-class maps derived from the selected peepholes, grouped by the outcome of the network (*correct* vs. *incorrect*) and by the output confidence level estimated through the maximum softmax probability (MSP). The three confidence regimes are defined as:

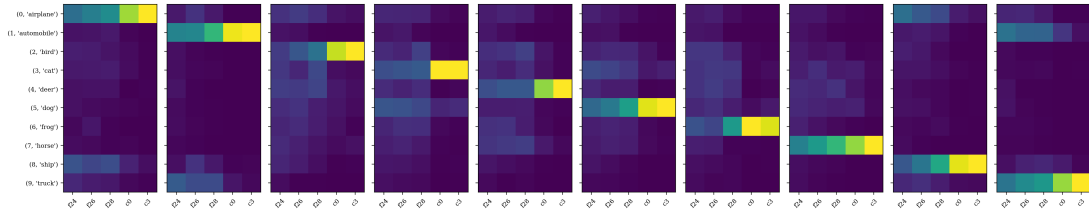
- **Very high confidence:** $\text{MSP} > 0.99$,
- **High confidence:** $0.9 < \text{MSP} \leq 0.99$,
- **Low confidence:** $\text{MSP} < 0.9$.

Each figure reports, for a given regime, a pair of examples: one correctly classified and one misclassified. In the figures, brighter intensities correspond to stronger class support.

Fig. 7.1 reports classification maps associated with very high output confidence. From the perspective of the softmax response alone, both decisions appear equally

7.2 TRACING DECISION DYNAMICS THROUGH LAYER-WISE EVIDENCE MAPS (CIFAR10)

(a) Correct prediction ($0.9 < \text{MSP} \leq 0.99$)



(b) Misclassified samples ($0.9 < \text{MSP} \leq 0.99$)

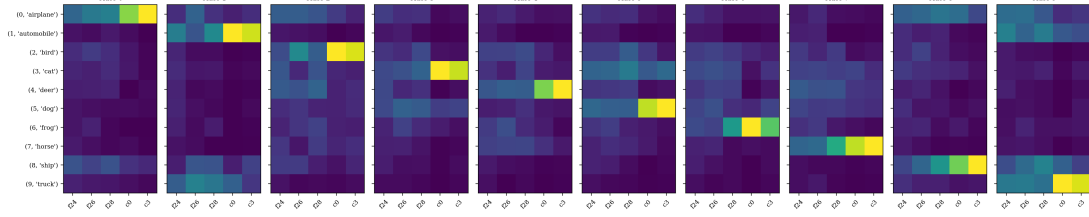


Figure 7.2. Classification maps at *high* confidence: proto-class maps across selected peepholes for (a) a correct prediction and (b) a confident misclassification. Brighter intensities denote stronger class support.

reliable. However, the decision trajectories induced by the inputs reveal substantially different geometric organizations across depth.

In the correctly classified case, early peepholes display moderate overlap among semantically related classes, reflecting shared low-level evidence. As depth increases, competing hypotheses are progressively attenuated and the proto-class support associated with the input increasingly concentrates on the target label. The deepest peepholes exhibit a clear consolidation of support on a single class row, indicating a coherent specialization process across layers. The sharp final prediction thus emerges as the endpoint of a gradual and structurally consistent evidence accumulation trajectory.

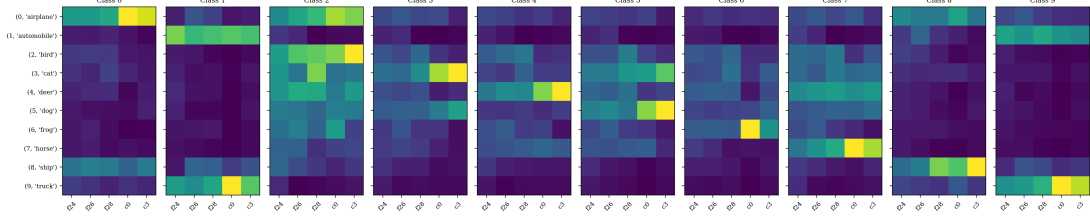
In the misclassification case, despite reaching equally high output confidence, the internal evolution follows a different path. Although the deepest layers display dominant support for a specific class, the consolidation process is less progressively structured, with intermediate peepholes revealing residual or fluctuating competing support. The network ultimately converges toward a semantically different attractor; yet this convergence arises from a weaker and less stable consolidation process.

The difference therefore lies not in the sharpness of the final decision, but in the trajectory that leads to it: distinct internal reasoning manifolds can produce equally confident outputs.

At slightly lower confidence (Fig. 7.2), the distinction between internal coherence

7.2 TRACING DECISION DYNAMICS THROUGH LAYER-WISE EVIDENCE MAPS (CIFAR10)

(a) Correct prediction (MSP < 0.9)



(b) Misclassified samples (MSP < 0.9)

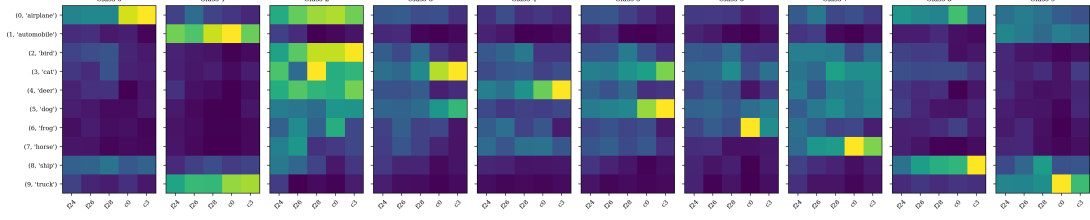


Figure 7.3. Classification maps at *low* confidence: proto-class maps across selected peepholes for (a) a correct prediction and (b) a misclassification. Brighter intensities denote stronger class support.

and output sharpness becomes more nuanced. In the correctly classified case, the decision trajectory still culminates in late-layer selectivity, with the proto-class support concentrating on the target label in the deepest peepholes. However, intermediate layers exhibit more persistent overlap among semantically related classes.

In datasets such as CIFAR10, pairs like cat/dog (classes 3/5) or automobile/truck (classes 1/9) share low- and mid-level visual features. Classification maps make visible how this shared representation is progressively disentangled across depth. Although the consolidation process appears slightly less abrupt than in the very-high-confidence regime, it remains structurally consistent, ultimately yielding a stable specialization toward the correct class manifold.

In the misclassification case, the same early semantic proximity does not resolve into clear separation. Support patterns fluctuate across intermediate peepholes, and competing classes are not progressively suppressed. Rather than a clean consolidation, the trajectory exhibits partial and unstable alignment, eventually favouring an alternative class without a fully structured build-up. The resulting prediction remains confident at the output level, yet its internal reasoning path reflects incomplete semantic disentanglement.

At low confidence (Fig. 7.3), the structural differences observed in the previous regimes become more pronounced. Across both cases, the evidence maps lose clear

hierarchical organization and late-layer consolidation weakens substantially. Proto-class support remains distributed across multiple competing classes even in the deepest peepholes, indicating that the network has not fully stabilized onto a single semantic manifold.

In the correctly classified case, a faint convergent tendency is still observable: although support remains diffuse, a mild concentration toward the target class gradually emerges. The reasoning trajectory can thus be described as fragile yet internally consistent, i.e., the network reaches the correct decision, but without a strong structural margin separating it from alternatives.

In the misclassification case, however, the trajectory exhibits markedly reduced coherence. Support patterns fluctuate across depth and competing classes persist without progressive suppression. Rather than a clean consolidation or even a structured drift, the evolution appears irregular and weakly organized, reflecting incomplete semantic disentanglement. The resulting prediction, whether correct or incorrect, lies near the boundary of the network’s decision manifold.

For this reason, all predictions with $\text{MSP} < 0.9$ are grouped under a common low-confidence regime. From a deployment perspective, such outputs would already warrant caution; classification maps provide structural evidence supporting this intuition by revealing reduced consolidation and increased internal competition.

Beyond post-hoc interpretation, this regime highlights the diagnostic potential of the proposed framework. Persistent diffuse patterns or repeated instability in specific class rows (e.g., class *bird* in Fig. 7.3) may indicate intrinsic representational difficulty, class overlap, or latent bias in the training data. Used during training or validation, evidence maps could therefore serve as structural monitors of semantic disentanglement, helping identify where the network struggles, how decision margins evolve across epochs, and whether specific classes require architectural or data-level refinement.

While CIFAR10 allows us to examine class-level consolidation patterns in a relatively low-dimensional semantic space, the transition to CIFAR100 tests whether the same structural tracing mechanism remains informative under increased class density and semantic granularity.

Each column of the resulting classification map corresponds to one of the selected peepholes (Table A.6), and each row to a dataset class. The map compactly visualizes how semantic categories are structured across depth, revealing emergent clusters and progressive separation among related classes.

This experiment demonstrates that the interpretive framework scales naturally to richer label spaces, preserving its ability to expose hierarchical consolidation patterns

at the dataset level.

7.3 Scaling Evidence Maps to Fine-Grained Classification (CIFAR100)

We now extend the analysis to a fine-grained classification setting. While CIFAR10 allows us to observe consolidation patterns in a relatively low-dimensional semantic space, CIFAR100 increases class density and semantic granularity, providing a more demanding test for layer-wise structural tracing.

In the following examples, evidence maps are instantiated as *classification-maps*, where rows correspond to dataset classes and columns to the selected instrumented layers. Each column therefore represents the peephole extracted at a given depth, and each row tracks the support assigned to a specific class across the network hierarchy.

For each input, the alignment between its layer-wise trajectory and the corresponding class prototype is quantified through the structural consistency measure defined in Eq. (4.4), denoted $s(\mathbf{x})$. The reported values of $s(\mathbf{x})$ should therefore be interpreted as a global indicator of how coherently the internal representation follows the expected class-specific manifold.

7.3.1 Nominal Case: Stable and Coherent Trajectory

Fig. 7.4 shows a correctly classified, in-distribution *sunflower* sample processed by the ViT-B16 model. The evidence map displays a smooth evolution of semantic evidence across depth: early layers activate broad visual primitives, intermediate layers refine the representation, and deeper layers consolidate evidence toward the *sunflower* tag. Interestingly, a secondary activation trail appears along the *bee* tag—an association that is also present in the corresponding prototype and is likely inherited from co-occurrence patterns in natural images, where flowers frequently appear alongside insects. This mild cross-tag activation is therefore *consistent* with the nominal behaviour learned during training, and the overall trajectory remains tightly aligned with the prototype, yielding a high score ($s(\mathbf{x}) = 0.9$).

7.3.2 Semantic Structure in VGG16 Classification Maps

The behaviour described above can be clearly observed in the VGG16 experiment on CIFAR-100. In this configuration, peepholes were attached to the last convolutional layer of each block and to the dense layers, producing evidence maps of dimension

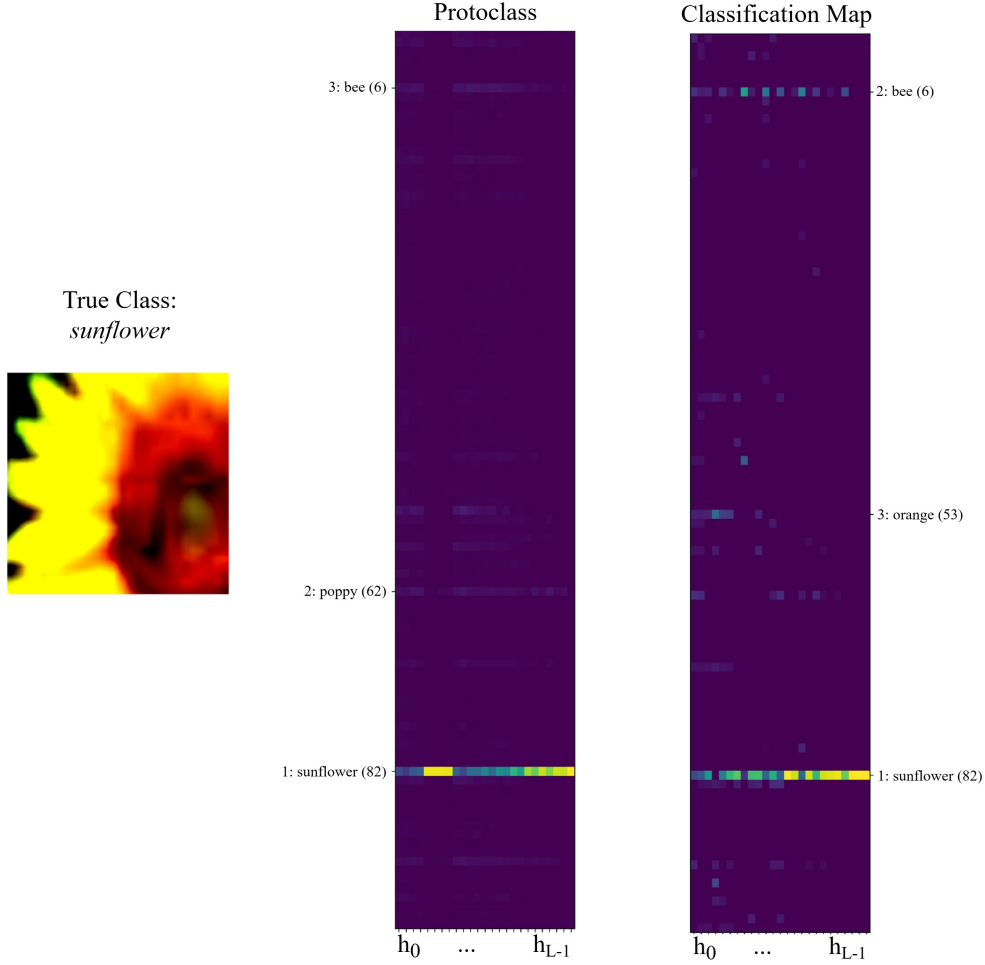


Figure 7.4. Nominal in-distribution *sunflower* sample processed by ViT-B16. The classification map reveals a stable and coherent layer-wise trajectory: evidence concentrates progressively toward the *sunflower* tag, with a mild, semantically plausible co-activation of the *bee* tag due to naturally occurring visual co-occurrence patterns. The resulting high $s(x)$ score reflects strong alignment between the sample-specific trajectory and the prototype for the predicted class.

$T \times L$, where T denotes the number of tracked high-level tags and L the number of instrumented layers.

Fig. 7.5 shows the classification map of a sample classified as *boy* alongside the corresponding prototype. At a glance, the two maps exhibit strong alignment in the upper and middle layers, indicating that the internal activations of this sample follow the expected reasoning manifold for the class. At the same time, mild activations appear in rows associated with semantically related categories such as *baby*, *girl*, and *man*. These co-activations reflect shared visual concepts (facial structure, skin texture, clothing

cues) that naturally span multiple human-related classes. Far from signalling confusion, this overlap reveals how the network reuses generalizable semantic components before specialising them in deeper layers.

This example reinforces a central insight of the peephole framework: trustworthy behaviour does not require perfectly isolated features, but structured interaction between shared and class-specific attributes across depth.

A trustworthy model does though display stable internal geometry, i.e., early-layer overlaps aligned with broad visual semantics, followed by progressive refinement toward the target class. Departures from this pattern, such as excessive activation in unrelated tags or missing evidence where the prototype expects it, may indicate abnormal reasoning or unreliable predictions.

7.3.3 Near-OOD Case: Output Stable, Internal Trajectory Degraded

Fig. 7.6 shows a visually plausible *near-OOD* input (from CIFAR-100C). The model assigns the same class as in the clean case and retains an extremely high softmax confidence (dropping only from 1.00 to 0.9999), suggesting a superficially stable prediction. However, the evidence map reveals that several mid- and deep-layer peepholes begin to drift away from the prototype trajectory: the evidence that should accumulate smoothly toward the predicted tag becomes fragmented, with certain layers activating unrelated semantic patterns.

These deviations translate into a noticeable drop in the structural consistency score (from $s(\mathbf{x}) \approx 0.85$ on the clean sample to $s(\mathbf{x}) \approx 0.75$ under corruption), indicating that the internal reasoning pathway is no longer consistent with nominal behaviour despite the negligible change in the output probabilities. This example illustrates how near-OOD shifts can leave the logits nearly untouched while disrupting the structure of the internal trajectory, a form of failure that the proposed score $s(\mathbf{x})$ is specifically designed to expose.

7.3.4 Adversarial Example: Imperceptible Change, Internal Collapse

Fig. 7.7 shows a ViT-B16 sample subjected to an imperceptible adversarial perturbation (e.g. PGD or CW). The visual content of the image is nearly identical to the clean one, and the softmax distribution often remains sharply peaked. However, the evidence map reveals a dramatic alteration of the internal trajectory: shallow layers remain partially aligned, but deeper layers exhibit a collapse of semantic evidence, leading to a score $s(\mathbf{x}) \approx 0$. This illustrates how adversarial attacks specifically target the internal reasoning chain, while the output layer hides the disruption.

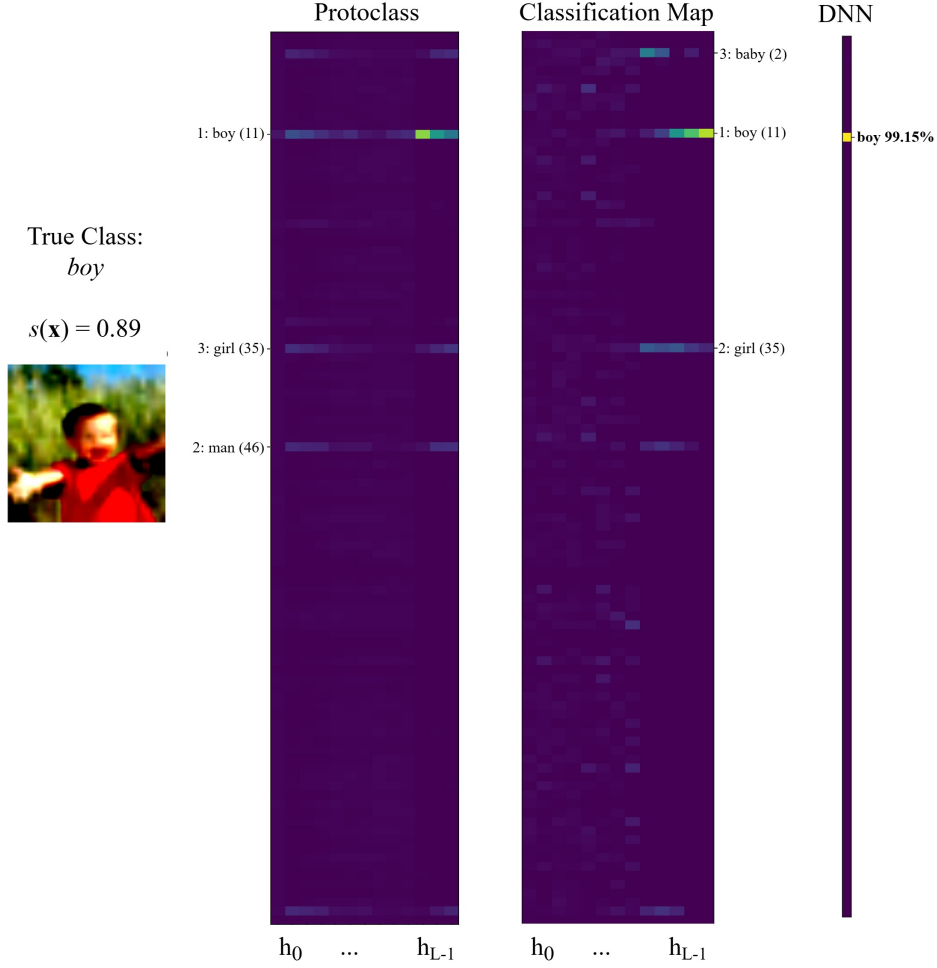


Figure 7.5. Classification map extracted from VGG16 on CIFAR-100 for a sample classified as *boy*. The dominant trajectory aligns with the predicted class, while shallow and intermediate layers display mild activations toward semantically related categories (*baby*, *girl*, *man*). These associations reflect shared visual features and gradually diminish as the representation specialises in deeper layers.

These qualitative examples highlight how the proposed methodology allows to detect distributional drift and adversarial perturbations at the *internal* representation level, even when output probabilities remain unchanged. This motivates the quantitative analyses that follow, where the score defined in Eq. 4.4 is evaluated systematically across architectures, datasets, and reliability dimensions.

For the same sample, the softmax confidence remains deceptively high under both attacks (from 1.00 on the clean input to 0.9998 for BIM and 0.9996 for PGD), whereas our structural consistency score drops sharply (from $s(\mathbf{x}) \approx 0.86$ to $s(\mathbf{x}) \approx 0.41$ for

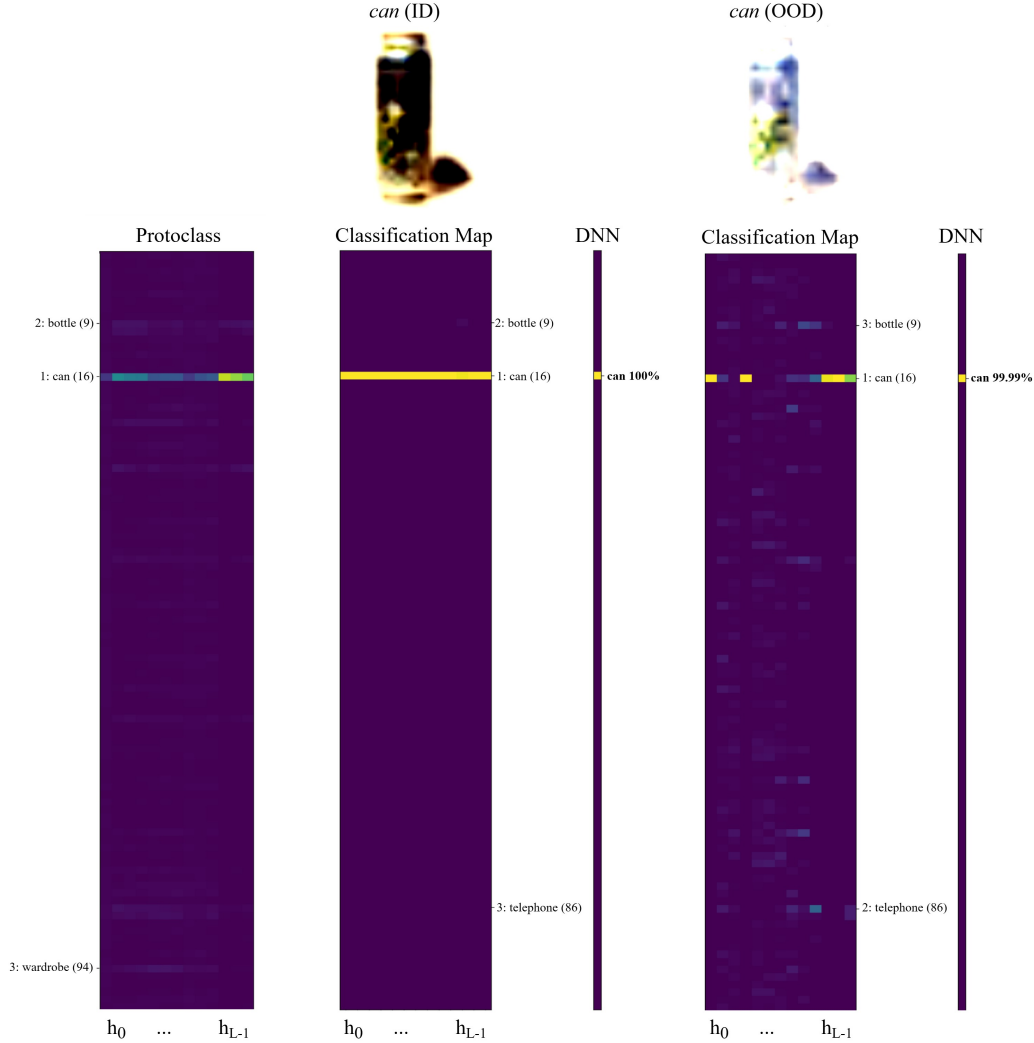


Figure 7.6. Near-OOD sample. The model’s softmax confidence remains virtually unchanged ($1.00 \rightarrow 0.9999$), yet the classification map reveals a degradation of the internal trajectory, with several layers drifting away from the prototype pattern. The structural consistency score drops accordingly (from $s(\mathbf{x}) \approx 0.85$ to $s(\mathbf{x}) \approx 0.75$), exposing inconsistencies that are not visible at the output level.

BIM and $s(\mathbf{x}) \approx 0.28$ for PGD). These values, reported in a more complete version in Sec. A.5, match the structural collapse visible in the classification maps: although the output distribution barely changes, late-layer peepholes deviate substantially from the prototype, revealing the disruption of the internal reasoning chain induced by the adversarial perturbation.

The qualitative evidence presented above illustrates a common pattern across architectures: inputs that follow the expected semantic trajectory maintain a high

7.3 SCALING EVIDENCE MAPS TO FINE-GRAINED CLASSIFICATION (CIFAR100)

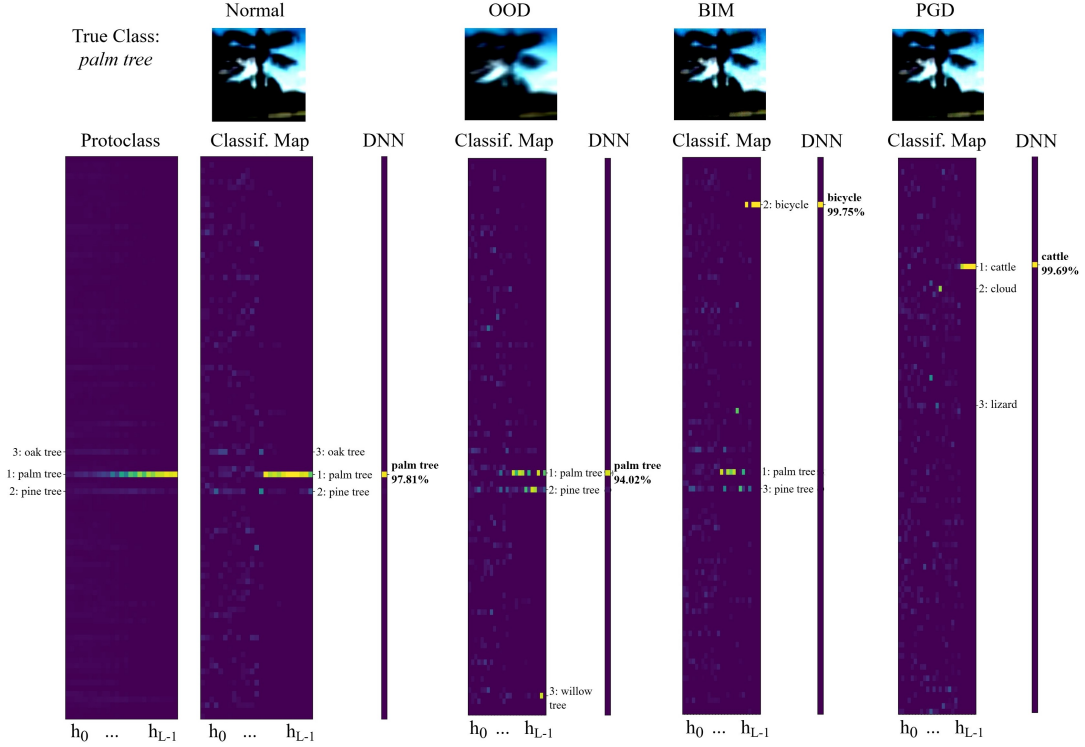


Figure 7.7. Adversarial example for ViT-B16. Although visually indistinguishable from the clean input and still producing a confident softmax prediction, the perturbed sample shows a sharp collapse of internal consistency in deeper layers. The proposed approach allows to detect this deviation immediately, yielding a low score $s(\mathbf{x})$ (Eq. 4.4) and exposing the hidden fragility of the model’s decision process.

degree of layer-wise alignment, whereas near-OOD or adversarial manipulations, often imperceptible at the pixel level, induce a gradual drift from the prototype signature. In all examples, the softmax output remains high, sometimes virtually unchanged, while the evidence maps reveal clear internal misalignment. This gap between output confidence and internal coherence is precisely what the $s(\mathbf{x})$ score is designed to quantify.

Having established the interpretive value of these depth-wise trajectories, we now turn to a systematic quantitative evaluation. The following sections examine how reliably the proposed consistency score detects these discrepancies across large datasets, multiple architectures, and controlled perturbation regimes, and how it compares with state-of-the-art post-hoc confidence estimators.

7.4 Quantitative Assessment of Model Trustworthiness

The qualitative examples presented in Sec. 7.3 illustrated, on a case-by-case basis, how internal structural deviations emerge under nominal, near-OOD, and adversarial conditions. While these instances provided intuition about the behaviour of the proposed score, they remain inherently local observations.

We now shift from sample-level inspection to dataset-level evaluation, analysing the behaviour of the score across complete test sets and controlled perturbation regimes.

Throughout this section, the proposed structural score is compared against representative post-hoc confidence estimators introduced in the background discussion: Maximum Softmax Probability (MSP) and Mahalanobis-based detectors (MD) (Sec. 5.1.4), as well as Rel-U, DOC, and Feature Squeezing (FS) (Sec. 5.4).

The analysis is structured along three complementary axes:

1. **Confidence estimation on in-distribution data**, measuring the separation between correct and incorrect predictions;
2. **Out-of-distribution (OOD) detection**, assessing sensitivity to both gradual corruptions and large domain shifts;
3. **Adversarial robustness**, quantifying the detectability of structurally inconsistent perturbations.

In OOD detection and adversarial robustness analysis, for the Mahalanobis approach, we distinguish between the supervised variant (MD-A), which exploits class-conditional statistics estimated using labelled data, and the unsupervised variant (MD-U), which relies solely on feature geometry without explicit OOD supervision.

In addition, in Sec. 7.8 we analyse computational cost to determine whether the proposed framework remains compatible with practical deployment constraints. The main objective is to determine under which conditions layer-wise structural coherence provides a meaningful and complementary reliability signal.

7.5 Confidence Estimation

We begin by evaluating whether layer-wise structural coherence provides a meaningful signal for distinguishing correct from incorrect predictions on in-distribution data. Rather than relying on output probabilities alone, the proposed score $s(\mathbf{x})$ (Eq. 4.4) measures the consistency of the internal semantic trajectory across depth.

This experiment does not aim to replace output-based confidence scores, but to assess whether structural alignment captures complementary information about prediction reliability.

Table 7.1 reports AUC and FPR@95 (Sec. 5.1.3) for the discrimination between correct and incorrect in-distribution predictions.

The proposed structural score remains competitive with established output-based confidence estimators across both architectures. On VGG16, the achieved AUC is slightly below MSP, DOC and Rel-U, but it achieves the lowest FPR@95 jointly with DOC. On ViT-B16, the gap in AUC remains limited, while FPR@95 matches the best-performing methods.

These results indicate that depth-wise structural coherence carries reliability-relevant information comparable to that extracted from output probabilities.

Although it does not consistently surpass softmax-based scores in raw AUC, its performance remains within a narrow margin without relying on logits, temperature scaling, or density modelling at the output layer.

Importantly, the structural score achieves this level of discrimination using only forward activations and precomputed geometric descriptors. This confirms that internal alignment across layers provides a measurable signal correlated with predictive correctness at dataset level.

Table 7.1. AUC (higher is better) and FPR@95 (lower is better) for in-distribution confidence estimation. Best results are shown in bold, second-best underlined. Equal values share the same rank.

Method	VGG16		ViT-B16	
	AUC	FPR@95	AUC	FPR@95
Ours	0.83	0.64	<u>0.87</u>	0.57
MSP	0.87	<u>0.65</u>	0.90	0.57
DOC	0.87	0.64	0.90	<u>0.59</u>
Rel-U	<u>0.85</u>	0.80	0.83	0.78
FS	0.84	0.76	0.85	0.74

Regarding limitations of the considered baselines, we observe that Rel-U exhibits architectural instability, deteriorating sharply on ViT-B16. FS remains sensitive to benign high-frequency components and is computationally heavy. By contrast, our method remains stable across architectures without additional supervision or input pre-processing.

7.6 Out-of-Distribution Detection

A reliable confidence indicator should decrease when an input departs from the distribution on which the model was trained. Such OOD samples may arise due to environmental changes, sensor degradation, or semantic mismatch, and constitute a critical failure mode: modern deep networks can remain sharply overconfident even when presented with unfamiliar data [Minderer et al. \(2021\)](#).

To evaluate the behaviour of the proposed structural score $s(\mathbf{x})$ under distribution shift, we consider two complementary scenarios:

- **near-OOD**: structured corruptions of the CIFAR-100 test set, as defined in CIFAR-100C;
- **far-OOD**: distinct input domains, represented by SVHN and PLACES365.

These regimes probe different reliability requirements: near-OOD tests sensitivity to gradual distortions that largely preserve semantics, while far-OOD evaluates the ability to detect a complete departure from the learned manifold.

7.6.1 Near-Out-of-Distribution Behaviour

Near-OOD evaluation is performed on CIFAR-100C [Hendrycks & Dietterich \(2019\)](#), which applies fifteen corruption types (noise, blur, weather, and digital distortions), each at five severity levels $\lambda \in \{1, \dots, 5\}$. Increasing λ induces a progressively stronger deviation from the clean distribution while keeping the underlying semantic content unchanged.

Across both VGG16 and ViT-B16, the proposed structural score $s(\mathbf{x})$ exhibits a monotonic increase in discriminative performance as corruption severity λ increases. While mild corruptions remain closer to the in-distribution manifold and are therefore harder to distinguish, stronger perturbations induce progressively larger deviations in the internal layer-wise trajectories, resulting in clearer separation between clean and corrupted samples.

This behaviour is consistent with the structural interpretation developed earlier: as corruption accumulates, a growing fraction of instrumented layers diverges from its nominal prototype alignment. The resulting reduction in structural agreement translates into improved detectability at the dataset level.

In contrast, MSP and DOC remain comparatively insensitive under moderate corruption. Rel-U is more rigid and provides limited separation across severities. FS can yield meaningful separation in some regimes, but at a significantly higher computational

Table 7.2. AUC values for near-OOD detection (CIFAR-100 vs. CIFAR-100C) across corruption severity λ . Best results per column are in bold and second-best are underlined.

Table 7.2.1. VGG16 Architecture					
Severity (λ)	1	2	3	4	5
MSP	0.59	0.63	0.67	0.69	0.73
DOC	0.60	0.64	0.68	0.70	0.73
Rel-U	0.61	0.65	0.68	0.71	0.74
FS	0.51	0.53	0.55	0.58	0.61
MD-A	<u>0.64</u>	<u>0.67</u>	<u>0.70</u>	<u>0.72</u>	<u>0.75</u>
Ours	0.70	0.72	0.74	0.76	0.78

Table 7.2.2. ViT-B16 Architecture					
Severity (λ)	1	2	3	4	5
MSP	0.65	0.67	0.70	0.73	0.76
DOC	0.64	0.66	0.69	0.72	0.75
Rel-U	0.65	0.67	0.70	0.73	0.76
FS	0.56	0.58	0.60	0.63	0.66
MD-A	<u>0.67</u>	<u>0.69</u>	<u>0.72</u>	<u>0.75</u>	<u>0.78</u>
Ours	0.75	0.77	0.79	0.81	0.82

cost due to repeated input transformations. MD-A is the most competitive, but its use is tied to modelling assumptions and to an estimation procedure that is not purely structure-driven in the same sense as the proposed score.

The quantitative evidence reported in Table 7.2 confirms this trend systematically. For both architectures and across all severity levels, the proposed score achieves the highest AUC, with the margin becoming more pronounced at higher corruption levels where structural deviations accumulate across depth. Importantly, this behaviour is obtained without input transformations, auxiliary detectors, or additional supervision.

7.6.2 Far-Out-of-Distribution Detection

To evaluate robustness under more pronounced domain shifts, we use two semantically distinct datasets: SVHN [Netzer et al. \(2011\)](#), which contains street-view digits, and PLACES365 [Zhou et al. \(2017\)](#), a large-scale scene recognition dataset. Both differ markedly from CIFAR-100 in content and low-level statistics.

Table 7.3 reports the AUC values under pronounced domain shifts. As expected, MD-A achieves the highest performance across most configurations, benefiting from explicit modelling of class-conditional feature distributions and access to labelled data during calibration.

Table 7.3. Far-OOD detection AUCs for VGG16 and ViT-B16 trained on CIFAR-100 (ID) and evaluated against PLACES365 (P365) and SVHN. Best results per column are in bold and second-best are underlined.

Method	VGG16 / P365	VGG16 / SVHN	ViT-B16 / P365	ViT-B16 / SVHN
Ours	<u>0.84</u>	0.85	0.86	0.93
MSP	0.83	0.82	0.83	0.91
DOC	<u>0.84</u>	0.83	0.84	0.92
Rel-U	0.74	0.65	0.67	0.64
FS	0.80	0.89	0.85	<u>0.94</u>
MD-A	0.99	<u>0.88</u>	0.99	0.95
MD-U	0.44	0.76	<u>0.87</u>	0.72

Among the unsupervised and purely post-hoc approaches, the proposed structural score $s(\mathbf{x})$ consistently ranks among the top three strongest methods. In particular, it matches or exceeds output-based baselines such as MSP, DOC and Rel-U, while remaining competitive with more computationally demanding techniques such as FS.

The results indicate that deviations in layer-wise structural coherence remain informative even under substantial semantic shifts.

Although Gaussian feature modelling can yield superior absolute AUC values in certain regimes, the proposed score provides a *fully unsupervised alternative* that does not rely on distributional assumptions, auxiliary OOD data, or gradient-based refinement.

7.7 Adversarial Robustness

Adversarial perturbations represent a stringent form of distributional shift: small, carefully crafted modifications can force a model to produce an incorrect prediction while keeping the input visually indistinguishable from the original. Following the definitions introduced in Sec. 5.2.1, we evaluate the ability of confidence scores to discriminate between clean and adversarially perturbed samples. Only correctly classified clean samples are considered, so that robustness reflects the degradation introduced by the attack rather than pre-existing errors.

Adversarial detection is framed as a binary discrimination task where higher AUC values indicate better separation between unperturbed and adversarial samples. Table 7.4 reports results for four standard first-order attacks, namely BIM, PGD, CW and DF, on both VGG16 and ViT-B16.

Across architectures and attack types, the structural score $s(\mathbf{x})$ exhibits consistently

Table 7.4. Detection performance (AUC) under different attacks on VGG16 and ViT-B16. Best results per column are in bold and second-best are underlined.

Method	VGG16				ViT-B16			
	BIM	CW	DF	PGD	BIM	CW	DF	PGD
Ours	0.99	0.99	0.88	0.98	<u>0.83</u>	0.99	0.92	<u>0.84</u>
MSP	0.46	0.90	<u>0.92</u>	0.44	0.57	0.91	0.90	0.55
DOC	0.48	0.87	0.91	0.44	0.58	0.88	0.89	0.55
Rel-U	0.42	0.83	0.79	0.40	0.48	0.76	0.72	0.48
FS	<u>0.87</u>	<u>0.94</u>	0.83	0.88	0.85	<u>0.96</u>	0.90	0.82
MD-A	0.78	0.63	0.93	<u>0.92</u>	0.82	0.59	0.98	0.98
MD-U	0.69	0.53	0.87	0.88	0.78	0.55	<u>0.97</u>	0.98

strong behaviour, although its relative ranking depends on both the model and the attack.

On VGG16, $s(\mathbf{x})$ achieves the highest AUC under BIM, CW and PGD, and remains competitive under DF, where the supervised Mahalanobis detector MD-A attains the best result. This indicates that layer-wise structural coherence is highly sensitive to first-order gradient-based perturbations that induce coordinated deviations across multiple layers.

On ViT-B16, the picture is more heterogeneous. The structural score remains the best performer under CW and second-best under BIM and PGD, while Mahalanobis-based methods dominate under DF and PGD. This behaviour reflects the different internal geometry of transformer architectures, where feature-space modelling can sometimes capture distributional shifts more sharply than depth-wise agreement alone.

Compared to purely output-based scores such as MSP and DOC, which degrade substantially under several attacks, $s(\mathbf{x})$ maintains stable and often superior discrimination. FS provides competitive results in multiple settings, but at the cost of repeated input transformations and significantly higher computational overhead.

Overall, the results confirm that adversarial perturbations frequently induce systematic inconsistencies in the internal layer-wise trajectory. The proposed structural score captures these deviations without supervision, distributional assumptions, or gradient-based optimisation, extending the same geometric principle previously observed in confidence estimation and OOD detection.

7.8 Computational Complexity

A meaningful post-hoc confidence estimator must balance predictive reliability with operational feasibility. In realistic deployments, ranging from large-scale inference pipelines to resource-constrained embedded or onboard systems, computational overhead rapidly becomes a limiting factor.

This section analyses the computational footprint of the proposed layer-wise approach, distinguishing between offline preparation costs and inference-time overhead, and compares them with representative state-of-the-art baselines.

7.8.1 Offline Computational Requirements

The pre-processing stage consists of a series of operations performed *once* for a fixed and already trained model. These steps construct the layer-wise low-rank bases and statistical descriptors required during inference. Since they are executed offline, they do not affect runtime latency.

(1) Singular Value Decomposition of each affine layer. For every selected layer ℓ , the augmented operator $\mathbf{A}_\ell \in \mathbb{R}^{m_\ell \times (n_\ell + 1)}$ (Eq. 2.17) is factorised via SVD. The theoretical worst-case cost is

$$\mathcal{O}\left(\min\{m_\ell n_\ell^2, n_\ell m_\ell^2\}\right),$$

matching the standard complexity of dense matrix decompositions. Convolutional layers with large channel banks incur higher one-time cost, while transformer layers remain substantially cheaper.

(2) Projection of training activations. Given the truncated basis of dimension κ_ℓ , each activation is projected as

$$\mathbf{v}_\ell = \mathbf{Q}_\ell^{(\kappa_\ell)\top} \bar{\mathbf{x}},$$

requiring a matrix–vector multiplication of complexity

$$\mathcal{O}(\kappa_\ell n_\ell)$$

per sample.

(3) Clustering of corevectors. Corevector distributions are modelled via unsupervised clustering with $N_{C,\ell}$ components in dimension κ_ℓ . For N_ℓ samples, the cost scales

as

$$\mathcal{O}(N_\ell N_{C,\ell} \kappa_\ell),$$

which typically dominates the offline stage.

(4) Prototype construction. Layerwise statistical descriptors (centroids, empirical frequencies, etc.) are computed through simple aggregation operations. Their cost is negligible compared to SVD and clustering.

The offline stage is executed once per model and requires neither retraining nor fine-tuning. The computational burden is therefore amortised across the entire deployment lifetime.

7.8.2 Online Computational Requirements

At inference time, the structural score reduces to lightweight algebraic operations applied to intermediate activations. No backpropagation, iterative optimisation, or gradient computation is required.

(1) Corevector extraction. Each activation $\mathbf{h}_\ell$ is projected onto the truncated basis:

$$\mathbf{v}_\ell(\mathbf{x}) = \mathbf{Q}_\ell^{(\kappa_\ell)\top} \bar{\mathbf{h}}_\ell(\mathbf{x}),$$

with cost

$$\mathcal{O}(\kappa_\ell n_\ell)$$

per layer.

(2) Distance evaluation. Distances between $\mathbf{v}_\ell$ and the $N_{C,\ell}$ centroids are computed:

$$\mathcal{O}(N_{C,\ell} \kappa_\ell),$$

followed by a softmax normalisation of linear complexity.

(3) Layer-wise similarity aggregation. The similarity score

$$s_\ell(\mathbf{x}) = \cos\left(\mathbf{p}_\ell(\mathbf{x}), \bar{\mathbf{G}}_{\hat{\ell}(\text{net}),\ell}\right)$$

requires dot products and norms of cost $\mathcal{O}(N_{C,\ell})$. Final aggregation across L layers is negligible.

Overall inference-time complexity scales as

$$\mathcal{O}\left(\sum_{\ell=0}^{L-1} [\kappa_{\ell} n_{\ell} + N_{C,\ell} \kappa_{\ell}]\right),$$

dominated by matrix–vector multiplications and linear-time operations.

The method therefore remains independent of backpropagation and scales sublinearly with model size.

Table 7.5. Average computation time overhead per sample (seconds). Timings taken on an NVIDIA H100 80 GB GPU.

Method	VGG16	ViT-B16
OURS	$(2.6 \pm 0.3) \times 10^{-3}$	$(1.5 \pm 0.5) \times 10^{-3}$
DOC	$(1.3 \pm 1.7) \times 10^{-7}$	$(1.3 \pm 1.8) \times 10^{-7}$
REL-U	$(3.7 \pm 2.6) \times 10^{-6}$	$(3.6 \pm 2.8) \times 10^{-6}$
FS	$(2.7 \pm 0.2) \times 10^{-1}$	$(2.8 \pm 0.1) \times 10^{-1}$
MD	$(2.3 \pm 0.1) \times 10^{-2}$	$(7.5 \pm 0.1) \times 10^{-2}$

7.8.3 Comparison with Baselines

Baseline methods exhibit markedly different computational profiles:

- **MSP and DOC** operate directly on logits and introduce virtually zero inference overhead. However, they rely exclusively on output-layer statistics and provide no internal diagnostic information.
- **REL-U** requires a single large matrix multiplication at inference time. While inexpensive, it remains confined to output-level transformations.
- **FS** applies repeated input transformations (e.g., non-local filtering), increasing both runtime and memory bandwidth requirements by orders of magnitude.
- **MD-A / MD-U** rely on class-conditional Gaussian modelling of deep features and require gradient computations during inference, significantly increasing overhead and memory footprint.

Compared to these alternatives, $s(\mathbf{x})$ occupies an intermediate regime: it is moderately more expensive than purely output-based scores, yet orders of magnitude lighter than gradient-based or transformation-based detectors.

Crucially, it provides a multi-layer structural diagnostic signal while keeping the computational cost compatible with deployment.

7.8.4 Architectural Considerations

Complexity depends primarily on activation dimensionality:

- **VGG16**: large spatial activation tensors (e.g., $512 \times 7 \times 7$) increase projection cost.
- **ViT-B16**: encoder outputs are lower-dimensional vectors (e.g., 768), reducing projection overhead despite increased depth.

This explains why transformer models exhibit slightly lower online overhead in Table 7.5.

7.8.5 Final Remark on Efficiency

The structural score provides internal diagnostic richness without requiring gradient computation, retraining, or heavy input transformations.

Its offline cost is amortised over deployment, while its online footprint remains compatible with real-time inference, including embedded or onboard scenarios where model size is typically smaller than the benchmark networks considered here.

It is worth noting that the experimental configuration adopted here intentionally reflects a conservative scenario. In convolutional architectures, full filter banks were unrolled and instrumented, leading to large projection operators; similarly, multiple transformer layers were included to stress-test depth-wise coverage.

These choices provide an upper-bound estimate of computational footprint. In practice, selective instrumentation, for instance, restricting the analysis to representative convolutional filters or a subset of layers, can substantially reduce projection dimensionality and memory requirements without compromising structural sensitivity.

Even under these deliberately demanding settings, the overhead remains moderate, confirming the intrinsic scalability of the approach.

The resulting balance between interpretability, robustness, and computational economy makes the method suitable not only for post-hoc analysis but also for continuous monitoring within AI systems.

Concluding Remarks

The quantitative analyses presented in this chapter extend the sample-level observations of Sec. 7.3 to dataset-scale evaluation. Rather than focusing on isolated examples,

we examined how the proposed structural score behaves across complete test sets, corruption regimes, and adversarial perturbations.

The results do not indicate universal dominance over all baselines in every scenario. In particular, supervised Mahalanobis variants achieve state-of-the-art performance in certain far-OOD and adversarial settings, and FS remains competitive under specific attacks. However, three consistent patterns emerge.

First, the structural score maintains competitive performance in in-distribution confidence estimation while relying exclusively on internal agreement across layers rather than on output probabilities. This confirms that internal coherence alone carries meaningful information about prediction reliability.

Second, under distribution shift, the score reflects the progressive disruption of the learned reasoning manifold. As corruption severity increases, the separation between clean and perturbed samples becomes more pronounced, indicating that structural deviations accumulate coherently across depth. Unlike purely output-based measures, the score remains informative even when logits exhibit saturation effects.

Third, adversarial perturbations frequently induce depth-wise inconsistencies that are detectable without supervision or gradient-based optimisation. Although feature-space modelling may outperform it in certain regimes, the structural signal consistently captures coordinated deviations across layers.

Taken together, these findings suggest that layer-wise structural coherence constitutes a distinct and complementary reliability dimension. It does not replace output-based or density-based detectors; rather, it provides an internal diagnostic signal grounded in the geometry of the model’s learned representations.

In this sense, the proposed framework reframes confidence estimation not as a property of the final output distribution alone, but as a measure of trajectory consistency across depth. Trustworthiness, therefore, emerges from agreement between what the model predicts and how coherently it reaches that prediction.

Future work may investigate principled fusion strategies combining structural coherence with output-based and density-based estimators, potentially yielding composite confidence mechanisms that leverage complementary reliability signals.

8

Applications in Constrained and Safety-Critical Domains: Autonomy in Space

Introduction

Modern spacecraft operate in environments that are simultaneously remote, resource-constrained, and safety-critical. Unlike terrestrial platforms, they must preserve health and stability over years of autonomous functioning under strict power and memory budgets, exposure to radiation, and limited communication windows. For this reason, contemporary avionics rely on layers of *Fault Detection, Isolation and Recovery* (FDIR) logic: deterministic, rule-based mechanisms designed to detect anomalies and trigger protective actions with minimal ground intervention.

Classical FDIR is deliberately conservative. It relies on Out-of-Limits (OOL) checks applied independently to individual telemetry channels. This simplicity supports transparency and verifiability, but it also limits sensitivity to correlated deviations, temporal signatures, and slow drifts that often precede faults in real spacecraft dynamics. As missions grow in complexity and autonomy requirements increase, these limitations increasingly constrain the achievable operational envelope.

In this chapter, we consider an *AI system* perspective: anomaly detection is not treated as a stand-alone model, but as a function embedded within an onboard chain that must satisfy determinism, traceability, bounded resource usage, and operational assurance. Accordingly, the focus is not only on detection performance, but on whether data-driven health monitoring can be made *deployable*, *auditable*, and *gated by runtime confidence* in a way that integrates with legacy safety logic.

The work presented here is enabled by access to real in-orbit telemetry from a safety-critical subsystem. Although the specific subsystem remains confidential, the data reflect genuine operational regimes, environmental perturbations and sensor behaviours encountered in flight, allowing a realistic assessment of Machine Learning (ML)-based health monitoring under authentic operating conditions.

The chapter proceeds in three stages. First, we assess whether nominal behaviour

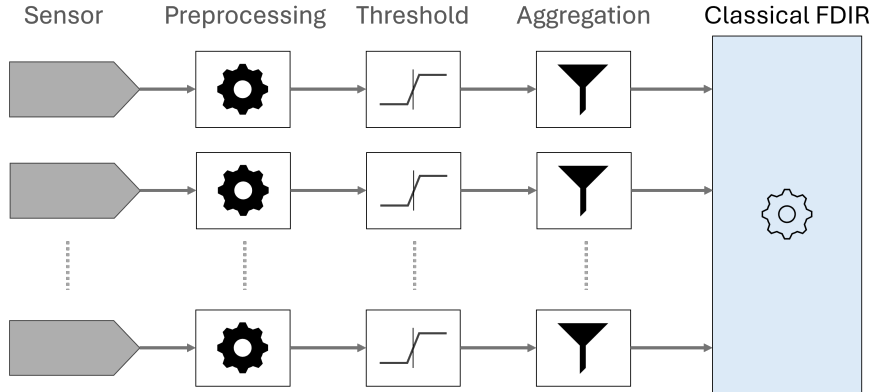


Figure 8.1. Classical PUS-based FDIR pipeline using thresholds and Boolean logic. Each signal is treated independently without global confidence awareness.

can be learned from multivariate telemetry and deviations detected reliably (Section 8.1). Second, we demonstrate that such detectors can be executed on space-qualified hardware and integrated into an existing FDIR chain (Section 8.2). Finally, we show how runtime confidence estimation and peephole-based structural interpretability extend naturally to telemetry-based autonomy, enabling conservative decision gating and actionable diagnostics (Sections 8.3 and 8.4).

The objective is not to replace classical FDIR, but to augment it: confidence-aware detection and compact structural evidence provide an operational bridge between learned normality and certifiable onboard autonomy.

8.1 From Thresholds to Learned Normality

Classical FDIR relies on deterministic thresholding (Fig. 8.1), an approach that, while transparent and predictable, cannot capture correlations across telemetry channels nor the temporal structures characteristic of gradual degradation. The first phase of this work examined whether a data-driven representation of nominal behaviour could improve sensitivity while respecting the computational and architectural constraints of onboard avionics.

Two complementary data regimes were considered. The first consists of multivariate telemetry acquired in orbit from a critical spacecraft subsystem. The second comprises controlled anomaly scenarios generated using a dedicated fault-injection suite, involving deviations such as drifts, spikes, saturations and compound distortions. This combination enabled a robust evaluation that accounted both for realistic environmental variability and for stress conditions not easily observed during nominal operations.

Motivation. Introducing learned normality in spacecraft health monitoring immediately raises an assurance question: not only *what* the detector outputs, but *when* that output should be trusted enough to influence onboard decisions. This motivates the chapter’s two-step transition: first, we move from univariate OOL supervision to multivariate, window-based structural monitoring; then, we attach lightweight runtime confidence and interpretable internal evidence to the detector, so that alarms can be gated and diagnostics can remain auditable within a safety-critical FDIR chain.

8.1.1 Design Constraints for Onboard AI Systems

Onboard health monitoring differs from terrestrial deployments because the surrounding *AI system* is tightly constrained by certification-driven engineering practices. In particular, any learning-based component must satisfy: (i) bounded execution time and memory footprint; (ii) deterministic scheduling and isolation within the avionics software architecture; (iii) robustness to non-idealities (including radiation-induced effects and operational regime changes); and (iv) traceability of decisions through compact logs and engineering-interpretable evidence. These constraints motivate the chapter’s emphasis on *runtime confidence* and *structural interpretability*: rather than aiming for richer explanations, the goal is to make the detector’s behaviour *observable and gateable* under flight constraints, while remaining compatible with legacy FDIR logic.

8.1.2 From Threshold Monitoring to Structural Assessment in Space Systems

Telemetry monitoring in spacecraft operations traditionally relies on Out-of-Limit (OOL) checks. Each telemetry channel is evaluated independently against predefined high and low bounds. If the sampled value lies within the admissible interval, the subsystem is declared nominal.

OOL monitoring operates on discretely sampled measurements $x_i[t] = x_i(t_{s_k})$, evaluated at inspection instants t_{s_k} . For illustrative purposes, Fig. 8.2 depicts continuous-time signal trajectories, while decisions are based solely on their sampled values.

Fig. 8.2 highlights a structural limitation of threshold-based supervision. Telemetry is evaluated independently and at discrete inspection times. If sampled values lie within predefined thresholds, the subsystem is declared nominal. This logic implicitly assumes that anomalies manifest as instantaneous limit violations.

However, degradation processes, offset drifts, transient excursions, or actuator saturations may develop between inspection instants while remaining undetected if the sampled values happen to fall within operational bounds. Consequently, a subsystem

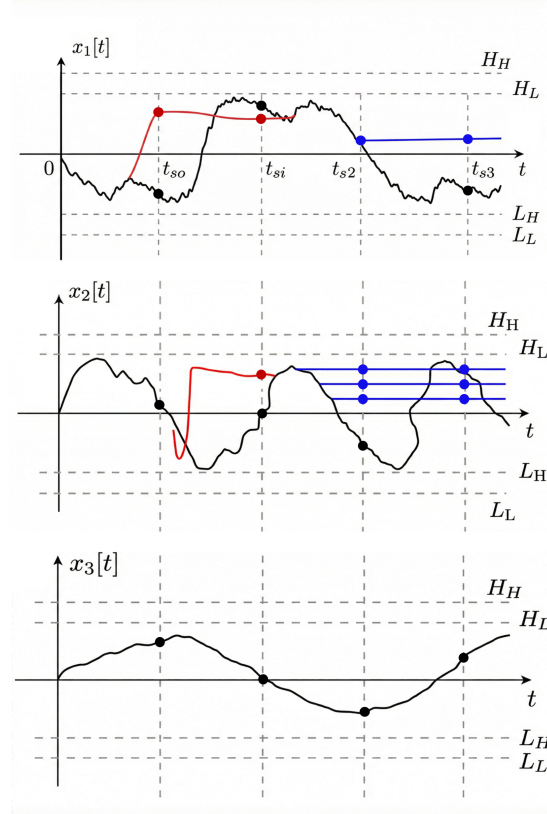


Figure 8.2. Examples of telemetry signals $(x_1[t], x_2[t], x_3[t])$ representative of sensor readings within a spacecraft subsystem. Black curves depict nominal behaviour, while red segments illustrate an offset/step-type deviation and blue segments a saturation effect. The vertical dashed lines denote discrete sampling instants $t_{s0}, t_{s1}, t_{s2}, t_{s3}$. Colored markers indicate the hypothetical sampled values under anomalous behaviour. At these inspection times, all measurements remain within the predefined operational bounds (outer limits H_H, L_L and inner limits H_L, L_H), hence no Out-of-Limit (OOL) alarm would be triggered despite the ongoing structural deviation.

may remain formally “within limits” while progressively departing from its nominal structural behaviour.

An additional complexity arises from the heterogeneous dynamics of telemetry signals. Different channels may evolve at distinct temporal scales, exhibit different bandwidths, or respond differently to perturbations. As a result, anomaly detectability becomes jointly determined by signal dynamics and sampling policy, rather than by threshold values alone. When the characteristic time constants of the underlying physical processes are comparable to or smaller than the inspection interval, relevant structural deviations may remain unobserved in the sampled sequence.

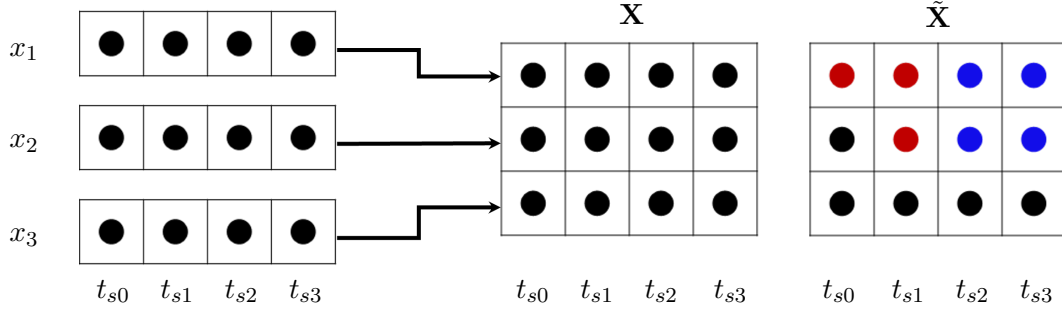


Figure 8.3. High-level schematic of windowed multichannel telemetry aggregation. Three telemetry channels (x_1, x_2, x_3) are sampled at discrete instants $\{t_{s0}, t_{s1}, t_{s2}, t_{s3}\}$ and arranged into a multivariate window $\mathbf{X}$. Off-nominal phenomena may affect one or multiple channels, producing a perturbed window $\tilde{\mathbf{X}}$. The red and blue samples in $\tilde{\mathbf{X}}$ qualitatively recall the offset/step and saturation behaviours illustrated in Fig. 8.2, emphasizing how localized anomalies translate into structured multivariate deviations at the window level.

In terrestrial systems, increasing the sampling rate or transmitting complete time series may partially mitigate this limitation. In space systems, however, telemetry bandwidth, onboard storage, computational resources, and downlink constraints impose strict limits on data transmission and inspection frequency. Such constraints become increasingly critical as the number of spacecraft, subsystem complexity, and degrees of onboard autonomy continue to grow.

These limitations are not restricted to ground-based monitoring. Within onboard FDIR architectures, threshold logic remains widely adopted due to its simplicity and determinism. Yet, as mission profiles evolve toward higher autonomy and larger constellations, reliance on purely univariate limit checks becomes progressively insufficient.

The limitation highlighted in Fig. 8.2 is therefore not merely temporal but structural. A subsystem is characterized by the joint evolution of its telemetry channels, not by isolated threshold checks. This observation motivates a transition from univariate limit supervision toward multivariate structural assessment, where subsystem consistency is evaluated in the joint state space of its variables.

8.1.3 Multivariate Structural Representation of Subsystem Behaviour

Following the sampling perspective introduced in the previous section, telemetry streams are here reformulated in terms of finite multivariate windows. Let $\mathbf{X} \in \mathbb{R}^{T \times d}$ denote a window of length T collecting d telemetry channels over a discrete inspection horizon. Rather than reasoning on individual samples in isolation, the subsystem state is assessed

through the joint structure of the window.

This representation makes explicit the transition from univariate threshold supervision to multivariate structural monitoring. While individual measurements may remain locally plausible, their joint arrangement within $\mathbf{X}$ can reveal coherent deviations that are not captured by independent out-of-limit checks. In this sense, the window becomes the elementary object of analysis for subsequent anomaly detection mechanisms.

A formal definition of the adopted windowing procedure, including indexing conventions and implementation details, is provided in [Ciancarelli et al. \(2023, 2022\)](#).

8.1.4 Model Families for Telemetry-Based Anomaly Detection

The investigation considered three methodological families.

Compression-based models such as Principal Component Analysis (PCA) [Jolliffe \(2002\)](#) and convolutional Autoencoders (AEs) [Hinton & Salakhutdinov \(2006\)](#) learn compact representations of nominal telemetry. Deviations from the learned manifold naturally translate into increased reconstruction error and thus into anomaly indications. These models are attractive for onboard use due to their predictable computational footprint and stable behaviour across noise regimes.

Prediction-based models, including autoregressive predictors (AR) [Box et al. \(2015\)](#) and Long Short-Term Memory (LSTM) networks [Hochreiter & Schmidhuber \(1997\)](#), capture the temporal structure of subsystem dynamics. An anomaly manifests as a mismatch between the predicted and observed behaviour and can therefore be detected through prediction residuals.

Reference models such as the One-Class SVM [Schölkopf et al. \(2001\)](#) and Local Outlier Factor (LOF) [Breunig et al. \(2000\)](#) treat anomalies as geometrically isolated or low-density points in feature space. These approaches provide a non-parametric baseline independent of architectural assumptions and serve as a useful comparison against more structured detectors.

All models were implemented in a unified and avionics-compatible C library supporting windowing, preprocessing, scoring and thresholding, and were used consistently across the studies reported in [Borgia et al. \(2025\)](#); [Ciancarelli et al. \(2023\)](#); [Manovi et al. \(2024c\)](#); [Leboffe et al. \(2024\)](#).

8.1.5 Empirical Findings Across Telemetry Scenarios

A coherent picture emerges across both synthetic and real telemetry. PCA consistently provides a robust linear baseline, outperforming classical AR models and offering a favourable robustness–complexity trade-off. Among neural architectures, convolutional

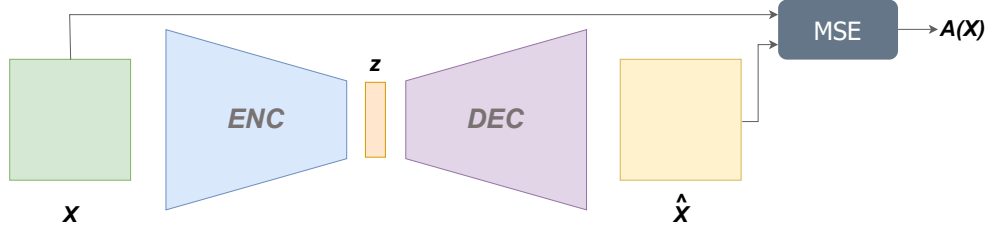


Figure 8.4. High-level architecture of an Autoencoder (AE)-based anomaly detector. Input sample $\mathbf{X}$ is compressed by the encoder (ENC) into a latent code $\mathbf{z}$, then reconstructed by the decoder (DEC) into $\hat{\mathbf{X}}$. The Mean Squared Error (MSE) between $\mathbf{X}$ and $\hat{\mathbf{X}}$ defines the base anomaly score $A(\mathbf{X})$.

AEs show the most stable performance, with higher sensitivity to subtle deviations and strong scalability with respect to window length, as discussed in [Manovi et al. \(2024c\)](#).

A central insight concerns the impact of the training objective. The recent study [Borgia et al. \(2025\)](#) compared two AE variants of equivalent architecture: one trained to minimise reconstruction error (AE-RECON) and one trained to maximise anomaly discrimination (AE-DETECT). The latter provided markedly superior separability, even though computational cost remained identical. This finding aligns with the information-theoretic characterisation of the reconstruction-detection trade-off presented by [Enttsel et al. \(2024a\)](#), which shows that representational capacity is allocated differently depending on the objective and that detection-oriented optimisation yields improved sensitivity under tight resource constraints.

Performance improved with longer temporal windows across all model families, reflecting the inherently dynamic and correlated structure of spacecraft telemetry. Taken together, these observations support the selection of detection-oriented convolutional AEs as the leading candidate for ML-enhanced FDIR, with PCA remaining a strong and lightweight alternative.

8.1.6 Autoencoders as Structural Models of Nominal Behaviour

Since AEs emerged as the most effective detectors among the candidate families, it is useful to formalise their structure. A high-level schematic depicting the general architecture of an AE designed for anomaly detection is provided in Fig. 8.4.

Given a telemetry window $\mathbf{X} \in \mathbb{R}^{C \times T}$, an AE consists of a parametrised encoder–

decoder pair $(\text{ENC}_\theta, \text{DEC}_\phi)$:

$$\mathbf{z} = \text{ENC}_\theta(\mathbf{X}) \in \mathbb{R}^k, \quad (8.1)$$

$$\hat{\mathbf{X}} = \text{DEC}_\phi(\mathbf{z}), \quad (8.2)$$

where $k \ll CT$ controls the dimensionality of the latent representation. The decoding function leading to the reconstructed version of the input mirrors the operations performed by the encoding stage, which produces the latent representation of that same input. Training minimises the reconstruction loss over a corpus of nominal windows $\mathcal{D}_{\text{nom}}$:

$$\min_{\theta, \phi} \mathbb{E}_{\mathbf{X} \sim \mathcal{D}_{\text{nom}}} \left[\frac{1}{CT} \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 \right]. \quad (8.3)$$

At inference time, the anomaly score is the normalised reconstruction error:

$$A(\mathbf{X}) = \frac{1}{CT} \|\mathbf{X} - \text{DEC}_\phi(\text{ENC}_\theta(\mathbf{X}))\|_F^2. \quad (8.4)$$

This formulation introduces two structural elements that will play a central role in the subsequent sections: the latent representation $\mathbf{z}$ and the activations $\mathbf{h}_\ell$ which are produced within the encoding and decoding stages. In addition to dense layers, the blocks of encoder ENC_θ and decoder DEC_ϕ are typically constituted by convolutional and transposed convolutional layers, respectively. Both will be used for runtime confidence estimation and for the peephole-based structural interpretation developed later in this chapter (Sections 8.3–8.4).

8.2 Deployability on Space-Qualified Hardware

Ensuring that ML-based detectors are not only effective but deployable under real spacecraft constraints was a fundamental objective of this work. The unified anomaly-detection library was compiled and executed on a GR740 board, the reference radiation-tolerant processor used in many European missions. The evaluation reported in [Leboffe et al. \(2024\)](#) shows that both PCA and convolutional AEs meet real-time execution budgets for representative window lengths and require memory footprints that remain comfortably within platform limits. Inference latency scales predictably with model dimensions, enabling integration into deterministic scheduling frameworks.

In parallel, the EDHPC work [Borgia et al. \(2025\)](#) detailed an end-to-end architecture connecting AE-based detection to preprocessing, scoring and PUS-based reporting, confirming architectural compatibility with existing FDIR chains. These results demon-

strate that multivariate ML-based health monitoring is not only viable at algorithmic level but *practically feasible* on current space-qualified hardware, thereby motivating the introduction of runtime confidence and structural interpretability as essential enablers for trustworthy onboard autonomy.

8.3 Toward Trustworthy Autonomy: Runtime Confidence

Demonstrating that ML-based detectors can run on-board is necessary but not sufficient for autonomy. In a safety-critical setting such as spacecraft avionics, correct detection alone does not guarantee safe operation: the system must also be able to assess the *credibility* of its own outputs. Classical FDIR embeds confidence implicitly through deterministic thresholds and fixed margins, whereas data-driven detectors do not provide such guarantees by construction. To be integrated into the autonomous decision loop, an ML module must therefore expose a lightweight and reliable notion of *runtime confidence*.

This requirement motivated the design of the confidence-estimation framework presented in [Manovi et al. \(2025\)](#). The framework is intentionally model-agnostic: it assumes a pre-trained detector producing an anomaly score $A(\mathbf{X})$ and attaches to it a complementary confidence score $R(\mathbf{X})$ without retraining and with negligible computational overhead. These two signals are combined through a dual-threshold decision rule

$$\text{RaiseAlarm}(\mathbf{X}) \iff (A(\mathbf{X}) > \tau_A) \wedge (R(\mathbf{X}) > \tau_R), \quad (8.5)$$

which plays the same role as classical FDIR logic: an alarm is issued only when the detector both identifies a deviation and declares itself reliable.

In practice, the dual-threshold logic can be an indicator of diverse scenarios, which are detailed in Sec. 8.3.1 below. Additionally, Sections 8.3.2-8.3.4 discuss possible implementations of the confidence score, differing in terms of cost and level of provided insight, when the detection logic is realised through an AE model (Sec. 8.1.6).

8.3.1 Boolean Abstraction and Operational Logic

For system-level reasoning, it is convenient to abstract the anomaly and confidence scores to Boolean indicators. Given $A(\mathbf{X})$ and $R(\mathbf{X})$ and their thresholds τ_A and τ_R ,

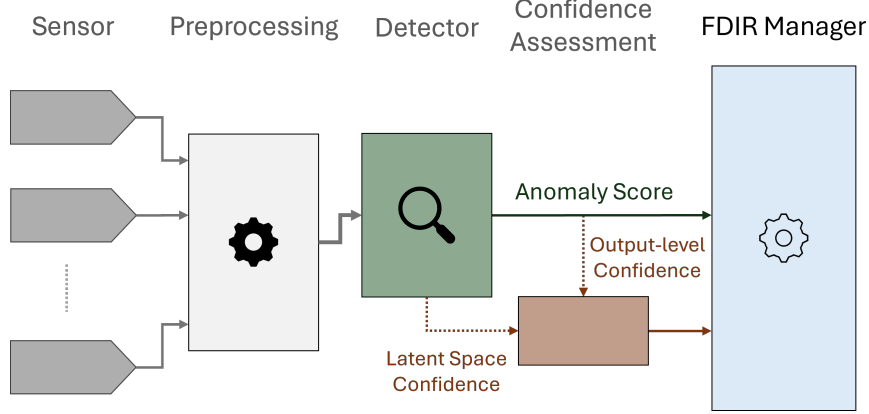


Figure 8.5. AI-augmented FDIR architecture. A data-driven detector computes an anomaly score $A(\mathbf{X})$, and a parallel module estimates a confidence score $R(\mathbf{X})$ to gate critical decisions.

we define

$$a(\mathbf{X}) = \mathbb{I}(A(\mathbf{X}) > \tau_A), \quad (8.6)$$

$$r(\mathbf{X}) = \mathbb{I}(R(\mathbf{X}) > \tau_R), \quad (8.7)$$

with $a, r \in \{0, 1\}$. The pair (a, r) captures four qualitatively different operational states, summarised in Table 8.1.

Table 8.1. Logical interpretation of anomaly and confidence indicators.

a	r	Operational interpretation
0	1	Nominal and reliable
1	1	Anomalous and reliable
0	0	Nominal but unreliable
1	0	Anomalous but unreliable

The regime $a = 0, r = 1$ indicates to a *nominal and reliable* state: the detector observes no deviation and its internal trajectory remains within familiar regions of the feature space. Operations can continue without interruption; only standard housekeeping telemetry needs to be logged.

The combination $a = 1, r = 1$ reflects an *anomalous and reliable* state. Here the detector identifies a deviation *and* has high confidence in its judgement. This is precisely the regime authorised by (8.5): fault reports can be raised autonomously and local diagnostics can be initiated. If configured, isolation or recovery sequences can also be triggered. As both scores are useful for diagnosis, the relevant window and the

associated peephole-based summaries should be logged to speed up root-cause analysis.

The case $a = 0, r = 0$ corresponds to state in which operation appears nominal but the devised confidence scoring mechanism is not able to certify that the outcome of detection is reliable. The detector does not flag an anomaly, yet its reasoning trajectory diverges from the nominal manifold. This may arise under distribution shift or atypical operating modes. In this situation, the absence of an alarm should not be taken as evidence of health: a prudent strategy is to maintain operation but escalate logging, raise a soft warning, or reduce the autonomy level (e.g., by tightening classical FDIR thresholds or increasing the frequency of ground checks). This state serves as an early indicator that the detector is operating outside its validated domain.

Finally, the case $a = 1, r = 0$ denotes an *anomalous but unreliable* outcome. The detector reacts to the input, but its internal trajectory suggests that the reasoning may not be trustworthy. This is the most delicate regime: a low-confidence alarm may indicate either a genuine but unusual fault or a spurious out-of-distribution event. A safe policy is to initiate diagnostics and detailed logging, but defer aggressive recovery actions. Control should be handed over to ground personnel as soon as possible, with the anomaly and confidence traces guiding the analysis. The peephole-derived summaries are particularly valuable in determining whether the model focused on physically meaningful patterns or artefacts unsupported by system knowledge. We will elaborate on this last point in 8.4.

The Boolean abstraction highlights that the dual-threshold rule defines only one of the four operational states in which autonomy is authorised. The remaining three states inform how trust is allocated, how strongly the system reacts, and how telemetry is prioritised for storage or downlink. This truth-table perspective therefore clarifies that anomaly detection and confidence estimation together form a *structured decision mechanism* rather than a single-threshold classifier.

Building on the concepts presented in [Manovi et al. \(2025\)](#), we will now explore potential approaches to defining a confidence score in this scenario.

8.3.2 Output-Level Confidence

Output-level confidence relies solely on the anomaly score and requires no access to intermediate features. Let $\{A_i\}$ denote the distribution of reconstruction errors (8.4) on a nominal validation set. An empirical confidence estimate is

$$R_o(\mathbf{X}) = 1 - \frac{|\{i : A_i \leq A(\mathbf{X})\}|}{N}, \quad (8.8)$$

which measures how atypical $A(\mathbf{X})$ is with respect to nominal behaviour. A Gaussian approximation may be used when the tail behaviour is well modelled. Although extremely lightweight, output-level confidence cannot detect inconsistencies in the internal reasoning process and therefore offers limited assurance in safety-critical contexts.

8.3.3 Latent-Space Confidence

A perhaps more informative viewpoint can be obtained from the latent representation $\mathbf{z} = \text{ENC}_\theta(\mathbf{X}) \in \mathbb{R}^k$. If the AE has learned a coherent nominal manifold, nominal windows cluster in a distribution characterized by mean and covariance $(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$, while anomalies are displaced. The Mahalanobis distance

$$d_M(\mathbf{z}) = (\mathbf{z} - \boldsymbol{\mu}_z)^\top \boldsymbol{\Sigma}_z^{-1} (\mathbf{z} - \boldsymbol{\mu}_z), \quad (8.9)$$

yields the confidence

$$R_z(\mathbf{X}) = 1 - F_{\chi_k^2}(d_M(\mathbf{z})), \quad (8.10)$$

with $F_{\chi_k^2}$ is the chi-squared cumulative distribution function with k degrees of freedom (Lee et al. (2018a)). Care must be taken to ensure that $\boldsymbol{\Sigma}_z$ is invertible and numerically stable onboard.

A Gaussian mixture alternative may also be adopted:

$$R_z^{\text{GMM}}(X) = \max_{j \leq K} \pi_j \mathcal{N}(\mathbf{z}; \mu_j, \boldsymbol{\Sigma}_j). \quad (8.11)$$

These latent-space measures are more sensitive to out-of-distribution behaviour than output-level ones and remain computationally affordable.

8.3.4 Deep Feature-Space Confidence via Peepholes

As argued extensively throughout the previous chapters of the manuscript, we claim that the richest notion of confidence arises from the *internal* activations of the network under analysis. Following the notation already introduced, for each hidden layer ℓ with activation $\mathbf{h}_\ell$, we form the augmented vector $\bar{\mathbf{h}}_\ell = [\mathbf{h}_\ell^\top \ 1]^\top$ and the corresponding augmented weight matrix $\mathbf{A}_\ell = [\mathbf{W}_\ell \ \mathbf{b}_\ell]$. A truncated singular value decomposition yields a set of dominant right singular vectors forming a peephole matrix $\mathbf{V}_{\ell,r}$. The reduced activation

$$\tilde{\mathbf{h}}_{\ell+1} = \mathbf{V}_{\ell,r}^\top \bar{\mathbf{h}}_\ell \quad (8.12)$$

captures the structural information flow at layer ℓ in a compact form. Applying the latent-space confidence machinery (Eq. 8.9, 8.10) to corevector $\tilde{\mathbf{h}}_{\ell+1}$ yields the deep feature-space confidence denoted as $R_h(\mathbf{X})$. Provided that the optimal dimension of this corevector is small enough to not introduce significant computational overhead, this score reflects whether the internal reasoning trajectory is consistent with nominal behaviour, resulting in both a robust confidence estimate and an interpretable structural signature at negligible computational cost.

It is worth noting that, given that the target models for this application will be limited in parameter size and will hardly approach the dimensions encountered in the results presented in the previous chapter, the assumption of very low dimensions of the corevectors or peepholes that can be extracted in this use case is reasonable. Table 8.2 provides an estimation of storage and computational cost (in terms of MAC operations) required to implement the mentioned confidence scores.

Table 8.2. Compute and storage cost summary for confidence scores.

Score	Stored params	Runtime cost
R_o	(μ_A, σ_A) or LUT	$O(1)$ or $O(\log N)$ lookup
$R_z^{\chi^2}$	$\mu_z \in \mathbb{R}^k, \Sigma_z^{-1} \in \mathbb{R}^{k \times k}$	$O(k^2)$
R_z^{GMM}	$\{\pi_j, \mu_j, \Sigma_j\}_{j=1}^K$	$O(K k^2)$
R_h	$\mathbf{V}_{\ell,r} \in \mathbb{R}^{(n+1) \times r}, (\Sigma_r)^{-1}$	$O(nr + r^2)$

Even though a score of the type 8.10/8.11 is considered robust for assessing model reliability when computed at intermediate internal processing steps, the focus naturally turns to investigating whether the framework described in Chap. 4 can be applied in the context of spacecraft autonomy. In the following section, we elaborate on the interpretability of the results obtained by adapting the approach involving semantic peepholes, and we present the preliminary outcomes obtained using real mission data, as discussed in [Capelli et al. \(2025\)](#).

8.4 Interpretability for Telemetry Diagnostics via Peepholes

In safety-critical avionics, confidence estimation alone may be insufficient: understanding the *reasoning path* that led the model to raise or suppress an alarm is often as important as the alarm itself. Operators must determine why a deviation was detected, how the model interpreted the multi-channel input, and which subsystem features contributed to the decision. Classical FDIR addresses these requirements only partially. While

its threshold-based rules offer transparent triggers and basic onboard routines support limited isolation, they do not articulate how a deviation developed nor how correlated channels co-evolved to cause a limit violation. Detailed identification is frequently deferred to ground, increasing latency and prolonging subsystem downtime.

The peephole framework introduced in Chap. 4 is designed to complement this process by providing structured, layer-wise evidence that *accelerates root-cause analysis* and makes the detector’s internal processing *auditable* in-flight.

Earlier in this chapter we showed that single-channel OOL logic has failed to detect the real anomaly in time, whereas joint monitoring of multiple channels of a homogeneous sub-assembly (here anonymised) exposed coordinated deviations earlier. This motivates the use of multivariate detectors. The interpretability mechanisms below ensure that their internal evidence is equally intelligible.

8.4.1 Experimental Setup and Peephole Construction

The experiments follow the semantic-peephole approach introduced in [Capelli et al. \(2025\)](#). Fig. 8.6 illustrates the functional blocks constituting the considered AE, where the peephole is extracted from a dense layer following the convolutional decoder.

The detector is a compact convolutional AE with a latent dimension of $k = 16$ and a final dense layer of dimension $d = 20$. The peephole is obtained by applying the SVD-based projection of Eq. 3.4 to this dense layer, keeping $\kappa = 6$ dominant directions. This results in a reduced representation $\tilde{\mathbf{h}} \in \mathbb{R}^\kappa$ that captures the principal structural components of the layer’s transformation.

To endow this reduced vector with semantic structure, a Gaussian Mixture Model (GMM) with $K = 5$ components is fitted over the collection of training-set corevectors, in line with Eq. 8.11. Here, rather than simply evaluating whether the activation projected onto the low-dimensional space of interest is equivalent to a sample drawn from a nominal distribution, we examine the inner workings of the neural detector. The aim is to identify not only abnormal behaviour, but also to predict whether it is associated with any anticipated anomalies. More specifically, we corrupt the validation dataset with synthetic anomaly families produced with the Wombat fault-injection suite [Enttsel et al. \(2024b\)](#), a set of tools designed to assess detection performance in time-series data. We build ad hoc datasets that balance the presence of normal instances with samples manifesting the effect of the presence of offset, step, impulse, White Gaussian Noise (WGN) and Principal Subspace Alteration (PSA) anomalies.

In line with the principles devised in Sec. 4.1.3, these anomaly types act as empirical semantic tags: the peephole posterior (soft cluster membership) provides an interpretable

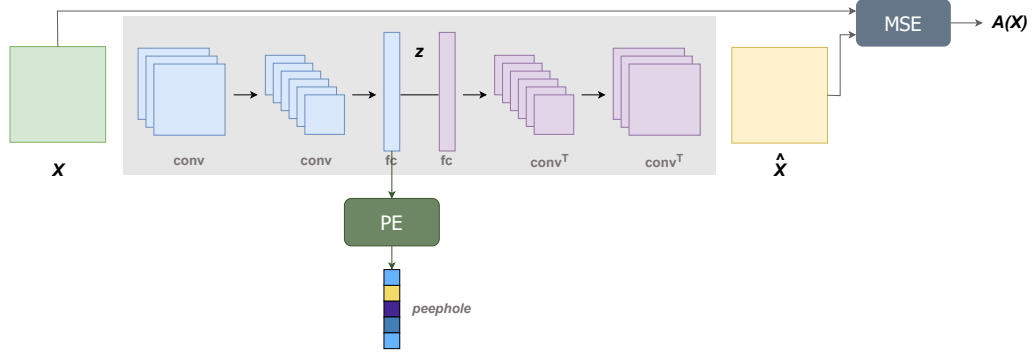


Figure 8.6. Schematic representation of the convolutional autoencoder and the subsequent peephole extraction (PE) applied to its final dense layer. The PE block corresponds to the reduced activation produced via the SVD-based projection of Eq. (3.4).

score indicating how closely the current activation resembles each anomaly family.

The AE-based detector was trained purely for reconstruction on nominal data. When evaluated on validation windows corrupted with Wombat-generated anomalies, it achieved approximately perfect discrimination in terms of AUC (Eq. 5.1) [Capelli et al. \(2025\)](#), ensuring that the semantic patterns extracted downstream are grounded on a detector with reliable separation capability.

8.4.2 Semantic Interpretation of Peepholes

The GMM posterior associated with $\tilde{\mathbf{h}}$ acts as an empirical high-level mapping: a soft categorical distribution over the anomaly families that best match the reduced activation. Because clusters are formed directly from the AE corevectors, this mapping is not imposed artificially but emerges from the geometry of the layer. While system monitoring is performed, it is not only the anomaly score $A(\mathbf{X})$ (8.4) that is observed. At each time interval, internal evidence of the detector’s reasoning is also inspected using semantic peepholes. This allows to build a graphical time evolution of the network’s internal state. A crucial property is that, under nominal conditions, the reduced activations lie near the center of the corevector distribution. Thus, all GMM components receive nearly equal, low-intensity scores, producing an “uniform dark” region in the produced time evolution heatmaps: this implies that the model does not align with any anomaly-related semantic structure.

In [Capelli et al. \(2025\)](#) we present an example of evolution of peephole evidence during a real anomaly, where the multivariate input $\mathbf{X} \in \mathbb{R}^{16 \times 16}$ represent the joint

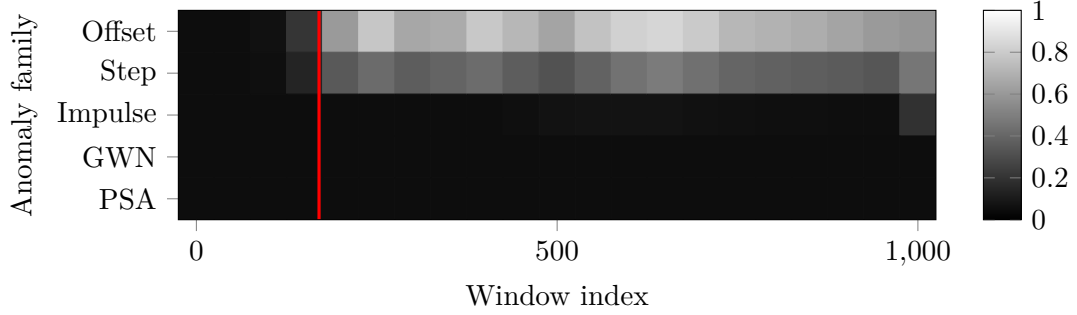


Figure 8.7. Stylised evolution of peephole-based semantic scores for a real anomaly case. Each column corresponds to one sliding window; each row to one anomaly family. At the onset (vertical red line), the semantic signature shifts toward offset and step anomalies, reflecting both the appearance and the incomplete recovery of the slow-varying displacement observed in telemetry.

monitoring of 16 telemetry channel over 16 time steps per sliding window. Fig. 8.7 shows a stylised evolution of the peephole vector $\mathbf{p}_\ell$ (one soft posterior per sliding window). Before the onset of the anomaly, the semantic scores are uniformly low across all families, as would be expected for nominal behaviour.

At the anomaly onset, the reduced activation shifts sharply into regions associated with slow-varying deviations. In this case, the physical event exhibited both a baseline shift and a gradual settling, behaviours that overlap with the offset and step anomaly families. Consistently, the corresponding rows of the peephole panel increase simultaneously, indicating that the model internally interprets the deviation as belonging to the two most similar anomaly shapes in the semantic space.

This evolution provides a compact, time-resolved account of *how the detector’s internal evidence evolves*, complementing the scalar anomaly score with an interpretable description of the anomaly type.

In the controlled setting presented by [Capelli et al. \(2025\)](#), anomalies were injected on individual channels of the homogeneous assembly. Even though the peephole is extracted from a single dense layer, the posterior distribution shifts consistently depending on which channel was perturbed. This provides weak but *actionable localisation capability*: the layer captures sub-component signatures implicitly through its learned representation. The same effect was observed in the real anomaly, where coordinated deviations across multiple channels preceded the model’s semantic transition. This strengthens the assumption that a setup as the presented one has potential to improve not only the detection, but also the isolation and identification steps performed by the classical FDIR mechanism.

In summary, peepholes provide a lightweight, structurally grounded tool for interpreting the internal computation of anomaly detectors. The GMM-based semantic mapping derived from Wombat synthetic anomalies gives each reduced activation a meaningful empirical interpretation, enabling anomaly-type identification, weak localisation, and time-resolved reasoning traces. By complementing confidence estimation with interpretable evidence, this mechanism supports faster operator diagnostics and strengthens the case for dependable autonomy in constrained onboard environments.

8.5 Discussion

The space domain provides a uniquely demanding environment in which to test the principles developed in this thesis. Across the studies summarised in this chapter, a coherent set of conclusions emerges.

ML-based detectors, and in particular detection-oriented convolutional Autoencoders, can model nominal spacecraft behaviour and identify deviations under realistic telemetry variability. Importantly, this capability is not confined to offline experimentation: the same models can be executed on space-qualified processors such as the GR740, with predictable timing behaviour and memory footprints compatible with flight constraints.

Beyond detection performance, the introduction of runtime confidence estimation proves essential. A principled confidence score can be attached to a pre-trained detector without retraining and with negligible computational overhead, enabling conservative decision gating through a dual-threshold rule that integrates cleanly with existing FDIR logic. This shifts anomaly detection from a stand-alone classifier to a structured decision component within an onboard AI system.

Finally, peephole-based structural analysis complements confidence estimation by exposing compact, layer-wise evidence of the network’s internal reasoning. Rather than providing post-hoc explanations, this mechanism makes the detector’s internal dynamics observable, allowing cross-validation with engineering expectations and supporting verification-oriented workflows.

Taken together, these results do not claim that learning-based health monitoring replaces classical FDIR. Rather, they demonstrate that learned normality, when coupled with runtime confidence and structurally grounded interpretability, can be embedded within safety-critical avionics in a controlled and auditable manner. This supports the broader thesis claim that trust in neural systems can be operationalised through analysis of their internal structure—even under stringent resource and certification constraints.

The combination of learned representations, confidence-aware decision logic, and

compact structural evidence therefore offers a technically credible pathway toward increased onboard autonomy in future space missions, in alignment with ongoing developments in spaceborne AI architectures [Barretta et al. \(2025\)](#).

Further validation across longer mission phases and heterogeneous subsystems remains an open direction for future work.

IV

Synthesis and Future Outlook

9

Synthesis and Outlook

Toward Human-Aligned Autonomy

Introduction

This concluding part steps back from individual experiments and architectural details to examine the broader trajectory of the research. Across domains as diverse as statistical signal processing, Deep Neural Networks (DNNs), and spacecraft autonomy, a single guiding question has shaped the work:

How can an artificial system produce decisions that are not only accurate, but structurally coherent, observable, and aligned with human expectations?

Rather than treating trust as a post-hoc certification label or a regulatory obligation, this thesis has explored a structural perspective: trustworthiness can emerge from internal organization that is mathematically grounded, geometrically interpretable, and operationally monitorable.

The peephole-based framework, progressively articulated through CNA, CNA-HC, and the layer-wise structural score developed in this work, was conceived not as a collection of isolated techniques, but as a unified approach to expose and quantify internal coherence.

The central contribution is therefore not merely the introduction of a new confidence indicator, but the demonstration that internal agreement across depth can be transformed into a measurable reliability signal. Visibility of representations, structured modelling of normality, and aggregation of depth-wise consistency together establish a bridge between interpretability and operational diagnostics.

This chapter synthesizes these contributions at a system level. It reframes the proposed methods within a broader engineering perspective, moving from model-centric performance optimization toward AI systems that are diagnosable, monitorable, and compatible with safety-critical deployment constraints. Finally, it outlines how structural observability may inform future design principles for trustworthy autonomy, where reliability is not inferred solely from outputs, but grounded in the coherence of the reasoning process itself.

9.1 A Design Philosophy for Trustworthy AI

Beyond the specific algorithms developed in this work, a broader design principle emerges: trustworthy AI systems should expose and preserve internal structure.

Throughout this dissertation, structure was treated not as an abstract mathematical artifact but as an operational signal. Whether through covariance analysis in classical signal processing or through layer-wise geometric tracing in deep networks, the central idea remained consistent: reliability depends on the stability of internal organization.

This view contrasts with purely performance-driven optimization, where incremental gains in accuracy are often pursued without regard for internal coherence. In safety-critical systems, however, marginal improvements in prediction metrics cannot compensate for opaque or unstable reasoning dynamics.

The results presented in this thesis suggest that structural coherence can serve as a design constraint: models should not only predict correctly, but do so through trajectories that remain interpretable, stable, and consistent across depth.

9.2 From Geometry to Operational Signals

The progressive transformation of geometric structure into operational indicators is a unifying theme throughout this dissertation. What began as an analysis of covariance operators and dominant modes in classical signal processing evolved into a framework capable of extracting structured descriptors from deep neural representations, ultimately resulting in a measurable signal of reliability.

Mathematically, the central idea is straightforward: structure can be revealed through dimensionality reduction and deviations from that structure can be quantified. Singular value decomposition isolates dominant directions, clustering identifies recurring patterns, and layer-wise projections reveal how representations consolidate or diverge across depth. Initially introduced as analytical instruments, these tools became the foundation for observing how learned models organise semantic information internally.

The peephole-based framework formalised this observation process. By projecting intermediate activations onto low-dimensional bases and modelling their distribution, hidden transformations were converted into structured, comparable descriptors. CNA and CNA-HC then quantified the dispersion and consolidation properties within these reduced spaces, thereby linking the internal geometry to predictive behaviour. The layer-wise structural score further refined this concept, establishing agreement across depth as a scalar indicator of trajectory consistency.

This progression marks an important conceptual shift. Rather than interpreting neural networks exclusively through their outputs, the framework treats the trajectory of the internal representation as the primary object of analysis. Therefore, prediction reliability is not inferred solely from output probabilities, but from the coherence of the transformation sequence that produces them.

From this perspective, a confidence score is not just a post-hoc statistic. Instead, it becomes an operational signal derived from structural consistency across layers. As it is based on geometric consistency rather than output sharpness alone, it remains informative in situations where logits may saturate or become unstable.

The same structural principle extends beyond controlled image benchmarks. In spacecraft telemetry and anomaly detection scenarios, normality is not an abstract concept, but rather a set of operational regimes that are governed by physics, control laws and environmental constraints. In this context, deviations from the normal structure correspond to measurable departures from expected behaviour. Therefore, a confidence score derived from internal geometric consistency can operate as a runtime health indicator, complementing rather than replacing traditional FDIR logic. In the classification setting, geometric displacement in representation space appears as an early signal of behavioural inconsistency in an engineering system. This translation of abstract geometry into operational diagnostics shows that the proposed framework is not just for model analysis, but can also be integrated into AI systems that are subject to real-time and reliability constraints.

Across domains, the underlying rationale remains consistent. Whether the object of analysis is an image classifier or a spacecraft telemetry stream, the central task is the same: to identify structured regularities, monitor their stability and detect meaningful deviations.

The geometric principles introduced at the beginning of this dissertation therefore culminate in an engineering instrument rather than a theoretical abstraction. Structural coherence becomes observable, measurable and operational. In this sense, trustworthiness does not arise from a declaration of confidence, but from sustained alignment between internal organisation and external behaviour.

9.3 Structural Diagnostics as an Engineering Instrument

One of the central shifts proposed in this dissertation concerns the role of interpretability. Rather than treating explanations as post-hoc narratives attached to predictions, the peephole-based framework elevates interpretability to the status of a structural

observable.

The evidence maps and layer-wise similarity measures developed in this study reveal how representations evolve in terms of depth. They reveal whether semantic support is consolidated progressively, whether competing classes remain entangled or whether instability emerges at specific abstraction levels. In this way, internal activations cease to be opaque tensors and become analysable signals.

This structural visibility provides a diagnostic perspective that extends beyond individual predictions. Persistent overlap between specific class manifolds may indicate dataset imbalance or insufficient separability. Abrupt fluctuations across layers may signal architectural bottlenecks. Systematic degradation under controlled perturbations may reveal fragilities that are not apparent at the output level.

The proposed framework therefore transforms interpretability into an engineering tool. It supports inspection during model validation, stress testing under distribution shift and monitoring during deployment. In safety-critical settings, such structural diagnostics provide an additional layer of assurance by offering insight into not only what the system predicts, but also how coherently it makes decisions.

This perspective is particularly relevant in environments where failure has operational consequences. When models operate within larger AI systems that are embedded in sensing chains, decision loops and control architectures, internal coherence becomes a measurable proxy for operational reliability.

9.4 From Model-Centric AI to AI Systems Engineering

Another implication of this work relates to the scale at which AI is designed and evaluated. Much contemporary research remains model-centric, focusing on making incremental improvements to predictive accuracy under benchmark conditions. While this kind of optimisation is scientifically valuable, it does not address the requirements of high-assurance deployment by itself.

In operational contexts, AI rarely functions as an isolated predictor. Instead, it becomes a component within a larger system comprising sensors, preprocessing chains, decision logic, communication layers and human supervision. Within this broader architecture, reliability cannot be assessed solely through top-1 accuracy or calibration curves. Rather, it must be evaluated in terms of stability, diagnosability, and integration with system-level safeguards.

The peephole-based framework facilitates this transition by incorporating internal observability as a design feature. Rather than treating confidence as an external

estimate added to a black-box model, the proposed structural score is derived from representation dynamics that are already present within the system. This makes it naturally compatible with monitoring pipelines, anomaly detection logic and hybrid decision-making architectures.

In safety-critical environments such as aerospace systems, this distinction is crucial. Autonomy must coexist with supervision, and learning-based components must interact with rule-based safeguards and physical constraints. Designing AI systems that expose structural self-assessment mechanisms is therefore a step towards human-aligned autonomy, where automated reasoning remains transparent, monitorable and interruptible.

9.5 Implications for Trustworthy Autonomy

The growing emphasis on trustworthy AI within regulatory and certification frameworks reflects a structural challenge rather than a purely ethical one. High-assurance systems require not only functional performance, but traceability of internal reasoning, and measurable indicators of operational stability.

The framework developed in this dissertation aligns with this requirement by design. Layer-wise structural coherence provides an auditable signal derived from internal model behaviour rather than from external heuristics. It supports transparency not through visualization alone, but through quantifiable agreement across depth.

Importantly, this approach does not claim to replace existing confidence estimators or density-based detectors. Instead, it introduces an additional reliability dimension grounded in representation geometry. In doing so, it bridges the analytical language of signal processing with the governance language of accountability and robustness.

Trustworthiness, in this view, is not asserted by declaration. It is supported by structural evidence.

9.6 Future Outlooks

The framework developed in this thesis opens several concrete research and engineering directions.

- **Refinement through selective structural instrumentation.** In this work, deliberately demanding configurations were adopted to stress-test scalability, including the unrolling of large convolutional filter banks in VGG16 and the

instrumentation of multiple encoder layers in ViT-B16. These choices maximized structural visibility but increased projection dimensionality and memory requirements.

Future implementations need not instrument every layer or all filters within a layer. Strategic selection of depth levels, representative channels, or adaptive instrumentation policies could preserve diagnostic power while significantly reducing computational footprint. Such refinement would facilitate integration in embedded and resource-constrained AI systems.

- **Integration within AI system architectures.** The proposed structural score can evolve from a post-hoc analytical tool into a native component of AI systems. In safety-critical settings, it could feed gating mechanisms, fallback policies, or human-in-the-loop supervision strategies. Rather than replacing existing reliability layers, it would complement them by providing an internal coherence signal grounded in representation geometry.
- **Standardisation of structural confidence metrics.** As regulatory frameworks increasingly require traceability and runtime monitoring of AI components, structurally grounded confidence indicators may contribute to the definition of auditable evaluation procedures. Future work could investigate threshold selection methodologies, statistical validation protocols, and reporting standards aligned with certification processes in aerospace and other high-assurance domains.
- **Extension to sensing-rich and multi-modal domains.** The notion of measurable structural normality extends beyond image classification. Telemetry analysis, industrial monitoring, and multi-modal perception systems provide natural test-beds for applying layer-wise consistency analysis to heterogeneous data streams. In such contexts, structural observability may support earlier anomaly detection and more interpretable fault localisation.

Conclusion

The research presented in this thesis advances a unified framework linking the statistical description of normality, the analysis of its deviations, and the quantification of trust in artificial intelligence systems. Beginning with the classical language of signal processing, covariance, dominant modes, structured approximations, it demonstrated that these principles remain not only relevant but foundational within modern deep neural architectures.

The peephole-based framework and its successive developments showed that interpretability can be embedded into representation itself rather than appended as an explanatory afterthought. By projecting intermediate activations onto structured low-dimensional spaces, modelling their distributions, and tracing their evolution across depth, internal transformations become observable and comparable. CNA and CNA-HC quantified dispersion and consolidation within these representations; the layer-wise structural score distilled their agreement into an operational reliability signal. Together, these tools form a coherent bridge between geometric structure and system-level diagnostics.

A central outcome of this work is the demonstration that performance and interpretability need not be opposing objectives. When internal organization is preserved and exposed, reliability emerges from coherence rather than from output sharpness alone. Confidence, in this sense, is no longer merely a property of the final probability distribution, but a reflection of how consistently a model constructs its decision.

Beyond methodological contributions, this thesis proposes a broader engineering perspective. AI systems operating in safety-critical contexts should not only produce accurate outputs, but also expose structured indicators of their own internal state. Such indicators provide a mechanism for oversight that does not require full knowledge of the underlying architecture. Just as a dashboard warning light conveys meaningful information without revealing the full mechanical complexity of an engine, a structurally grounded confidence signal can inform operators about the health and coherence of an AI system.

This shift is particularly relevant in domains where autonomy must coexist with human supervision. Bridging the gap between machine reasoning and human understanding does not require translating every internal computation into natural language. It requires the design of observables that humans can learn to interpret, trust, and act upon. Structural coherence, as developed in this work, represents one such observable.

Looking forward, the continued integration of geometry, system theory, and AI engineering may contribute to a generation of intelligent systems that are not only powerful, but diagnosable, monitorable, and accountable by design.

Ultimately, trustworthiness is not declared by a system. Rather, it is earned through sustained alignment between internal organization and external behaviour.

For future AI systems to operate with increasing autonomy, their legitimacy will depend not only on their performance, but also on how transparently and coherently they achieve it. The principles developed in this work suggest that geometry, structure and observability may become the foundational design elements of the next generation of AI systems, where autonomy is coupled with intelligibility and power with accountability.

Reducing the gap between machine intelligence and human understanding does not require models to be simplified, but rather structured so that their internal logic remains observable. Therefore, bridging this gap is less a matter of post-hoc explanation and more a matter of deliberate design.

In this sense, the trajectory traced in this dissertation does not end with a confidence score. Instead, it points towards a broader engineering paradigm in which intelligent systems are designed from the outset to be observable and self-diagnosing entities that remain compatible with human reasoning and governance.

V

Appendix



Experimental and Implementation Details

A.1 Models

To showcase the results with the CNA and CNA-HC methods, we focus on a VGG8B architecture [Nokland & Eidnes \(2019\)](#). Models of the Visual Geometric Group (VGG) are classical convolutional networks with varying depth. VGG8B has 6 convolutional layers dedicated to feature extraction and 2 dense layers for classification. Refer to Table A.1 for the complete architecture and parameter sizes. Fig. A.1 shows a well-known variant of this architecture and provides an overview on how the model structures its feature extraction and classification parts.

Two additional architectures are analysed to verify the generality of the proposed layer-wise analysis framework:

- **VGG16**: a classical convolutional network with 13 convolutional and 3 fully connected layers. This architecture is widely targeted in the context of XAI. We use it as a stress test given the high dimensionalities involved in its intermediate activations.
- **ViTB16**: a Vision Transformer model with 12 encoder blocks, each containing multi-head self-attention (MSA) and MLP submodules.

Each convolutional block increases the number of feature channels while halving the spatial resolution, leading to a progressive compression of the input manifold into a hierarchy of latent representations. The parameter tensors of the convolutional layers, $\mathbf{W}_l \in \mathbb{R}^{k \times k \times C_{\text{in}} \times C_{\text{out}}}$, grow quadratically with the kernel size k and linearly with the channel dimensions. Bias vectors $\mathbf{b}_l \in \mathbb{R}^{C_{\text{out}}}$ add a single offset per feature map.

In the context of our *peephole* analysis, these high-dimensional parameter spaces are locally linearized or projected into compact subspaces, producing descriptors of size $d \ll |\mathbf{W}_l|$. Each peephole thus represents a low-dimensional embedding of the layer’s parameter or activation manifold, retaining the salient structure while discarding redundancy. By mapping $\mathbf{W}_l$ or its activation responses $\mathbf{A}_l \in \mathbb{R}^{H_l \times W_l \times C_l}$ into a

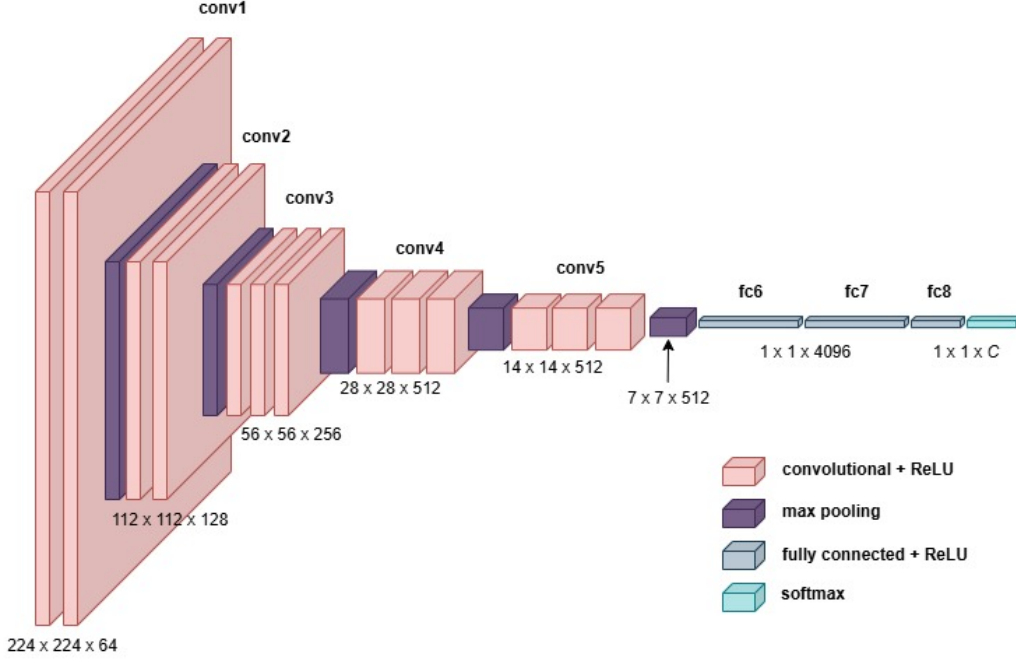


Figure A.1. VGG16 architecture (3-D stacked view).

lower-rank form, we obtain interpretable coordinates that expose how complexity and abstraction evolve across depth.

With a different input size $H \times W$ or patch size P , the token count is $N = (H/P) \cdot (W/P)$ (assuming divisibility), while D , M and h stay as configured. In our peephole view, each parameter bundle (e.g., $\mathbf{W}_{qkv} \in \mathbb{R}^{D \times 3D}$ or $\mathbf{W}_1 \in \mathbb{R}^{D \times M}$) or activation block $((N+1) \times D$ tokens) is projected to a compact descriptor of size $d \ll \text{dim}$, preserving local structure (e.g., spectra, covariance, or concept axes) and enabling cross-depth, cross-layer comparisons.

Both architectures (VGG16 and ViT-B16) are pretrained on ImageNet and fine-tuned on the domain datasets used in this work. For interpretability analysis, we select representative layers that cover distinct depth levels and functional roles—spatial feature extraction (deep convolutional filters) versus semantic abstraction (fully connected or MLP projections).

For VGG16, the Conv5_3 layer represents the deepest convolutional kernel stage before flattening; each filter bank is a $3 \times 3 \times 512$ tensor, giving an equivalent weight matrix of 4608×512 when unfolded channel-wise. The FC2 layer corresponds to a dense transformation $\mathbf{W} \in \mathbb{R}^{4096 \times 4096}$ at the top of the classifier head.

For ViT-B16, the analyzed layers correspond to the two linear mappings in the MLP

Table A.1. VGG8B for CIFAR (padding=1 for all 3×3 convs).

Layer	Type / Kernel / Stride	Output size	Parameters
Input	RGB image	$32 \times 32 \times 3$	–
Conv1_1	Conv 3×3 , s=1, 64	$32 \times 32 \times 64$	$(3 \times 3 \times 3) \times 64 + 64$
Conv1_2	Conv 3×3 , s=1, 64	$32 \times 32 \times 64$	$(3 \times 3 \times 64) \times 64 + 64$
MaxPool1	2×2 , s=2	$16 \times 16 \times 64$	–
Conv2_1	Conv 3×3 , s=1, 128	$16 \times 16 \times 128$	$(3 \times 3 \times 64) \times 128 + 128$
Conv2_2	Conv 3×3 , s=1, 128	$16 \times 16 \times 128$	$(3 \times 3 \times 128) \times 128 + 128$
MaxPool2	2×2 , s=2	$8 \times 8 \times 128$	–
Conv3_1	Conv 3×3 , s=1, 256	$8 \times 8 \times 256$	$(3 \times 3 \times 128) \times 256 + 256$
Conv3_2	Conv 3×3 , s=1, 256	$8 \times 8 \times 256$	$(3 \times 3 \times 256) \times 256 + 256$
MaxPool3	2×2 , s=2	$4 \times 4 \times 256$	–
Flatten	–	4096	–
FC1	Dense(512)	512	$4096 \times 512 + 512$
FC2	Dense(C)	C	$512 \times C + C$

sub-blocks of selected encoders. Both reflect the internal feed-forward transformations over the token embeddings.

In the peephole framework, these high-dimensional tensors are vectorized or spectrally decomposed to generate local descriptors $\mathbf{p}_j \in \mathbb{R}^d$ with $d \ll \dim(\mathbf{W}_l)$, capturing structure-preserving summaries of layer complexity and orientation in the learned representation space.

For each layer, we compute SVD truncation ranks κ and number of clusters C empirically chosen to balance interpretability and statistical coverage.

A.2 Dataset Overview

Unless otherwise specified, all datasets are standardized using mean and variance normalization.

A.3 Layers Overview

The layers selected for analysis correspond to representative stages of feature abstraction within the examined architectures. While for VGG8B we examined layer FC1 mentioned in Table A.1, for VGG16 and ViT-B16 we report the layer type and the corresponding input and activation dimensions, following the notation introduced in Chap. 7.

Table A.2. VGG16 architecture and tensor dimensions.

Layer	Type / Kernel / Stride	Output size	Parameters
Input	RGB image	$224 \times 224 \times 3$	–
Conv1_1	Conv 3×3 , 64	$224 \times 224 \times 64$	$(3 \times 3 \times 3) \times 64 + 64$
Conv1_2	Conv 3×3 , 64	$224 \times 224 \times 64$	$(3 \times 3 \times 64) \times 64 + 64$
MaxPool1	2×2 , s=2	$112 \times 112 \times 64$	–
Conv2_1	Conv 3×3 , 128	$112 \times 112 \times 128$	$(3 \times 3 \times 64) \times 128 + 128$
Conv2_2	Conv 3×3 , 128	$112 \times 112 \times 128$	$(3 \times 3 \times 128) \times 128 + 128$
MaxPool2	2×2 , s=2	$56 \times 56 \times 128$	–
Conv3_1	Conv 3×3 , 256	$56 \times 56 \times 256$	$(3 \times 3 \times 128) \times 256 + 256$
Conv3_2	Conv 3×3 , 256	$56 \times 56 \times 256$	$(3 \times 3 \times 256) \times 256 + 256$
Conv3_3	Conv 3×3 , 256	$56 \times 56 \times 256$	$(3 \times 3 \times 256) \times 256 + 256$
MaxPool3	2×2 , s=2	$28 \times 28 \times 256$	–
Conv4_1	Conv 3×3 , 512	$28 \times 28 \times 512$	$(3 \times 3 \times 256) \times 512 + 512$
Conv4_2	Conv 3×3 , 512	$28 \times 28 \times 512$	$(3 \times 3 \times 512) \times 512 + 512$
Conv4_3	Conv 3×3 , 512	$28 \times 28 \times 512$	$(3 \times 3 \times 512) \times 512 + 512$
MaxPool4	2×2 , s=2	$14 \times 14 \times 512$	–
Conv5_1	Conv 3×3 , 512	$14 \times 14 \times 512$	$(3 \times 3 \times 512) \times 512 + 512$
Conv5_2	Conv 3×3 , 512	$14 \times 14 \times 512$	$(3 \times 3 \times 512) \times 512 + 512$
Conv5_3	Conv 3×3 , 512	$14 \times 14 \times 512$	$(3 \times 3 \times 512) \times 512 + 512$
MaxPool5	2×2 , s=2	$7 \times 7 \times 512$	–
Flatten	–	25,088	–
FC1	Dense(4096)	4096	$(25,088) \times 4096 + 4096$
FC2	Dense(4096)	4096	$(4096) \times 4096 + 4096$
FC3	Dense(C)	C	$(4096) \times C + C$

A.4 Hyper-parameters and Tuning

For the VGG16 model trained on the CIFAR10 dataset we used the rejection-aware criterion described in Sec. 7.1.1. Table A.7 summarizes the considered layers and the related corevector dimensionality κ and number of clusters C based on the score used for tuning.

Another line of reasoning guides the configuration of the dimensionality-reduction and clustering stages corresponding to the results presented in Sec. 7.4 and reports the hyper-parameters adopted in the adversarial sample generation procedures. The latter were optimized using the RAY TUNE framework, jointly searching over the corevector dimensionality κ and the number of clusters C when GMM is the reference clustering method. Tables A.8 and A.9 summarize the best configurations identified during this process.

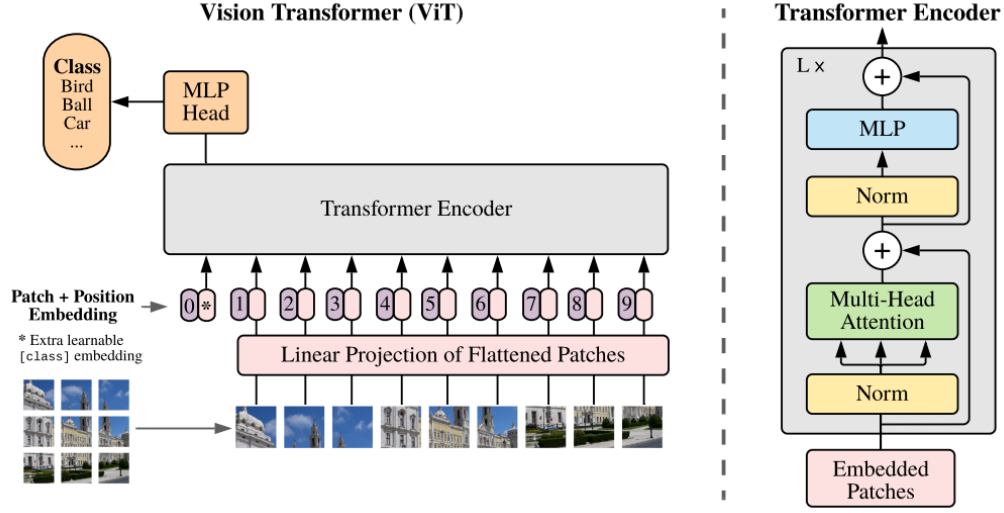


Figure A.2. General ViT architecture [Jain & Wallace \(2019\)](#).

Table A.3. ViT-B16 architecture and main tensor dimensions. Default setup: patch size $P=16$, hidden size $D=768$, MLP size $M=3072$, heads $h=12$, layers $L=12$.

Stage	Operation / Details	Output shape	Main parameter shapes
Input	RGB image	$224 \times 224 \times 3$	–
Patch embedding	16×16 conv / linear $\rightarrow D$	196×768	$(768) \times 768 + 768$
Class token + Pos. embed	add 1 token + positions	197×768	$(197) \times 768$
<i>Transformer Encoder (repeated $L=12$ times)</i>			
LayerNorm (pre)	normalize tokens	197×768	$\gamma, \beta \in \mathbb{R}^{768}$
Multi-head Self-Attn	$h=12, d_h=64$	197×768	$W_{qkv} \in \mathbb{R}^{768 \times 2304}, W_o \in \mathbb{R}^{768 \times 768}$
Residual Add	skip connection	197×768	–
LayerNorm	normalize tokens	197×768	$\gamma', \beta' \in \mathbb{R}^{768}$
MLP block	GELU, $2 \times \text{Linear} (768 \rightarrow 3072 \rightarrow 768)$	197×768	$W_1 \in \mathbb{R}^{768 \times 3072}, W_2 \in \mathbb{R}^{3072 \times 768}$
Residual Add	skip connection	197×768	–
Head (CLS)	take CLS token $\rightarrow$ Linear $D \rightarrow C$	C	$W_{\text{head}} \in \mathbb{R}^{768 \times C}$

The results obtained for VGG16 exhibit a clear trend in the optimal corevector dimensionality. Early convolutional layers require relatively large values ($\kappa \approx 500$), reflecting the high spatial dimensionality and the abundance of low-level features encoded at those depths. As the network progresses toward deeper convolutional layers, where spatial resolution decreases and feature abstraction increases, the optimal value gradually drops to $\kappa \approx 350$. For the dense layers (FC1-FC3), the optimal dimension becomes significantly smaller ($\kappa \approx 50$), consistent with their reduced input dimensionality (Table A.6) and the more compact nature of high-level feature representations.

For ViT-B16, a different pattern emerges. Layers belonging to the *first* branch of the MLP block (denoted here as MLP-X-0) generally require smaller κ values than

Table A.4. Summary of the datasets used in the experiments. Each dataset is split into training ($\mathcal{D}^{\text{train}}$), validation ($\mathcal{D}^{\text{val}}$), and test ($\mathcal{D}^{\text{test}}$) partitions. $\mathcal{D}^{\text{train}}$ is used for model fitting and Gaussian Mixture Model (GMM) estimation; $\mathcal{D}^{\text{val}}$ for tuning κ and C ; and $\mathcal{D}^{\text{test}}$ for framework evaluation.

Dataset	Input Shape (X)	$ \mathcal{D}^{\text{train}} $	$ \mathcal{D}^{\text{val}} $	$ \mathcal{D}^{\text{test}} $
CIFAR10	$32 \times 32 \times 3$	40 000	10 000	10 000
CIFAR100	$32 \times 32 \times 3$	40 000	10 000	10 000
CIFAR100C	$32 \times 32 \times 3$	–	10 000	10 000
PLACES365	$256 \times 256 \times 3$	–	10 000	10 000
SVHN	$32 \times 32 \times 3$	–	10 000	10 000

Table A.5. Dimensions and Types of Selected Layers in ViT-B16 for each of the 12 encoders.

Layer	Type	n
mlp.-.0	Dense	768
mlp.-.3	Dense	3072
heads.head	Dense	768

their MLP- X -3 counterparts. This behaviour is coherent with the ViT architecture: the first MLP projection acts as an expansion layer (projection to a higher-dimensional hidden space), whereas the second projection compresses the representation. The dimensionality-reduction step thus retains more information when applied after the expansion (MLP- X -3), explaining the systematically higher optimal κ values observed in those layers.

The optimal number of clusters C displays larger variance across layers, which is expected: increasing C partitions the corevector space more finely, allowing the model to capture more localized patterns at the cost of increased fragmentation. Nevertheless, it is notable that C rarely exceeds 2,500 for most layers, with only a few ViT layers reaching higher values. We speculate that this upper range is constrained by the diversity and density of the available training samples: excessively large C values result in sparse *empirical posterior* distributions and provide diminishing returns.

Finally, the obtained C values offer an empirical proxy for estimating the number of distinct low level features encoded at each layer, a property that may be exploited in future work to characterize feature complexity across architectures.

Table A.6. Dimensions and Types of Selected Layers in VGG16.

Layer	Type	$H \times W \times C_{\text{in}}$	n
Conv1_2	Conv2D	$112 \times 112 \times 128$	1,605,632
Conv2_2	Conv2D	$56 \times 56 \times 128$	401,408
Conv3_2	Conv2D	$56 \times 56 \times 256$	802,816
Conv3_3	Conv2D	$56 \times 56 \times 256$	802,816
Conv4_1	Conv2D	$28 \times 28 \times 256$	200,704
Conv4_2	Conv2D	$28 \times 28 \times 512$	401,408
Conv4_3	Conv2D	$28 \times 28 \times 512$	401,408
Conv5_1	Conv2D	$14 \times 14 \times 512$	100,352
Conv5_2	Conv2D	$14 \times 14 \times 512$	100,352
Conv5_3	Conv2D	$14 \times 14 \times 512$	100,352
FC1	Dense	-	25,088
FC2	Dense	-	4,096
FC3	Dense	-	4,096

Table A.7. Layer-wise tuning configurations under the rejection-aware criterion for VGG16 on CIFAR10. The score $s_\ell(\cdot)$ is defined in Eqs. (7.1)–(7.2).

Score s_ℓ	Clusterer	Layer	(κ, C)	Rejected Fraction
$s_\ell^{\max}$	k-Means	FC2	(64, 64)	0.0675
$s_\ell^{\max}$	k-Means	FC1	(32, 32)	0.1056
$s_\ell^{\max}$	k-Means	Conv5_3	(300, 64)	0.2466
$s_\ell^{\max}$	k-Means	Conv5_2	(128, 64)	0.3245
$s_\ell^{\max}$	k-Means	Conv5_1	(256, 32)	0.4434
s_ℓ^{ent}	k-Means	FC2	(256, 64)	0.0721
s_ℓ^{ent}	k-Means	FC1	(32, 32)	0.0997
s_ℓ^{ent}	k-Means	Conv5_3	(256, 128)	0.2293
s_ℓ^{ent}	k-Means	Conv5_2	(128, 64)	0.3096
s_ℓ^{ent}	k-Means	Conv5_1	(512, 16)	0.3857

A.4.1 Adversarial Generation Parameters

The distance metrics used for the generation of adversarial examples are ℓ_2 for DeepFool (DF) and Carlini & Wagner (CW), and ℓ_∞ for Projected Gradient Descent (PGD) and Basic Iterative Method (BIM). All attacks were tuned to reach 95-98% success rates within perceptual similarity thresholds.

Table A.8. Best κ - C configurations obtained through hyper-parameter tuning for VGG16.

Layer	κ	C
Conv1_2	500	1360
Conv2_2	479	2239
Conv3_2	442	2159
Conv3_3	496	524
Conv4_1	356	773
Conv4_2	499	1326
Conv4_3	489	851
Conv5_1	323	1035
Conv5_2	351	838
Conv5_3	377	1481
FC1	50	1717
FC2	53	1627
FC3	51	1706

A.5 Additional Results on Confidence and Robustness

The results reported in Table A.10 complement the qualitative discussion in Sec. 7.3.4, in particular with reference to the scenario illustrated in Fig. 7.7. In that example, the ViT-B16 model is evaluated under four input conditions: nominal, partially corrupted (near-OOD), and adversarially perturbed using BIM and PGD. We compare the proposed internal consistency score against MSP and the competing post-hoc confidence estimators DOC, Rel-U, and FS (Sec. 5.1).

In the nominal case, the network correctly classifies the image and assigns an almost unitary MSP (0.98). All considered scores, including the proposed one, remain high, reflecting a coherent internal activation pattern consistent with the predicted class. This behaviour confirms that, under standard operating conditions, the model’s internal representations follow the expected decision trajectory encoded in the proto-class maps.

In the near-OOD case, the input is partially corrupted but still correctly classified. While MSP and the competing scores remain high (all above 0.89), the proposed score drops significantly (0.57). This reduction indicates that, although the output label is correct, the internal classification map deviates from the nominal pattern typically associated with that class. In other words, the decision is correct but structurally atypical. This behaviour highlights the sensitivity of the method to subtle distributional shifts that do not yet cause misclassification but already alter the internal reasoning geometry.

Table A.9. Best κ - C configurations obtained through hyper-parameter tuning for ViT-B16.

Layer	(κ, C)	Layer	(κ, C)
MLP-0-0	(448, 1694)	MLP-6-0	(254, 1945)
MLP-1-0	(52, 2436)	MLP-7-0	(253, 1234)
MLP-2-0	(390, 1527)	MLP-8-0	(376, 1025)
MLP-3-0	(100, 1881)	MLP-9-0	(500, 918)
MLP-4-0	(156, 2295)	MLP-10-0	(258, 1060)
MLP-5-0	(174, 1761)	MLP-11-0	(378, 3714)
MLP-0-3	(54, 1417)	MLP-6-3	(410, 2381)
MLP-1-3	(83, 3252)	MLP-7-3	(479, 1766)
MLP-2-3	(469, 1423)	MLP-8-3	(346, 1486)
MLP-3-3	(333, 2921)	MLP-9-3	(441, 1263)
MLP-4-3	(405, 1606)	MLP-10-3	(403, 2473)
MLP-5-3	(398, 1321)	MLP-11-3	(153, 2373)
Head	(52, 4519)		

The contrast becomes even more pronounced under adversarial attacks. For both BIM and PGD, the network assigns maximum confidence (MSP equal to 1.00) to an incorrect class, demonstrating the well-known overconfidence of softmax-based measures. DOC, Rel-U, and FS also remain close to saturation, failing to reflect the structural inconsistency of the decision. In contrast, the proposed score decreases substantially (0.42 for BIM and 0.72 for PGD), signalling a clear mismatch between the observed classification map and the proto-class typically associated with the predicted label.

Overall, these results reinforce the central premise of the framework: confidence should not only reflect output-level certainty but also internal geometric coherence. By explicitly measuring the alignment between a sample’s layer-wise activation trajectory and the model’s learned nominal patterns, the proposed score provides a structurally grounded indicator of reliability that remains sensitive to subtle shifts and adversarial manipulations, even when competing methods remain overconfident.

A.6 Software and Computational Environment

All experiments were implemented in Python using the PyTorch framework, with GPU acceleration enabled through CUDA-compatible NVIDIA hardware. Dimensionality reduction, clustering, and auxiliary numerical routines rely on standard linear algebra and scientific computing libraries. The proposed methodology is independent of specific software versions and does not rely on framework-dependent features. Its core

Table A.10. Quantitative comparison of confidence and robustness scores for ViT-B16 under different input conditions. The internal consistency score quantifies the mismatch between each input classification map and its corresponding proto-class (Eq. 4.4).

Score	Normal	OOD	BIM	PGD
MSP	0.98 (✓)	0.94 (✓)	1.00 (×)	1.00 (×)
Ours (Eq. 4.4)	0.91	0.57	0.42	0.72
DOC Lin et al. (2021)	0.96	0.89	1.00	0.99
Rel-U Liu et al. (2022)	0.99	0.98	1.00	1.00
FS Xu et al. (2018)	1.00	0.99	0.92	0.97

components are based on classical matrix decompositions, probabilistic clustering, and similarity measures, ensuring portability across computational environments.

Bibliography

- Alvarez-Melis, D. & Jaakkola, T. S. 2018, *Towards Robust Interpretability with Self-Explaining Neural Networks*, in Advances in Neural Information Processing Systems (NeurIPS)
- Avizienis, A., Laprie, J.-C., Randell, B. & Landwehr, C. 2004, *Basic concepts and taxonomy of dependable and secure computing*, IEEE Transactions on Dependable and Secure Computing, 1, 11
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R. & Samek, W. 2015, *On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation*, PloS one, 10, e0130140
- Bahdanau, D., Cho, K. & Bengio, Y. 2015, *Neural Machine Translation by Jointly Learning to Align and Translate*, in International Conference on Learning Representations (ICLR)
- Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R. & Herrera, F. 2020, *Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI*, Information Fusion, 58, 82
- Barretta, D., Marini, L., Razzano, F., Guerrisi, G., Sandberg, S., Del Prete, R., Zancan, V., Mariotti, E., Furano, G., Manovi, L., Guo, J., Ullo, S. L., Longépé, N. & Meoni, G. 2025, *Towards Real-Time Earth System Intelligence through Advancements in Onboard AI Processing*. IEEE Geoscience and Remote Sensing Magazine
- Bau, D., Zhou, B., Khosla, A., Oliva, A. & Torralba, A. 2017, *Network dissection: Quantifying interpretability of deep visual representations*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 6541–6549
- Borgia, C., Ciancarelli, C., Corallo, F., Fiaschetti, A., Franchi, L., Di Ienno, D., Leboffe, A., Pinci, I., Manovi, L., Aldrigo, F., De Tommasi, M., Marchioni, A., Mangia, M.,

BIBLIOGRAPHY

- Rovatti, R., Furano, G. & Mariotti, E. 2025, *Advanced Health Monitoring and Failure Identification in Satellite Avionics using Machine Learning*, in 2025 European Data Handling Data Processing Conference (EDHPC)
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. 2015, *Time Series Analysis: Forecasting and Control* (Wiley)
- Breunig, M., Kriegel, H., Ng, R. & Sander, J. 2000, *LOF: Identifying density-based local outliers*, ACM SIGMOD Record, 29, 93
- Capelli, L., De Souza Rosa, L., Sartor, A. S., Manovi, L., Mangia, M. & Rovatti, R. 2025, *Peephole-Based Interpretability for Onboard Anomaly Detection*, in Explainable AI in Safety-Critical Systems Workshop (ECAI 2025)
- Carlini, N. & Wagner, D. 2017, *Towards Evaluating the Robustness of Neural Networks*, in 2017 IEEE Symposium on Security and Privacy (SP) (Los Alamitos, CA, USA: IEEE Computer Society), 39–57
- Ciancarelli, C., Corallo, F., Cognetta, S., Mariotti, E., Mangia, M., Marchioni, A., Manovi, L., Setti, G., Pareschi, F. & Rovatti, R. 2022, *New Concepts Of Automated Anomaly Detection In Space Operations Through ML-Based Techniques*, in IAC 2022 Congress Proceedings, 73rd edn., Paris, France
- Ciancarelli, C., Mariotti, E., Corallo, F., Cognetta, S., Manovi, L., Marchioni, A., Mangia, M., Rovatti, R. & Furano, G. 2023, *Innovative ML-based Methods for Automated On-board Spacecraft Anomaly Detection*, in The Use of Artificial Intelligence for Space Applications, ed. C. Ieracitano, N. Mammone, M. Di Clemente, M. Mahmud, R. Furfaro, & F. C. Morabito (Cham: Springer Nature Switzerland), 213–228
- Dadalto, E., Romanelli, M., Pichler, G. & Piantanida, P. 2024, *A Data-Driven Measure of Relative Uncertainty for Misclassification Detection*
- Davis, J. & Goadrich, M. 2006, *The relationship between Precision-Recall and ROC curves*, in Proceedings of the 23rd International Conference on Machine Learning, ICML '06 (New York, NY, USA: Association for Computing Machinery), 233–240
- Doshi-Velez, F. & Kim, B. 2017, *Towards a Rigorous Science of Interpretable Machine Learning*, arXiv preprint arXiv:1702.08608
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houlsby, N. 2021,

- An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, in ICLR
- Eckart, C. & Young, G. 1936, *The Approximation of One Matrix by Another of Lower Rank*, Psychometrika, 1, 211
- Enttsel, A., Marchioni, A., Manovi, L., Nikpali, R., Ravaglia, G., Setti, G., Rovatti, R. & Mangia, M. 2024a, *Anomaly Detection-Reconstruction Trade-Off in Autoencoder-Based Compression*, 1957–1961
- Enttsel, A., Onofri, S., Marchioni, A., Mangia, M., Setti, G. & Rovatti, R. 2024b, *A General Framework for the Assessment of Detectors of Anomalies in Time Series*, IEEE Transactions on Industrial Informatics, 20, 12051
- European Union. 2024, *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 on establishing harmonised rules on artificial intelligence and amending Regulations (EC) No 45/2001 and (EU) 2019/1802 and Directives (EU) 2020/1828 and (EU) 2022/2486*
- Ghorbani, A., Wexler, J., Zou, J. & Kim, B. 2019, *Towards Automatic Concept-based Explanations*, in Advances in Neural Information Processing Systems (NeurIPS)
- Golub, G. H. & Van Loan, C. F. 2013, *Matrix Computations*, 4th edn. (Baltimore, MD: Johns Hopkins University Press)
- Goodfellow, I., Shlens, J. & Szegedy, C. 2015, *Explaining and Harnessing Adversarial Examples*, in International Conference on Learning Representations
- Granese, F., Romanelli, M., Gorla, D., Palamidessi, C. & Piantanida, P. 2021, *DOCTOR: A Simple Method for Detecting Misclassification Errors*, in Proceedings, Proceedings, Virtual event, United States, 5669–5681
- Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. 2017, *On calibration of modern neural networks*, in ICML’17, Vol. 70, Proceedings of the 34th International Conference on Machine Learning - Volume 70 (JMLR.org), 1321–1330
- Hendrycks, D. & Dietterich, T. G. 2019, *Benchmarking Neural Network Robustness to Common Corruptions and Perturbations*, CoRR, abs/1903.12261
- Hendrycks, D. & Gimpel, K. 2017, *A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks*, in International Conference on Learning Representations

BIBLIOGRAPHY

- High-Level Expert Group on AI. 2019, *Ethics Guidelines for Trustworthy AI*, Policy and investment recommendations, European Commission, Brussels
- Hinton, G. E. & Salakhutdinov, R. R. 2006, *Reducing the Dimensionality of Data with Neural Networks*, Science, 313, 504
- Hochreiter, S. & Schmidhuber, J. 1997, *Long Short-Term Memory*, Neural Computation, 9, 1735
- Hubert, L. & Arabie, P. 1985, *Comparing partitions*, Journal of Classification, 2, 193
- Hüllermeier, E. & Waegeman, W. 2021, *Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods*, Machine Learning, 110, 457–506
- Jain, S. & Wallace, B. C. 2019, *Attention is not Explanation*, in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)
- Jolliffe, I. 2002, *Principal Component Analysis*, Springer Series in Statistics
- Kay, S. M. 1998, Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory (Upper Saddle River, NJ: Prentice Hall)
- Kendall, A. & Gal, Y. 2017, *What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?*, CoRR, abs/1703.04977
- Kim, B., Wattenberg, M., Gilmer, J., Cai, C. J., Wexler, J., Viegas, F. & Sayres, R. 2018, *Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)*, in International Conference on Machine Learning (ICML)
- Kornblith, S., Norouzi, M., Lee, H. & Hinton, G. 2019, *Similarity of Neural Network Representations Revisited*, in International Conference on Machine Learning (ICML)
- Kurakin, A., Goodfellow, I. J. & Bengio, S. 2018, in Artificial intelligence safety and security (Chapman and Hall/CRC), 99–112
- Leboffe, A., Ienno, D. D., Pinci, I., Corallo, F., Ciancarelli, C., Mangia, M., Manovi, L., Marchioni, A., Rovatti, R., Mariotti, E. & Furano, G. 2024, *Optimized Machine Learning-based Strategies for On-board S/C Failure Detection: Software Integration and Testing on a Space-qualified Processor*, in Proceedings of the 75th International Astronautical Congress (IAC-24), International Astronautical Federation (IAF), Milan, Italy

- Lee, K., Lee, K., Lee, H. & Shin, J. 2018a, *A simple unified framework for detecting out-of-distribution samples and adversarial attacks*, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18 (Red Hook, NY, USA: Curran Associates Inc.), 7167–7177
- Lee, K., Lee, K., Lee, H. & Shin, J. 2018b, *A Simple Unified Framework for Detecting Out-of-Distribution and Adversarial Samples*, in Advances in Neural Information Processing Systems (NeurIPS)
- Li, X., Liu, Y., Han, B., Zhang, S., Xie, X., Yuan, C., Wang, S. & Song, D. 2023, *Trustworthy Artificial Intelligence: A Survey and Taxonomy*, ACM Computing Surveys, 55, 1
- Lin, S., Roy, A. & Fu, H. 2021, *DOCTOR: Detecting Out-of-distribution and Misclassification Errors in Neural Networks*, IEEE Transactions on Neural Networks and Learning Systems
- Lipton, Z. C. 2018, *Mythos of Model Interpretability*, Queue, 16, 31
- Liu, W., Jia, X., Lyu, Y., Li, S. & Grosse, R. 2022, *Relative Uncertainty for Out-of-Distribution Detection*, in Advances in Neural Information Processing Systems (NeurIPS)
- Lloyd, S. 1982, *Least squares quantization in PCM*, IEEE Transactions on Information Theory, 28, 129
- Longo, L., Brcic, M., Cabitza, F., Choi, J., Confalonieri, R., Ser, J. D., Guidotti, R., Hayashi, Y., Herrera, F., Holzinger, A., Jiang, R., Khosravi, H., Lecue, F., Malgieri, G., Páez, A., Samek, W., Schneider, J., Speith, T. & Stumpf, S. 2024, *Explainable Artificial Intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions*, Information Fusion, 106, 102301
- Lundberg, S. M. & Lee, S.-I. 2017, *A unified approach to interpreting model predictions*, Advances in neural information processing systems, 4768–4777
- MacQueen, J. B. 1967, *Some Methods for Classification and Analysis of Multivariate Observations*, in Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1 (University of California Press), 281–297
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D. & Vladu, A. 2017, *Towards deep learning models resistant to adversarial attacks*

BIBLIOGRAPHY

- Maennel, H., Alabdulmohsin, I., Tolstikhin, I., Baldock, R. J. N., Bousquet, O., Gelly, S. & Keyzers, D. 2020, *What Do Neural Networks Learn When Trained With Random Labels?*
- Manovi, L., Capelli, L., Marchioni, A., Martinini, F., Setti, G., Mangia, M. & Rovatti, R. 2024a, *SVD-based Peephole and Clustering to Enhance Trustworthiness in DNN Classifiers*, in 2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS), 129–133
- Manovi, L., Capelli, L., Marchioni, A., Martinini, F., Setti, G., Mangia, M. & Rovatti, R. 2024b, *SVD-based Peephole and Clustering to Enhance Trustworthiness in DNN Classifiers*, in 2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS), 129–133
- Manovi, L., Gallon, R., Furano, G., Rovatti, R., Mangia, M., Menicucci, A. & Schiemenz, F. 2025, *Beyond the Black Box: Advancing Transparency, Explainability, and Trust in AI for Space*, in Proceedings of the 2025 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT) (Barcelona, Spain: IEEE)
- Manovi, L., Leboffe, A., Marchioni, A., Mariotti, E., Mangia, M., Corallo, F., Di Ienno, D., Pinci, I., Rovatti, R., Ciancarelli, C. & Furano, G. 2024c, *Onboard AI for Enhanced FDIR: Revolutionizing Spacecraft Operations with Anomaly Detection*, 2024-10-03, Zenodo
- Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D. & Lucic, M. 2021, *Revisiting the calibration of modern neural networks*, in Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21 (Red Hook, NY, USA: Curran Associates Inc.)
- Mirsky, L. 1960, *Symmetric Gauge Functions and Unitarily Invariant Norms*, Quarterly Journal of Mathematics, 11, 50
- Molnar, C. 2020, Interpretable Machine Learning (Lulu.com)
- Montavon, G., Binder, A., Lapuschkin, S., Samek, W. & Müller, K.-R. 2017, *Explaining Nonlinear Classification Decisions with Deep Taylor Decomposition*, Pattern Recognition, 65, 211
- Montavon, G., Samek, W. & Müller, K.-R. 2018, *Methods for Interpreting and Understanding Deep Neural Networks*, Digital Signal Processing, 73, 1

- Montavon, G., Samek, W. & Müller, K.-R. 2019, in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (Springer), 193–209
- Moosavi-Dezfooli, S.-M., Fawzi, A. & Frossard, P. 2016, *DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
- Morcos, A. S., Raghu, M. & Bengio, S. 2018, *Insights on Representational Similarity in Neural Networks with Canonical Correlation*, in *Advances in Neural Information Processing Systems (NeurIPS)*
- Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R. & Yu, B. 2019, *Interpretable Machine Learning: Definitions, Methods, and Applications*, *Proceedings of the National Academy of Sciences (PNAS)*
- Murtagh, F. & Legendre, P. 2014, *Ward’s Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward’s Criterion?*, *Journal of Classification*, 31, 274
- Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y. & others. 2011, *Reading digits in natural images with unsupervised feature learning*, Granada, 7
- Nokland, A. & Eidnes, L. H. 2019, *Training Neural Networks with Local Error Signals*, arXiv preprint arXiv:1901.06656
- Olah, C., Mordvintsev, A. & Schubert, L. 2017, *Feature Visualization*, *Distill*, 2
- Papernot, N. & McDaniel, P. 2018, *Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning*
- Papayan, V., Han, X. & Donoho, D. L. 2020, *The Prevalence of Neural Collapse During the Terminal Phase of Deep Learning Training*, *Proceedings of the National Academy of Sciences (PNAS)*
- Raghu, M., Gilmer, J., Yosinski, J. & Sohl-Dickstein, J. 2017, *SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics*, in *Advances in Neural Information Processing Systems (NeurIPS)*
- Reynolds, D. A. 2009, *Gaussian Mixture Models*, *Encyclopedia of Biometrics*, 659
- Ribeiro, M. T., Singh, S. & Guestrin, C. 2016, “Why should I trust you?” *Explaining the predictions of any classifier*, in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, KDD ’16* (New York, NY, USA: Association for Computing Machinery), 1135–1144

BIBLIOGRAPHY

- Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K. & Müller, K.-R., eds. 2021, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (Springer)
- Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J. & Williamson, R. C. 2001, *Estimating the Support of a High-Dimensional Distribution*, *Neural Computation*, 13, 1443
- Sedghi, H., Gupta, V. & Long, P. M. 2018, *The Singular Values of Convolutional Layers*, arXiv preprint arXiv:1805.10408
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. & Batra, D. 2017, *Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization*, in 2017 IEEE International Conference on Computer Vision (ICCV), 618–626
- Simonyan, K., Vedaldi, A. & Zisserman, A. 2014, *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*, in International Conference on Learning Representations (ICLR Workshop)
- Smilkov, D., Thorat, N., Kim, B., Viégas, F. & Wattenberg, M. 2017, *SmoothGrad: Removing Noise by Adding Noise*, in ICML Workshop on Visualization for Deep Learning
- Springenberg, J. T., Dosovitskiy, A., Brox, T. & Riedmiller, M. 2015, *Striving for Simplicity: The All Convolutional Net*, in International Conference on Learning Representations (ICLR Workshop)
- Sundararajan, M., Taly, A. & Yan, Q. 2017, *Axiomatic Attribution for Deep Networks*, in Proceedings of Machine Learning Research, Vol. 70, Proceedings of the 34th International Conference on Machine Learning, ed. D. Precup & Y. W. Teh (PMLR), 3319–3328
- Tabassi, E. 2023, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST Trustworthy and Responsible AI, National Institute of Standards and Technology, Gaithersburg, MD
- Trees, H. L. V. 2001, *Detection, Estimation, and Modulation Theory, Part I*, 2nd edn. (New York: Wiley)
- Xu, W., Evans, D. & Qi, Y. 2018, *Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks*, in Proceedings 2018 Network and Distributed System Security Symposium (Internet Society)

- Zhang, X.-Y., Xie, G.-S., Li, X., Mei, T. & Liu, C.-L. 2023, *A Survey on Learning to Reject*, Proceedings of the IEEE, 111, 185
- Zhou, B., Lapedriza, A., Khosla, A., Oliva, A. & Torralba, A. 2017, *Places: A 10 million Image Database for Scene Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence

Acknowledgements

First and foremost, to my family.

For your steady presence, for trusting in me through every orbit of this path and for doing so even when the trajectory wasn't clear. This work carries your patience more than my strength.

I am sincerely grateful to Prof. Riccardo Rovatti for his guidance, patience and curiosity that continually inspired this work, and to Prof. Mauro Mangia for his vision, contagious enthusiasm and for betting on me from the very beginning.

My sincere thanks extend to Dr. Gianluca Furano and to the European Space Agency for welcoming me and for offering an exceptional environment where research meets real-world vision.

To those who crossed this path professionally and personally, contributing in ways large or small - discussions, challenges, disagreements, or shared effort - thank you. Growth rarely comes from comfort alone.

To music, for always breaking my noise and tuning my thoughts when silence felt too heavy.

To love, that one constant variable behind every motion. Love for the people who saw me even and especially when I couldn't see myself and accepted me anyway. Love as fuel, as chaos, as clarity. Love that is lost, found, and never fully explained.

Looking back now at the beginning of this road, I recall that question *does this path have a heart?* I've wondered it often. I would say it surely did.

To be a rock and not to roll

