

Learning for Transductive Threshold Calibration in Open-World Recognition

Qin Zhang¹, Dongsheng An¹, Tianjun Xiao², Tong He², Qingming Tang³, Ying Nian Wu¹,
Joseph Tighe¹, Yifan Xing¹, Stefano Soatto¹

¹ AWS AI Labs ² Amazon Web Services ³ Alexa AI

{qzaamz, andongsh, tianjux, htong, qmtang, wunyun, yifax, soattos}@amazon.com, jtighe@cs.unc.edu

Abstract

In deep metric learning for visual recognition, the calibration of distance thresholds is crucial for achieving desired model performance in the true positive rates (TPR) or true negative rates (TNR). However, calibrating this threshold presents challenges in open-world scenarios, where the test classes can be entirely disjoint from those encountered during training. We define the problem of finding distance thresholds for a trained embedding model to achieve target performance metrics over unseen open-world test classes as **open-world threshold calibration**. Existing posthoc threshold calibration methods, reliant on inductive inference and requiring a calibration dataset with a similar distance distribution as the test data, often prove ineffective in open-world scenarios. To address this, we introduce **OpenGCN**, a Graph Neural Network-based transductive threshold calibration method with enhanced adaptability and robustness. OpenGCN learns to predict pairwise connectivity for the unlabeled test instances embedded in a graph to determine its TPR and TNR at various distance thresholds, allowing for transductive inference of the distance thresholds which also incorporates test-time information. Extensive experiments across open-world visual recognition benchmarks validate OpenGCN’s superiority over existing posthoc calibration methods for open-world threshold calibration.

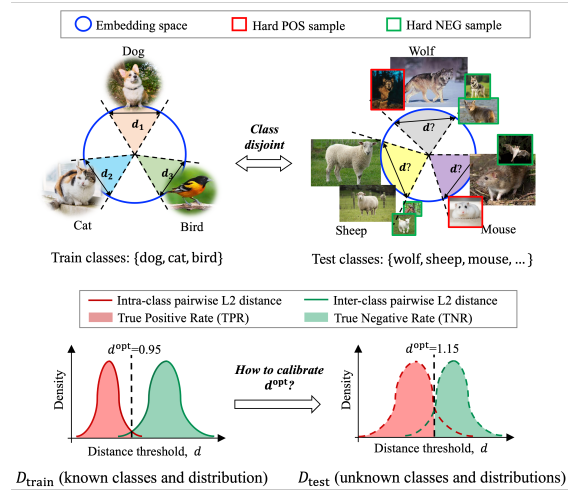


Figure 1. This figure illustrates the **open-world threshold calibration** problem. In open-world recognition, the embedding model is trained on closed-set classes but tested on distinct open-world classes. When applying the model to open-world classes, it often produces less compact embeddings than those encountered during training, necessitating the calibration of the distance threshold for achieving the desired TPR and TNR trade-off. However, the absence of prior knowledge about open-world test classes and distributions makes it challenging to find the optimal distance threshold, denoted as d^{opt} . *Best viewed in color.*

1. Introduction

In deep metric learning (DML) for visual recognition, distance calibration plays a critical role in determining the user-perceived model performance. Unlike confidence calibration in closed-set classification settings which focuses on aligning confidence probabilities with true likelihood of correctness in a fixed label space [27, 33], distance calibration in DML aims to pinpoint an optimal distance threshold to achieve a target true positive rate (TPR) or true negative rate (TNR) for diverse test-time distributions [26]. This calibration is vital because, even with a highly effective

embedding model, an inappropriate distance threshold can significantly degrade user experience. The issue becomes more pronounced in open-world recognition, where the embedding model, trained on a closed set of classes (e.g., dog, cat, bird), is tested on an open collection of unseen classes (e.g., wolf, sheep, mouse, ...). These open-world classes may have very different intra-class and inter-class representation structures, typically being less compact, compared to the training classes. In such open-world scenarios, where prior knowledge about test-time classes and distributions are absent, calibrating the distance threshold becomes a challenging task, as illustrated in Fig. 1. We term this task of distance calibration in an open-world scenario as **open-world threshold calibration**.

Existing posthoc calibration methods, such as [16, 24, 34, 37, 53, 54], typically utilize a fully-labeled calibration dataset that has a similar distribution as the test data [35, 42, 56] to learn general calibration rules for test distributions. However, this approach has a key limitation: it heavily relies on the assumption of identical distributions between test and calibration data for effective calibration. In open-world scenarios, this assumption becomes unreliable, posing significant challenges to threshold calibration, including:

1. **The open-world challenge** The test data may exclusively contain open-world classes, which exhibit different relationships between distance thresholds and TPR or TNR compared to those encountered during the embedding model training [26]. Meanwhile, test data composition and quality can vary significantly, potentially exhibiting substantial class imbalances and data corruptions.
2. **Non-stationary data** In real-world testing environments, the test distribution can be infinitely varied and highly dynamic, rendering the assumption of similar distribution between calibration and test data obsolete.
3. **Deployment Scalability** Real-world systems require calibration methods that can adapt to diverse user distributions without individual recalibration. Existing methods lack deployment scalability as they frequently require dedicated calibration data and the creation of specific calibration functions for each user. Imagine a scenario with 1,000 user profiles with distinct classes and data distributions – creating and deploying custom calibration datasets and functions for each would be impractical.

Addressing these challenges is crucial for the reliability of DML-based open-world recognition systems. Current posthoc calibration methods are ill-suited for this purpose, as they are inherently inductive and prone to failure when confronted with test data with different distance distributions from the calibration data. To address this, we adopt a fresh perspective on distance threshold calibration, treating it as a transductive inference process, where the calibration method incorporates the information of the unlabeled test samples along with the learned calibration rules to make better threshold estimations. Our proposed method, *OpenGCN*, employs a Graph Neural Network (GNN), known for its generalization capabilities [3, 5, 9, 10, 49, 51, 52], to jointly predict pairwise connectivity and two instance-wise representation densities for test data, where the predicted pairwise connectivity is used to compute the TPR and TNR of the test data at each distance threshold to enable transductive threshold calibration. *OpenGCN* is tailored for the task of open-world threshold calibration through a carefully crafted learning process, which accurately estimates the mapping between performance metrics and pairwise distance thresholds in open-world scenarios. In particular, the multi-task learning of connectivities and representation densities facilitates infor-

mation sharing, which helps enhance the model’s generalization to open-world scenarios [49, 52]. Additionally, our joint prediction design incorporates two types of density metrics, addressing both intra-class and inter-class connectivity estimations. This approach, as opposed to using a single density metric, is shown to enhance calibration performance, as illustrated in Sec. 4.3. Furthermore, *OpenGCN* adopts a two-stage training process. It pre-trains on a large closed-world dataset, followed by fine-tuning on a small open-world calibration dataset with disjoint classes to both the closed-world and test data, to adapt the model to be aware of the open-world context. By these design choices, *OpenGCN* sidesteps the requirement for calibration data to have a similar distance distribution¹ as the test data, significantly improving calibration performance in open-world scenarios. To summarize, our contributions are as follows:

1. We are, to the best of our knowledge, the first to formally define the *open-world threshold calibration* problem.
2. We propose *Transductive Threshold Calibration (TTC)*, a new threshold calibration paradigm that diverges from traditional inductive posthoc calibration methods, which does not rely on the assumption of similar distance distributions between the test and calibration data.
3. We introduce *OpenGCN*, a GNN-based TTC method tailored for open-world threshold calibration against diverse test distributions. We build comprehensive evaluation protocols with and without distance distribution shifts to assess *OpenGCN*’s performance. The evaluation result underscores *OpenGCN*’s effectiveness and robustness in real-world testing environments.

2. Problem Definition and Related Works

We first introduce some notations and formalize the open-world threshold calibration problem. Let D_{labeled} be a labeled dataset consisting of two disjoint subsets: D_{train} and D_{cal} , and let D_{test} be an unlabeled dataset. In open-world scenarios, the class sets of D_{train} , D_{cal} , and D_{test} , denoted as C_{train} , C_{cal} , and C_{test} , are disjoint, i.e., $C_{\text{train}} \cap C_{\text{cal}} = C_{\text{train}} \cap C_{\text{test}} = C_{\text{cal}} \cap C_{\text{test}} = \emptyset$. The goal of open-world threshold calibration is to find a suitable distance threshold that achieves the target TPR and TNR for D_{test} , given an embedding model trained on D_{train} . We approach this as a constrained optimization task, with the objective being maximizing the metric of interest. Take optimizing for TNR with a minimum TPR requirement as an example, this problem can be formulated as follows:

$$\underset{d}{\text{maximize}} \text{ TNR}_{\text{test}}, \text{ subject to } \text{TPR}_{\text{test}}(d) \geq \alpha \quad (1)$$

where d is the distance threshold, and α is the minimum performance requirement for TPR_{test} . Due to the inherent

¹We use “distance distribution” to refer to the distribution of pairwise distances between L2-normalized embeddings from a trained DML model.

trade-off between TPR and TNR, the objective in Eq. (1) is equivalent to finding an optimal distance threshold d^{opt} for which $\text{TPR}_{\text{test}}(d^{\text{opt}}) = \alpha$. To solve this, we express TPR_{test} and TNR_{test} at a distance threshold d as follows:

$$\text{TPR}_{\text{test}}(d) = \frac{\sum_{i,j \in D_{\text{test}}} 1_{y_i=y_j} \cdot 1_{d_{ij} < d}}{\sum_{i,j \in D_{\text{test}}} 1_{y_i=y_j}} \quad (2)$$

$$\text{TNR}_{\text{test}}(d) = \frac{\sum_{i,j \in D_{\text{test}}} 1_{y_i \neq y_j} \cdot 1_{d_{ij} > d}}{\sum_{i,j \in D_{\text{test}}} 1_{y_i \neq y_j}} \quad (3)$$

where d_{ij} is the L2 distance between the embeddings of samples i and j , and y_i is the label for sample i . The symbol $1_{\text{condition}}$ represents the indicator function which equals 1 if the condition is met, otherwise 0. With TPR_{test} and TNR_{test} calculated at each distance threshold, we can optimize for the optimal distance threshold d^{opt} to achieve the target performance metrics, as described in Eq. (1).

2.1. Related Works

Open-world Recognition [29] aims to learn discriminative representations that align distances between representations with their semantic similarities. This allows for effective generalization to diverse, previously unseen open-world classes during testing, setting it apart from closed-set classification where training and testing classes are the same. Popular recognition losses [6, 12, 36] typically encourage compact intra-class representations, promoting strong affinity within each class while maintaining separation from other classes. However, it is widely observed that these losses tend to produce highly varied intra-class and inter-class representation structures across classes and distributions [32, 39, 55], necessitating threshold calibration to ensure consistent performance across diverse users.

Posthoc Calibration We focus on posthoc calibration methods which are more relevant to our research. Generally, existing posthoc calibration methods fall into two categories: (i) non-parametric methods like isotonic regression [54] and histogram binning [34, 53]; and (ii) parametric methods such as Platt scaling [37] and temperature scaling [16]. These methods are inductive: they rely on a hold-out calibration set with similar distribution as the test data to derive general rules for fine-tuning the decision threshold, aiming to align the performance metrics with a predefined target. While effective in closed-set classification, these methods struggle in scenarios with significant distribution differences between test and calibration data. Diverging from traditional methods, another group of methods such as conformal prediction [4, 15, 40, 43] or Prediction-Powered Inference [1] emphasize confidence coverage guarantees, and has been shown applicable even beyond the setting of exchangeable data [14, 15]. However, these methods inherently assume a closed-set setting, making them unsuitable for open-world scenarios. Currently, open-world posthoc calibration remains largely under-explored.

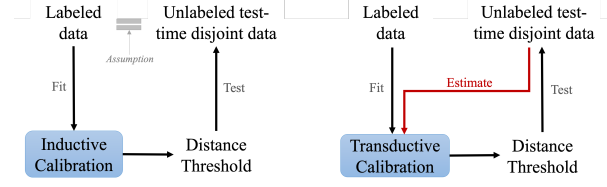


Figure 2. This figure distinguishes between (left) inductive and (right) transductive threshold calibration methods in open-world scenarios with disjoint test-time classes. Inductive methods rely on a labeled hold-out dataset with the same distance distribution as the test data to learn general calibration rules. Transductive methods, however, also use the test information for more specific calibration, as indicated by the red arrow. *Best viewed in color.*

Transductive Inference Transduction is the reasoning from observed, specific (training) cases to specific (test) cases [45]. Such an approach is desirable as it alleviates the problem of overfitting on limited support set since information from the test data is also used for inference. This is also known as increasing VC-dimension for structural risk minimization in classical statistical learning [19]. Recently, a large body of works investigated transductive inference for few-shot and open-world recognition tasks [8, 18, 28, 38], where significant increases in performances have been reported. Given the relevance of these tasks, it is worthwhile to reconsider existing inductive posthoc calibration methods for distance threshold calibration in open-world scenarios.

3. Methodology

3.1. Transductive Threshold Calibration

Traditional calibration methods are inherently inductive – they rely on a calibration dataset to learn general calibration rules under the assumption of identically distributed data. However, in open-world scenarios, this assumption seldom holds, as the test distribution is unknown and can be infinitely varied and highly dynamic. To improve calibration specificity in the open world, it is natural to adopt a transductive approach, where the TPR and TNR estimations directly involve the test data, rather than relying on a separate calibration dataset that might not accurately represent the test data. As illustrated in Fig. 2, a transductive approach allows the calibration model to “see” the unlabeled test data when deciding on the distance threshold, contrasting with the traditional inductive methods which are “blind” to the test data. We term this approach as *Transductive Threshold Calibration (TTC)*, and the traditional inductive calibration methods as *Inductive Threshold Calibration (ITC)*.

To overcome the limitations of ITC methods, we propose OpenGCN, a GNN-based TTC method with enhanced adaptability and robustness for open-world scenarios with diverse concepts and distance distributions. We highlight the key differences between OpenGCN and conventional

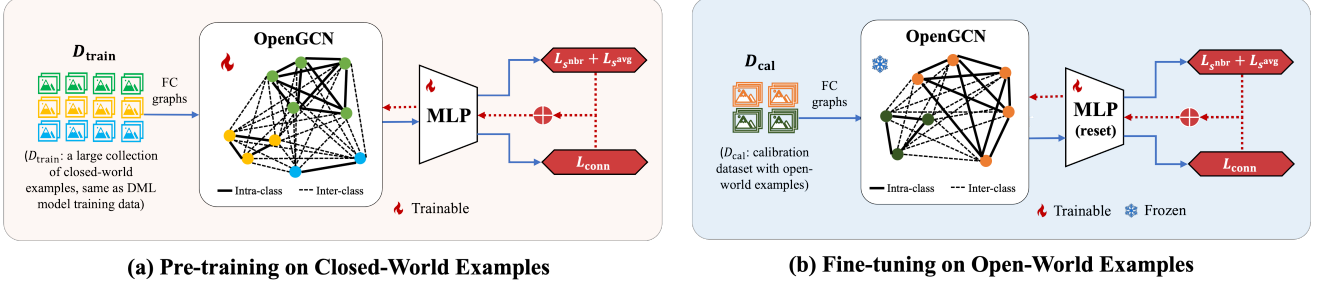


Figure 3. OpenGCN training workflow: (a) During pre-training, OpenGCN jointly optimizes pairwise connectivity, and instance-specific neighborhood and average densities. (b) During fine-tuning, the 2-layer MLP is reset for fine-tuning, while the other weights remain frozen. Solid blue and dashed red arrows represent forward and backward propagation, respectively. At test time, we employ the trained OpenGCN model and MLP head to predict the TPR and TNR as functions of each distance threshold specifically for each test distribution. We then follow Eq. (1) and use grid search to find the optimal distance threshold for each test dataset. *Best viewed in color.*

ITC methods. First, OpenGCN, as a transductive method, derives distance thresholds by leveraging information directly from the test data. This empowers it to adapt to the characteristics of the test data, thereby eliminating the requirement for the calibration data to share a similar distribution with the test data. Second, OpenGCN is engineered to integrate useful information from both closed-world and open-world data sources. This is achieved through a two-stage training process, as illustrated in Fig. 3. We first pretrain OpenGCN on a closed-world dataset, which is the same dataset used to train the DML embedding model. Afterwards, we fine-tune it on a smaller calibration dataset. This calibration dataset contains open-world classes that do not overlap with those in the test data or the closed-world pretraining data. This approach allows the model to smoothly transition from a closed-world context to open-world scenarios, effectively utilizing closed-world knowledge to enhance its transductive reasoning capabilities in the dynamic and unknown open world. In the next section, we delve into the details of OpenGCN, elaborating on how it enables effective TTC for open-world scenarios.

3.2. OpenGCN: Learning for Effective TTC

OpenGCN Inference Workflow A straight-forward way to estimate TPR_{test} and TNR_{test} , as defined in Eqs. (2) and (3), is to model the true pairwise connectivities with edge connectivity probability [52]. This probability, denoted as p_{ij} , quantifies the likelihood that two samples have the same label. By setting a proper connectivity threshold τ , we can approximate TPR_{test} and TNR_{test} as follows:

$$\hat{\text{TPR}}_{\text{test}}(d) = \frac{\sum_{i,j \in D_{\text{test}}} 1_{p_{ij} > \tau} \cdot 1_{d_{ij} < d}}{\sum_{i,j \in D_{\text{test}}} 1_{p_{ij} > \tau}} \quad (4)$$

$$\hat{\text{TNR}}_{\text{test}}(d) = \frac{\sum_{i,j \in D_{\text{test}}} 1_{p_{ij} \leq \tau} \cdot 1_{d_{ij} > d}}{\sum_{i,j \in D_{\text{test}}} 1_{p_{ij} \leq \tau}} \quad (5)$$

These formulations offer a TTC solution that centers on precisely predicting pairwise connectivities for open-world test distributions, a problem well-suited for modern

deep learning algorithms. Specifically, as shown in Fig. 3, OpenGCN is designed as a GNN-based method for predicting pairwise connectivities over graph data constructed from the unlabeled test samples. We adopt a GNN architecture, specifically a Graph Attention Network (GAT) [46], due to its demonstrated effectiveness in generalizing to open-world scenarios [3, 5, 9, 10, 49, 51, 52]. Additionally, we use fully connected graphs to ensure that in-graph pairwise distance distribution is representative of the overall pairwise distance distribution. For inference, nodal features extracted by the GAT encoder are concatenated with the original DML embedding features [52] and passed through a 2-layer MLP to predict pairwise connectivities. The connectivity predictions are then used to transductively estimate TPR and TNR at each distance threshold for the test distributions, following the formulations in Eqs. (4) and (5), where the connectivity threshold τ is selected by 10-fold cross validation on D_{cal} . Due to the typically large size of the test data, for efficient inference, we randomly sample subsets from D_{test} to construct fully connected sub-graphs for connectivity inference, repeating this process until the TPR and TNR estimations converge.

Joint Connectivity and Density Estimations Using representation density prediction as an auxiliary task to enhance connectivity prediction is widely used in clustering tasks [2, 7, 31]. This approach is based on the idea that a cluster typically exists within a contiguous region of high sample density, separated from other clusters. Recent supervised visual clustering works also leverage density as a key modeling parameter to enhance clustering performance by encouraging information sharing between the tasks [49, 50, 52]. Driven by the intrinsic connections between density and connectivity, we adopt a multi-task approach, where we simultaneously learn for pairwise edge connectivity and instance-wise representation densities. However, unlike previous works which only consider one density metric, we simultaneously learn two density

metrics: the average density (s^{avg}), and the neighborhood density (s^{nbr}). Formally, these two density metrics, defined in [52], can be expressed as follows²:

$$s_i^{\text{avg}} = \frac{\sum_{j \in \mathcal{N}_i} a_{ij} \cdot 1_{y_i=y_j}}{|\mathcal{N}_i|}, \quad s_i^{\text{nbr}} = \frac{\sum_{j \in \mathcal{N}_i} a_{ij} \cdot (1_{y_i=y_j} - 1_{y_i \neq y_j})}{|\mathcal{N}_i|} \quad (6)$$

where $\mathcal{N}_i$ denotes the neighbourhood of a sample i , and a_{ij} represents the cosine similarity between the original embedding features of sample i and sample j .

To illustrate the motivation of utilizing both density metrics instead of just one, we first introduce two metrics adapted from prior works [20, 41], namely the class-specific TPR and TNR scores, denoted as TPR^k and TNR^k , respectively. Let f_i denote the L_2 -normalized embeddings of an image in a dataset D . For a given class k , its class-specific TPR and TNR scores can be expressed as:

$$\text{TPR}^k = \frac{\|\sum_{i \in D} f_i \cdot 1_{y_i=k}\|}{\sum_{i \in D} 1_{y_i=k}}, \quad \text{TNR}^k = \frac{\sum_{i,j \in D} (1 - a_{ij}) \cdot 1_{y_j \neq y_i=k}}{\sum_{i,j \in D} 1_{y_j \neq y_i=k}} \quad (7)$$

The subsequent theorems formally establish a connection between two density metrics defined in Eq. (6) and the class-specific TPR and TNR scores.

Theorem 1 (Correspondence between s^{avg} and TPR^k)

Let $\mathcal{N}$ be a cluster with high purity, where the majority class is k . For each sample $i \in \mathcal{N}$, when both $|\mathcal{N}|$ and $|\mathcal{N}_i|$ are sufficiently large, TPR^k can be approximated as:

$$\lim_{|\mathcal{N}_i| \rightarrow \infty} \text{TPR}^k = \left(\frac{|\mathcal{N}_i|}{2|\mathcal{N}|} \cdot \left(\underbrace{\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} s_i^{\text{nbr}}}_{\text{avg } s^{\text{nbr}}} + \underbrace{\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} a_i^{\text{avg}}}_{\text{avg } a^{\text{avg}}} \right) \right)^{1/2} \quad (8)$$

$2 \times \text{avg } s^{\text{avg}}$

where $a_i^{\text{avg}} = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} a_{ij}$, and a^{avg} is the mean of average cosine similarity of all vertices in $\mathcal{N}_i$.

Theorem 2 (Correspondence between $s^{\text{avg}} - s^{\text{nbr}}$ and TNR^k) Under the same assumptions in Theorem 1, for a given class k , its TNR^k can be approximated as:

$$\lim_{|\mathcal{N}_i| \rightarrow \infty} \text{TNR}^k = 1 - \frac{|\mathcal{N}|}{|\mathcal{N}|_{k-}} \cdot \left(\underbrace{\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} s_i^{\text{avg}} - \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} s_i^{\text{nbr}}}_{\text{average } (s^{\text{avg}} - s^{\text{nbr}})} \right) \quad (9)$$

where $|\mathcal{N}|_{k-}$ denotes the number of negative pairs in $\mathcal{N}$ where one sample of each negative pair must have label k .

Based on the theorems, when the neighborhood size is sufficiently large, considering both density metrics effectively encapsulates both class-specific TPR and TNR within

²Although the original definition of s^{avg} in [52] requires a neighborhood size that includes all samples belonging to a given class, it can be shown by stochastic convergence of random variables that our definition is a tight approximation for [52] when $|\mathcal{N}_i|$ is sufficiently large.

this neighbourhood. As open-world threshold calibration aims to balance the TPR and TNR trade-off for unknown test distributions, it is crucial to capture both aspects to improve within-class and cross-class connectivity predictions. Furthermore, the class-specific nature of these metrics grants them the versatility to adapt to varying class compositions. In Sec. 4.3, we provide an ablation study comparing the use of a single density metric versus both densities, where jointly predicting both densities along with connectivity yields better calibration performance. Thus, we introduce predictions of both density metrics, s^{avg} and s^{nbr} , as auxiliary tasks to enhance the generalization of connectivity prediction. This leads to the following learning objective for training OpenGCN:

$$\mathcal{L}_{\text{overall}} = \underbrace{\mathcal{L}_{\text{conn}}}_{\text{main task}} + \lambda \cdot \underbrace{(\mathcal{L}_{s^{\text{nbr}}} + \mathcal{L}_{s^{\text{avg}}})}_{\text{auxiliary task}} \quad (10)$$

where $\mathcal{L}_{\text{conn}}$ is the balanced cross-entropy loss for pairwise edge connectivity and $\mathcal{L}_{s^{\text{nbr}}}$ and $\mathcal{L}_{s^{\text{avg}}}$ are the mean squared error losses for s^{nbr} and s^{avg} , respectively. Specifically, we define $\mathcal{L}_{\text{conn}}$ as follows to ensure equal importance for both within-class and cross-class connectivities:

$$\mathcal{L}_{\text{conn}} = \frac{\sum_{i,j \in V} 1_{y_i=y_j} \cdot \log(p_{ij})}{\sum_{i,j \in V} 1_{y_i=y_j}} + \frac{\sum_{i,j \in V} 1_{y_i \neq y_j} \cdot \log(1 - p_{ij})}{\sum_{i,j \in V} 1_{y_i \neq y_j}} \quad (11)$$

Meanwhile, $\mathcal{L}_{s^{\text{nbr}}}$ and $\mathcal{L}_{s^{\text{avg}}}$ can be expressed as:

$$\mathcal{L}_{s^{\text{avg}}} = \frac{\sum_{i \in V} (s_i^{\text{avg}} - \hat{s}_i^{\text{avg}})^2}{|V|}, \quad \mathcal{L}_{s^{\text{nbr}}} = \frac{\sum_{i \in V} (s_i^{\text{nbr}} - \hat{s}_i^{\text{nbr}})^2}{|V|} \quad (12)$$

where V represents the node vertices in the graph data, and $\hat{s}_i$ is the estimated density for each sample based on p_{ij} .

Two-stage Training for Adaptability The DML embedding model, trained on D_{train} (closed-set examples), tends to produce more compact embeddings for these examples than those of open-world classes. If OpenGCN is trained solely on D_{train} , its ability to generalize to the open-world scenarios will be limited. On the other hand, if OpenGCN is trained solely on D_{cal} , its knowledge may be very narrow since the calibration dataset is typically small and lacks diverse concepts. To tackle this, we borrow established experience in domain generalization and adaptation [11, 21, 48], and adopt a two-stage training strategy. First, we pretrain OpenGCN on D_{train} , which consists of a large collection of closed-set examples. After this, we reset the 2-layer MLP while keeping the other parameters frozen. Subsequently, we fine-tune the MLP on D_{cal} , a small open-world calibration dataset containing disjoint classes to the test data, to adapt the pretrained model to open-world scenarios. In Sec. 4.3, we conduct an ablation study to provide further support for this two-stage training approach. We choose to

fine-tune only the MLP based on the practical observations that D_{cal} is typically limited in size, and fine-tuning the entire model on such a small dataset may lead to overfitting. It is worth reiterating that this approach does not require additional training data, as the closed-set data is already in place for training the DML embedding model, and the separate open-world calibration dataset is required for conventional inductive posthoc calibration methods as well.

4. Experiment and Result

We experiment on public recognition benchmarks including iNaturalist-2018 [44], CUB-200 [47] and Cars-196 [23]. Below, we outline our setup and present the results. Further experiments can be found in the supplementary materials.

4.1. Dataset and Implementation Details

Datasets To simulate real-world testing environments, we consider three calibration scenarios: SameDist, ShiftDist and DiffDist. The SameDist scenario involves cases where D_{cal} and D_{test} share similar distance distributions, the ShiftDist scenario accounts for test-time non-semantic distance distribution shifts, and the DiffDist scenario represents out-of-distribution calibration, where D_{cal} and D_{test} have very different distance distributions. Note that in all three scenarios, we adhere to the open-world setting where $C_{\text{train}} \cap C_{\text{cal}} = C_{\text{train}} \cap C_{\text{test}} = C_{\text{cal}} \cap C_{\text{test}} = \emptyset$. Below, we elaborate on the setup for each calibration scenario:

- **SameDist** For iNaturalist, the training and testing classes are distinct, so we directly use the training partition as D_{train} . To create D_{cal} , we randomly select 10% of the test classes, leaving the remaining classes for D_{test} . For CUB and Cars, where there is overlap between training and testing classes, we divide them into train / cal / test subsets. The train set comprises the first half of the class indices, while the cal / test sets are randomly chosen from the remaining classes with a 1/9 ratio. As D_{cal} and D_{test} are randomly split from the same dataset, they are expected to have similar distance distributions.
- **ShiftDist** We consider 13 common image corruption and perturbation types, including noise, blur, weather, and digital distortions, to assess the robustness of the calibration methods under varied adversities. We follow the setups in [17] and apply the changes to D_{test} only, while leaving D_{cal} and D_{train} unchanged.
- **DiffDist** To induce significant distance distribution shifts between D_{cal} and D_{test} , we employ the following treatments. For iNaturalist, characterized by a long-tailed distribution, we divide its test classes into two sets based on cluster size, each containing approximately the same number of images. For calibration purposes, we use the set with a higher number of images per class (“head” set, denoted as D_{head}) as D_{cal} and the set with fewer images per class (“tail” set, denoted as D_{tail}) as D_{test} to simu-

Table 1. Detailed statistics of the datasets.

Setting	Dataset	Partition	# img	# cls	# img/cls
SameDist	Cars	D_{train}	7,961	98	81.2
		D_{cal}	866	10	86.6
		D_{test}	7,356	88	83.6
	CUB	D_{train}	5,802	99	58.6
		D_{cal}	599	10	59.9
		D_{test}	5,385	91	59.2
	iNat	D_{train}	324,418	5,690	57.0
		D_{cal}	12,613	245	51.5
		D_{test}	123,047	2,207	55.8
ShiftDist	Cars	SameDist except for corruption on D_{test}			
DiffDist	iNat	D_{train}	iNat SameDist D_{train}		
		D_{head}	70,057	200	350.3
		D_{tail}	66,036	2,252	29.3
	Cars	SameDist except for sketchifying D_{test}			
		D_{train}	iNat SameDist D_{train}		
		D_{cal}	iNat SameDist D_{cal}		
	iNat/CUB (cross dataset)	D_{test}	Entire CUB dataset		

late a calibration for the long tail scenario. In addition, we also explore two out-of-domain calibration scenarios. First, for Cars, we transform D_{test} into sketches while leaving D_{train} and D_{cal} untouched. Second, we consider cross-dataset calibration, where the OpenGCN model is pretrained and fine-tuned on iNaturalist (general natural species images) but evaluated on CUB (bird images).

Evaluation Metrics For a comprehensive evaluation, we consider two approaches to assess calibration performance:

- **Global Evaluation:** Since we define open-world threshold calibration as the accurate prediction of both TPR and TNR at each distance threshold to meet specific TPR or TNR performance requirements of diverse test-time users, it is natural to employ the combined Mean Absolute Errors (MAE) for both TPR and TNR predictions across the entire distance range as our evaluation metric. Formally, this metric can be expressed as:

$$\text{MAE}_{\text{comb}} = \frac{1}{2} \int_0^2 (|\hat{\text{TPR}}(d) - \text{TPR}(d)| + |\hat{\text{TNR}}(d) - \text{TNR}(d)|) dd \quad (13)$$

- **Point-wise Evaluation:** We first set a performance target and compute the optimal distance threshold, denoted as $\hat{d}^{\text{opt}}$, based on the TPR or TNR estimations. We then compute the Absolute Error (AE) between the actual performance at $\hat{d}^{\text{opt}}$ and the target, denoted as $\text{AE}_{\text{TPR}} = |\text{TPR}(\hat{d}^{\text{opt}}) - \text{TPR}_{\text{target}}|$ and $\text{AE}_{\text{TNR}} = |\text{TNR}(\hat{d}^{\text{opt}}) - \text{TNR}_{\text{target}}|$ for TPR and TNR, respectively.

Baseline Methods We consider the most representative inductive posthoc calibration methods including Platt Scaling [37], Histogram Calibration [53], Isotonic Calibration [54] and Beta Calibration [24]. Additionally, we explore pseudolabel-based baselines, including traditional clustering methods such as DBSCAN [13] and the state-of-the-art method in GNN-based clustering, HiLANDER [49]. For clustering-based methods, we follow

Table 2. Evaluation in the SameDist scenario using pointwise metrics of AE_{TPR} (optimize for TPR) and AE_{TNR} (optimize for TNR). The smaller the metric, the better. For each dataset, the best and second best results are marked in **Red** and **Blue**, respectively. Shading in the Table: Gray for posthoc calibration baselines, **Cyan** for clustering baselines, and **Blue** for our OpenGCN method. *Best viewed in color.*

Method	Optimize for TPR=80%			Optimize for TPR=90%			Optimize for TNR=80%			Optimize for TNR=90%			Rank
	Cars	CUB	Inat	Cars	CUB	Inat	Cars	CUB	Inat	Cars	CUB	Inat	
Platt scaling [37]	1.35%	5.10%	6.08%	0.44%	2.63%	4.63%	2.83%	2.02%	7.54%	2.93%	6.49%	0.92%	6
Beta calibration [24]	1.13%	5.16%	5.51%	0.02%	2.91%	3.26%	2.94%	1.41%	7.57%	2.78%	6.43%	0.93%	5
Isotonic regression [54]	0.82%	5.28%	4.53%	0.90%	2.56%	3.54%	1.94%	1.00%	5.78%	1.26%	4.65%	0.65%	3
Histogram Calibration [53]	0.82%	5.28%	4.53%	0.90%	2.56%	3.54%	1.94%	1.00%	5.78%	1.26%	4.65%	0.65%	4
DBSCAN [13]	43.11%	18.87%	0.45%	34.57%	9.18%	1.85%	4.09%	13.77%	12.90%	1.60%	9.32%	9.32%	7
Hi-LANDER [49]	3.44%	1.36%	10.54%	2.02%	0.93%	7.00%	0.06%	0.38%	2.35%	0.10%	2.20%	0.21%	2
OpenGCN (ours)	0.33%	0.74%	1.59%	0.72%	1.41%	2.37%	0.61%	0.09%	0.74%	0.58%	0.72%	0.10%	1

their original clustering decoding inference workflows to estimate pseudo labels, and use these pseudo labels to compute TPR_{test} and TNR_{test} for finding d^{opt} .

Implementation Details In all experiments, we train ResNet-50 models with 128-dimensional embeddings on D_{train} using the setups in [6]. The embedding models are then used to extract the embeddings for D_{train} , D_{cal} and D_{test} . For training OpenGCN, as implied in Theorems 1 and 2, the neighborhood size needs to be sufficiently large to encapsulate both intra-class and inter-class representation structures. Thus, we use a batch sizes of 256 for graph construction during training. We use the Adam optimizer [22] with a cosine annealing schedule [30]. For traditional calibration methods, we use the official codebase from [25] to map the ground truth TPR (or TNR) as a function of the distance threshold from D_{train} to D_{cal} . When doing point-wise evaluation, the optimal distance threshold d^{opt} is solved with grid search at a grid size of 0.01. Further details are provided in the supplementary materials.

4.2. Evaluation Results

SameDist Calibration We present the global and pointwise evaluation results for the SameDist scenario in Tab. 3 and Tab. 2, respectively. For pointwise evaluation, we evaluate at multiple target values (TPR=80%, 90% and TNR=80%, 90%) to provide a comprehensive assessment. Our results reveal that no single calibration method consistently excels across all distance thresholds and datasets. However, on average, OpenGCN achieves the highest rank. This underscores the importance of TTC in open-world scenarios, where calibration is conducted based on the characteristics of D_{test} rather than relying on a calibration dataset that may not accurately represent D_{test} . Additionally, the global metrics in Tab. 3 show that, compared to the best baseline method, OpenGCN significantly reduces global error rates by 59.30%, 66.49%, and 59.15% for Cars, CUB, and iNaturalist, respectively. Among the baseline methods, we observe that DBSCAN performs worse than the traditional posthoc calibration methods, while Hi-LANDER outperforms traditional posthoc methods on Cars and CUB but underperforms on iNaturalist. In contrast, OpenGCN con-

Table 3. Evaluation in the SameDist scenario using the global error metric of MAE_{comb} . For each benchmark, the best and second best results are marked in **Red** and **Blue**, respectively. We also report the improvement in error reduction of OpenGCN over the best baseline method. *Best viewed in color.*

Method	Cars	CUB	iNat	Rank
Platt scaling	1.55e-2	3.59e-2	1.23e-2	6
Beta calibration	1.53e-2	3.59e-2	1.18e-2	5
Isotonic regression	1.38e-2	3.61e-2	1.18e-2	3
Histogram calibration	1.38e-2	3.62e-2	1.18e-2	4
DBSCAN	1.02e-1	1.10e-1	3.65e-2	7
Hi-LANDER	1.29e-2	1.94e-2	2.14e-2	2
OpenGCN (ours)	5.25e-3	6.50e-3	4.82e-3	1
Imp. over top baseline $\uparrow$	59.30%	66.49%	59.15%	69.14% (avg.)

sistently performs well across all three datasets.

ShiftDist Calibration In Tab. 4, we report the global error metric MAE_{comb} for each corruption type across various calibration methods. Among the baseline methods, Isotonic Regression and Histogram Calibration appear to be the most effective in the presence of image corruptions. However, it is evident that OpenGCN consistently outperform these baseline methods across all corruption types, achieving an average error reduction of 55.03% compared to the best baseline method. This robust performance against image corruptions can be attributed to the model’s pretraining stage, where it was exposed to closed-set data with similar types of corruptions. Additionally, it is observed that, among the various corruption categories, OpenGCN exhibits the most improvement in the weather category, while showing the least improvement in the blur category.

DiffDist Calibration We present the DiffDist calibration results in Tab. 5. As observed, in this scenario characterized by a substantial shift in distance distributions between D_{cal} and D_{test} , all calibration methods display elevated errors compared to the SameDist scenario. However, OpenGCN demonstrates superior performance compared to the other calibration methods in both out-of-domain settings (sketch and cross-dataset) and the long-tail calibration setting, achieving an average relative reduction in the global error MAE_{comb} of 43.99%. In particular, we observe significant improvement in the cross-dataset setting (pretrained

Table 4. Evaluation on the Cars-196 dataset in the ShiftDist scenario across 13 common corruption and perturbation types using combined global error metric of MAE_{comb} . The best results are marked in **Red**.

Method	Noise			Blur			Weather			Digital				Rank
	Gauss	Shot	Impulse	Defocus	Motion	Zoom	Snow	Fog	Bright	Contrast	Elastic	Pixel	JPEG	
Platt scaling	2.95e-2	2.99e-2	3.41e-2	2.66e-2	2.62e-2	5.02e-2	4.24e-2	4.37e-2	2.16e-2	4.61e-2	2.16e-2	2.24e-2	2.03e-2	4
Beta calibration	2.94e-2	2.97e-2	3.41e-2	2.67e-2	2.69e-2	5.06e-2	4.32e-2	4.37e-2	2.18e-2	4.61e-2	2.20e-2	2.23e-2	2.02e-2	5
Isotonic regression	2.88e-2	2.85e-2	3.38e-2	2.37e-2	2.31e-2	4.85e-2	4.07e-2	4.34e-2	1.83e-2	4.59e-2	1.85e-2	2.03e-2	1.80e-2	2
Histogram calibration	2.88e-2	2.85e-2	3.38e-2	2.37e-2	2.31e-2	4.85e-2	4.07e-2	4.34e-2	1.83e-2	4.59e-2	1.85e-2	2.03e-2	1.80e-2	3
DBSCAN	4.96e-2	6.02e-2	7.79e-2	9.81e-2	1.13e-1	1.22e-1	1.19e-1	4.02e-2	9.27e-2	4.53e-2	1.09e-1	1.04e-1	8.21e-2	7
Hi-LANDER	7.65e-2	6.30e-2	6.59e-2	3.98e-2	5.33e-2	4.48e-2	5.94e-2	7.16e-2	5.09e-2	9.45e-2	4.42e-2	9.48e-2	6.91e-2	6
OpenGCN (ours)	1.33e-2	5.87e-3	1.66e-2	1.50e-2	1.71e-2	3.92e-2	1.42e-2	7.32e-3	6.73e-3	7.08e-3	5.34e-3	1.15e-2	1.68e-2	1
Imp. over top baseline $\uparrow$	53.82%	79.40%	50.89%	36.71%	25.97%	12.50%	65.11%	81.79%	63.22%	84.37%	71.14%	43.35%	6.67%	55.03% (avg)



Table 5. Evaluation in the DiffDist scenario using the global error metric MAE_{comb} . The best results are highlighted in **Red**.

Method	Cars: Sketch	CUB: Cross-dataset	iNat: Longtail
Platt scaling	1.08e-1	1.15e-1	2.09e-2
Beta calibration	1.08e-1	1.15e-1	2.12e-2
Isotonic regression	1.08e-1	1.15e-1	2.11e-2
Histogram Calibration	1.08e-1	1.15e-1	2.11e-2
DBSCAN	5.16e-2	1.60e-1	7.21e-2
Hi-LANDER	6.67e-2	1.30e-1	6.26e-2
OpenGCN (ours)	3.54e-2	1.42e-2	1.82e-2
Imp. over top baseline $\uparrow$	31.40%	87.65%	12.92%

Table 6. Impact of multi-task learning on global error metric MAE_{comb} on iNaturalist-2018. We use $\lambda = 10$ for all experiments.

	Best baseline	OpenGCN loss ablations			
		$\mathcal{L}_{conn}$	$+\lambda \cdot \mathcal{L}_{s^{avg}}$	$+\lambda \cdot \mathcal{L}_{s^{nbr}}$	$+\lambda \cdot (\mathcal{L}_{s^{avg}} + \mathcal{L}_{s^{nbr}})$
MAE_{comb}	1.18e-2	6.25e-3	5.37e-3	5.12e-3	4.82e-3

Table 7. Impact of fine-tuning on open-world calibration dataset on global error metric MAE_{comb} . PT: pretraining FT: finetuning. Numbers in the bracket show the relative improvement over PT.

Method	Cars	CUB	iNat
OpenGCN (PT)	2.90e-2	2.52e-2	3.55e-2
OpenGCN (PT+FT)	5.25e-3 (81.9%)	6.50e-3 (74.2%)	4.82e-3 (86.4%)

and finetuned on the iNaturalist-2018 nature species dataset but tested on the CUB birds dataset), where OpenGCN achieves a notable error reduction of 87.65%.

4.3. Ablation Studies

Importance of Multi-task Learning We assess the impact of multi-task learning on MAE_{comb} . As shown in Tab. 6, compared to predicting connectivity only, employing a single density metric in conjunction with connectivity prediction helps reduce MAE_{comb} from 6.25e-3 to 5.37e-3 for s^{avg} and to 5.12e-3 for s^{nbr} , respectively. However, by utilizing both density metrics, we further decrease this error to 4.82e-3. This supports our choice to incorporate both density metrics, allowing us to capture both intra-class compactness and inter-class separation while facilitating information sharing for improved connectivity prediction.

Importance of Two-stage Training We assess the impact

of two-stage training on OpenGCN by comparing MAE_{comb} before and after fine-tuning on D_{cal} across all three benchmarks. The comparison in Tab. 7 reveals significant error reduction of up to 86.4% after fine-tuning on the open-world calibration dataset. This results supports our choice of two-stage training in adapting the calibration model from the closed-world context to the open-world scenarios.

5. Conclusions

In this work, we formally define the open-world threshold calibration problem for DML-based open-world visual recognition systems. To address this problem, we introduce OpenGCN, a GNN-based transductive threshold calibration method designed to enhance adaptability in open-world scenarios. Unlike traditional posthoc calibration methods, OpenGCN does not rely on the common assumption of matching distance distributions between D_{cal} and D_{test} . Instead, it leverages the information of the unlabeled test instances along with learnt calibration rules to predict pairwise connectivity of the test data, via a GNN, to enable effective transductive threshold calibration in open-world scenarios. Our evaluations demonstrate that OpenGCN outperforms both traditional posthoc calibration methods and pseudolabel-based calibration techniques. When assessed using global error metrics, OpenGCN exhibits significant improvements, achieving average error reductions of 69.14%, 40.85%, and 22.58% for SameDist, ShiftDist, and DiffDist calibration scenarios, respectively, compared to the best baseline method. Overall, our results underscore OpenGCN’s robustness across different distance distribution patterns between D_{cal} and D_{test} , highlighting its practical applicability for threshold calibration in DML-based open-world recognition applications.

Limitations OpenGCN is computationally less efficient and more susceptible to over-parameterization compared to traditional posthoc calibration methods. Furthermore, OpenGCN is not a calibration-data-free method as it still requires some calibration data in addition to the closed-world data used for training the embedding model.

References

- [1] Anastasios N. Angelopoulos, Stephen Bates, Clara Fan-njiang, Michael I. Jordan, and Tijana Zrnic. Prediction-powered inference, 2023.
- [2] 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.
- [3] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagan-nath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. *arXiv preprint arXiv:2102.06966*, 2021.
- [4] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ram-das, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.
- [5] Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. Size-invariant graph representations for graph classification extrapolations. In *International Conference on Machine Learning*, pages 837–851. PMLR, 2021.
- [6] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. *CoRR*, abs/2007.12163, 2020.
- [7] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining: 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II 17*, pages 160–172. Springer, 2013.
- [8] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. *arXiv preprint arXiv:2102.03526*, 2021.
- [9] 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.
- [10] Ching-Yao Chuang and Stefanie Jegelka. Tree mover’s distance: Bridging graph metrics and stability of graph neural networks. *Advances in Neural Information Processing Systems*, 35:2944–2957, 2022.
- [11] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.
- [12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.
- [13] 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.
- [14] Shai Feldman, Stephen Bates, and Yaniv Romano. Conformalized online learning: Online calibration without a hold-out set. *arXiv preprint arXiv:2205.09095*, 2022.
- [15] Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.
- [16] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 1321–1330. JMLR.org, 2017.
- [17] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.
- [18] Shell Xu Hu, Pablo G Moreno, Yang Xiao, Xi Shen, Guillaume Obozinski, Neil D Lawrence, and Andreas Damianou. Empirical bayes transductive meta-learning with synthetic gradients. *arXiv preprint arXiv:2004.12696*, 2020.
- [19] Thorsten Joachims et al. Transductive inference for text classification using support vector machines. In *Icml*, pages 200–209, 1999.
- [20] V. Mardia Kanti and Peter E. Jupp. Directional statistics. Wiley, 2000.
- [21] Donghyun Kim, Kaihong Wang, Stan Sclaroff, and Kate Saenko. A broad study of pre-training for domain generalization and adaptation. In *European Conference on Computer Vision*, pages 621–638. Springer, 2022.
- [22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.
- [24] 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.
- [25] Fabian Küppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020, pages 10124–10132, 2021.
- [26] Jiaheng Liu, Zhipeng Yu, Haoyu Qin, Yichao Wu, Ding Liang, Gangming Zhao, and Ke Xu. Oneface: one threshold for all. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, pages 545–561. Springer, 2022.
- [27] Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Generalized zero-shot learning with deep calibration network. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018.
- [28] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv preprint arXiv:1805.10002*, 2018.
- [29] Vincent Lonij, Amrith Rawat, and Maria-Irina Nicolae. Open-world visual recognition using knowledge graphs. *arXiv preprint arXiv:1708.08310*, 2017.

- [30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
- [31] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.
- [32] Timo Milbich, Karsten Roth, Samarth Sinha, Ludwig Schmidt, Marzyeh Ghassemi, and Bjorn Ommer. Characterizing generalization under out-of-distribution shifts in deep metric learning. *Advances in Neural Information Processing Systems*, 34:25006–25018, 2021.
- [33] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020.
- [34] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, page 2901–2907, 2015.
- [35] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.
- [36] Yash Patel, Giorgos Tolias, and Jiri Matas. Recall@k surrogate loss with large batches and similarity mixup. *CVPR*, 2022.
- [37] John Platt. Probabilistic outputs for svms and comparisons to regularized likelihood methods. *Advances in Large-Margin Classifiers*, 10(3):61–74, 1999.
- [38] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3603–3612, 2019.
- [39] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. *arXiv preprint arXiv:1511.05939*, 2015.
- [40] Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.
- [41] S. Sra. A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of $i_s(x)$. 2012.
- [42] Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.
- [43] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.
- [44] Grant Van Horn, Oisín 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.
- [45] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.
- [46] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [47] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [48] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.
- [49] 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.
- [50] Xiaopeng Yan, Riquan Chen, Litong Feng, Jingkan Yang, Huabin Zheng, and Wayne Zhang. Progressive representative labeling for deep semi-supervised learning. *arXiv preprint arXiv:2108.06070*, 2021.
- [51] Chenxiao Yang, Qitian Wu, Jiahua Wang, and Junchi Yan. Graph neural networks are inherently good generalizers: Insights by bridging gnns and mlps. *arXiv preprint arXiv:2212.09034*, 2022.
- [52] 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.
- [53] 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.
- [54] 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.
- [55] Qin Zhang, Linghan Xu, Qingming Tang, Jun Fang, Ying Nian Wu, Joe Tighe, and Yifan Xing. Threshold-consistent margin loss for open-world deep metric learning, 2024.
- [56] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16489–16498, 2021.