Practical Work: Out-of-Distribution Detection, OOD Scoring Methods, and Neural Collapse

Nicolas RINCON VIJA

Nicolas LOPEZ NIETO

nicolas.rincon@ensta.fr

nicolas.lopez@ensta.fr

École Nationale Supérieure de Techniques Avancées - ENSTA
ROB313 – Deep Learning in Computer Vision
Palaiseau, France

Academic Year 2025-2026

Contents

List of Figures

1 Part I — Training pipeline (`model.py`)

1.1 Computational environment

All experiments were executed on the ENSTA SLURM cluster. The training job was submitted to the `ENSTA-l40s` partition and requested one NVIDIA L40S GPU, 8 CPU cores, 32 GB of RAM, and a 4-hour time limit. The software environment was a local Python virtual environment, and paths were controlled via environment variables to ensure reproducibility: `DATA_DIR` was set to the dataset folder and `OUT_DIR` was set to a run-specific output directory using the SLURM job ID, so that each training run writes its checkpoints, logs, and figures into an isolated folder.

The implementation uses Python 3.12.3 with PyTorch and torchvision compiled with CUDA support. The in-distribution dataset is CIFAR-100.

The dataset was used as the in-distribution dataset. The official training split was used entirely for optimization. The official test split was divided into two disjoint subsets, a validation set of 5,000 images and a test set of 5,000 images.

1.2 Model architecture

We implemented a ResNet-18 architecture adapted to CIFAR resolution. In particular, the initial convolution uses a 3×3 kernel with stride 1 and no initial max-pooling layer. The network consists of four residual stages with channel dimensions $64 \to 128 \to 256 \to 512$, followed by global average pooling and a fully connected layer producing 100 logits.

The model was trained using cross-entropy loss.

1.3 Optimization and learning rate schedule

Optimization was performed using stochastic gradient descent with momentum 0.95 and weight decay 10^{-4}. A OneCycle learning rate policy was employed over 300 epochs with:

$$\text{max\_lr} = 0.2, \quad \text{pct\_start} = 0.43,$$

and a three-phase schedule. The batch size was set to 512.

1.4 Training dynamics

Training converged smoothly without instability. At the end of training:

- Final training accuracy: 99.98%
- Best validation accuracy: 76.04%
- Final test accuracy: 76.52%
- Final test loss: 1.6894

While the training accuracy approaches 100%, validation and test accuracy stabilize around 76%. The best validation checkpoint was automatically saved.

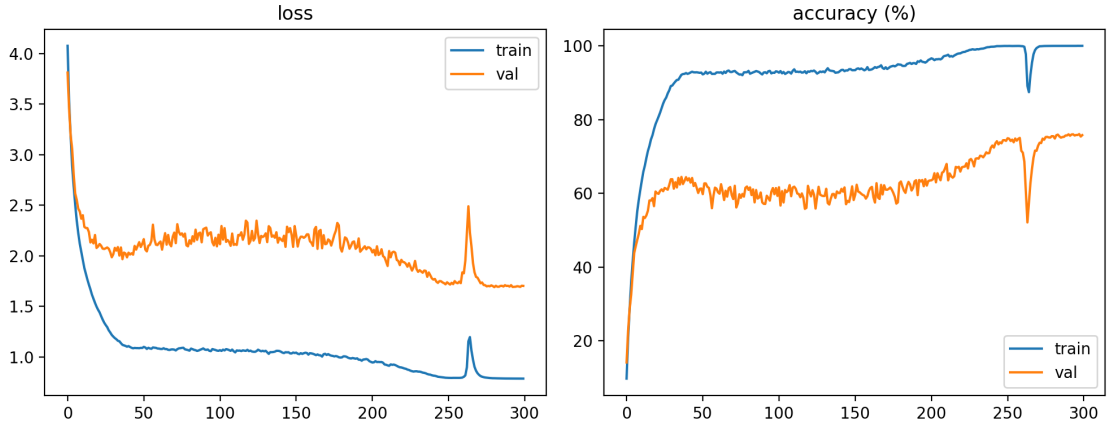


Figure 1: Training and validation curves over 300 epochs (loss and accuracy).

1.5 Saved artifacts for post-hoc analysis

Beyond classification performance, the training stage was designed to produce all statistics required for the second part of the TP (OOD detection and Neural Collapse analysis). In particular:

- The classifier head parameters (W, b) were saved in `classifier_head.pt`.
- Penultimate-layer features and labels for the entire training set were extracted using a forward hook on the fully connected layer and stored in `train_penultimate_feats_labels.pt`.
- Class-wise feature means and within-class variance statistics were computed and saved in `class_stats_train.pt`, enabling Mahalanobis-based scoring.
- PCA parameters (mean vector and principal directions with $d = 64$ components) were fitted on training features and stored in `neco_params.pt`, enabling the NECO score computation.

This separation between training (`model.py`) and analysis (`calculate.py`) ensures that all subsequent OOD and Neural Collapse experiments are purely post-hoc, without retraining the network.

2 Part II — OOD scoring and evaluation (`calculate.py`)

2.1 Evaluation protocol (ID/OOD, features, and metrics)

All post-hoc computations are performed by calculate.py using the best checkpoint saved during training. CIFAR-100 is treated as the in-distribution (ID) dataset, and SVHN is used as th out-of-distribution dataset. Both datasets are processed with the same normalization statistics; SVHN images are resized to 32×32 before normalization to match CIFAR-100 input size.

For each sample x, the network outputs logits $z(x) \in \mathbb{R}^C$ with $C = 100$, and we also extract penultimate features $f(x) \in \mathbb{R}^d$ by registering a forward hook on the fully connected layer and collecting its input. For each method we compute a scalar score $s(x)$ and follow the convention used in our implementation: higher score indicates the sample is more likely to be in-distribution. We report AUROC, AUPR, and FPR@95TPR (defined as the fraction of OOD samples whose score exceeds the 5th percentile of ID scores, i.e., the false positive rate when the ID true positive rate is fixed to 95%).

2.2 OOD score definitions

Max Softmax Probability (MSP). The MSP score is the maximum predicted probability:

$$s_{\text{MSP}}(x) = \max_c \ \text{softmax}(z(x))_c.$$

The corresponding ID/OOD score histogram is shown in Figure 2a.

Maximum Logit Score. The MaxLogit score uses the largest raw logit:

$$s_{\text{MaxLogit}}(x) = \max_c \ z_c(x).$$

The corresponding ID/OOD score histogram is shown in Figure 2b.

Energy score. We use the (temperature-1) log-sum-exp energy score:

$$s_{\text{Energy}}(x) = \log \sum_{c=1}^{C} \exp(z_c(x)).$$

With our scoring convention, ID samples tend to produce larger values than OOD samples. The corresponding ID/OOD score histogram is shown in Figure 2c.

Mahalanobis score (feature-based). We fit class means and a shared covariance on ID training penultimate features. Let μ_c be the mean of class c, and let Σ be the pooled within-class covariance estimated from centered features. A small ridge term is added for numerical stability: $\Sigma \leftarrow \Sigma + 10^{-4} I$. For a test feature $f(x)$, we compute the minimum squared Mahalanobis distance to the class means:

$$d_{\text{Maha}}(x) = \min_c \left(f(x) - \mu_c\right)^\top \Sigma^{-1} \left(f(x) - \mu_c\right),$$

and convert it to an ID score by negation:

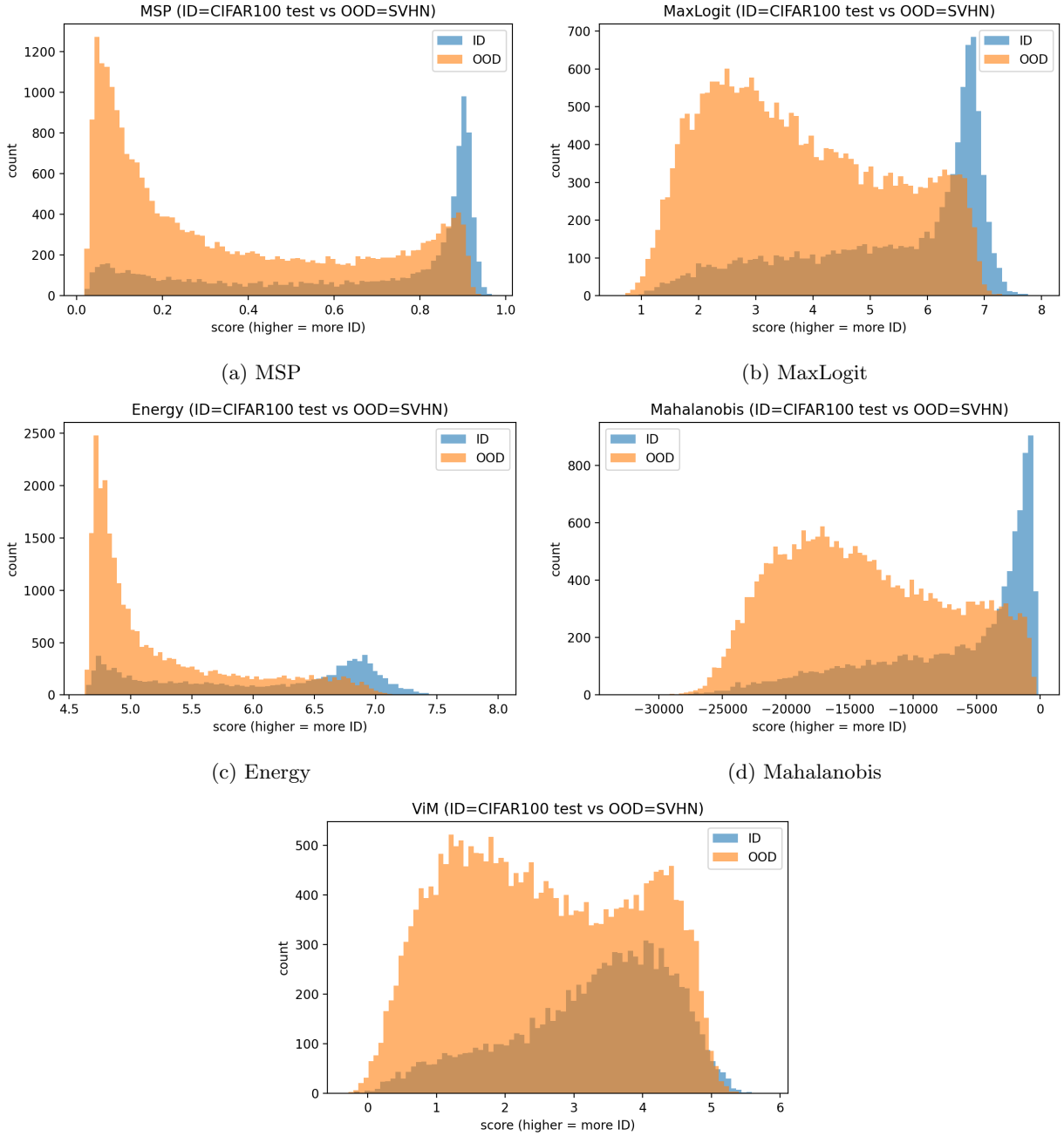
$$s_{\text{Maha}}(x) = -d_{\text{Maha}}(x).$$

The corresponding ID/OOD score histogram is shown in Figure 2d.

ViM score. We use a vanilla ViM formulation that combines a residual term in feature space with a logit aggregation term. First, we compute a reference vector $o \in \mathbb{R}^d$ using the classifier parameters, defined as $o = -W^\dagger b$ where $W^\dagger$ denotes the Moore Penrose pseudoinverse of the final layer weight matrix and b is the bias vector. We then center training features as $x = f - o$ and fit a PCA subspace spanned by the top D principal components. For a sample, we compute the residual norm in the orthogonal complement of that subspace, and we define a scaled residual term $v(x) = \alpha \|x - UU^\top x\|_2$, where α is fitted on training data as the ratio between the mean max-logit and the mean residual norm. ViM OODness is computed as $v(x) - \log \sum_c \exp(z_c(x))$. Since our convention is that higher scores indicate more in-distribution, we report the negative of this OODness, namely

$$s_{\text{ViM}}(x) = \log \sum_{c=1}^{C} \exp(z_c(x)) - \alpha \left\|x - UU^\top x\right\|_2, \qquad x = f(x) - o.$$

The corresponding ID and OOD score histogram is shown in Figure 2e.



(a) MSP

(b) MaxLogit

(c) Energy

(d) Mahalanobis

(e) ViM

Figure 2: ID vs OOD score distributions for each compared method (SVHN as OOD).

Figure 2 shows the empirical score distributions for ID and OOD. A substantial overlap is still visible across methods, which explains why FPR@95TPR can remain high when the threshold is chosen to preserve 95% ID recall. However, the degree of separation varies noticeably between scores. Mahalanobis and Energy typically produce a clearer shift between ID and OOD, and the updated ViM implementation also exhibits a more consistent separation, with ID scores shifted toward higher values and OOD scores concentrated at lower values. This behavior matches the intended design of ViM, which penalizes samples whose features

have a large residual component outside the principal subspace fitted on ID data.

Among the compared approaches, Mahalanobis and Energy show the most pronounced separation. The ID distribution is shifted toward higher scores while the OOD distribution concentrates at lower values, yielding both a stronger rank ordering, which increases AUROC, and improved precision recall behavior, which increases AUPR. MSP and MaxLogit behave similarly, which is expected since both are confidence-based scores derived from the same logits. The main difference is that MSP applies a softmax normalization, which can compress large logit magnitudes and reduce contrast in highly confident regions.

In contrast, ViM exhibits the weakest separation, with a broad overlap between ID and OOD. In our implementation, ViM penalizes the maximum logit by a residual norm outside a classifier-induced subspace. If the residual magnitude is not reliably smaller for ID than for OOD, or if it correlates weakly with semantic mismatch for this dataset pair, the penalty can blur the confidence signal and lead to poor ranking performance and a degraded AUPR.

2.3 Quantitative results

Table 1 highlights different trade-offs across scoring rules. Mahalanobis achieves the best AUROC and the best FPR@95TPR among the compared methods, which suggests that distances to class-conditional feature statistics provide a relatively robust separation between ID and OOD. Energy and MaxLogit are close behind and behave similarly to MSP, which is expected since all three are derived from logits and mainly capture model confidence.

VIM9 yields the best AUPR, which indicates improved precision over a wide range of recall values when ranking samples by score. However, ViM also produces a very high FPR@95TPR. This means that when the threshold is set to keep 95% of ID samples, a large fraction of OOD samples still remains above the threshold. This behavior is consistent with Figure 2e, where the two distributions are shifted but still overlap significantly in the high-score region, leading to a heavy OOD tail that dominates the FPR@95TPR metric.

Table 1: OOD detection metrics using SVHN as OOD (higher score indicates more ID).

Method	AUROC (↑)	AUPR (↑)	FPR@95 (↓)
MSP	0.7486	0.6047	0.8458
MaxLogit	0.7525	0.6074	0.8453
Energy	0.7581	0.6111	0.8359
Mahalanobis	0.7707	0.6188	0.8189
ViM	0.7230	0.6310	0.9385

3 Point 3 — Neural Collapse at the end of training (NC1–NC4)

3.1 Motivation and intuition

Neural Collapse refrs to a set of geometric regularities observed in the termnal phase of training of deep classifiers. The main idea is that, as training converges, th representation geometry becomes highly structured. Samples from the same class concentrate around a single class center, which corresponds to NC1. Class centers become maximally symmetric, which corresponds to NC2. Classifier weigths align with class centers, which corresponds to NC3. The classifier behaves similarly to a nearest class center rule in feature space, which corresponds to NC4.

In our experiments, NC metrics are computed on the penultimate-layer features of the ID training set. Class means are computed per class, and the classifier weights are taken from the final fully connected layer.

3.2 Definitions as measured in our implementation

Let $f(x) \in \mathbb{R}^d$ denote penultimate features and let μ_c be the mean feature vector for class c. Let $\bar{\mu}$ be the global mean across classes, and let $W \in \mathbb{R}^{C \times d}$ be the classifier weight matrix with rows w_c.

NC1 within-class variability collapse. We compute the within-class variance as the average squared distance betwen each feature and its class mean. NC1 is reported as the ratio

$$\text{NC1} = \frac{\text{within\_var}}{\text{between\_var}}.$$

Smaller values indicate that features are tightly concentrated around class means compared to the separation between class means.

NC2 simplex ETF structure of class means. NC2 states that normalized class means tend to form an equiangular configuration, also known as a simplex ETF. We normalize class means and compute their cosine Gram matrix. NC2 is summarized by the mean and standard deviation of off-diagonal cosine similarities. In the ideal ETF limit, off-diagonal cosines concentrate around $-1/(C-1)$.

NC3 self-duality alignment between classifier weights and class means. NC3 captures the alignment between classifier weights and class means. We normalize both w_c and μ_c and compute $\cos(w_c, \mu_c)$ for each class. NC3 is summarized by the mean and standard deviation across classes. Values close to 1 indicate strong alignment.

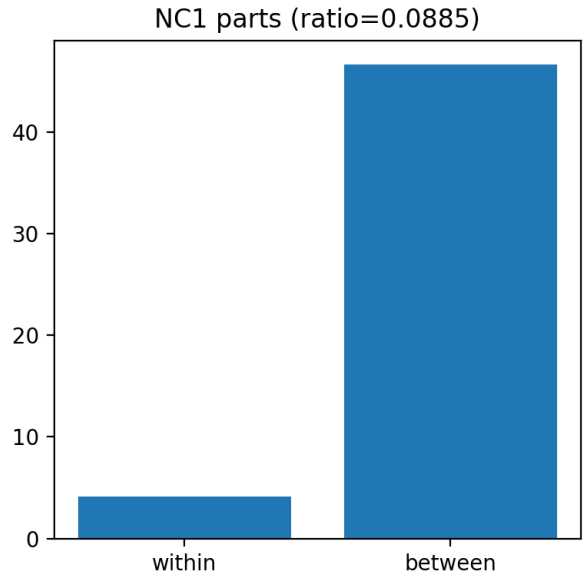
NC4 classifier symmetry and nearest class center behavior. NC4 reflects the symmetry of the classifier geometry and its consistency with a nearest class center rule. We compute the cosine Gram matrix of normalized classifier weights and summarize the off-diagonal entries by their mean and standard deviation. As with NC2, the ETF-like regime corresponds to off-diagonal values concentrating around $-1/(C-1)$.

3.3 Results and supporting plots

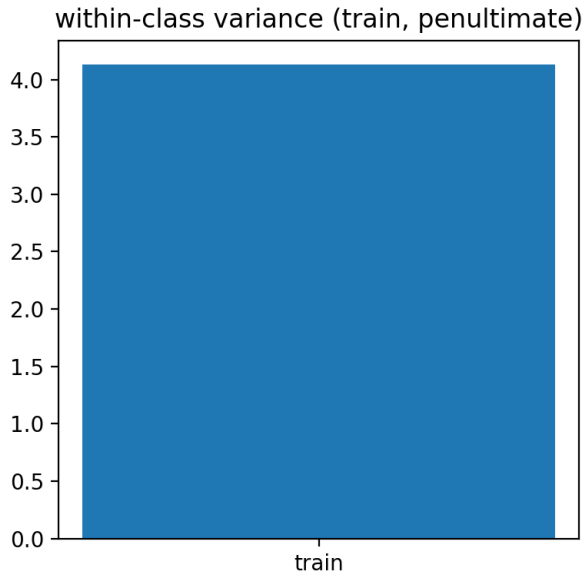
At convergence, using penultimate features on the ID training set, we obtain

$$\begin{aligned} \text{NC1} &= 0.08848, \\ \text{NC2 mean/std} &= -0.01007/0.02973, \\ \text{NC3 mean/std} &= 0.89644/0.01508, \\ \text{NC4 mean/std} &= -0.01008/0.01735. \end{aligned}$$

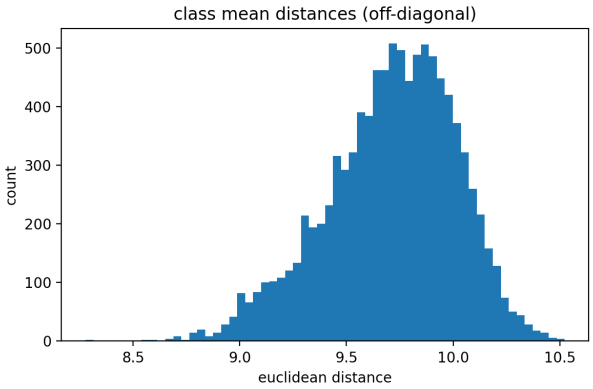
Figures 3–4 provide visual support. NC1 is illustrated by comparing within-class and between-class variance. NC2 is illustrated by the distribution of pairwise distances and the off-diagonal cosine similarity distribution between class means. NC3 is illustrated by the distribution of $\cos(w_c, \mu_c)$ across classes. NC4 is illustrated by the off-diagonal cosine similarity distribution between classifier weights.



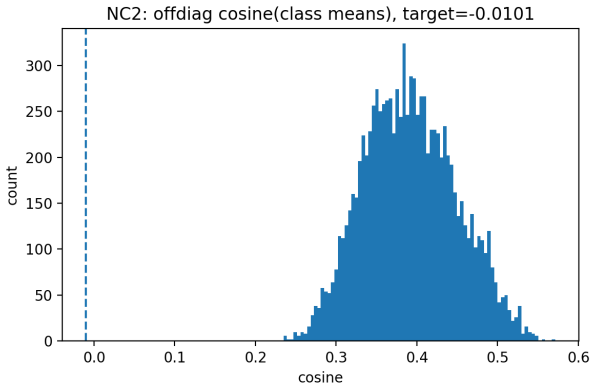
(a) NC1 within-class and between-class variability



(b) Within-class variance on training penultimate features

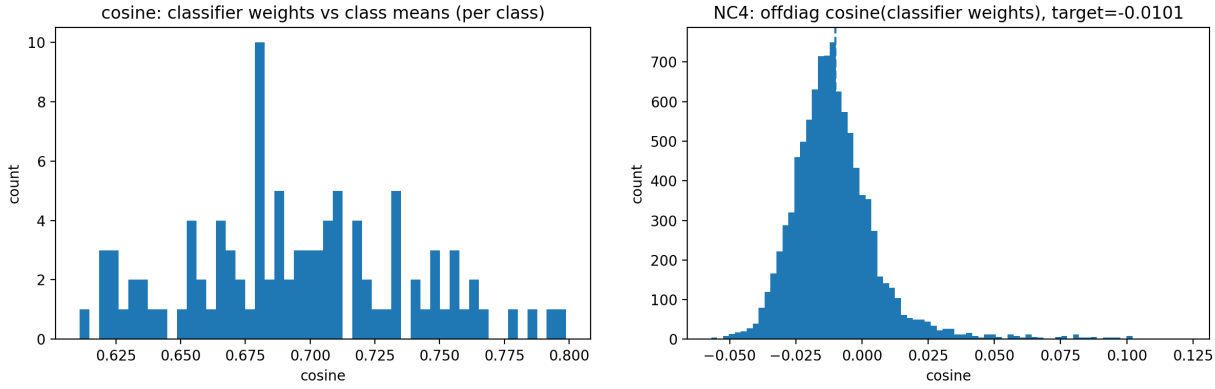


(c) Pairwise class mean distances, off-diagonal entries



(d) NC2 off-diagonal cosine similarities between class means

Figure 3: Neural Collapse evidence for NC1 and NC2



(a) NC3 cosine similarities between w_c and μ_c across classes

(b) NC4 off-diagonal cosine similarities between classifier weights

Figure 4: Neural Collapse evidence for NC3 and NC4

3.4 Short discussion

NC1 is NC1 = 0.08848, which means that points from the same class are much closer to each other than class centers are between themselves. This is easy to see in Figure 3, because the between-class bar is much larger than the within-class bar. The ratio is not close to zero, so features do not collapse to a single point per class. This is normal in practice, since the network is finite and training is not perfectly ideal.

NC2 gives the clearest sign of Neural Collapse. For $C = 100$, the ETF target is about -0.0101, and our NC2 mean is -0.01007, which is almost the same. In Figure 3, the histogram of off-diagonal cosines is centered near the target line. This means that, on average, class means are arranged in a very symmetric way, with similar angles between different classes. The histogram is not a sharp spike and it has a positive tail. This shows that some pairs of classes are still closer than expected in the ideal case, which can happen because some classes are naturally more similar and because we have finite data.

The histogram of distances between class means has one main peak, but it is not very narrow. This means class centers are not all at exactly the same distance from each other, even if their average angle pattern is close to the ETF target. This is expected because ETF mainly describes angles, while distances can vary when feature norms and remaining variability are not perfectly uniform.

NC3 shows a strong alignment between the classifier weights and the class means. The mean cosine is 0.89644 and the spread is small, so most classes behave similarly. The values are below 1, which means the alignment is very good but not perfect, which is again normal in a realistic setting.

NC4 also matches the ETF target, with mean -0.01008. In Figure 4, the weight cosine histogram is concentrated near the target line, which means the classifier weights are also arranged in a very symmetric way. The remaining spread and the positive tail show that the symmetry is not perfect for every pair of classes, which is consistent with small differences in how classes are separated.

4 Part IV — NC5

Neural Collapse properties NC1–NC4 describe the terminal geometry of the last hidden features on in-distribution (ID) data. In this part, we study an additional NC5 indicator at the end of training, based on the Gram matrix of the centered class means. The intuition is that, in the Neural Collapse regime, class means become almost equiangular: after centering and normalization, their pairwise inner products should be close to a constant negative value (the ETF target). Therefore, NC5 can be studied by comparing the off-diagonal entries of this Gram matrix to the theoretical target and by analyzing the spectrum (eigenvalues) of the Gram matrix.

4.1 Definition and measurement protocol

Let $f(x) \in \mathbb{R}^d$ denote the penultimate feature vector of the trained network. For each class c, we compute the empirical class mean

$$\mu_c = \frac{1}{N_c} \sum_{i:y_i=c} f(x_i),$$

using the ID training data.

We then compute the global mean of all class means,

$$\bar{\mu} = \frac{1}{C} \sum_{c=1}^{C} \mu_c,$$

and center each class mean as $\tilde{\mu}_c = \mu_c - \bar{\mu}$. Each centered mean is normalized to unit norm:

$$\hat{\mu}_c = \frac{\tilde{\mu}_c}{\|\tilde{\mu}_c\|}.$$

We build the Gram matrix $G \in \mathbb{R}^{C \times C}$ defined by

$$G_{ij} = \hat{\mu}_i^\top \hat{\mu}_j.$$

In the ideal Neural Collapse regime, the class means form a Simplex Equiangular Tight Frame (ETF). In that case, the diagonal entries satisfy $G_{ii} = 1$, and the off-diagonal entries satisfy

$$G_{ij} = -\frac{1}{C-1}, \quad i \neq j.$$

To quantify how close we are to this ideal structure, we compare the off-diagonal entries of G to the theoretical target value and report: (i) the mean absolute deviation from the target, and (ii) the Frobenius norm of the deviation matrix. We also analyze the eigenvalues of G to study its spectral structure.

4.2 Results

The theoretical ETF off-diagonal target for $C = 100$ classes is

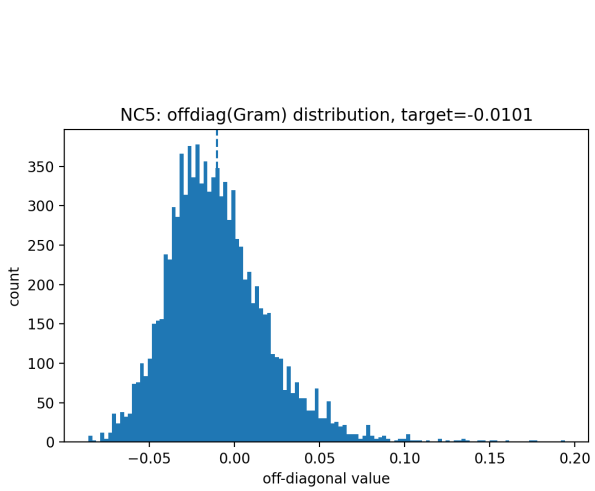
$$-\frac{1}{C-1} = -0.01010.$$

From our implementation, we obtain:

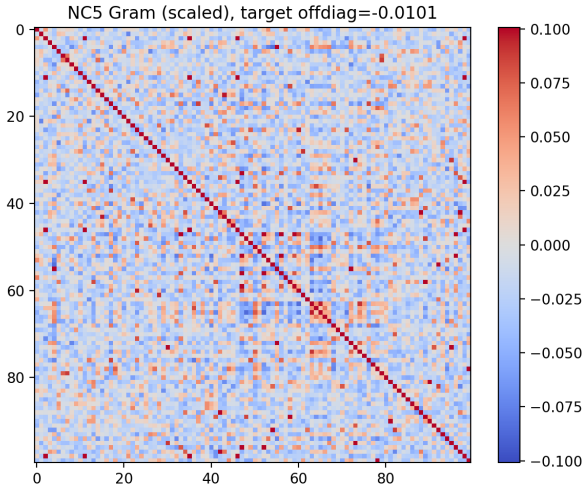
$$\text{NC5 target} = -0.01010, \quad \text{abs dev mean} = 0.02260, \quad \text{fro dev} = 0.02958.$$

The mean absolute deviation from the target is relatively small compared to the magnitude of the entries, which indicates that the class means are close to an equiangular configuration. The Frobenius deviation is also low, suggesting that the overall Gram matrix is close to the ideal ETF structure.

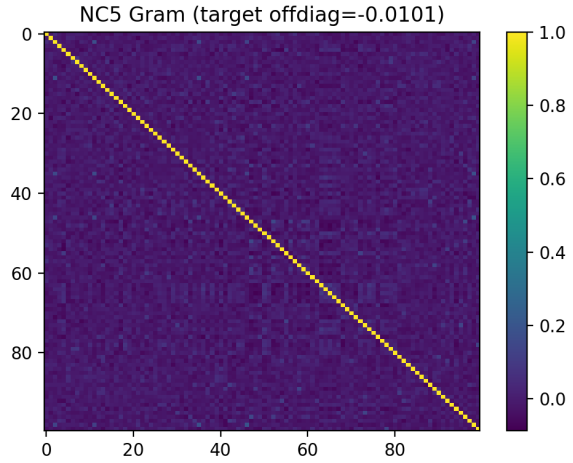
Figure 5 provides additional insight. The histogram of off-diagonal entries shows that most values concentrate around the theoretical target, with moderate dispersion. The scaled Gram matrix presents a strong diagonal structure with off-diagonal values centered around small negative numbers. Finally, the eigenvalue distribution indicates a structured spectrum, consistent with a near-simplex geometry rather than a random configuration.



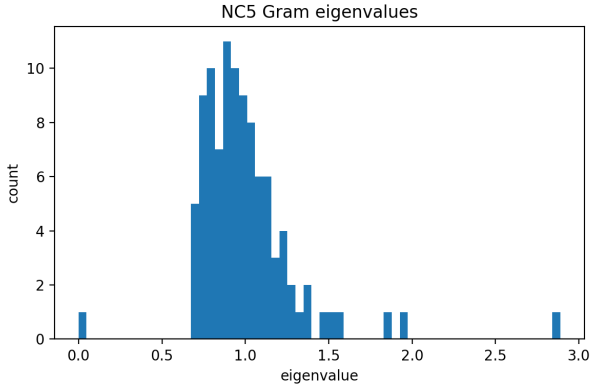
(a) NC5 off-diagonal histogram



(b) Scaled Gram matrix



(c) Gram matrix



(d) Eigenvalues

Figure 5: NC5 visualizations.

4.3 Analysis

Overall, the results suggest that the network is close to the Neural Collapse regime at the end of training. The off-diagonal entries of the Gram matrix are centered around the theoretical ETF target, which indicates that the class means are approximately equiangular after centering and normalization. This behavior is consistent with the expected simplex structure described in Neural Collapse theory.

However, the deviations are not exactly zero. This is normal in practice, since we work with finite data, stochastic optimization, and a model that may not be perfectly converged. The small dispersion observed in the histogram reflects this realistic setting. The eigenvalue distribution also confirms that the representation is highly structured, but not perfectly ideal.

These observations show that the penultimate features have reached a geometrically organized configuration. This geometric structure will be important in the next section, where we use it to design an OOD detection score inspired by Neural Collapse.

5 Part V — NECO

After observing in Part IV that the penultimate features exhibit a structured Neural Collapse geometry, we now investigate whether this geometric structure can be used for OOD detection. The main idea behind NECO (Neural Collapse Inspired OOD) is to exploit the low-dimensional organization of ID features. If ID features concentrate in a specific structured subspace, then OOD features are expected to deviate from this configuration. Therefore, measuring how much a feature vector aligns with the ID subspace can provide a useful OOD score.

5.1 Algorithm (as implemented)

We compute the NECO score using penultimate features and the model logits. The procedure is:

1. Fit NECO parameters on ID training features: We estimate the ID feature mean μ and fit PCA on centered ID penultimate features. We keep the PCA directions V_d that define the ID subspace.

2. Center and project test features: For a test sample with penultimate feature $f(x)$, we center it as $X = f(x) - \mu$ and project it onto the PCA subspace: $\text{proj} = XV_d$.

3. Compute a normalized projection ratio: We compute

$$r(x) = \frac{\|\text{proj}\|_2}{\|X\|_2 + \varepsilon},$$

   which measures how much of the feature energy lies inside the ID subspace.

4. Fuse with confidence (MaxLogit): We compute the maximum logit $\max_c z_c(x)$ and define the final score as

$$s_{\text{NECO}}(x) = r(x) \max_c z_c(x).$$

In our experiments, higher NECO scores indicate samples that are more likely to be in-distribution.

5.2 Geometric intuition

NECO is motivated by the fact that, near the Neural Collapse regime, ID penultimate features become organized in a structured way and concentrate around a low-dimensional configuration. PCA provides a simple way to capture this dominant ID subspace. For an ID sample, a large part of its centered feature vector should lie inside this subspace, which leads to a high projection ratio $r(x)$. In contrast, OOD samples are expected to have feature directions that do not match the ID geometry, so their energy is less aligned with the PCA subspace and the ratio becomes smaller. Multiplying by MaxLogit further penalizes samples that also receive low model confidence. This explains why the NECO score tends to be higher for ID and lower for OOD, as illustrated by the score histograms.

This behavior is visible in Figure 6, where the ID scores are shifted to larger values compared to OOD.

5.3 Empirical evaluation

We evaluate NECO on CIFAR100 (ID) versus SVHN (OOD). Higher scores indicate that a sample is more likely to be in-distribution.

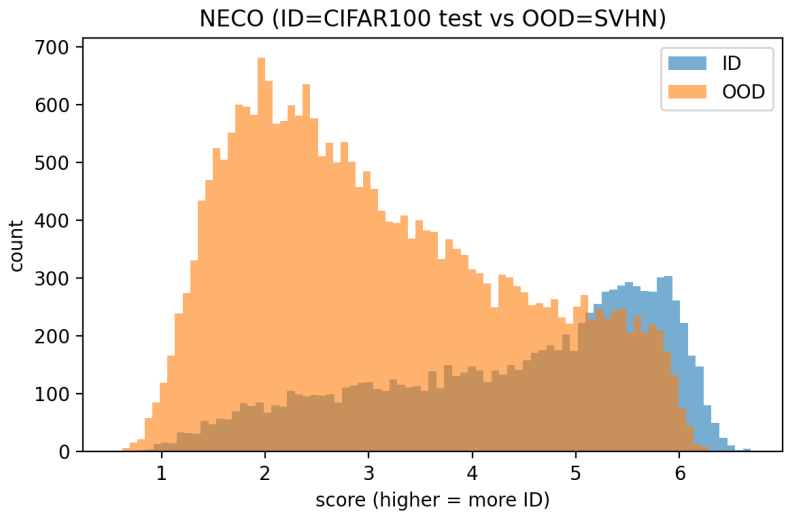


Figure 6: Distribution of NECO scores for ID (CIFAR100 test) and OOD (SVHN).

Figure 6 shows a visible shift between ID and OOD score distributions. ID samples tend to produce higher NECO scores, while OOD samples are concentrated at lower values. Although there is some overlap, the separation is consistent with the geometric intuition described previously.

Quantitatively, NECO achieves an AUROC of 0.7461, an AUPR of 0.5545, and an FPR@95 of 0.8184. Its performance is comparable to MSP and MaxLogit, and slightly below Mahalanobis in AUROC. However, NECO improves over MSP in terms of FPR@95 and clearly outperforms ViM in this setting. These results indicate that exploiting Neural Collapse geometry provides a competitive OOD detection signal, even without additional training.

6 Bonus — Neural Collapse Across Layers

In previous sections, Neural Collapse was analyzed using penultimate features, where the effect is expected to be strongest at the end of training. In this bonus experiment, we extend the analysis to intermediate layers of the network in order to study how Neural Collapse metrics evolve from early to deep representations. The goal is to understand whether the collapse behavior appears gradually through the network or only emerges in the final layers.

6.1 Protocol

We study how Neural Collapse metrics evolve across the depth of the network by analyzing the representations of `layer1`, `layer2`, `layer3`, and `layer4`. For each layer, we apply the same pipeline:

- **Feature extraction:** We record the layer output with a forward hook and apply global average pooling to obtain a feature vector for each sample.
- **Class statistics:** Using ID training data, we compute the class means and the within-class variance for that layer representation.
- **NC1 and NC2:** We compute NC1 as the ratio *within-class variance / between-class variance*. We compute NC2 as the mean off-diagonal cosine similarity between class means.

- **Linear probe for NC3 and NC4:** We fit a linear classifier on the layer features using least-squares regression (with a small ridge regularization). Then we compute NC3 as the mean cosine similarity between the probe weights and the class means, and NC4 as the mean off-diagonal cosine similarity between probe weight vectors.

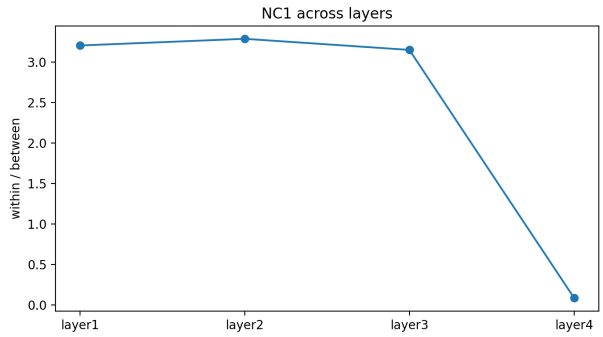
This protocol provides a consistent comparison of NC metrics from early to deep layers.

6.2 Results

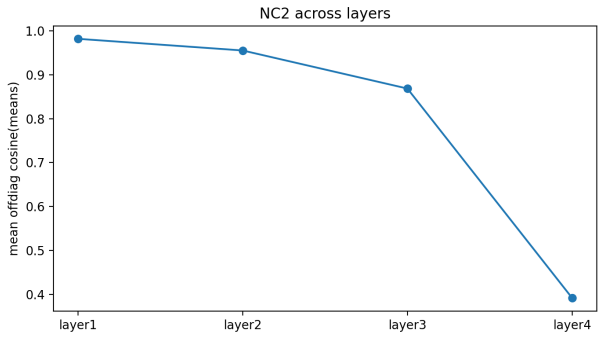
Figure 7 summarizes the evolution of NC metrics across `layer1` to `layer4`. The main trends are:

- NC1 is very large in early layers (about 3.2 in `layer1--2`), and then drops sharply in `layer4` (about 0.09). This indicates that strong collapse mainly appears in the deepest representation.
- NC3 increases with depth (from near 0 in `layer1` to about 0.47 in `layer4`), meaning that the linear probe weights become progressively more aligned with the class means.
- NC4 moves closer to the ETF target $-1/(C-1) \approx -0.0101$ as depth increases, with the closest value observed in `layer4`.
- NC2 also changes significantly across layers: early layers show high off-diagonal cosine values, while deeper layers show a more structured configuration consistent with Neural Collapse behavior.

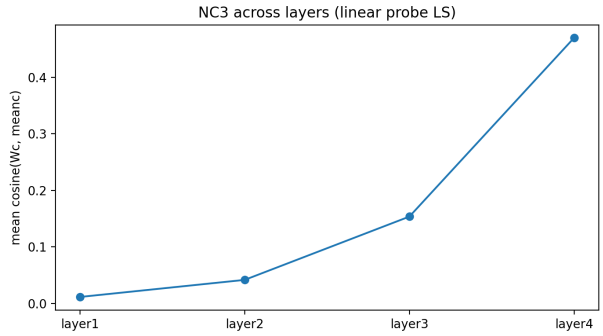
Overall, the results suggest that Neural Collapse is not present in early layers and becomes most pronounced in the last block of the network.



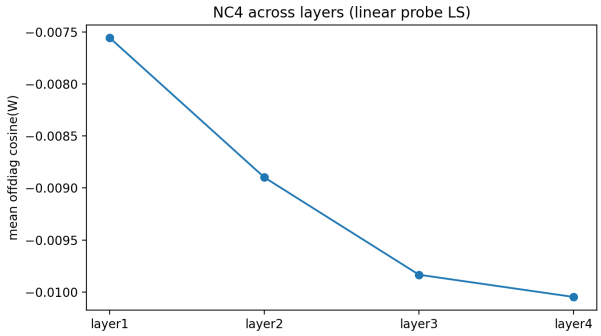
(a) NC1 across layers



(b) NC2 across layers



(c) NC3 across layers



(d) NC4 across layers

Figure 7: Bonus: NC metrics across intermediate layers.

6.3 Analysis

The progressive behavior observed across layers is consistent with the theoretical understanding of Neural Collapse. Early layers mainly extract low-level visual features, and therefore class representations are still highly mixed and not well separated. This explains the large NC1 values and the weak alignment between classifier weights and class means.

As depth increases, representations become more class-specific. The network gradually reduces within-class variance and organizes class means into a more structured geometric configuration. In the final layer, the strong decrease of NC1 and the improved alignment (NC3 and NC4) indicate that the representation has entered a regime close to Neural Collapse.

These results support the idea that Neural Collapse is a late-stage phenomenon that emerges in deep layers after sufficient training, rather than being present uniformly throughout the network.