Supplementary Materials

A. Supplementary to the Analysis of Hidden Classifier

A.1. Proofs for the properties of hidden classifier

Notation (detailed) Each hidden layer feature $\mathbf{a}^{(l)}$ is defined by consecutive computation of the post-activated feature vector

$$\mathbf{a}^{(l)} = \sigma(\mathbf{W}^{(l)T}\mathbf{a}^{(l-1)}) \tag{11}$$

from the input layer $l = 0$ to the last hidden layer $l = L$. The pre-activated features satisfy $\mathbf{a}^{(l)} = \sigma(\mathbf{z}^{(l)})$, where the activation function σ is a rectifier (*e.g.* ReLU, GeLU, Leaky ReLU). The penultimate embedding is $g(\mathbf{x}) = \mathbf{U}^T\mathbf{a}^{(L)}$, which computes the network classification logit $\psi(\mathbf{x}) \in \mathbb{R}^K$ by

$$\psi(\mathbf{x}) = \mathbf{W}^T g(\mathbf{x}). \tag{12}$$

$\mathbf{W}$ is the weight matrix for the classification layer. For notation simplicity, let $\mathbf{W}^{(L+1)} := \mathbf{U}\mathbf{W}$ such that $\psi(\mathbf{x}) = \mathbf{W}^{(L+1)}\mathbf{a}^{(L)}$. The sign function $\operatorname{sign}(\cdot)$, on the other hand, that binarizes a scalar to either 1 or -1 is applied point-wise.

Note For the embedding computation, $\mathbf{U}$ is a fixed identity matrix in supervised models, while $\mathbf{U}$ serves as a learnable parameters for self-supervised models with projection head [5].

Proposition 1. *The final logit is represented by*

$$\psi(\mathbf{x}) = \mathbf{C}^{(l)}\mathbf{a}^{(l)} \tag{13}$$

for each hidden layer l, where

$$\mathbf{C}^{(l)} = \left(\prod_{k=0}^{L-l-1} \mathbf{W}^{(L+1-k)T}\mathbf{D}^{(L-k)}\right)\mathbf{W}^{(l+1)T} \tag{14}$$

with $\mathbf{D}^{(l)} = \operatorname{diag}(\frac{\sigma(z_1)}{z_1}, \ldots, \frac{\sigma(z_{d_l})}{z_{d_l}})$ *with the convention* $\frac{\cdot}{0} = 0$. $\mathbf{C}^{(l)} = \mathbf{C}^{(l)}(\mathbf{x}) \in \mathbb{R}^{K\times d_l}$ *depends on* $\mathbf{x}$.

Proof. Observe inductively that

$$\begin{aligned}
\psi(\mathbf{x}) &= \mathbf{W}^{(L+1)T}\mathbf{a}^{(L)} && (15)\\
&= \mathbf{W}^{(L+1)T}\mathbf{D}^{(L)}\mathbf{z}^{(L)} && (16)\\
&= \mathbf{W}^{(L+1)T}\mathbf{D}^{(L)}\mathbf{W}^{(L)T}\mathbf{a}^{(L-1)} && (17)\\
&= \mathbf{W}^{(L+1)T}\mathbf{D}^{(L)}\mathbf{W}^{(L)T}\mathbf{D}^{(L-1)}\mathbf{z}^{(L-1)} && (18)\\
&= \cdots, && (19)
\end{aligned}$$

obtaining

$$\psi(\mathbf{x}) = \left(\prod_{k=0}^{L-l-1} \mathbf{W}^{(L+1-k)T}\mathbf{D}^{(L-k)}\right)\mathbf{W}^{(l+1)T}\mathbf{a}^{(l)} \tag{20}$$

□

Remark. We note that both $\mathbf{D}^{(l)} = \mathbf{D}^{(l)}(\mathbf{x})$ and $\mathbf{C}^{(l)} = \mathbf{C}^{(l)}(\mathbf{x})$ depend on $\mathbf{x}$ as they depend on $\mathbf{a}^{(l)}$. Also, note that the dimension of $\mathbf{C}^{(l)}$ is $K \times d_l$.

Recall that $\mathbf{C}^{(l)} = [\mathbf{c}_1^{(l)}, \ldots, \mathbf{c}_K^{(l)}]^T$.

Proposition 2. *Let* $(\mathbf{x}, y)$ *be arbitrary labeled ID sample. Suppose that* $\psi_y(\mathbf{x})$ *is maximized in a manner to reduce the angle between* $\mathbf{c}_y^{(l)}$ *and* $\mathbf{a}^{(l)}$ *sufficiently that* $\operatorname{sign}(\mathbf{c}_y^{(l)}) = \operatorname{sign}(\mathbf{a}^{(l)})$. *Suppose that* $\psi_k(\mathbf{x})$ *is minimized in a manner to increase the angle between* $\mathbf{c}_k^{(l)}$ *and* $\mathbf{a}^{(l)}$ *sufficiently that* $\measuredangle(\operatorname{sign}(\mathbf{c}_k^{(l)}), \mathbf{a}^{(l)}) > \pi/2$. *Then,* $\overline{\psi}^{(l)}$ *becomes a discriminative classifier with* $\overline{\psi}_y^{(l)}(\mathbf{x}) > \overline{\psi}_k^{(l)}(\mathbf{x})$.

Proof. For notational simplicity, ignore the superscript index l, and let $\mathbf{a} = \mathbf{a}^{(l)}$, $\mathbf{b}_k = \mathbf{b}_k^{(l)}$, $\mathbf{c}_k = \mathbf{c}_k^{(l)}$, and $\overline{\psi} = \overline{\psi}^{(l)}$. First, observe $\mathbf{b}_y = \text{sign}(\mathbf{c}_y) = \text{sign}(\mathbf{a})$ implies $0 \leq \measuredangle(\mathbf{b}_y, \mathbf{a}) < \pi/2$. Therefore,

$$\overline{\psi}_y(\mathbf{x}) = \mathbf{b}_y \cdot \mathbf{a} = \|\mathbf{b}_y\|_2 \|\mathbf{a}\|_2 \cos(\measuredangle(\mathbf{b}_y, \mathbf{a})) > 0. \tag{21}$$

On the other hand, $\measuredangle(\text{sign}(\mathbf{c}_k), \mathbf{a}) > \pi/2$ means $\pi \geq \measuredangle(\mathbf{b}_k, \mathbf{a}) > \pi/2$ by the definition of $\mathbf{b}_k$ for $k \neq y$. Therefore,

$$\overline{\psi}_k(\mathbf{x}) = \mathbf{b}_k \cdot \mathbf{a} = \|\mathbf{b}_k\|_2 \|\mathbf{a}\|_2 \cos(\measuredangle(\mathbf{b}_k, \mathbf{a})) < 0. \tag{22}$$

Since $(\mathbf{x}, y)$ was arbitrary, we have proved the desired. □

The main message of Prop. 2 is that the discriminative optimization of the original classifier should be powerful enough to optimize the *angle* between the hidden layer feature and the binary weight. Then, in this case, the hidden classifier becomes discriminative.

Theorem 3. *Under the sufficient condition of Prop. 2, for any labeled ID sample* $(\mathbf{x}, y)$*,*

$$\|\mathbf{a}^{(l)}\|_1 \textit{ converges to } \overline{\psi}_y^{(l)}(\mathbf{x}) = \max_k \overline{\psi}_k^{(l)}(\mathbf{x}) \tag{23}$$

in which case $\text{sign}(\mathbf{a}^{(l)}) = \mathbf{b}_y^{(l)}$*. In general, for any* k *and for any sample* $\mathbf{x}$ *(either ID or OOD),*

$$0 \leq \|\mathbf{a}^{(l)}\|_1 - \overline{\psi}_k^{(l)}(\mathbf{x}) \leq \|\mathbf{a}^{(l)}\|_\infty \|\text{sign}(\mathbf{a}^{(l)}) - \mathbf{b}_k^{(l)}\|_1. \tag{24}$$

Proof. For notational simplicity, ignore the superscript index l, and let $\mathbf{a} = \mathbf{a}^{(l)}$, $\mathbf{b}_k = \mathbf{b}_k^{(l)}$, $\mathbf{c}_k = \mathbf{c}_k^{(l)}$, and $\overline{\psi} = \overline{\psi}^{(l)}$. First, observe that

$$\|\mathbf{a}\|_1 = \sum_i |a_i| \geq \sum_i b_{ki} a_i = \mathbf{b}_k \cdot \mathbf{a} = \overline{\psi}_k(\mathbf{x}) \tag{25}$$

where $\mathbf{b}_k = (b_{k1}, \ldots, b_{kd_l}) \in \{-1, 1\}^{d_l}$. This proves that $\|\mathbf{a}\|_1 \geq \overline{\psi}_k(\mathbf{x})$ for all k.

Now, observe that $|a_i| = \text{sign}(a_i) a_i$. Therefore,

$$\|\mathbf{a}\|_1 - \overline{\psi}_k(\mathbf{x}) = \sum_i (\text{sign}(a_i) - b_{ki}) a_i \leq \sum_i |\text{sign}(a_i) - b_{ki}||a_i| \leq \|\mathbf{a}\|_\infty \|\text{sign}(\mathbf{a}) - \mathbf{b}_k\|_1, \tag{26}$$

proving a general upper bound of the difference between the hidden classifier output and the feature norm.

Now, under the sufficient condition of Prop. 2, the binary weight becomes the activation pattern by the assumption; $\text{sign}(\mathbf{a}) = \mathbf{b}_y$. Therefore, in this case,

$$0 \leq \|\mathbf{a}\|_1 - \overline{\psi}_y(\mathbf{x}) \leq \|\mathbf{a}\|_\infty \cdot 0 = 0, \tag{27}$$

proving the desired. □

Corollary 4. *If* $\max_k \overline{\psi}_k^{(l)}(\mathbf{x}_{ood})$ *is sufficiently small such that*

$$\max_k \overline{\psi}_k^{(l)}(\mathbf{x}_{ood}) + \delta < \max_k \overline{\psi}_k^{(l)}(\mathbf{x}_{ind}) \tag{28}$$

for all ID samples $\mathbf{x}_{ind}$ *where*

$$\delta \geq \|\mathbf{a}^{(l)}(\mathbf{x}_{ood})\|_\infty \cdot \|\text{sign}(\mathbf{a}^{(l)}(\mathbf{x}_{ood})) - \mathbf{b}_{k_0}^{(l)}\|_1 \tag{29}$$

and $k_0 = \arg\max_k \overline{\psi}_k^{(l)}(\mathbf{x}_{ood})$*, then*

$$\|\mathbf{a}^{(l)}(\mathbf{x}_{ood})\|_1 < \|\mathbf{a}^{(l)}(\mathbf{x}_{ind})\|_1 \tag{30}$$

for all ID samples $\mathbf{x}_{ind}$*.*

Proof. By Thm. 3,

$$\|\mathbf{a}^{(l)}(\mathbf{x}_{ood})\|_1 \leq \overline{\psi}_{k_0}^{(l)}(\mathbf{x}_{ood}) + \|\mathbf{a}^{(l)}(\mathbf{x}_{ood})\|_\infty \|\text{sign}(\mathbf{a}^{(l)})(\mathbf{x}_{ood}) - \mathbf{b}_{k_0}^{(l)}\|_1 < \max_k \overline{\psi}_k^{(l)}(\mathbf{x}_{ind}) \leq \|\mathbf{a}^{(l)}(\mathbf{x}_{ind})\|_1. \tag{31}$$

□

A.2. Additional Theoretical Consideration

We present additional results of the theoretical analysis on the hidden classifier.

A.2.1 Relation to General l_p-norms

We have proved that l_1-norm can differentiate OOD from ID. This capability of l_1-norm extends to the general l_p-norm by Holder's inequality.

Theorem 5 (Holder's inequality)**.** *For* $0 < p \leq q < \infty$ *and* $\mathbf{x} \in \mathbb{R}^d$*,*

$$\|\mathbf{x}\|_q \leq \|\mathbf{x}\|_p \leq d^{1/p-1/q}\|\mathbf{x}\|_q. \tag{32}$$

Thus, for an activation vector $\mathbf{a}^{(l)} \in \mathbb{R}^{d_l}$ and for $p > 1$, we have

$$\|\mathbf{a}^{(l)}\|_p \leq \|\mathbf{a}^{(l)}\|_1 \leq d_l^{-1/p}\|\mathbf{a}^{(l)}\|_p \tag{33}$$

Therefore, if $\|\mathbf{a}^{(l)}\|_1$ is large or small, then $\|\mathbf{a}^{(l)}\|_p$ is also large or small, respectively. Thus, different l_p-norms have similar mechanisms for OOD detection. Note, however, that different l_p-norms have different priors on the computation of units in the activation vector. Accordingly, the OOD detection performance will vary depending on which l_p-norm is used.

A.2.2 Extension to Pre-Activation Layer

Extending the framework in Sec. 3 to the pre-activation layer feature vector $\mathbf{z}^{(l)}$ is trivial, where the pre-activation layer feature is the vector satisfying $\mathbf{a}^{(l)} = \sigma(\mathbf{z}^{(l)})$ with the activation function σ. Here, we provide the properties of the pre-activation layer that correspond to the ones given in Sec. 3.

Proposition 6. *The final logit is represented by*

$$\psi(\mathbf{x}) = \widehat{\mathbf{C}}^{(l)}\mathbf{z}^{(l)} \tag{34}$$

for each hidden layer l, where

$$\widehat{\mathbf{C}}^{(l)} = \left(\prod_{k=0}^{L-l} \mathbf{W}^{(L+1-k)T}\mathbf{D}^{(L-k)}\right) \tag{35}$$

with $\mathbf{D}^{(l)} = \mathrm{diag}(\frac{\sigma(z_1)}{z_1}, \ldots, \frac{\sigma(z_{d_l})}{z_{d_l}})$ *with the convention* $\frac{\cdot}{0} = 0$*.* $\widehat{\mathbf{C}}^{(l)} = \widehat{\mathbf{C}}^{(l)}(\mathbf{x}) \in \mathbb{R}^{K \times d_l}$ *depends on* $\mathbf{x}$*.*

Define a hidden classifier corresponding to $\mathbf{z}^{(l)}$ by

$$\widehat{\psi}(\mathbf{x}) := \mathrm{sign}(\widehat{C}^{(l)})\mathbf{z}^{(l)} = \widehat{\mathbf{B}}^{(l)}\mathbf{z}^{(l)} \tag{36}$$

where $\widehat{\mathbf{C}}^{(l)} = [\widehat{\mathbf{c}}_1^{(l)}, \ldots, \widehat{\mathbf{c}}_K^{(l)}]^T$ and $\widehat{\mathbf{B}}^{(l)} = [\widehat{\mathbf{b}}_1^{(l)}, \ldots, \widehat{\mathbf{b}}_K^{(l)}]^T$.

Proposition 7. *Let* $(\mathbf{x}, y)$ *be an arbitrary labeled sample. Suppose that* $\psi_y(\mathbf{x})$ *is maximized in a manner to reduce the angle between* $\widehat{\mathbf{c}}_y^{(l)}$ *and* $\mathbf{z}^{(l)}$ *sufficiently that* $\mathrm{sign}(\widehat{\mathbf{c}}_y^{(l)}) = \mathrm{sign}(\mathbf{z}^{(l)})$*. Suppose that* $\psi_k(\mathbf{x})$ *is minimized in a manner to increase the angle between* $\widehat{\mathbf{c}}_k^{(l)}$ *and* $\mathbf{z}^{(l)}$ *sufficiently that* $\measuredangle(\mathrm{sign}(\widehat{\mathbf{c}}_k^{(l)}), \mathbf{z}^{(l)}) > \pi/2$*. Then,* $\widehat{\psi}^{(l)}$ *becomes a discriminative classifier with* $\widehat{\psi}_y^{(l)}(\mathbf{x}) > \widehat{\psi}_k^{(l)}(\mathbf{x})$*.*

Theorem 8. *Under the sufficient condition of Prop. 7,*

$$\|\mathbf{z}^{(l)}\|_1 \textit{ converges to } \widehat{\psi}_y^{(l)}(\mathbf{x}) = \max_k \widehat{\psi}_k^{(l)}(\mathbf{x}) \tag{37}$$

in which case $\mathrm{sign}(\mathbf{z}^{(l)}) = \widehat{\mathbf{b}}_y^{(l)}$*. In general, for any k*

$$0 \leq \|\mathbf{z}^{(l)}\|_1 - \widehat{\psi}_k(\mathbf{x}) \leq \|\mathbf{z}^{(l)}\|_\infty \|\mathrm{sign}(\mathbf{z}^{(l)}) - \widehat{\mathbf{b}}_k^{(l)}\|_1 \tag{38}$$

A.2.3 On Bias

In Sec. 3, we ignored the bias in the computation of features for simplicity. We can preserve the properties of features given in Sec. 3 while including the bias terms. To observe this, consider

$$\mathbf{a}^{(l)} = \sigma(\mathbf{W}^{(l)T}\mathbf{a}^{(l-1)} + \boldsymbol{\beta}^{(l)}) = \mathbf{D}^{(l)}\mathbf{W}^{(l)T}\mathbf{a}^{(l-1)} + \mathbf{D}^{(l)}\boldsymbol{\beta}^{(l)}. \tag{39}$$

Thus, if Ψ denotes the logit computed with bias, then

$$\Psi(x) = \mathbf{C}^{(l)}\mathbf{a}^{(l)} + \sum_{j=l}^{L} \widehat{\mathbf{C}}^{(j+1)}\boldsymbol{\beta}^{(j+1)} = \psi(x) + \boldsymbol{\Gamma} \tag{40}$$

with $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}(l, \mathbf{x}) = \sum_{j=l}^{L} \widehat{\mathbf{C}}^{(j+1)}\boldsymbol{\beta}^{(j+1)}$ and the convention that $\widehat{\mathbf{C}}^{(L+1)} = \mathbf{I}$. Hence, if the discriminative learning of Ψ is not trivially achieved by the optimization of the bias term $\boldsymbol{\Gamma}$, and if the discriminative learning of ψ is thus sufficiently powerful, then the properties in Sec. 3 hold.

A.2.4 On Cosine Similarity Logit

We assumed that the classification logit is the output of the inner product in Sec. 3. Here, we show that changing the inner product logit by a (scaled) cosine similarity logit does not alter the major behavior of discriminative learning, and hence they are equivalent in our theoretical consideration. Thus, the theory developed in the inner-product logit also holds in the (scaled) cosine similarity logit.

To observe this, note that the scaled cosine similarity logit is defined as

$$\phi_k(\mathbf{x}) = \frac{1}{T}\frac{\mathbf{w}_k}{\|\mathbf{w}_k\|_2} \cdot \frac{g(\mathbf{x})}{\|g(\mathbf{x})\|_2} \tag{41}$$

where $\mathbf{w}_k$ are class weight vectors (prototypes) of trainable parameters and $g(\mathbf{x}) = \mathbf{U}^T\mathbf{a}^{(L)}$ with a matrix $\mathbf{U}$ of trainable parameters. T is the temperature that modifies the scale of similarity. Without loss of generality, we assume $T = 1$. Let $\psi_k(\mathbf{x}) = \mathbf{w}_k \cdot g(\mathbf{x})$ denote the inner-product logit that we originally used. Thus, we have

$$\phi_k(x) = \psi_k(x)(\|\mathbf{w}_k\|_2\|g(\mathbf{x})\|_2)^{-1}. \tag{42}$$

During discriminative learning, the model maximizes

$$(-1)^{1_{y\neq k}}\phi_k(\mathbf{x}) = (-1)^{1_{y\neq k}}\psi_k(\mathbf{x})(\|\mathbf{w}_k\|_2\|g(\mathbf{x})\|_2)^{-1}. \tag{43}$$

Assuming $\psi_y(\mathbf{x}) = \mathbf{w}_y \cdot g(\mathbf{x}) > 0$ and $\psi_k(\mathbf{x}) = \mathbf{w}_k \cdot g(\mathbf{x}) < 0$, the above maximization is equivalent to minimizing its negative log

$$-\log((-1)^{1_{y\neq k}}\phi_k(\mathbf{x})) = -\log\left((-1)^{1_{y\neq k}}\psi_k(\mathbf{x})\right) + \log\left(\|\mathbf{w}_k\|_2\|g(\mathbf{x})\|_2\right), \tag{44}$$

which can be considered as the constrained minimization of

$$-\log\left((-1)^{1_{y\neq k}}\psi_k(\mathbf{x})\right) \equiv -(-1)^{1_{y=k}}\psi_k(\mathbf{x}) \tag{45}$$

constraint to

$$\|\mathbf{w}_k\|_2\|g(\mathbf{x})\|_2 \leq e^{\eta_0} = \eta \tag{46}$$

for some η. Thus, optimization of the cosine similarity logit is equivalent to the constrained optimization of the inner product logit.

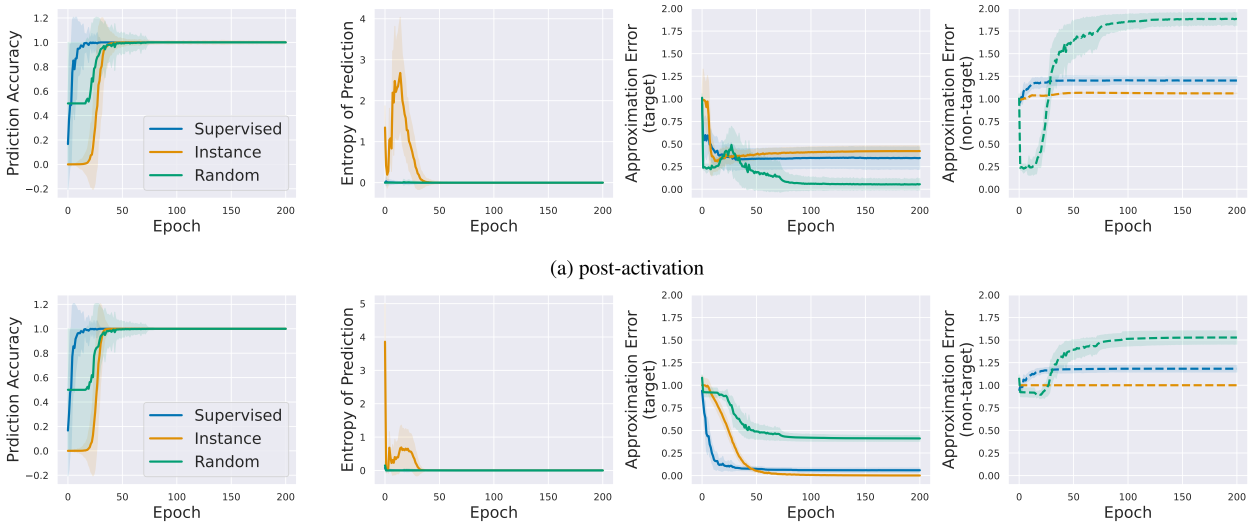
Proposition 9. *The maximization*

$$\max_{\phi} \quad (-1)^{1_{y\neq k}}\phi_k(\mathbf{x}) \tag{47}$$

is equivalent to

$$\begin{aligned} &\max_{\psi} && (-1)^{1_{y=k}}\psi_k(\mathbf{x}) \\ &\textit{subject to} && \|\mathbf{w}_k\|_2\|g(\mathbf{x})\|_2 \leq \eta \end{aligned} \tag{48}$$

for some $\eta > 0$ if $\psi_y > 0$ and $\psi_k < 0$.



(a) post-activation

(b) pre-activation

Figure 8: Results of hidden classifiers of ResNet-18 with different class labeling schemes on CIFAR-10. The approximation error on the target unit measures the normalized error $(\|\mathbf{a}\|_1 - \overline{\psi}_y(\mathbf{x}))/\|\mathbf{a}\|_1$, while the approximation error on the non-target unit is the average of $(\|\mathbf{a}\|_1 - \overline{\psi}_k(\mathbf{x}))/\|\mathbf{a}\|_1$ with respect to $k \neq y$. In the case of post-activation, the vector $\mathbf{a}$ is $\mathbf{a} = \mathbf{a}^{(L)}$. In the case of pre-activation, the vector $\mathbf{a}$ is $\mathbf{a} = \mathbf{z}^{(l)}$

A.3. Supplementary to empirical validation of hidden classifier

Here, we provide a detailed description of the experiments conducted to validate the theoretical analysis presented in Sec. 3.

A.3.1 On MLP

Setup We train an MLP with 5 hidden layers. The hidden layer dimension is fixed to 512, and likewise for the embedding layer dimension. The embedding is normalized, and the cosine similarity logit is divided by a temperature of 0.1. The model is trained by AdamW for 200 epochs with batch size 256. The learning rate decays from 0.001 to 0 by the cosine scheduler. Other setups follow the default setting in PyTorch.

Results The results are given in Fig. 12, 13, and 14. They have similar trends that we expected and thus verify our theoretical claims.

A.3.2 On Convolutional Network

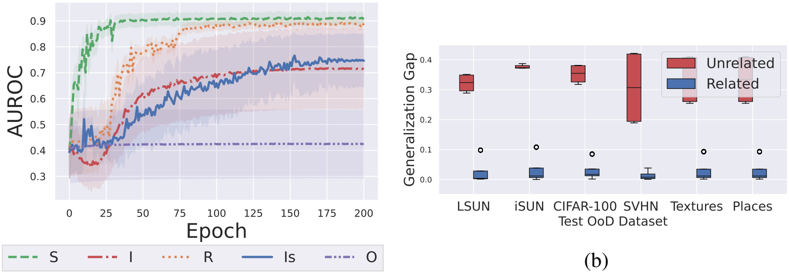
Setup The experiment setup is given as in Sec. B.

Results In the cases of both instance discrimination (I), supervised learning (S), and random binary label discrimination (R), the hidden classifier of the last hidden layer in ResNet-18 is trained to be discriminative (Fig. 8).

B. Supplementary to the Analysis of Feature Norm's Class Agnosticity

Setup. We train a ResNet-18 on CIFAR-10. We add an MLP projection head as in MoCo-v2 [5]. The embedding is normalized, and the cosine similarity logit is divided by a temperature of 0.1. The model is trained for 200 epochs and batch size 256 with the SGD optimizer, cosine learning rate (0.06 to 0), and momentum 0.9. Each model is trained in a different manner based on a different class labeling scheme:

- **S**: The class labels y_i are supervised labels (*e.g.* plane, dog, cat, ...). No data augmentation is applied.



(a)

(b)

Figure 9: (a) The detection performance of NAN versus the learning epoch across different types of training schemes (b) The generalization gap of NAN based on the intra-class semantics.

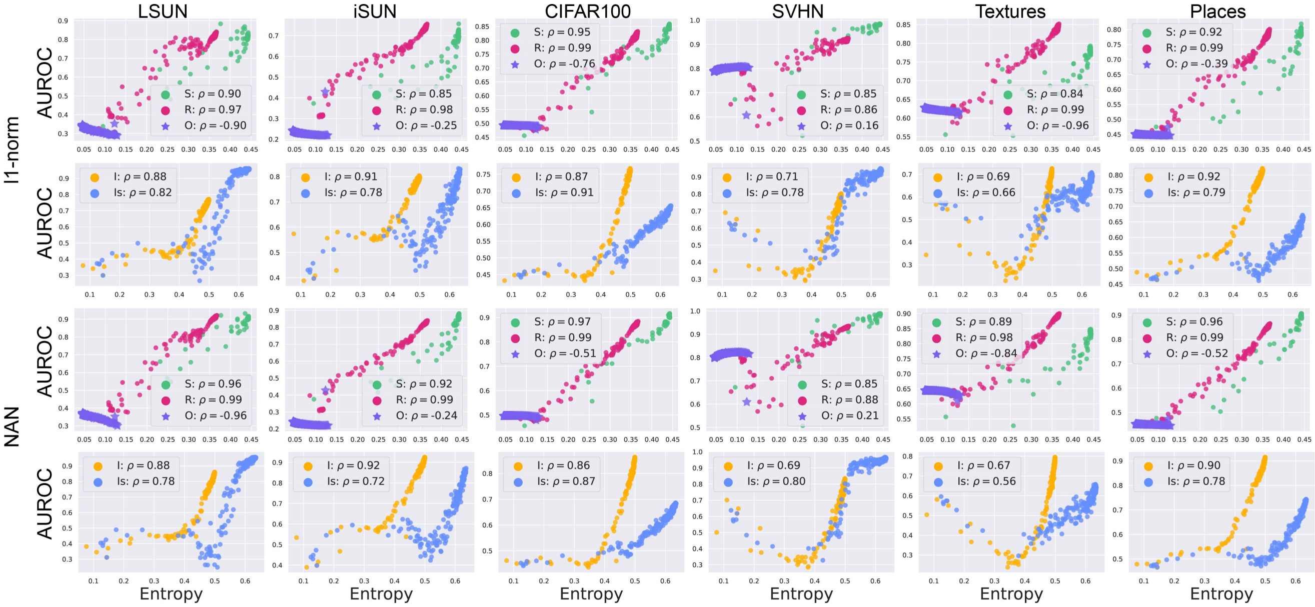


Figure 10: The graph of the detection performance versus the activation entropy. The performance is measured at every training epoch.

- **I**: The class labels y_i are instance labels $y_i = i$. No data augmentation is applied such that each instance class has only one intra-class sample.
- **Is**: The class labels y_i are instance labels $y_i = i$. Data augmentation is applied such that each instance class has multiple intra-class samples.
- **R**: The class labels y_i are labeled randomly by a binary number $y_i \in \{0, 1\}$.
- **O**: The class labels y_i are labeled with a single label $y_i = 0$ such that every sample is in the same class.

Other setups follow the default setting in PyTorch.

Full results on the impact of inter/intra-class learning The additional results on NAN is given in Fig. 9, which NAN exhibits the same trend of memorization and generalization as the conventional feature norm.

Full results on the relation to entropy The full results on the relation between the activation entropy and the detection performance is given in Fig. 10.

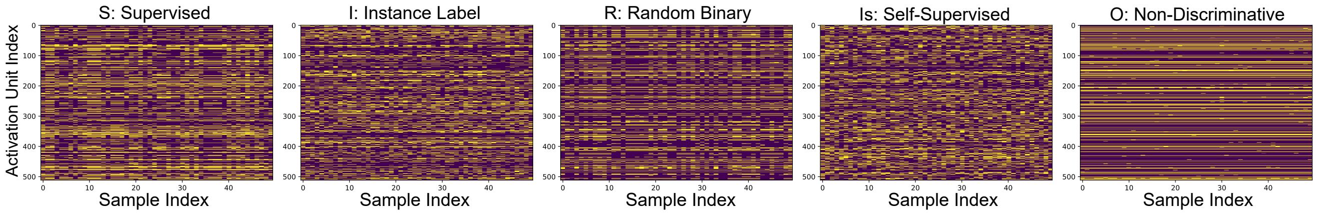


Figure 11: **Activation patterns** of randomly chosen 50 ID samples after training. Each column corresponds to the activation pattern $\text{sign}(\mathbf{a}^{(L)}){\in}\{-1,1\}^{512}$ of an ID sample. Discriminative training $\{$S,R,I,Is$\}$ results in *diverse* activation patterns, while the activation pattern *collapses* for the non-discriminative model O.

On the activation pattern If the model is trained in a non-discriminative manner with a single class, then the entropy of activation is diminished. In this case, the activation pattern collapses as shown in Fig. 11.

C. The Detailed Setup for the Experiments on NAN

C.1. Setup

Setup: ImageNet-1k For the supervised model trained by the cross entropy, we utilize the ResNet-50 backbone trained on ImageNet-1k. The model is provided by the PyTorch model zoo.

For the supervised model trained by the contrastive loss (thanks to the authors of [42]), we utilize the pretrained ResNet-50 model provided from the official GitHub page of KNN [42], which is trained on ImageNet-1k by the supervised contrastive loss [23] with the MLP projection head.

For the self-supervised contrastive model trained without the supervised labels of ID, thanks to the authors of MoCo-v2, we utilize the pretrained MoCo-v2 model provided from the official GitHub page of MoCo-v2 (the one with 71.1 accuracies on ImageNet-1k).

Setup: OOD CIFAR-10 For the evaluation results of OOD detection 'with supervised labels of ID' in Table 3, we train a cross-entropy model with supervised labels of CIFAR-10. The model has trained on CIFAR-10 over 800 epochs with the SGD optimizer and its momentum is 0.9. The learning rate decays to 0 from 0.03 by the cosine scheduler. The batch size is 512. The backbone is ResNet-18, accompanied by an MLP projection head on top of the encoder as in MoCo-v2. The embedding is normalized, and the cosine similarity logit is divided by the temperature 0.1.

For the evaluation results of OOD detection 'without supervised labels of ID' in Table 3, we train MoCo-v2 on CIFAR-10. The model is trained over 800 epochs with the SGD optimizer and its momentum 0.9. The batch size is 512. The learning rate is decayed by the cosine scheduler from 0.06 to 0. The model backbone is ResNet-18 combined with an MLP projection head. For the other configurations, we follow those given in the link[1]. After training the MoCo-v2 model, the NAN score is computed over multiple (9 overall) translated images of the test sample including the original image, and the scores are aggregated by average [43]. This aggregation technique is used exclusively for the model trained by MoCo-v2.

Setup: OOD CIFAR-10 The model training configuration for OCC is similar to that of label-free OOD detection on CIFAR-10 except that the train dataset is augmented randomly with 90-degree rotations. During the inference, the rotation is not used.

C.2. Score Fusion

A distance-based score $S_{dist}(\mathbf{x}) = d(X_{ind}, \mathbf{x})$ (*e.g.* KNN, SSD, or Mahalanobis) can be combined with NAN in a simple manner by

$$S_{dist+NAN}(\mathbf{x}) = d(X_{ind}, \mathbf{x}) / \|\mathbf{a}^{(L)}\|_{\text{NAN}}. \tag{49}$$

[1] https://colab.research.google.com/github/facebookresearch/moco/blob/colab-notebook/colab/moco_cifar10_demo.ipynb

			AUROC↑	FPR95↓
ID	Architecture	Last hidden layer $\mathbf{a}^{(L)}$	l_1-norm / NAN	l_1-norm / NAN
CIFAR-10	ResNet-18	average pool	93.27 / 93.56 (**+0.29**)	40.42 / 38.86 (**-1.56**)
	ResNet-18 + projection head	hidden layer in projection head	92.43 / 94.94 (**+2.51**)	43.02 / 30.08 (**-12.94**)
ImageNet-1k	ResNet-50	average pool	87.09 / 86.33 (**-0.76**)	44.67 / 46.56 (**+1.89**)
	ResNet-50 + projection head	hidden layer in projection head	57.99 / 92.32 (**+34.33**)	95.22 / 31.59 (**-63.63**)

Table 6: **Ablation of NAN with respect to the *projection head*.** The sparsity term in NAN is particularly effective when applied to the network architecture that contains the MLP projection head. Note that the l_1-norm here refers to the NAN score without the sparsity term. The reported performance here is obtained by averaging over all test OOD datasets.

	ReLU		Leaky ReLU		GeLU	
	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓
NAN w/o sparsity term (l_1-norm)	92.43	43.02	92.40	44.65	92.68	43.84
NAN	**94.94**	**30.08**	**94.92**	**30.56**	**94.05**	**35.02**

Table 7: **Ablation of NAN with respect to the *activation functions*** used in the last hidden layer. The ID data is CIFAR-10. The results indicate two aspects: (1) The performance of NAN is fairly robust with different choices of the activation function. (2) The sparsity term in NAN is always effective. The reported performance here is obtained by averaging over all test OOD datasets.

ID		ImageNet-1k			CIFAR-10		
	Formula	d	AUROC↑	FPR95↓	d	AUROC↑	FPR95↓
embedding magnitude	$\|\|g(\mathbf{x})\|\|_2$	128	84.09	72.85	128	93.00	43.40
NAN w/o sparsity term	$\|\|\mathbf{a}^{(L)}\|\|_1$	2048	57.99	95.22	512	92.40	43.00
NAN	$\|\|\mathbf{a}^{(L)}\|\|_{\text{NAN}}$	2048	**92.32**	**31.59**	512	**94.90**	**30.10**

Table 8: Comparison of NAN with the embedding magnitude. The embedding magnitude has been widely used in previous works. Here d indicates the dimension of the corresponding layer. The dimension of the embedding layer is often chosen small for effective training of the model. Due to its small layer dimension, the embedding magnitude may not fully capture the activation patterns, and hence can be sub-optimal. The reported performance here is obtained by averaging over all test OOD datasets.

D. Further Analysis on NAN

Setup We follow the same setup given in Sec. 6. When CIFAR-10 is the ID data, the test OOD datasets are LSUN-fix, ImageNet-fix, CIFAR-100, SVHN, and Places. When ImageNet-1k is the ID data, the test OOD datasets are iNaturalist, SUN, Places, and Texture.

D.1. Analysis on Projection Head

We analyze NAN with respect to **the projection head**. Table 6 indicates that NAN is more effective when it is applied to the hidden layer of the projection head rather than the average pooling layer.

NAN (*i.e.* particularly its sparsity term) becomes effective when the network learns to increase the number of deactivated units of ID samples (or have a relatively larger number of deactivated units for ID samples than OOD instances). Due to the entanglement of the feature map units in the average pooling layer, the network may not effectively increase the number of deactivated units in the average pooling layer. Hence, NAN can be sub-optimal for the average pooling layer.

D.2. Analysis on Activation Function

We evaluate NAN with **different activation functions**. We follow the same experimental protocol given in Sec. 6.3. We apply different activation functions in the hidden layer of the projection head. The results given in Table 7 shows that NAN is robust with respect to the choice of the activation function.

D.3. Comparison with Embedding Magnitude

For the sake of extensiveness, we compare NAN with the **embedding magnitude**. The embedding magnitude has been widely used in prior works for OOD detection-related tasks. The dimension of the embedding layer is often chosen to be a small number to avoid the curse of dimensionality during training. This may have a trade-off to OOD detection as the

OOD	iNaturalist		SUN		Places		Texture		Average		ID ACC
	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	
MSP	89.63	50.57	80.64	75.54	79.78	76.24	82.98	65.14	83.26	66.87	81.07
Energy	83.76	49.68	56.50	75.22	54.77	78.38	72.44	65.09	66.87	67.09	81.07
Mahalanobis	91.96	43.76	75.62	86.01	61.50	89.74	84.60	67.93	78.42	71.86	81.07
KNN	91.43	50.04	83.45	75.76	79.46	78.41	89.25	50.78	**85.90**	63.75	81.07
embedding magnitude	81.26	66.16	78.64	67.44	75.81	69.37	82.93	57.11	79.66	65.02	81.07
NAN w/o sparsity term (*i.e.* l_1-norm)	54.93	83.98	67.05	80.47	65.25	81.01	67.87	72.54	63.78	79.50	81.07
NAN	92.46	45.82	82.11	67.62	80.46	69.66	87.24	57.77	85.57	**60.22**	81.07

Table 9: **Results on ImageNet-1k (ID) with *ViT-B/16*.**

test OOD datasets		LSUN-fix		ImageNet-fix		CIFAR-100		SVHN		Places		Average	
Score	Formula	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓
hidden classifier confidence	$\max_k \overline{\psi}_k^{(L)}(\mathbf{x})$	95.06	33.35	94.54	35.92	92.17	45.10	94.66	39.91	94.66	30.15	94.22	36.89

Table 10: Results on CIFAR-10 (ID) with ResNet-18. **The hidden classifier confidence** is evaluated as a score function for OOD detection. The results shows that the hidden classifier confidence is capable of OOD detection.

embedding of a small dimension may not capture diverse activation patterns of embedding layer units and therefore its norm may not effectively differentiate OOD from ID. This hypothesis seems consistent to the results given in Table 8.

D.4. Evaluation of NAN on ViT

We evaluate NAN on the **vision transformer ViT**. We utilize ViT-B/16 pretrained on ImageNet-1k, which can be downloaded from PyTorch[2]. Analogous to the observations in Sec. D.1, direct usage of NAN on the pretrained ViT can be sub-optimal because the class token output of ViT is the LayerNorm layer, which can cancel out the norm information therein. Therefore, we add an MLP projection head on top of the pretrained ViT, and fine-tune the projection head while freezing the pretrained ViT backbone. The MLP projection head consists of a single hidden layer whose dimension is 786 and its activation function is ReLU. The embedding of the projection head is normalized and divided by the temperature 0.2, and trained by the cross entropy with 10 epochs under SGD, using the learning rate 0.03 that decays to 0 by the cosine scheduler.

For comparison, the KNN and Mahalanobis scores are applied on the original class token output of the pretrained ViT, and hence are independent of the projection head fine-tuning. Other OOD detection scores (MSP, Energy, and embedding magnitude) are applied to the fine-tuned classifier of the projection head. NAN utilizes the hidden layer in the projection head as this layer is the last hidden layer that involves the activation function computation.

Table 9 shows that NAN is effective for the ViT network as well. In addition, NAN is comparable to the state-of-the-art OOD detection scores.

Note on the ViT performance of KNN Note that the performance of KNN in Table 9 is lower than that of KNN reported in [42]. This is because the KNN we implemented is applied on ViT pretrained on ImageNet-1k, while the KNN reported in [42] is applied on ViT pretrained on ImageNet-21k.

D.5. Evaluation of Hidden Classifier for OOD Detection

We evaluate the **hidden classifier for OOD detection**. NAN's numerator is the l_1-norm of the activation vector, which we proved is a confidence value of the hidden classifier. We test this numerator component by testing the OOD detection capability of this hidden classifier confidence. Table 10 shows the hidden classifier confidence is capable of OOD detection.

D.6. Evaluation of NAN on CIDER

CIDER [29] is a training framework that is particularly effective for the KNN score. We evaluate NAN's compatibility to the KNN score from the model trained by CIDER. The results shown in Table 11 indicates that NAN can effectively enhance the KNN score of CIDER.

	SVHN		Places365		iSUN		Texture		LSUN		Average	
	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
NAN	73.82	90.46	26.33	94.65	25.47	96.46	25.35	95.21	1.17	99.45	30.43	95.25
KNN	4.44	99.36	37.88	92.97	22.94	96.16	17.27	97.15	9.85	98.21	18.48	96.77
NAN+KNN	5.70	98.62	21.79	95.32	14.01	97.64	16.21	96.61	0.95	99.68	**11.73**	**97.57**

Table 11: The results of the OOD detection scores (KNN, NAN, NAN+KNN) on the model trained by CIDER on CIFAR-10 (ID).

	iNaturalist		SUN		Places		Texture		ImageNet-O		OpenImage-O		Species		Average	
	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC	FPR95	AUROC
l_1-norm	97.52	52.06	95.58	59.40	95.65	61.30	92.11	59.21	88.20	67.97	92.43	63.10	95.83	59.42	93.90	60.35
$1/l_0$-norm	15.66	96.58	33.38	91.83	39.10	90.37	44.36	87.41	88.60	56.76	41.29	88.58	64.04	79.55	46.63	84.44
Residual	28.74	95.09	46.88	89.76	58.91	85.77	11.28	96.45	63.50	84.24	34.96	93.34	74.43	73.72	45.53	**88.34**
NAN	15.86	96.94	29.81	92.77	37.21	91.46	43.46	88.09	**87.95**	69.74	38.12	92.44	64.56	80.09	**45.28**	87.36
with ReAct:																
l_1-norm	98.07	37.19	96.37	46.97	96.90	45.47	85.44	61.21	84.95	74.80	93.54	54.48	98.81	41.25	93.44	51.62
$1/l_0$-norm	21.19	95.60	36.56	90.81	41.28	89.63	52.16	82.23	90.35	53.37	49.25	85.85	61.45	81.89	50.32	82.77
Residual	28.59	95.06	39.40	91.95	51.02	88.18	12.11	96.87	68.30	83.01	36.67	92.62	72.27	75.03	44.05	**88.96**
NAN	13.86	97.37	24.90	94.69	33.31	92.52	34.02	91.44	84.10	71.72	37.27	92.02	63.68	81.10	**41.59**	88.69

Table 12: The comparison of NAN with various forms of vector norms on ImageNet-1k (ID).

D.7. Comparison of NAN to various forms of vector norms

To further highlight the effectiveness of NAN, we compare NAN with various forms of vectors norms; namely, l_1-norm, the reciprocal of l_0-norm, and the residual of ViM which is the l_2-norm of the orthogonal projection. The experiment protocol follows [51], and the OOD datasets can be downloaded from its GitHub repository.

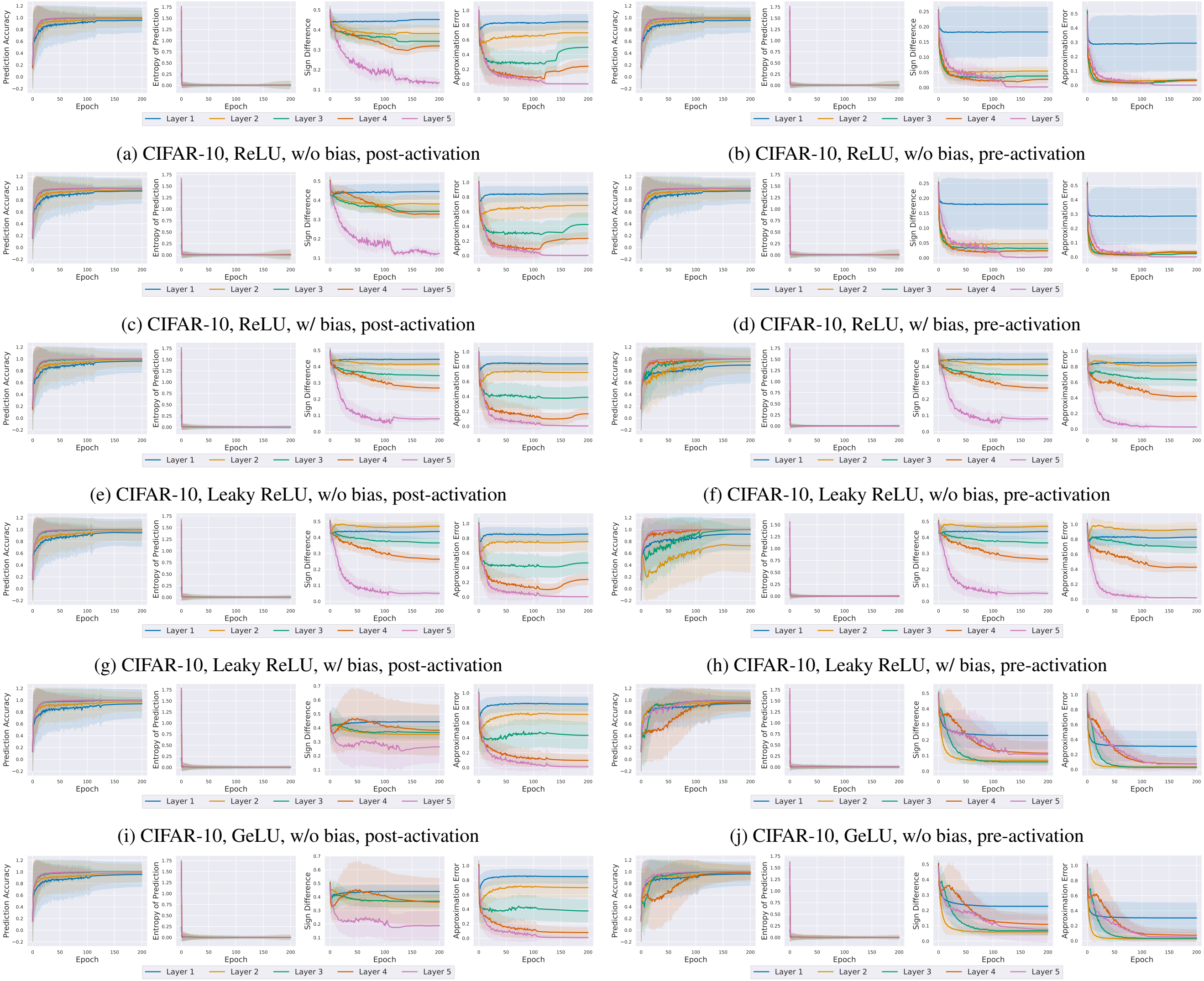
The results in Table 12 indicate that NAN is significantly better than the l_1-norm and the reciprocal of l_0-norm. We note that l_1-norm does not capture deactivation, while the reciprocal of l_0-norm captures only deactivation. Hence, the superiority of NAN over these vector norms indicate that capturing both activation and deactivation is crucial.

Compared to the residual of ViM, on the other hand, NAN is notably superior with respect to the FPR95 metric when ReAct is applied on the model, while NAN is comparable to the residual when without ReAct. We note, however, that the residual of ViM requires eigen decomposition of the bankset features, while the computation of NAN is done by a single forward pass of the network.

E. Limitation of NAN

Based on our theoretical observations, NAN is intrinsically a classifier output and hence may inherit the weaknesses of classifier-based OOD detectors that have been recently found in [8, 9]. In addition, as observed in Sec. D.1, the optimal usage of NAN requires networks that involve the MLP projection head.

[2] https://pytorch.org/vision/main/models/generated/torchvision.models.vit_b_16.html



(a) CIFAR-10, ReLU, w/o bias, post-activation

(b) CIFAR-10, ReLU, w/o bias, pre-activation

(c) CIFAR-10, ReLU, w/ bias, post-activation

(d) CIFAR-10, ReLU, w/ bias, pre-activation

(e) CIFAR-10, Leaky ReLU, w/o bias, post-activation

(f) CIFAR-10, Leaky ReLU, w/o bias, pre-activation

(g) CIFAR-10, Leaky ReLU, w/ bias, post-activation

(h) CIFAR-10, Leaky ReLU, w/ bias, pre-activation

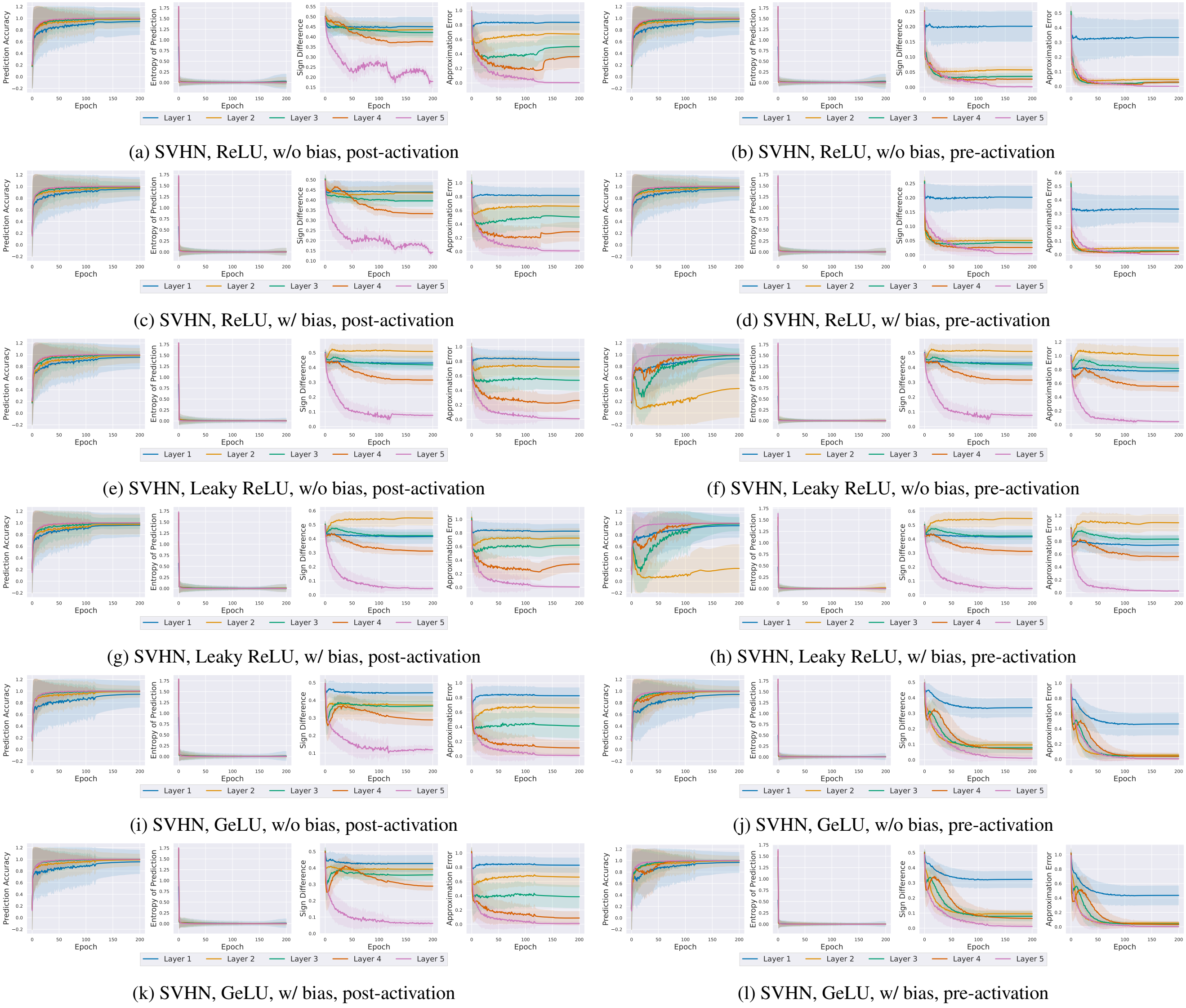
(i) CIFAR-10, GeLU, w/o bias, post-activation

(j) CIFAR-10, GeLU, w/o bias, pre-activation

(k) CIFAR-10, GeLU, w/ bias, post-activation

(l) CIFAR-10, GeLU, w/ bias, pre-activation

Figure 12: Results of hidden classifiers with different activation functions (ReLU, Leaky ReLU, and GeLU) on CIFAR-10.



(a) SVHN, ReLU, w/o bias, post-activation

(b) SVHN, ReLU, w/o bias, pre-activation

(c) SVHN, ReLU, w/ bias, post-activation

(d) SVHN, ReLU, w/ bias, pre-activation

(e) SVHN, Leaky ReLU, w/o bias, post-activation

(f) SVHN, Leaky ReLU, w/o bias, pre-activation

(g) SVHN, Leaky ReLU, w/ bias, post-activation

(h) SVHN, Leaky ReLU, w/ bias, pre-activation

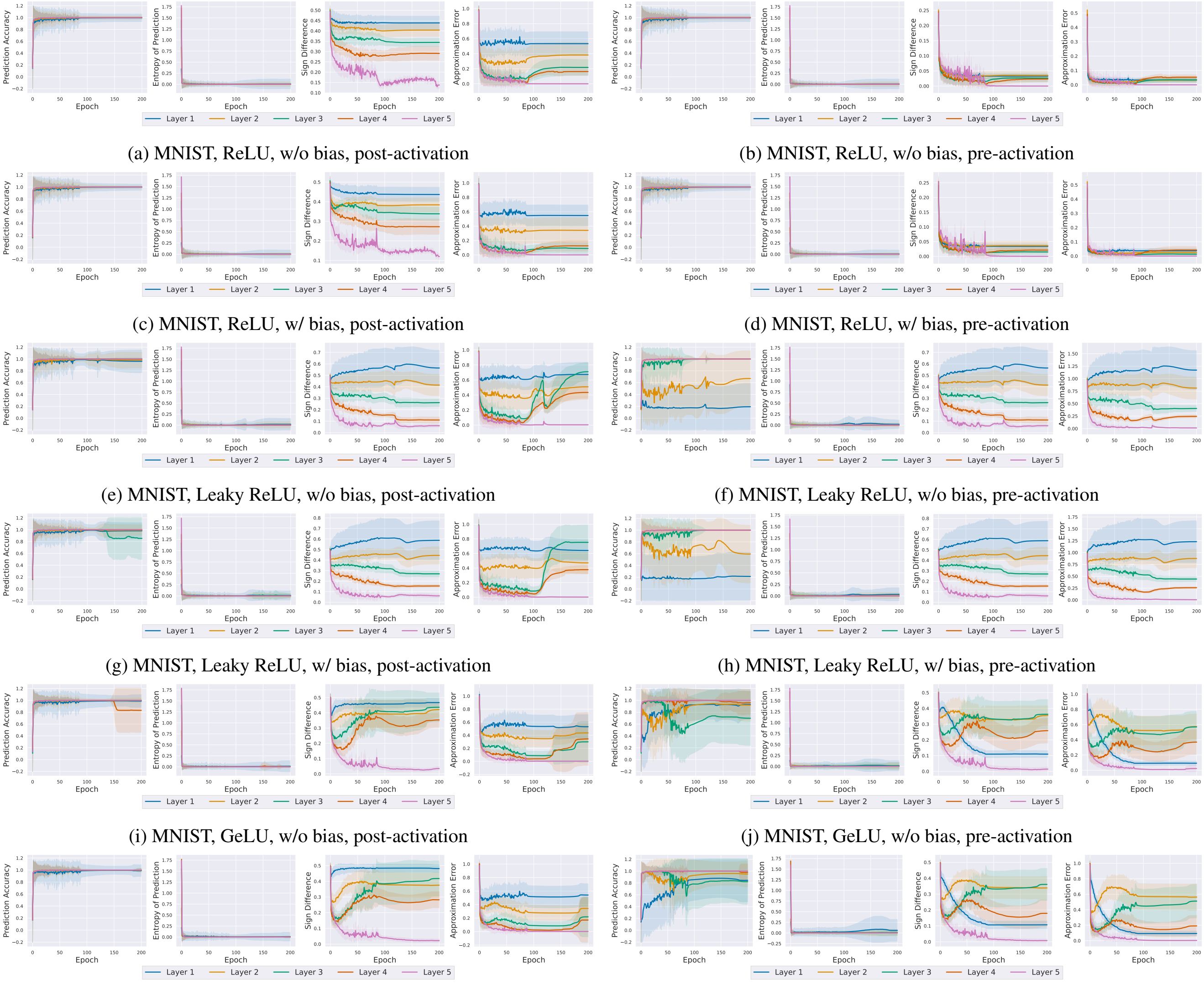
(i) SVHN, GeLU, w/o bias, post-activation

(j) SVHN, GeLU, w/o bias, pre-activation

(k) SVHN, GeLU, w/ bias, post-activation

(l) SVHN, GeLU, w/ bias, pre-activation

Figure 13: Results of hidden classifiers with different activation functions (ReLU, Leaky ReLU, and GeLU) on SVHN.



(a) MNIST, ReLU, w/o bias, post-activation

(b) MNIST, ReLU, w/o bias, pre-activation

(c) MNIST, ReLU, w/ bias, post-activation

(d) MNIST, ReLU, w/ bias, pre-activation

(e) MNIST, Leaky ReLU, w/o bias, post-activation

(f) MNIST, Leaky ReLU, w/o bias, pre-activation

(g) MNIST, Leaky ReLU, w/ bias, post-activation

(h) MNIST, Leaky ReLU, w/ bias, pre-activation

(i) MNIST, GeLU, w/o bias, post-activation

(j) MNIST, GeLU, w/o bias, pre-activation

(k) MNIST, GeLU, w/ bias, post-activation

(l) MNIST, GeLU, w/ bias, pre-activation

Figure 14: Results of hidden classifiers with different activation functions (ReLU, Leaky ReLU, and GeLU) on MNIST.