Supplementary Materials for Learning for Transductive Threshold Calibration in Open-World Recognition

In the supplementary materials, we provide: 1) mathematical derivations and proofs in Appendix A using the same notations as in the main paper, 2) additional visualizations and ablation study results in Appendix B, 3) more implementation details in Appendix C, and 4) A list of frequently asked questions (FAQ) in Appendix D.

A. Detailed Proof and Derivation

A.1. Definitions of TPR^k and TNR^k in Paper Eq.(7)

We first provide a formal definition of TPR^k introduced in Paper Eq.(7). Let $f_1, f_2, ...$ be M-dimensional embeddings for a dataset D projected by an trained DML model whose outputs are L$_2$ normalized. For a given class k, its embedding vectors tend to form a same-class sample cloud, while being away from embeddings of other classes. In this model, the class-specific TPR score reflects intra-class representation compactness, while the class-specific TNR score represents inter-class representation separation. Such structure is often modeled with the von Mises-Fisher (vMF) distribution as in [8, 10, 13, 20, 23, 27, 29, 34], where a concentration parameter κ is used to indicate the representation compactness for said class. According to the Sra's approximation [24], the vMF concentration κ can be estimated as:

$$\kappa = \frac{|c| \cdot (M - |c|^2)}{1 - |c|^2} \tag{1}$$

where $c = \frac{\sum_{i \in D} f_i \cdot 1_{y_i=k}}{\sum_{i \in D} 1_{y_i=k}}$ is the average of all embeddings belonging to this class, and $|c|$ is the L2 norm of c. Given a fixed embedding dimension M, κ is only a function of $|c|$, where $|\cdot|$ denotes the L$_2$ norm. Recall that in the paper we define TPR^k the same as $|c|$. Hence, TPR^k can be regarded as a class-specific TPR indicator with the following property: TPR^k $\rightarrow 1$ when the intra-class embeddings are in close proximity to each other (dense), and TPR^k $\rightarrow 0$ when the intra-class embeddings are far apart (loose).

We also explain the definition of TNR^k. In general, for a well-trained embedding model, true negative pairs tend to exhibit small inter-class cosine similarities, whereas false negative pairs tend to have larger inter-class cosine similarities. Consequently, the numerator of TNR^k $= \frac{\sum_{i,j \in D} (1-a_{ij}) \cdot 1_{y_j \neq y_i = k}}{\sum_{i,j \in D} 1_{y_j \neq y_i = k}}$ captures the true negative pairs while the denominator gives all negative pairs. This effectively approximates the TNR of this class within the dataset D.

A.2. Proof for Paper Theorem 1

We first show that TNR^k can be expressed as a function of the sum of cosine similarities for all positive pairs of the given class (whose proof is given below), and give the mathematical proof for paper Theorem 1. First, let $f_1, f_2, ..., f_n$ be image embeddings belonging to class k where $n = \sum_{i \in D} 1_{y_i=k}$, TPR^k can be expanded as follows:

$$\begin{aligned}
\text{TPR}^k &= \frac{\| \sum_{i \in D} f_i \cdot 1_{y_i=k} \|}{\sum_{i \in D} 1_{y_i=k}} \\
&= \frac{1}{n} \cdot \sqrt{\| \sum_{t=1}^{n} f_t \|^2} \\
&= \frac{1}{n} \cdot \sqrt{2 \sum_{i,j=1,2,\ldots,n;\, i<j} \langle f_i, f_j \rangle + \sum_{t=1}^{n} \|f_t\|^2} \\
&= \frac{1}{n} \cdot \sqrt{2 \sum_{i,j=1,2,\ldots,n;\, i<j} \cos(\theta_{ij}) + n} \\
&= \frac{1}{n} \cdot \sqrt{2 \sum_{a_{ij} \in A} a_{ij} + n}
\end{aligned} \tag{2}$$

where A is the collection of all pair-wise cosine similarities for nodes belonging to said class. Note that, we only count a_{ij} and a_{ji} once in A, i.e $i < j$ for all a_{ij}.

Now, we provide the derivation for Paper Theorem 1. Let $\mathcal{N}^{\text{same}}$ be the set of nodes which contains all instances belonging to a given class. For each vertex $i \in \mathcal{N}^{\text{same}}$ whose neighbourhood is $\mathcal{N}_i$, if $\mathcal{N}_i$ is large enough to cover all same-class vertices for vertex i (i.e., $\mathcal{N}^{\text{same}} \subset \mathcal{N}_i$), TPR^k

can be expressed as:

$$
\begin{aligned}
\mathrm{TPR}^k &= \sqrt{\frac{2 \sum\limits_{i,j \in \mathcal{N}^{\text{same}}, i<j} 1_{y_i=y_j} \cdot a_{ij} + |\mathcal{N}^{\text{same}}|}{|\mathcal{N}^{\text{same}}|^2}} \\
&= \sqrt{\frac{\sum\limits_{i \in \mathcal{N}^{\text{same}}} \sum\limits_{j \in \mathcal{N}_i} 1_{y_i=y_j} \cdot a_{ij} + |\mathcal{N}^{\text{same}}|}{|\mathcal{N}^{\text{same}}|^2}} \\
&= \sqrt{\frac{|\mathcal{N}_i|}{2|\mathcal{N}^{\text{same}}|} \cdot \underbrace{\frac{2 \sum\limits_{i \in \mathcal{N}^{\text{same}}} s_i^{avg}}{|\mathcal{N}^{\text{same}}|}}_{\text{average avg density}} + \frac{1}{|\mathcal{N}^{\text{same}}|}} \\
&= \sqrt{\frac{|\mathcal{N}_i|}{2|\mathcal{N}^{\text{same}}|} \cdot \frac{\sum\limits_{i \in \mathcal{N}^{\text{same}}} \sum\limits_{j \in \mathcal{N}_i} (\cdot 1_{y_i=y_j} - 1_{y_i \neq y_j} + 1) \cdot a_{ij}}{|\mathcal{N}^{\text{same}}| \cdot |\mathcal{N}_i|} + \frac{1}{|\mathcal{N}^{\text{same}}|}} \\
&= \sqrt{\frac{|\mathcal{N}_i|}{2|\mathcal{N}^{\text{same}}|} \cdot (\underbrace{\frac{\sum\limits_{i \in \mathcal{N}^{\text{same}}} s_i^{nbr}}{|\mathcal{N}^{\text{same}}|}}_{\text{average nbr density}} + \underbrace{\frac{\sum\limits_{i \in \mathcal{N}^{\text{same}}} \sum\limits_{j \in \mathcal{N}_i} a_{ij}}{|\mathcal{N}^{\text{same}}| \cdot |\mathcal{N}_i|}}_{\text{average affinity}}) + \frac{1}{|\mathcal{N}^{\text{same}}|}}
\end{aligned}
\tag{3}
$$

Given a reasonable clustering model and $|\mathcal{N}_i| \gg 1$, we let $\mathcal{N}$ be a subset of $\mathcal{N}^{\text{same}}$, due to the stochastic convergence of TPR^k provided by the vMF model, when $|\mathcal{N}|$ is sufficiently large, we have:

$$
\begin{aligned}
\mathrm{TPR}^k &\approx \sqrt{\frac{|\mathcal{N}_i|}{2|\mathcal{N}|} \cdot (\frac{\sum\limits_{i \in \mathcal{N}} s_i^{nbr}}{|\mathcal{N}|} + \frac{\sum\limits_{i \in \mathcal{N}} \left(\frac{1}{|\mathcal{N}_i|} \sum\limits_{j \in \mathcal{N}_i} a_{ij}\right)}{|\mathcal{N}|}) + \frac{1}{|\mathcal{N}|}} \\
&= \sqrt{\frac{|\mathcal{N}_i|}{2|\mathcal{N}|} \cdot (\frac{\sum\limits_{i \in \mathcal{N}} s_i^{nbr}}{|\mathcal{N}|} + \frac{\sum\limits_{i \in \mathcal{N}} \mathfrak{a}_i^{avg}}{|\mathcal{N}|}) + \frac{1}{|\mathcal{N}|}}
\end{aligned}
\tag{4}
$$

As $|\mathcal{N}| \gg 1$, we further simplify the above equation by dropping the last term, yielding:

$$
\mathrm{TPR}^k \approx \left(\frac{|\mathcal{N}_i|}{2|\mathcal{N}|} \cdot \left(\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} s_i^{nbr} + \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \mathfrak{a}_i^{avg}\right)\right)^{1/2} \tag{5}
$$

Note that the s_i^{nbr} and $\mathfrak{a}_i^{avg}$ here are defined with respect to $\mathcal{N}_i$ as shown in paper Eq. (6) and Eq. (8). So far, we have proven Theorem 1 in the main paper.

A.3. Proof for Paper Theorem 2

By definition in Paper Eq.(6), we can express the difference between the average density and the neighborhood average density for a given sample i as follows:

$$
s_i^{\text{avg}} - s_i^{\text{nbr}} = \frac{\sum\limits_{j \in \mathcal{N}_i} a_{ij} \cdot 1_{y_i \neq y_j}}{|\mathcal{N}_i|} \tag{6}
$$

Assuming cluster $\mathcal{N}$ has high purity with the majority class being k, we express the average difference between the average density and the neighborhood average density in cluster $\mathcal{N}$ as follows:

$$
\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} (s_i^{\text{avg}} - s_i^{\text{nbr}}) = \frac{\sum\limits_{i \in \mathcal{N}} \sum\limits_{j \in \mathcal{N}_i} a_{ij} \cdot 1_{y_j \neq y_i = k}}{|\mathcal{N}| \cdot |\mathcal{N}_i|} \tag{7}
$$

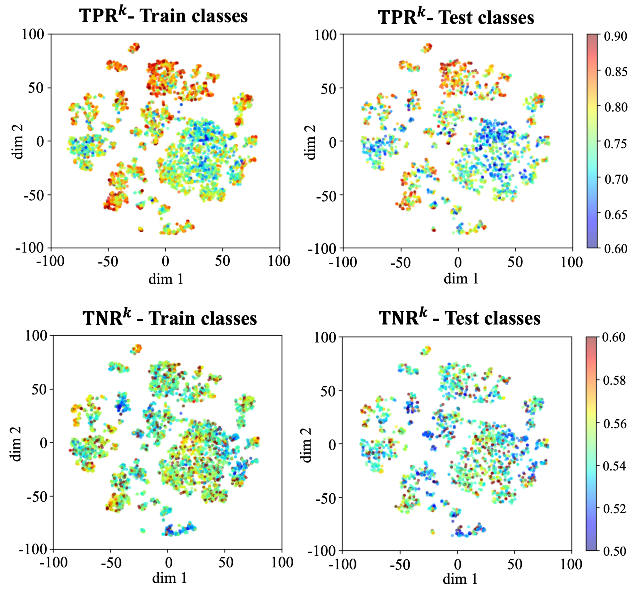


Figure 1. This T-SNE visualization depicts class-specific TPR and TNR scores for disjoint train and test classes in the iNaturalist-2018 dataset. Each point represents a class, with color indicating TPR^k in the top row and TNR^k in the bottom row. The results reveal a strong class-to-class neighbourhood similarity in both intra-class compactness and inter-class separation. Despite a magnitude offset between the scores of the train and test classes, this neighbourhood similarity generalizes to previously unseen test classes in the open world, as demonstrated by the consistent patterns between training and testing. *Best viewed in color.*

When N_i is sufficiently large, i.e., $|N_i| \to |N| \gg 1$, the above equation becomes:

$$
\begin{aligned}
\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} (s_i^{\text{avg}} - s_i^{\text{nbr}}) &\approx \frac{\sum\limits_{j \in \mathcal{N}} a_{ij} \cdot 1_{y_j \neq y_i = k}}{|\mathcal{N}|} \\
&= \frac{|\mathcal{N}|_{k-}}{|\mathcal{N}|} \frac{\sum\limits_{j \in \mathcal{N}} a_{ij} \cdot 1_{y_j \neq y_i = k}}{|\mathcal{N}|_{k-}} \\
&= \frac{|\mathcal{N}|_{k-}}{|\mathcal{N}|} \left(1 - \underbrace{\frac{\sum\limits_{j \in \mathcal{N}} (1 - a_{ij}) \cdot 1_{y_j \neq y_i = k}}{|\mathcal{N}|_{k-}}}_{\text{same as } \mathrm{TNR}^k \text{ in cluster } \mathcal{N}}\right)
\end{aligned}
\tag{8}
$$

By rearranging the above equation, we have:

$$
\begin{aligned}
\mathrm{TNR}^k &\approx 1 - \frac{1}{|\mathcal{N}|_{k-}} \cdot \sum_{i \in \mathcal{N}} (s_i^{\text{avg}} - s_i^{\text{nbr}}) \\
&= 1 - \frac{|\mathcal{N}|}{|\mathcal{N}|_{k-}} \cdot \left(\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} s_i^{\text{avg}} - \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} s_i^{\text{nbr}}\right)
\end{aligned}
\tag{9}
$$

Thus, we have proven Theorem 2 in the main paper.

B. Additional Visualizations / Ablation Studies

B.1. Further Justification for Using a GNN

To further justify the use of a GNN architecture, we explore the intra-class and inter-class representation structures on

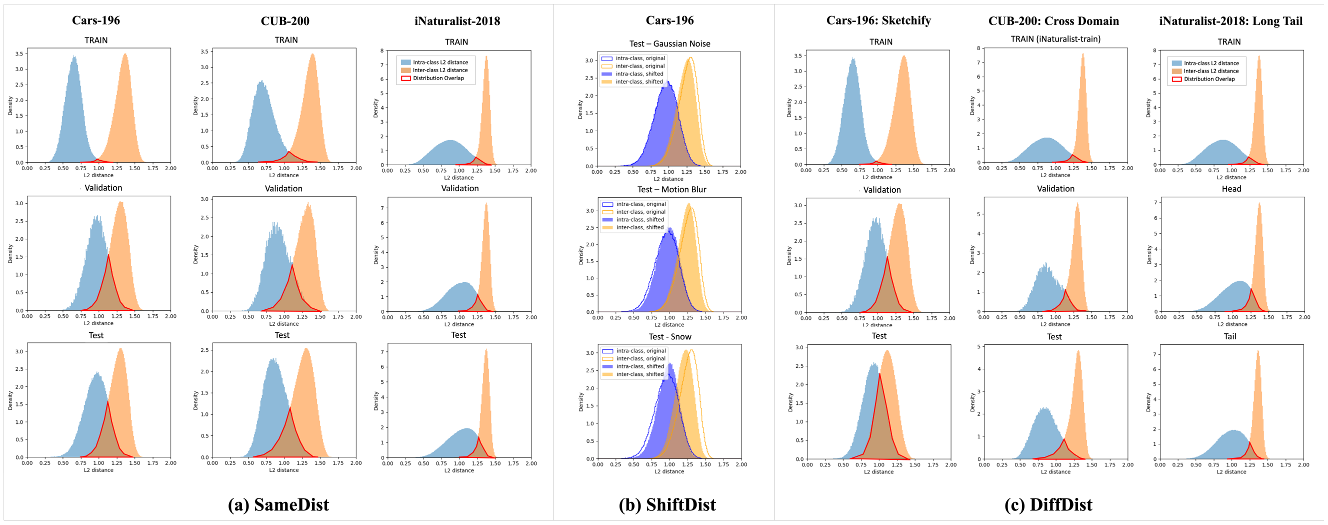


Figure 2. Intra-class and inter-class L2 distance distributions for Cars-196, CUB-200 and iNaturalist-2018 datasets in (a) SameDist, (b) ShiftDist, and (c) DiffDist scenarios. In the ShiftDist scenario, we select three representative corruption types (Gaussian Noise, Motion Blur, Snow) for visualization. *Best viewed in color.*

iNaturalist-2018 [26], one of the largest image recognition benchmarks. We follow the open-world recognition setting by training a DML embedding model on the training partition and extracting embeddings for both training and testing partitions. For each class k, we compute its class centroid, along with TPR^k and TNR^k, and subsequently generate a T-SNE [25] plot using these class centroids. It's worth noting that in this T-SNE plot, each point represents a class instead of a data sample. The visualization in Fig. 1 demonstrates that proximate classes in the embedding space exhibit similarities in their class-specific TPR and TNR scores, as highlighted by the colors. This observation aligns with prior research [7, 11, 12], which has argued for the existence of neighborhood similarity in representation structures. Meanwhile, this similarity pattern, despite a magnitude offset between closed and open worlds, extends to unseen test classes. These findings highlight the rich neighborhood information within the representation structures, supporting our choice of a GNN architecture, known for its capabilities in capturing the neighbourhood information in representation structures [2, 9, 22, 30, 31].

B.2. Further Justification for Using Both Densities

OpenGCN uses a single GAT encoder to jointly estimate graph edge connectivity, as well as s^{nbr} and s^{avg} as two node attributes to harness the mutual information among these properties. In Section 4.3 of the paper, we present an ablation study where predicting both densities in conjunction with the connectivity yields the best calibration performance. Additionally, in alignment with the advantage of capturing both class-specific TPR and TNR simultaneously, as mentioned in Paper Sec.3.2, we demonstrate that simultaneously evaluating $(s^{\text{avg}}, s^{\text{nbr}})$ guarantees a higher amount of entropy (denoted as H) compared to evaluating s^{nbr} alone, as shown in the entropic inequality below:

$$\mathrm{H}(s^{\text{avg}}, s^{\text{nbr}}) = \mathrm{H}(s^{\text{nbr}}) + \mathrm{H}(\mathfrak{a}^{\text{avg}}|s^{\text{nbr}}) \geq \mathrm{H}(s^{\text{nbr}}) \quad (10)$$

This increased information assists OpenGCN in better capturing the representation structure for distance calibration.

B.3. Additional Experiment Details

Distance Distribution Visualizations for SameDist, ShiftDist and DiffDist Scenarios In Fig. 2, we present visualizations of the intra-class and inter-class L2 distance distributions for D_{train}, D_{cal}, and D_{test} in the SameDist, ShiftDist, and DiffDist calibration scenarios. For the ShiftDist scenario, we select three representative corruption types: Gaussian noise, motion blur, and snow, for illustration. Notably, these corruptions are applied only to D_{test}, while the distance distributions for D_{train} and D_{cal} remain the same as in the SameDist scenario. As depicted in the figure, in the SameDist calibration scenario, D_{cal} and D_{test} exhibit similar distance distributions for each benchmark. Meanwhile, as the embedding model is exclusively trained on D_{train}, the distance distribution for D_{train} is more compact and discriminative when compared to those for D_{cal} and D_{test}. In the ShiftDist scenario, there is a slight distribution shift, with the magnitude varying depending on the corruption type. In the DiffDist scenario, even more pronounced distance distribution shifts are observed between D_{cal} and D_{test}, characterized by differing geometries and locations of the overlapping area, highlighted in red.

Visualizations for TPR and TNR Predictions In Fig. 3, we present visualizations of the TPR and TNR prediction

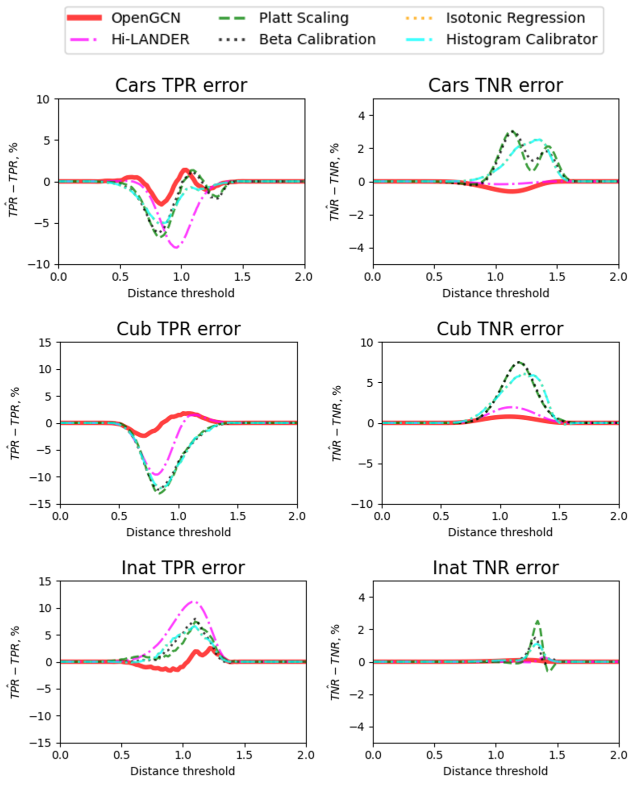


Figure 3. TPR and TNR prediction errors for various threshold calibration methods in the SameDist calibration scenario. Values closer to 0 indicate better accuracy. *Best viewed in color.*

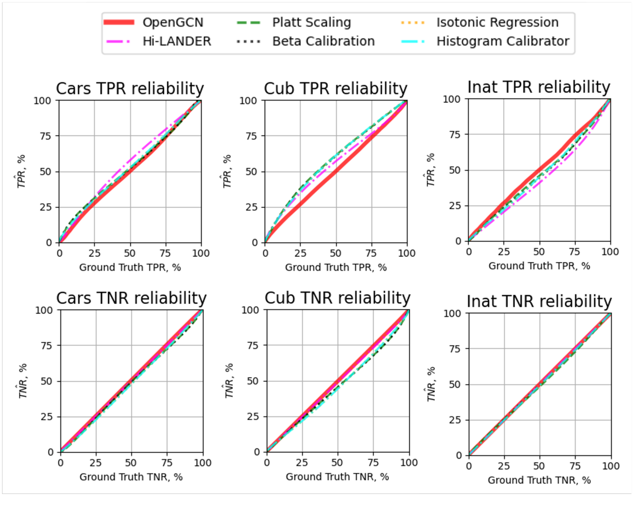


Figure 4. Reliability plots for TPR and TNR predictions for various calibration methods in the SameDist calibration scenario. In each plot, the x-axis is the ground truth TPR or TNR values, and y-axis is the predicted TPR or TNR values. Curves that are closer to $y = x$ indicate better accuracy. *Best viewed in color.*

errors at each distance threshold for all three benchmarks. Values closer to zero indicate higher accuracy in prediction.

Table 1. Evaluation results in the SameDist scenario for additional baseline methods using the global error metric of $\mathrm{MAE}_{\mathrm{comb}}$. In the gray-shaded methods, the notation "oracle" refers to using the ground truth number of test classes as the cluster number for estimating the pseudolabels to compute TPR and TNR. In the lightcyan-shaded methods, an asterisk (*) denotes that, when fitting the calibration function for each baseline method, both D_{train} and D_{cal} were used as the calibration dataset, as opposed to the standard approach of using only D_{cal}. Methods with pink shading are consistent with those presented in the main table.

Method	Cars	CUB	iNat	Avearge
K-Means (oracle)	1.93e-2	2.39e-2	7.08e-2	3.80e-2
Spectral (oracle)	9.55e-3	1.74e-2	4.82e-2	2.51e-2
OPTICS	1.91e-1	1.21e-1	1.47e-1	1.53e-1
FINCH	1.22e-1	1.81e-1	5.67e-2	1.20e-1
Platt Scaling*	1.68e-1	2.34e-1	9.74e-2	1.66e-1
Isotonic Regression*	1.68e-1	2.34e-1	9.74e-2	1.66e-1
Hi-LANDER*	5.66e-3	1.53e-2	1.05e-2	1.05e-2
Platt scaling	1.55e-2	3.59e-2	1.23e-2	2.12e-2
Isotonic regression	1.38e-2	3.61e-2	1.18e-2	2.06e-2
Hi-LANDER	1.29e-2	1.94e-2	2.14e-2	1.79e-2
OpenGCN	**5.25e-3**	**6.50e-3**	**4.82e-3**	**5.52e-3**

Additionally, in Fig. 4, we present reliability plots depicting the relationship between ground truth TPR and TNR (on the x-axis) and the predicted TPR and TNR values (on the y-axis) for each method, where methods with curves closer to $y = x$ indicates better accuracy. As shown in the figures, our openGAN method demonstrates the best overall calibration performance by accurately predicting TPR and TNR metrics across various distance thresholds. However, it is worth noting that, similar to what was observed in the pointwise evaluation results presented in Paper Table (2), OpenGCN may not excel at every individual threshold, despite its overall excellent performance. In cases where a specific calibration distance threshold range is of interest, further adjustments, such as reweighting the importance of pairwise connectivities for pairs within a particular cosine distance range, can be applied to improve OpenGCN's point-wise calibration accuracy for a specific target.

Additional Baseline Methods In Table 1, we present additional baseline results to further validate the effectiveness of our proposed OpenGCN method. First, we consider several standard clustering baselines, marked in gray shade, including K-Means [16], Spectral clustering [17], OPTICS [1], and FINCH [21]. For K-Means and Spectral clustering, we provide them with an unfair advantage by assuming knowledge of the ground truth number of classes, denoted with "(oracle)" in the table. This is based on the consideration that obtaining this hyperparameter via cross-validation in open-world scenarios is practically infeasible. Additionally, we provide results for Platt Scaling [19], Isotonic Regression [33] and Hi-LANDER [30], marked in lightcyan shade, where both D_{train} and D_{cal} are used as the calibration dataset to fit the calibration function, marked with * in the table. This approach serves as a data-fair comparison with our OpenGCN method, which utilizes both D_{train} and D_{cal} during training. As demonstrated, among the gray-shaded ones, the methods labeled as "oracle" outperform those lacking knowledge of the ground truth number of test classes. However, even with access to this information, they still lag behind our OpenGCN method. Regarding the traditional calibration methods that utilize $D_{\text{train}} + D_{\text{cal}}$ for calibration, such as Platt Scaling* and Isotonic Regression*, we observed a significant increase in calibration error by an order of magnitude. This can be ascribed to the fact that using $D_{\text{train}} + D_{\text{cal}}$ as the calibration dataset further violates the assumption of similar distance distribution between the calibration and test datasets. For Hi-LANDER, we noticed an improvement in calibration performance when introducing D_{cal}, although it is still significantly worse than our OpenGCN method. We hypothesize that this improvement is because Hi-LANDER is also a GNN-based method, which exhibits better generalization and adaptability compared to traditional posthoc calibration methods.

B.4. OpenGCN Design Ablations

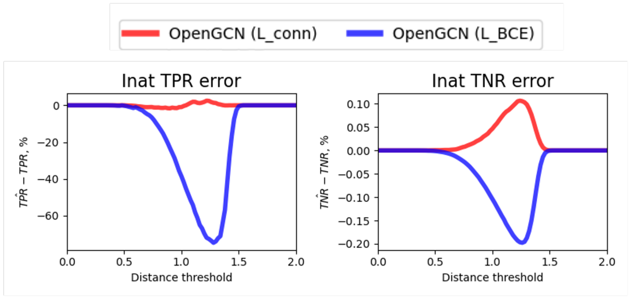
Unless otherwise stated, all experiments in this section are conducted on the iNaturalist-2018 [26] dataset.

Joint Training vs Two-stage Training In Table 7 of the paper, we compare the calibration performances between OpenGCN pretrained on D_{train} (without finetuning) and OpenGCN pretrained on D_{train} and finetuned on D_{cal}. For more comprehensive comparison, we also provide an additional experiment where OpenGCN is pretrained on the joint set of D_{train} and D_{cal}, denoted as "JT" in Tab. 2. The results in the table indicate mixed outcomes when introducing the additional open-world dataset D_{cal} in the pretraining stage: while it improves calibration performance on Cars and iNat, it worsens it on CUB. However, as suggested by the results of "OpenGCN (PT+FT)," introducing D_{cal} via finetuning the MLP head significantly reduced the global calibration error by nearly one order of magnitude across all three benchmarks. This results supports our choice of two-stage training in adapting the calibration model from the closed-world context to the open-world scenarios.

Table 2. Performance of OpenGCN in the SameDist scenario with various pretraining and finetuning strategies, evaluated using the global error metric MAE_{comb}. PT: pretraining with D_{train}, FT: finetuning with D_{cal}, JT: joint-training with both D_{train} and D_{cal}.

Method	Cars	CUB	iNat	Avearge
OpenGCN (PT)	2.90e-2	2.52e-2	3.55e-2	2.99e-2
OpenGCN (JT)	1.80e-2	2.67e-2	2.08e-2	2.18e-2
OpenGCN (PT+FT)	**5.25e-3**	**6.50e-3**	**4.82e-3**	**5.52e-3**

Ablation of Loss for Connectivity Prediction We validate our choice of the connectivity prediction loss, $\mathcal{L}_{\text{conn}}$ (defined in Paper Eq.(11)), by replacing it with the standard Binary Cross Entropy loss (BCE) while retaining the same auxiliary losses for density predictions. As depicted in Fig. 5, implementing a balancing strategy in connectivity prediction significantly improves calibration performance, especially in accurately predicting True Positive Rates (TPR) at various distance thresholds. In our experiment, we observed that without balancing, the connectivity distribution is dominated by the negative pairs, leading to TPR prediction errors as high as 70%. This is primarily because in visual recognition, the number of classes can be very large, and as OpenGCN randomly samples subsets to build fully connected graphs, it generates a substantially higher number of negative pairs compared to positive pairs. Without balancing, the model tends to struggle in learning and identifying positive connectivities.



	$\mathcal{L}_{\text{BCE}} + \lambda \cdot (\mathcal{L}_{s^{\text{avg}}} + \mathcal{L}_{s^{\text{nbr}}})$	$\mathcal{L}_{\text{conn}} + \lambda \cdot (\mathcal{L}_{s^{\text{avg}}} + \mathcal{L}_{s^{\text{nbr}}})$
MAE_{comb}	1.59e-1	**4.82e-3**

Figure 5. Comparison between Binary Cross Entropy Loss and Our Proposed $\mathcal{L}_{\text{conn}}$ for Connectivity Prediction.

Robustness to Calibration Data Size In Fig. 6, we perform an ablation study to evaluate how the size of the calibration dataset, measured as the percentage of classes in D_{cal} used, affects the calibration performance of the OpenGCN model. We also provide a comparison with Isotonic Regression, which is the best baseline method, under the same conditions. The figure demonstrates that OpenGCN consistently achieves lower calibration errors across all considered percentages of classes (5%, 10%, 20%, 50%, 100%). While a reasonably large D_{cal} is preferred for OpenGCN to perform well, it outperforms Isotonic Regression in all cases.

C. Additional Implementation Details

C.1. Implementation for Baseline Methods

Traditional calibration methods We consider several traditional posthoc calibration baselines including the Platt scaling [19], Beta calibration [14] and Isotonic regression [33], where the calibration methods learn a map from D_{train} to D_{val} about the evaluation metrics at the same distance threshold. The implementation is based on the *net:cal calibration framework* [15]. Below, we detail the setup for each

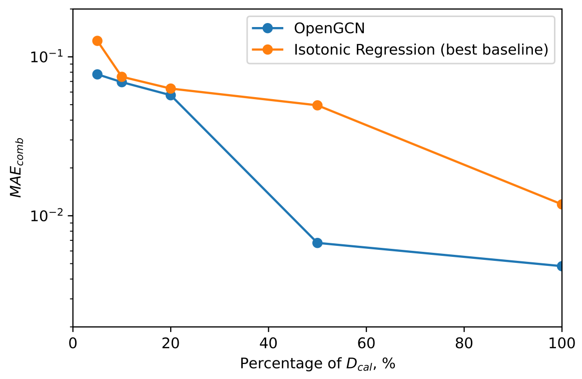


Figure 6. Comparing OpenGCN's Robustness to Calibration Dataset Size with the Best Baseline Method (Isotonic Regression). The x-axis represents the percentage of classes in D_{cal} used for fitting the calibration functions.

baseline method:

- For Platt scaling [19], we aim to find the optimal scalar parameters A, B of the $y = (1 + e^{Ax+B})^{-1}$ function to map from $\{x$: the performance metric of interest for $D_{\text{train}}\}$ to $\{y$: the performance metric of interest for $D_{\text{cal}}\}$ at the same d, so as to calibrate the target performance metric to distance threshold curve obtained from D_{train} for better generalization. The optimal solution is found with maximum likelihood estimate.
- Beta calibration [14] is a variant of Platt scaling that solve the regression problem of $y = \frac{1}{1+e^{z(x)}}$, where $z(x)$ is a function parameterized by the Beta function. Isotonic regression [33] fits a piecewise-constant non-decreasing function from the training curve to the calibration curve.
- For isotonic regression, to prevent overfitting, , we split the d-axis into $n = 5$ bins with equal mass.
- For histogram calibration [32], we use a histogram bin number of 100 to ensure a sufficient grid resolution.

Pseudolabel-based calibration methods For pseudolabel-based baselines, including DBSCAN [6] and HiLANDER [30], we follow previous works [30, 31] and exclusively train the calibration model on D_{train} and evaluate it on D_{test} using the pseudolabels decoded from the clustering model. In the case of DBSCAN, we fine-tune the ϵ parameter on D_{cal} through cross-validation. For Hi-LANDER, we use its official codebase for both training and inference.

C.2. Details to OpenGCN Training and Inference

Input Embedding Preparation Same as [4, 28, 30, 31], we utilize deep CNN features as GNN input to compute the affinity between nearby nodes using the pairwise cosine similarity of the visual embeddings. For all datasets, following [3], we use ImageNet [5]-pretrained ResNet50, whose final softmax layers are replaced with one linear layer, to train a recognition model for each benchmark. For CUB and Cars, we use an embedding dimension of 128; for iNaturalist-2018, we use an embedding dimension of 512. As we focus on calibration for open-world recognition, the training data for the visual embedding model follows the same open-set train-test split as calibration where the classes in training and testing are disjoint to each other. After training, we use the embedding model to extract embeddings, which are normalized to unit vectors following the standard practice in visual recognition.

Sub-graph Construction During training, we build mini-batches containing 256 samples. Specifically, in the pretraining stage, we follow the data balancing strategy in [3, 18], where we divide D_{train} into batches, each containing 4 images per class. In the fine-tuning stage, we generate random batches from D_{cal}, to simulate the unknown open-world test compositions. To ensure that the pairwise distance distribution within each batch accurately reflects the actual pairwise distance distribution, we construct fully connected sub-graphs for each data batch during pretraining, finetuning and inference.

Inference for Calibration After acquiring TPR and TNR estimates at each distance threshold from the GNN output, we solve for $\hat{d}_{\text{opt}}$ as a constrained optimization problem as outlined in Paper Eq.(1), where the objective can be customized for each use case. For global evaluation, we assume that α is unknown and can assume any possible value. Thus, we use the global error metric MAE_{comb} for evaluation. For point-wise evaluation, we set a specific α value (e.g., TPR $= 80\%$) and use grid search across the global distance threshold of $[0, 2]$ with a grid size of 0.01 to solve for d^{opt}. Once $\hat{d}_{\text{opt}}$ is acquired, we apply it to D_{test} and use the ground-truth labels to examine the calibration performance using the pointwise metrics AE_{TPR} and AE_{TNR}.

D. FAQ

OpenGCN is less computationally efficient compared to traditional post-hoc calibration methods, would this make OpenGCN less preferable for application deployment in practice? Though less computationally efficient compared to traditional inductive post-hoc calibration methods such as Platt-scaling and Isotonic regression, OpenGCN mitigates the need to collect one calibration dataset for each application data domain as in traditional calibration methods. Such manual collection of calibration datasets with human annotation is costly, time-consuming and often prohibitively expensive in an open-world environment where test distributions are highly dynamic and can infinitely vary. However, with OpenGCN, one model can be deployed and used without the need to collect and annotate test-domain specific calibration datasets, and the distance calibration is done transductively over the incoming unlabeled test instances through a GNN model inference. This renders OpenGCN more suitable for deployment in practical applications, particularly in environments that inherently embody an open-world scenario.

Why are traditional calibration evaluation metrics such as ECE not used in the evaluation? Our problem setting - open-world threshold calibration, aims at finding the distance threshold for achieving the desired TPR and TNR performance values for deep metric learning in open-world recognition. This is different to closed-world confidence calibration, where the goal is to align output softmax scores with the empirical correctness of the model. As a result, traditional calibration metrics like Expected Calibration Error (ECE), which assess closed-world confidence calibration, do not apply to our problem setting and are not used to evaluate our proposed method and baselines.

References

[1] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999. 4

[2] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 3

[3] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. *CoRR*, abs/2007.12163, 2020. 6

[4] Tianyue Cao, Yongxin Wang, Yifan Xing, Tianjun Xiao, Tong He, Zheng Zhang, Hao Zhou, and Joseph Tighe. Pss: Progressive sample selection for open-world visual representation learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pages 278–294. Springer, 2022. 6

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, pages 226–231, 1996. 6

[7] Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *International conference on machine learning*, 2019. 3

[8] Siddharth Gopal and Yiming Yang. Von mises-fisher clustering models. In *International Conference on Machine Learning*, pages 154–162. PMLR, 2014. 1

[9] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017. 3

[10] Md Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, Liming Chen, et al. von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017. 1

[11] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8849–8858, 2020. 3

[12] Shaogang Gong Jiabo Huang, Qi Dong and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *Proceedings of the International Conference on machine learning (ICML)*, 2019. 3

[13] V. Mardia Kanti and Peter E. Jupp. Directional statistics. *Wiley*, 2000. 1

[14] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial Intelligence and Statistics*, pages 623–631. PMLR, 2017. 5, 6

[15] Fabian Küppers, Jonas Schneider, and Anselm Haselhoff. Parametric and multivariate uncertainty calibration for regression and object detection. In *European Conference on Computer Vision (ECCV) Workshops*. Springer, 2022. 5

[16] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 4

[17] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001. 4

[18] Yash Patel, Giorgos Tolias, and Jiri Matas. Recall@k surrogate loss with large batches and similarity mixup. *CVPR*, 2022. 6

[19] John Platt. Probabilistic outputs for svms and comparisons to regularized likelihood methods. *Advances in Large-Margin Classifiers*, 10(3):61—-74, 1999. 4, 5, 6

[20] Karsten Roth, Oriol Vinyals, and Zeynep Akata. Non-isotropy regularization for proxy-based deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7420–7430, 2022. 1

[21] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8934–8943, 2019. 4

[22] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1): 61–80, 2008. 3

[23] Tyler R Scott, Andrew C Gallagher, and Michael C Mozer. von mises-fisher loss: An exploration of embedding geometries for supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10612–10622, 2021. 1

[24] S. Sra. A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of $i_s(x)$. 2012. 1

[25] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 3

[26] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 3, 5

[27] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017. 1

[28] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1117–1125, 2019. 6

[29] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 748–756. IEEE, 2018. 1

[30] Yifan Xing, Tong He, Tianjun Xiao, Yongxin Wang, Yuanjun Xiong, Wei Xia, David Wipf, Zheng Zhang, and Stefano Soatto. Learning hierarchical graph neural networks for image clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3467–3477, 2021. 3, 4, 6

[31] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. Learning to cluster faces via confidence and connectivity estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3, 6

[32] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, page 609–616, 2001. 6

[33] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 694–699, 2002. 4, 5, 6

[34] Xuefei Zhe, Shifeng Chen, and Hong Yan. Directional statistics-based deep metric learning for image classification and retrieval. *Pattern Recognition*, 93:113–123, 2019. 1