ADAGEO: ADAPTIVE LOCAL GEOMETRY MODELING FOR LONG-TAIL OUT-OF-DISTRIBUTION DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

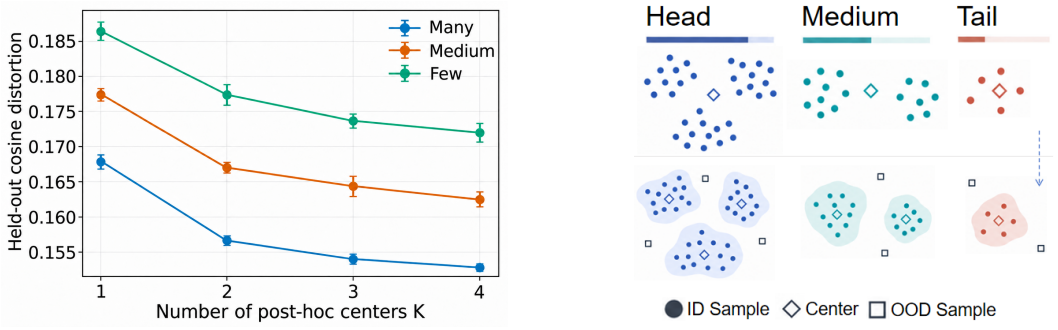
Long-tailed class imbalance presents a fundamental challenge for out-of-distribution (OOD) detection. Limited supervision yields sparse, unreliable representations for tail classes, and large intra-class variation prevents class-level representations from fully capturing the geometric structure of head classes. We observe that these head-class features often form multiple compact local structures rather than concentrating around a single global prototype. This motivates us to reconsider long-tailed OOD detection from the perspective of geometry modeling. Specifically, we propose **AdaGeo**, an adaptive local geometry framework that represents each class with multiple geometric components and jointly exploits their location, compactness, and empirical support. Based on this representation, we introduce a density-aware geometry learning objective that assigns stronger supervision to samples that deviate from their matched centers relative to the local geometric scale. Further, to keep this learned geometry aligned with the available evidence, we propose an adaptive mechanism that removes redundant local components according to their effective support and compactness. At the inference phase, we combine the global energy-based confidence with local geometric evidence, including intra-class cohesion and inter-class separation, to obtain a geometry-aware OOD score. Experiments on CIFAR-10-LT, CIFAR-100-LT, and ImageNet-LT demonstrate that AdaGeo improves in-distribution (ID) classification accuracy over the evaluated baselines and achieves competitive OOD detection performance, with the best average OOD results on CIFAR-100-LT and ImageNet-LT.

1 INTRODUCTION

Long-tailed out-of-distribution (OOD) detection requires a model to recognize the underrepresented in-distribution (ID) classes while rejecting unfamiliar inputs. However, long-tailed class imbalance poses a great challenge to this task. Several representative methods (e.g., PASCL (Wang et al., 2022b), BERL (Choi et al., 2023), EAT (Wei et al., 2024), PATT (He et al., 2025)) have been proposed to address this problem, yet their performance remains unsatisfactory. This is mainly because limited tail-class supervision makes the extent of ID feature regions difficult to estimate, while intra-class variation can make a single class-level summary insufficient to describe those regions. A useful representation must therefore accommodate variation within ID classes while accounting for the unequal evidence available to characterize it.

To examine the representational value of local centers, we analyze normalized features from a frozen ResNet-18 checkpoint trained on CIFAR-100-LT. For each class, spherical k-means fits centers to 50 test images; the remaining 50 images evaluate the mean cosine distance to the nearest center. Details of the evaluation protocol are provided in Appendix A.1. As shown in Fig. 1a, increasing the number of centers from one to three reduces the average held-out cosine distortion by 7.51%, where held-out refers to samples excluded from center fitting and reserved for evaluation. The reduction occurs across all frequency groups: 8.18% for many-shot, 7.37% for medium-shot, and 7.02% for few-shot classes. These results support the use of multiple centers to characterize intra-class variation, including that within tail classes. Further increasing the number of centers to four leads to only marginal additional reduction, indicating diminishing representational returns.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Empirical motivation.

(b) Intuition behind our idea.

Figure 1: Motivation and intuition behind our idea. (a) Multiple centers reduce representation error across frequency groups. (b) A single center cannot cover the multiple feature clusters (see the upper row), while adaptive multi-center modeling forms more components for frequent classes and fewer for rare ones respectively (see the lower row), leaving OOD samples outside all covered regions.

The benefits of a finer representation raise a second question: how much local complexity can the training data reliably support? The diagnostic analysis above uses equal numbers of fitting samples per class, whereas long-tailed training yields highly uneven amounts of evidence across classes. Assigning many components to all classes uniformly can leave some components poorly supported; restricting every class to one center can obscure useful local variation. This motivates adapting representational complexity to the available evidence. Moreover, empirical support alone does not describe local geometry: a populated component may be dispersed, while a smaller one may be compact. Component location, compactness, and empirical support thus provide complementary information for modeling in-distribution (ID) feature regions. In this work, our focus is to control local representational complexity under long-tailed supervision and use the resulting geometric statistics as shared evidence for both learning and OOD inference.

To achieve the above goal, we propose **AdaGeo**, an adaptive local geometry framework for long-tailed OOD detection. AdaGeo maintains class-wise centers together with local scale and assignment-support statistics; see Fig. 1b for an illustration. In our framework, frequency-based initialization, support- and compactness-aware pruning control the number of active components. During the training phase, local feature–center consistency guides geometric supervision, while local cohesion and inter-class separation complement global energy-based confidence during the inference phase. To summarize, our contributions are as follows:

- We introduce **Adaptive Multi-Center Geometry Modeling**, which couples online estimation of local component statistics with class-dependent complexity control to accommodate imbalanced training support.
- We couple **Density-Aware Geometry Learning** with **Geometry-Aware OOD Scoring**, leveraging a shared local representation to guide training supervision and augment energy-based detection with cohesion and separation evidence.
- We conduct extensive experiments on standard long-tailed benchmarks. The results demonstrate improved ID classification and competitive OOD detection. Moreover, AdaGeo requires less training time than PATT, as it does not leverage outlier evidence during training.

2 Related Work

Out-of-Distribution Detection. OOD detection identifies inputs outside the training distribution. There are several types of methods: confidence-based (e.g., MSP (Hendrycks & Gimpel, 2017) and ODIN (Liang et al., 2018)), energy-based (e.g., Energy (Liu et al., 2020)), and others (e.g., OE (Hendrycks et al., 2019), Mahalanobis scoring (Lee et al., 2018), ReAct (Sun et al., 2021), DICE (Sun & Li, 2022), and ViM (Wang et al., 2022a)). Particularly, OpenOOD (Yang et al., 2022) standardizes their evaluation. For large-scale evaluation, GradNorm (Huang et al., 2021) establishes benchmarks for near-OOD detection on semantically similar datasets such as iNaturalist,

SUN, Places365, and DTD. These general-purpose methods do not explicitly address severe ID class imbalance, which is the core challenge targeted by our work.

Long-Tailed Recognition. Addressing class imbalance is important in many learning tasks. Long-tailed recognition can counter class-frequency bias through margins, calibrated logits, or decoupled optimization; representative techniques include LDAM-DRW (Cao et al., 2019), Balanced Meta-Softmax (Ren et al., 2020), Logit Adjustment (Menon et al., 2021), and decoupled training (Kang et al., 2020). Furthermore, several typical representation learning methods including SupCon (Khosla et al., 2020), hybrid contrastive networks (Wang et al., 2021), and balanced contrastive learning (Zhu et al., 2022) can generate more balanced embeddings. All these works, however, do not address OOD detection, and our method for class imbalance complements them by modeling local multi-modal geometry within a class.

Long-Tail OOD Detection. Long-tail OOD detection needs to distinguish true OOD inputs from sparsely represented tail ID samples. There are already some representative works. For example, PASCL (Wang et al., 2022b) uses asymmetric contrastive learning, whereas BERL (Choi et al., 2023) balances energy regularization over auxiliary outliers; EAT (Wei et al., 2024) uses virtual abstention classes and tail-context augmentation; PATT (He et al., 2025) combines vMF-based contrastive augmentation with post-hoc calibration. Although these works achieve great improvements, they bear a common fact, i.e., they use a single center for each class. In contrast, our method models local intra-class modes and uses their geometric statistics in both optimization and OOD scoring. Our work is orthogonal to existing long-tail OOD methods.

Multi-Prototype Sub-Center Learning. Multi-prototype learning represents a class by several local centers rather than one global centroid. Sub-center ArcFace (Deng et al., 2020) uses fixed sub-centers for noisy face recognition, while P-ODN (Shu et al., 2020) jointly learns prototypes and radii for open-set recognition. MGPL (Liu et al., 2023) represents complex classes with multiple Gaussian prototypes, and diversified multi-prototype contrastive learning (Jia et al., 2025) improves OOD embeddings with fine-grained prototypes. All of these works do not cover the problem of long-tail OOD detection. Furthermore, our setting additionally requires the number of useful modes to vary with class frequency; hence, we maintain online sub-center statistics and prune weak centers with a BIC-inspired criterion (Schwarz, 1978).

3 PRELIMINARIES

Problem Definition. Following (He et al., 2025), let $\mathcal{D}_{\text{in}} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the long-tailed in-distribution (ID) training set, where x_i is an image and $y_i \in \mathcal{Y} = \{1, \dots, C\}$ is its class label. Each class $c \in \mathcal{Y}$ contains N_c training samples, with $N = \sum_{c \in \mathcal{Y}} N_c$ and highly imbalanced class counts. The task combines multi-class ID classification with binary OOD detection. We learn an encoder f_θ and a classifier φ, together with an OOD score $S(x)$, such that for any test sample x: if x is drawn from the ID distribution, the classifier assigns it the correct class; otherwise, the score $S(x)$ identifies x as OOD.

Logit Adjustment. Logit Adjustment (LA) (Menon et al., 2021) calibrates logits with class prior probabilities to mitigate long-tailed classification bias. For an encoded feature $\mathbf{z} = f_\theta(x)$ with ground-truth label y, its loss is:

$$\mathcal{L}_{\text{LA}}(\mathbf{z}, y) = -\log \frac{\pi_y \, e^{\varphi_y(\mathbf{z})}}{\sum_{y' \in \mathcal{Y}} \pi_{y'} \, e^{\varphi_{y'}(\mathbf{z})}}, \tag{1}$$

where $\pi_y = N_y/N$ is the prior of label y, and $\varphi_y(\mathbf{z})$ is the logit of class y produced by classifier φ.

4 METHODOLOGY

4.1 OVERVIEW

The overall framework of AdaGeo is illustrated in Fig. 2. We address long-tailed OOD detection from a geometric perspective. The core idea is to maintain per-class adaptive local geometry memory: a small collection of local centers with scales and support counts, which is reused for both training supervision and inference scoring. AdaGeo comprises three core elements:

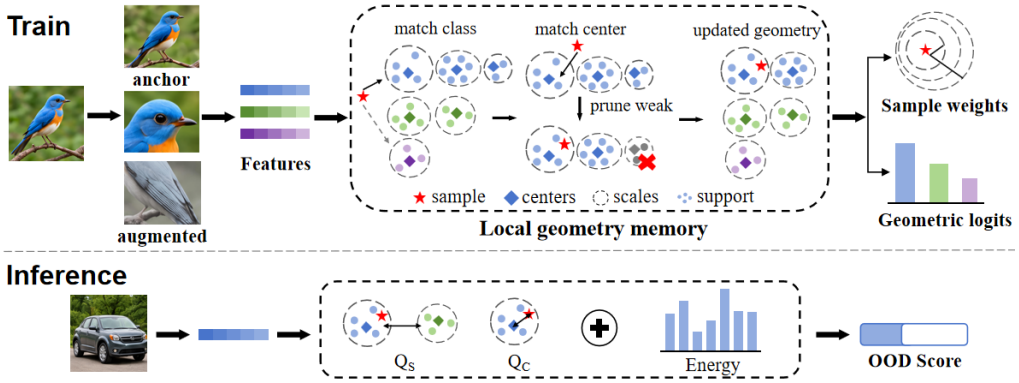


Figure 2: AdaGeo overview. **Train.** Each input image is augmented into three views; their features update the shared local geometry memory through class matching, center matching, and weak-component pruning, producing geometry-aware logits. **Inference.** A test feature is scored by combining inter-class separation, intra-class cohesion, and the energy score into the final OOD score.

1. **Adaptive multi-center geometry modeling.** It first initializes per-class capacity from class frequency, then updates the local centers and compactness online, and prunes weakly supported components;
2. **Density-aware geometry learning.** It uses local geometric reliability to weight samples and combines feature–center similarities with component support to construct geometric logits, providing multi-view supervision for the encoder and classifier; and
3. **Geometry-aware OOD scoring.** It uses the fixed memory at inference to combine local cohesion and inter-class separation into a geometric confidence that complements the energy score.

Training couples the geometry loss $\mathcal{L}_{\text{geo}}$ with logit adjustment loss $\mathcal{L}_{\text{LA}}$ to handle class imbalance. The loss function is: $\mathcal{L} = \lambda_{\text{geo}}\mathcal{L}_{\text{geo}} + \lambda_{\text{LA}}\mathcal{L}_{\text{LA}}$, where λ_{geo} and λ_{LA} are their weights, respectively. (The complete training procedure can refer to Appendix A.5.) Next, we cover these three core elements in detail (Sections 4.2∼4.4).

4.2 Adaptive Multi-Center Geometry Modeling

It aims to represent intra-class variation while matching geometric capacity to the unequal training evidence across classes. Our idea is to maintain a bounded, variable number of active components per-class and use their assignment support and compactness to determine which remain active. We achieve this by introducing a concept of *geometry memory*. We need to perform initialization, online updates, and adaptive component pruning on it.

Multi-center representation. For each class c, we reserve a fixed *memory* of K component slots, of which $k_c \le K$ are currently active; the active components are indexed by $k \in \mathcal{K}_c = \{1, \dots, k_c\}$ and parameterized by centers $\mathbf{m}_{c,k}$ on the unit hypersphere. We dub this geometry-aware memory as *geometry memory*. Each component is associated with a cumulative assignment count $n_{c,k}$ and a compactness statistic $r_{c,k}$. The center specifies the location of a local feature mode, while $r_{c,k}$ measures the average cosine distance of its assigned features. Together with the empirical assignment proportion, these quantities characterize the location, compactness, and support of each component.

Initialization and competitive assignment. The initial active count k_c^{init} for class c is determined by its sample size:

$$k_c^{\text{init}} = \min\left(\left\lfloor \frac{N_c}{N_{\min} \cdot k_{\text{base}}} \right\rfloor + 1, K\right), \tag{2}$$

where $N_{\min}$ is the minimum class size, and k_{base} controls the initial granularity. This initialization allocates greater representational capacity to head classes while preserving a compact representation for tail classes. Given a normalized feature $\mathbf{f} = \text{norm}(f_\theta(x))$ from class c, where $\text{norm}(\mathbf{z}) = \mathbf{z}/\|\mathbf{z}\|_2$ denotes ℓ_2 normalization for nonzero $\mathbf{z}$, we assign it to the most compatible/competitive

active component:

$$k^* = \arg\max_{k \in \mathcal{K}_c} \mathbf{f}^\top \mathbf{m}_{c,k}, \tag{3}$$

The assigned features are then used to update each component's information.

Online updates. We need to update geometry memory information including support, center (i.e., location), and compactness. As for support it is trivially achieved as it is just the count of features assigned to the component. As for the latter two, they are achieved by exponential moving average (EMA) and its variant. Specifically, let $\mathcal{F}_{c,k}$ denote the anchor features assigned to component (c, k) in the current mini-batch. For a non-empty $\mathcal{F}_{c,k}$, we update the center by EMA:

$$\mathbf{m}_{c,k} \leftarrow \text{norm}\left(\beta_m \mathbf{m}_{c,k} + (1-\beta_m)\bar{\mathbf{f}}_{c,k}\right), \tag{4}$$

where $\bar{\mathbf{f}}_{c,k}$ is the ℓ_2-normalized mean of the features in $\mathcal{F}_{c,k}$, β_m controls the update momentum. Using the updated center, we can estimate local compactness as the EMA of the mean cosine distance:

$$r_{c,k} \leftarrow \beta_r r_{c,k} + (1-\beta_r)\frac{1}{|\mathcal{F}_{c,k}|}\sum_{\mathbf{f} \in \mathcal{F}_{c,k}} \left(1 - \mathbf{f}^\top \mathbf{m}_{c,k}\right), \tag{5}$$

where β_r is the compactness update momentum, components with no assigned features retain their previous states, and smaller $r_{c,k}$ indicates a more compact local structure. These updated centers and compactness statistics are then shared by the subsequent learning and inference stages.

Adaptive component pruning. Different classes exhibit distinct degrees of representational power. We evaluate the contribution of each component as

$$\gamma_{c,k} = \pi_{c,k}(1 - r_{c,k}), \tag{6}$$

where $\pi_{c,k} = \frac{n_{c,k}}{\sum_{k' \in \mathcal{K}_c} n_{c,k'}}$ denotes the normalized support of component (c, k). The contribution jointly reflects the prevalence and compactness of a local component. We can prune weak components using a BIC (Bayesian Information Criterion)-inspired threshold:

$$\min_{k \in \mathcal{K}_c} \gamma_{c,k} < \lambda_{\text{BIC}} \frac{\log(\max(2, N_c))}{\max(1, N_c)}, \tag{7}$$

where λ_{BIC} controls the pruning strength. The class-dependent threshold decreases as the available sample size grows, enabling head classes to preserve finer local structures while restricting excessive representational power for tail classes. The weakest component is removed iteratively until all remaining components satisfy the criterion or only one component remains.

4.3 Density-Aware Geometry Learning

It converts the geometry memory built above into supervisory signals. Particularly, it concentrates training effort on samples where the current geometry exhibits low reliability: samples deviating from their matched component relative to the component's scale are more difficult to model and are thus enforced with stronger supervisory signals. We achieve this by converting component statistics into a *local geometric reliability score*, which we use to reweight samples and construct geometry-aware logits. Further, we optimize the geometry-aware objective across multiple views.

Local geometric reliability score. For a normalized feature $\mathbf{f}$ from class c, the local geometric reliability score is computed as:

$$\rho_{\text{geo}}(\mathbf{f}, c) = \sigma\left(\frac{\mathbf{f}^\top \mathbf{m}_{c,k^*}}{\tau}\right) \exp\left(-\frac{1 - \mathbf{f}^\top \mathbf{m}_{c,k^*}}{r_{c,k^*}}\right), \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function, τ is a shared temperature parameter. The sigmoid term increases with feature–center similarity, while the exponential term penalizes cosine distance relative to the component's local radius. At the same distance, a smaller radius produces a stronger penalty, reflecting the tighter concentration of a compact component. Together, the two terms measure how well a feature aligns with its matched component.

Reliability-aware reweighting. The reliability score above is a geometric reliability measure, not a normalized probability density. We need to convert it into a normalized sample weight:

$$w(\mathbf{f}, y) = \frac{1 - \rho_{\text{geo}}(\mathbf{f}, y)}{\frac{1}{B}\sum_{i=1}^{B}\left[1 - \rho_{\text{geo}}(\mathbf{f}_i, y_i)\right]}, \tag{9}$$

where B is the mini-batch size. The weight is evaluated on anchor features and treated as constant during backpropagation. The denominator normalizes their batch mean to one, so samples with lower geometric reliability receive greater relative emphasis.

Geometry-aware logits. The above learned local components provide class-level geometric evidence. To achieve geometry-aware classification, we first normalize the log-counts within each class: $b_{c,k} = \log\max(n_{c,k}, 1) - \max_{k' \in \mathcal{K}_c} \log\max(n_{c,k'}, 1)$. This normalization computes support relative to the most populated active component in the same class, removing the class-wise maximum log-count offset. For each class c, we then aggregate responses from its active components by summing their exponentiated, log-count-normalized similarities:

$$\ell_c^{\text{geo}}(\mathbf{f}) = \log\left(\frac{1}{K}\sum_{k \in \mathcal{K}_c} \exp\left(b_{c,k} + \frac{\mathbf{f}^\top \mathbf{m}_{c,k}}{\tau}\right)\right). \tag{10}$$

Finally, we combine this geometric evidence with the classifier logits as follows:

$$\ell_c(\mathbf{z}) = (1-\mu)\varphi_c(\mathbf{z}) + \mu\ell_c^{\text{geo}}(\text{norm}(\mathbf{z})) + \log\pi_c, \tag{11}$$

where μ is a hyperparameter for controlling the weight of the geometric evidence.

Multi-view geometry learning. To improve the robustness of the learned geometry under feature perturbations, we optimize the geometry-aware objective across multiple augmentations of each training image. Given $V \geq 2$ correlated views, the first view serves as the anchor for geometry memory updates and sample weighting, while the remaining views receive geometry-aware supervision. For each optimized view v, the density-aware geometry loss is computed as:

$$\mathcal{L}_{\text{geo}}^{(1\to v)} = \frac{1}{B}\sum_{i=1}^{B} w(\mathbf{f}_i^{(1)}, y_i)\,\text{CE}\left(\boldsymbol{\ell}(\mathbf{z}_i^{(v)}), y_i\right). \tag{12}$$

The same anchor features are reused across positive views, but the geometry memory is updated before each view is evaluated. Note that, both w and $\boldsymbol{\ell}$ above are evaluated using the current memory state, so the anchor-derived weights need not be identical across views. The final objective is:

$$\mathcal{L}_{\text{geo}} = \frac{1}{V-1}\sum_{v=2}^{V} \mathcal{L}_{\text{geo}}^{(1\to v)}. \tag{13}$$

4.4 Geometry-Aware OOD Scoring

It uses the shared geometry memory to evaluate test samples at inference. Our basic idea is to leverage global and local evidences together: the global energy score measures how far a sample lies from every class, while local geometric confidence measures how well it aligns with the structure of the class it claims. Generally, we first get the local geometric confidence from two concepts: local cohesion and inter-class separation, and then merge it with the global energy score.

Local geometric confidence. To evaluate whether a test feature is geometrically consistent with the learned local structures, we define a geometric confidence score $Q(\mathbf{f}) \in [0, 1]$ by combining two complementary signals: *local cohesion* and *inter-class separation*.

Local Cohesion. Given the predicted class $\hat{c}$, we match $\mathbf{f}$ to the most similar active component k^* of that class using Eq. 3, and take the reliability of Eq. 8 as the local cohesion score: $Q_{\text{C}} = \rho_{\text{geo}}(\mathbf{f}, \hat{c})$. A larger Q_{C} indicates that the feature aligns closely with its matched center and lies within the corresponding local radius.

Inter-Class Separation. Local cohesion alone cannot distinguish a compact but geometrically ambiguous region from a well-separated one. We therefore measure the separation of the matched component from components belonging to other classes:

$$Q_{\text{S}} = \frac{1}{2}\left(1 - \max_{\substack{c' \in \mathcal{Y} \\ c' \neq \hat{c}}} \max_{k' \in \mathcal{K}_{c'}} \mathbf{m}_{\hat{c},k^*}^\top \mathbf{m}_{c',k'}\right), \tag{14}$$

Here a larger Q_{S} indicates that the matched local component is better separated from competing class geometries. We combine local cohesion and inter-class separation using an unweighted arithmetic

mean: $Q(\mathbf{f}) = \frac{1}{2}\left(Q_{\mathrm{C}} + Q_{\mathrm{S}}\right)$. Both terms lie in $[0, 1]$, so their average preserves this range without introducing an additional mixing parameter.

Geometry-energy fusion. The learned geometry provides local structural evidence complementary to the global discriminative confidence captured by the energy score. We adopt the standard energy score (Liu et al., 2020): $E(x) = -\log \sum_{c \in \mathcal{Y}} e^{\varphi_c(\mathbf{z})}$, where $\varphi_c(\mathbf{z})$ is the classifier logit used for OOD scoring. The final OOD criterion fuses the two complementary signals as

$$S(x) = E(x) + \alpha\big(1 - Q(\mathbf{f})\big). \tag{15}$$

where α is a parameter for controlling the weight of geometric inconsistency, and is automatically selected using ID and auxiliary OOD calibration data. The term $1 - Q(\mathbf{f})$ increases the OOD score for samples that are poorly aligned with the learned local geometry, while the energy term captures global classifier uncertainty.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate AdaGeo on the standard long-tailed benchmarks CIFAR-10-LT and CIFAR-100-LT with an imbalance ratio of $\rho = 0.01$ (Cao et al., 2019). For OOD evaluation, we follow the SC-OOD benchmark protocol (Yang et al., 2021) and evaluate against six OOD datasets: Describable Textures Dataset (DTD) (Cimpoi et al., 2014), SVHN (Netzer et al., 2011), CIFAR-100/CIFAR-10 as cross-dataset OOD sets (Fort et al., 2021), TinyImageNet (Le & Yang, 2015), LSUN (Yu et al., 2015), and Places365 (Zhou et al., 2017). These datasets cover diverse distribution shifts and provide a comprehensive evaluation of OOD detection performance under long-tailed training distributions.

We additionally evaluate on the large-scale ImageNet-LT benchmark (Liu et al., 2019). Following the large-scale OOD benchmark (Huang et al., 2021), we use four near-OOD datasets that are semantically close to ImageNet, namely iNaturalist (Van Horn et al., 2018), SUN (Xiao et al., 2010), Places365 (Zhou et al., 2017), and DTD (Cimpoi et al., 2014).

Baselines and Evaluation Metrics. We compare against PATT (He et al., 2025) together with its accompanying baselines such as EAT, PASCL, OE, MGPL, MSP, and EnergyOE. Unless stated otherwise, we adopt PATT as the reference baseline. Following (Wang et al., 2022b; Yang et al., 2021; He et al., 2025), we report ID accuracy (ACC), AUROC for ID/OOD discrimination, AUPR-In/Out with ID/OOD as the positive class, and FPR95 at 95% OOD recall, respectively. Higher is better except for FPR95. OOD metrics are averaged equally over six datasets for CIFAR and four for ImageNet-LT.

Implementation Details. On CIFAR-10-LT and CIFAR-100-LT, we train ResNet-18 (He et al., 2016) with Adam (Kingma & Ba, 2015) for 100 epochs using only ID images. Geometry learning uses one anchor and two positive views, with $K = 3$, shared temperature $\tau = 0.5$, and geometry-logit weight $\mu = 0.5$. Following PATT (He et al., 2025), post-hoc feature calibration (FC) uses class-balanced ID features and auxiliary TinyImages features; geometric confidence uses pre-FC features. The fusion weight is automatically calibrated on ID training features and auxiliary TinyImages features before evaluation. On ImageNet-LT, we train ResNet-50 with SGD for 100 epochs, using only ID images and the same multi-view geometry settings as on CIFAR. The fusion weight is determined by the automatic calibration protocol. (More details about datasets, full optimization, augmentation, calibration, score conventions, and hardware settings are provided in Appendix A.)

5.2 Main Results

On CIFAR-10-LT, AdaGeo outperforms PATT across all detection metrics on four of the six OOD datasets (see the rows for CIFAR-100, LSUN, Places365, and TinyImageNet in Table 1). Further, on CIFAR-100-LT it outperforms PATT across all metrics on five of the six OOD datasets (see the rows for CIFAR-100, LSUN, Places365, DTD and TinyImageNet in Table 1). In addition, we can see from the last row in Table 1 (i.e., aggregate performance on average) that AdaGeo performs better than PATT for all these metrics on both CIFAR-10-LT and CIFAR-100-LT datasets. Table 2 compares against the baselines on average OOD detection and ID classification performance. In terms of ID classification, AdaGeo performs the best among all these competitors on both CIFAR-10-LT

Table 1: Per-dataset OOD detection results on CIFAR-10-LT and CIFAR-100-LT.

		CIFAR-10-LT				CIFAR-100-LT			
$\mathcal{D}_{\text{out}}^{\text{test}}$	Method	AUROC↑	AUPR-In↑	AUPR-Out↑	FPR95↓	AUROC↑	AUPR-In↑	AUPR-Out↑	FPR95↓
CIFAR-100 CIFAR-10	OE	83.40±0.3	84.06±0.3	80.93±0.6	56.96±0.9	62.23±0.3	66.16±0.4	57.57±0.3	80.64±1.0
	PATT	85.75±0.2	86.50±0.2	83.80±0.2	51.12±0.9	61.07±0.6	62.39±1.2	57.77±0.3	85.74±1.6
	Ours	86.10±0.2	87.07±0.2	83.94±0.1	47.54±0.5	65.15±0.2	67.31±0.1	61.04±0.2	80.36±0.4
LSUN	OE	91.35±0.2	93.06±0.3	87.62±0.8	27.86±0.7	77.10±0.6	85.33±0.6	61.42±1.0	63.98±1.4
	PATT	92.43±0.3	93.35±0.3	90.54±0.4	29.58±1.6	75.58±1.8	84.72±1.4	60.46±1.9	65.33±3.3
	Ours	93.07±0.2	94.24±0.1	91.37±0.4	25.54±0.3	78.02±0.6	86.39±0.4	63.20±0.8	61.42±0.9
Places365	OE	90.07±0.3	82.09±0.4	95.15±0.2	34.04±0.9	75.80±0.5	60.99±0.6	86.68±0.4	65.72±0.9
	PATT	91.80±0.2	83.70±0.8	96.43±0.1	31.57±1.2	73.72±0.5	57.10±0.8	85.99±0.3	70.87±0.9
	Ours	92.38±0.1	85.70±0.2	96.70±0.1	28.51±0.3	76.46±0.4	60.32±0.4	87.62±0.3	67.38±0.4
SVHN	OE	95.10±1.0	91.59±0.5	97.14±0.8	16.15±1.5	77.61±3.3	73.25±1.5	86.82±2.5	58.04±4.8
	PATT	98.53±0.1	97.60±0.1	98.93±0.1	4.95±0.1	96.12±0.7	93.01±1.3	98.01±0.3	16.85±3.3
	Ours	97.48±0.2	95.61±0.3	98.50±0.1	9.84±0.6	90.07±0.3	82.69±0.4	95.07±0.2	37.54±0.6
DTD	OE	92.59±0.4	96.01±1.4	83.32±1.7	25.10±1.1	76.71±1.2	85.28±0.9	58.79±1.4	68.28±1.5
	PATT	92.66±0.1	95.79±0.1	85.89±0.1	29.23±1.0	75.50±0.6	84.29±0.2	59.75±1.1	70.48±0.4
	Ours	91.96±0.3	95.32±0.1	85.26±0.8	32.44±1.1	78.81±1.4	87.26±1.0	64.61±2.1	62.24±3.0
Tiny ImageNet	OE	86.14±0.3	89.88±0.7	79.33±0.7	47.78±0.7	68.04±0.4	79.36±0.3	51.66±0.5	76.66±0.5
	PATT	87.83±0.3	91.02±0.3	81.98±0.3	41.52±1.4	70.16±0.6	80.46±0.4	54.75±0.5	74.83±0.9
	Ours	87.99±0.1	91.37±0.1	81.98±0.1	39.41±0.5	71.88±0.3	82.00±0.2	56.37±0.3	71.88±0.3
Average	OE	89.77±0.3	89.45±0.4	87.25±0.6	34.65±0.5	72.92±0.7	75.06±0.6	67.16±0.6	68.89±1.1
	PATT	91.50±0.1	91.33±0.3	89.60±0.2	31.33±0.7	75.36±0.2	77.00±0.3	69.46±0.1	64.01±0.3
	Ours	91.50±0.1	91.55±0.1	89.63±0.1	30.55±0.1	76.73±0.1	77.66±0.1	71.32±0.3	63.47±0.4

Table 2: Average OOD detection and ID classification results on CIFAR-10-LT and CIFAR-100-LT. ± denotes the standard deviation over three runs.

Method	CIFAR-10-LT					CIFAR-100-LT				
	ACC↑	AUROC↑	AUPR-In↑	AUPR-Out↑	FPR95↓	ACC↑	AUROC↑	AUPR-In↑	AUPR-Out↑	FPR95↓
MSP	72.34	72.28	73.96	70.27	66.07	40.97	61.00	64.52	57.54	82.01
EnergyOE	74.68	89.31	91.01	88.92	40.88	39.05	71.10	75.42	67.23	71.78
MGPL	77.75±0.3	65.65±1.6	66.73±1.0	63.43±1.1	72.23±1.3	44.18±0.4	56.96±0.5	58.89±1.2	55.02±0.6	85.85±1.5
OE	73.84±0.8	89.77±0.3	89.45±0.4	87.25±0.6	34.65±0.5	39.04±0.4	72.92±0.7	75.06±0.6	67.16±0.6	68.89±1.1
PASCL	77.08±1.0	90.99±0.2	90.18±0.4	89.24±0.3	33.36±0.8	43.10±0.5	73.32±0.3	74.52±0.4	67.18±0.1	67.44±0.6
EAT	81.31±0.3	91.73±0.3	91.46±0.5	90.35±0.5	30.30±1.2	46.13±0.3	74.41±0.9	76.04±0.7	69.57±0.9	65.05±1.5
PATT	84.69±0.2	91.50±0.1	91.33±0.3	89.60±0.2	31.33±0.7	49.51±0.5	75.36±0.2	77.00±0.3	69.46±0.1	64.01±0.3
Ours	86.11±0.3	91.50±0.1	91.55±0.1	89.63±0.1	30.55±0.1	52.05±0.2	76.73±0.1	77.66±0.1	71.32±0.3	63.47±0.4

and CIFAR-100-LT datasets. As for OOD detection, it performs best among all these competitors on CIFAR-100-LT, and it is also competitive on CIFAR-10-LT dataset. These results together demonstrate the superiority and competitiveness of AdaGeo.

On the large-scale dataset ImageNet-LT, the per-dataset gains are concentrated on DTD and iNaturalist, and AdaGeo performs the best on the average OOD metrics (see the last row in Table 3 or 4). Relative to PATT, AUROC increases by 1.82 percentage points and FPR95 decreases by 5.30 percentage points, which constitutes a non-trivial performance gain. This set of results further verifies the superiority of AdaGeo.

5.3 Ablation Studies

To study the effectiveness of multi-center geometry modeling, density-aware learning, and geometry-aware OOD scoring, respectively (recall Sections 4.2, 4.3, and 4.4), Table 5 compares progressive component configurations on both CIFAR-10-LT and CIFAR-100-LT benchmarks.

Replacing a single center with multiple centers increases AUROC by 1.07 and 0.26 points on CIFAR-10-LT and CIFAR-100-LT respectively, and reduces FPR95 by 1.25 and 0.69 points; ACC changes by +0.32 and -0.03 points. This set of results demonstrates multiple centers yield modest OOD detection gains, while it retains competitive classification performance. Particularly, we observe that adding density-aware supervision produces the largest incremental AUROC gains in the progressive configurations: 2.97 and 2.59 points on CIFAR-10-LT and CIFAR-100-LT respectively,

Table 3: Per-dataset OOD detection results.

$\mathcal{D}_{\text{out}}^{\text{test}}$	Method	AUROC↑	AUPR-In↑	AUPR-Out↑	FPR95↓
iNaturalist	MSP	73.09	84.42	51.59	74.62
	PASCL	74.02	85.71	49.88	70.86
	PATT	88.68±1.23	93.97±0.70	80.68±1.91	44.46±2.74
	Ours	91.04±0.40	95.12±0.26	84.84±0.71	40.15±0.76
SUN	MSP	71.84	82.83	51.55	78.09
	PASCL	76.45	87.36	52.65	63.88
	PATT	88.35±0.39	93.26±0.28	79.52±0.71	45.73±1.12
	Ours	86.50±0.41	92.22±0.14	76.26±0.91	49.44±1.20
Places365	MSP	72.07	83.41	52.16	75.14
	PASCL	76.21	87.15	53.14	64.42
	PATT	85.07±0.39	91.09±0.27	74.80±0.66	56.79±1.22
	Ours	82.62±0.40	89.92±0.10	69.97±1.19	60.33±0.41
DTD	MSP	63.03	86.27	27.99	81.91
	PASCL	62.45	86.35	26.57	81.14
	PATT	83.64±0.40	93.94±0.14	66.06±0.67	60.70±0.83
	Ours	92.88±0.27	97.53±0.13	83.23±0.40	36.56±1.11
Average	MSP	70.01	84.23	45.82	77.44
	PASCL	72.28	86.64	45.56	70.07
	PATT	86.44±0.42	93.07±0.30	75.27±0.83	51.92±1.06
	Ours	88.26±0.23	93.70±0.05	78.57±0.55	46.62±0.40

Table 4: Average results over the four OOD test datasets.

Method	ACC↑	AUROC↑	AUPR-In↑	AUPR-Out↑	FPR95↓
MSP	42.69	70.01	84.23	45.82	77.44
PASCL	47.28	72.28	86.64	45.56	70.07
PATT	55.35±0.60	86.44±0.42	93.07±0.30	75.27±0.83	51.92±1.06
Ours	56.09±0.32	88.26±0.23	93.70±0.05	78.57±0.55	46.62±0.40

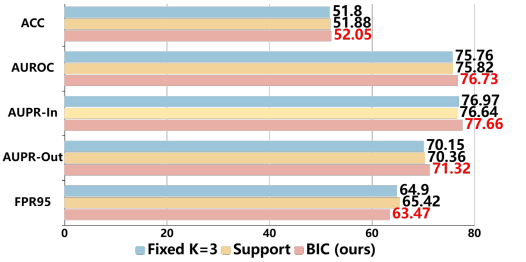


Figure 3: Capacity and pruning-rule ablations on CIFAR-100-LT.

Table 5: Component configurations on the CIFAR benchmarks. The full pipeline includes post-hoc FC and automatic fusion-weight calibration.

ID Dataset	Multi-center	Density-aware	Geometric Score	ACC↑	AUROC↑	AUPR-In↑	AUPR-Out↑	FPR95↓
CIFAR-10-LT	×	×	×	83.43	87.05	86.11	85.31	42.99
	✓	×	×	83.75	88.12	87.21	86.90	41.74
	✓	✓	×	86.11	91.09	91.19	88.86	31.54
	✓	✓	✓	**86.11**	**91.50**	**91.55**	**89.63**	**30.55**
CIFAR-100-LT	×	×	×	51.14	73.27	74.70	67.29	68.53
	✓	×	×	51.11	73.53	75.05	67.40	67.84
	✓	✓	×	52.05	76.12	76.84	70.94	64.82
	✓	✓	✓	**52.05**	**76.73**	**77.66**	**71.32**	**63.47**

and FPR95 decreases by 10.20 and 3.02; ACC improves on both datasets. This set of results supports weighting the supervision by local geometric consistency. The full configuration combines automatically calibrated geometric scoring with post-hoc FC. It retains the density-aware configuration's ACC of 86.11% and 52.05% on CIFAR-10-LT and CIFAR-100-LT respectively, while achieving the best OOD metrics. This set of results justifies the effectiveness of geometry-aware OOD scoring.

Additionally, Fig. 3 compares the BIC-inspired rule with fixed-capacity and support-only variants on CIFAR-100-LT. The results show that BIC-inspired pruning achieves the best ACC and OOD metrics, confirming the benefit of frequency-scaled adaptive pruning. (Other important experimental results are included in Appendix B, due to space limit.)

6 Conclusion

In this paper we propose AdaGeo that adapts local geometric components to long-tailed data, and leverages their statistics for density-aware geometry learning and geometry-aware OOD scoring. AdaGeo achieves the highest ID accuracy among the evaluated baselines on all three benchmarks. As for OOD detection, it performs best among all these competitors on CIFAR-100-LT dataset and large-scale dataset ImageNet-LT. AdaGeo is also competitive on CIFAR-10-LT dataset, it outperforms PATT across all detection metrics on four of the six OOD datasets. Through ablation studies we also verify the effectiveness of major elements/techniques presented in the paper. Our study also yields further findings — Improved tail-class accuracy does not consistently yield better tail-class OOD detection, and fusion calibration requires auxiliary OOD data whose distribution may affect the selected weight. In the future, we will address tail-class separation and ID-only calibration.

AI USE STATEMENT

The authors used generative AI tools for language editing and coding assistance. The authors are responsible for verifying AI-assisted content and code and for the final manuscript and results.

ETHICS STATEMENT

This work uses publicly available image benchmarks and does not collect or process human-subject data, personally identifiable information, or private records. As with other visual recognition methods, deployment on imbalanced data may reproduce biases present in the underlying datasets; practitioners should evaluate performance across relevant populations before use in high-stakes settings.

REPRODUCIBILITY STATEMENT

The paper specifies the model, objective, data protocol, evaluation metrics, and training procedure. The accompanying source tree contains the CIFAR data loaders, training entry point, evaluation entry point, and the result files used for the CIFAR tables. Appendix A details the experimental protocol, including state updates, component initialization, numerical safeguards, and pruning in Appendix A.5. Results are reported over three random runs where indicated in the tables.

REFERENCES

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

Hyunjun Choi, Hawook Jeong, and Jin Young Choi. Balanced energy regularization loss for out-of-distribution detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

Jiankang Deng, Jia Guo, Jing Yang, Nikolaos Zafeiriou, Yuxiang Zhou, and Shuzhong Ding. Subcenter ArcFace: Boosting face recognition by large-scale noisy web faces. In *European Conference on Computer Vision*, 2020.

Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Yina He, Lei Peng, Yongcun Zhang, Juanjuan Weng, Shaozi Li, and Zhiming Luo. Long-tailed out-of-distribution detection: Prioritizing attention to tail. In *AAAI Conference on Artificial Intelligence*, 2025.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples. In *International Conference on Learning Representations*, 2017.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271, 2021.

Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild (GradNorm). In *Advances in Neural Information Processing Systems*, 2021.

Yulong Jia, Jiaming Li, Ganlong Zhao, Shuangyin Liu, Weijun Sun, Liang Lin, and Guanbin Li. Enhancing out-of-distribution detection via diversified multi-prototype contrastive learning. *Pattern Recognition*, 161:111214, 2025.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge, 2015. CS 231N.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 2018.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

Jiaming Liu, Jun Tian, Wei Han, Zhili Qin, Yulu Fan, and Junming Shao. Learning multiple gaussian prototypes for open-set recognition. *Information Sciences*, 626:738–753, 2023.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 2020.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems*, 2020.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

Yu Shu, Yemin Shi, Yaowei Wang, Tiejun Huang, and Yonghong Tian. P-ODN: Prototype-based open deep network for open set recognition. *Scientific Reports*, 10:7146, 2020.

Yiyou Sun and Yixuan Li. DICE: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision*, 2022.

Yiyou Sun, Chuan Guo, and Yixuan Li. ReAct: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems*, 2021.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. ViM: Out-of-distribution with virtual-logit matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022a.

Haotao Wang, Aston Zhang, Yi Zhu, Shuai Zheng, Mu Li, Alex J. Smola, and Zhangyang Wang. Partial and asymmetric contrastive learning for out-of-distribution detection in long-tailed recognition. In *International Conference on Machine Learning*, 2022b.

Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang. Contrastive learning based hybrid networks for long-tailed image classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Tong Wei, Bo-Lin Wang, and Min-Ling Zhang. EAT: Towards long-tailed out-of-distribution detection. In *AAAI Conference on Artificial Intelligence*, 2024.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2010.

Jingkang Yang, Haoqi Wang, Litong Feng, Xiaopeng Yan, Huabin Zheng, Wayne Zhang, and Ziwei Liu. Semantically coherent out-of-distribution detection. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. OpenOOD: Benchmarking generalized out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 2022.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017.

Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

A Experimental Protocol and Reproducibility

A.1 Evaluation Details for the Held-Out Geometry Diagnostic

Results shown in Fig. 1a follow the following evaluation settings. Specifically, it uses one frozen ResNet-18 checkpoint trained on CIFAR-100-LT with $\rho = 0.01$. Normalized test features are partitioned within each class into 50 fitting and 50 evaluation samples using split seeds 0, 1, 2. All tested center counts $K \in \{1, 2, 3, 4\}$ use the same class-wise partitions. For $K = 1$, the center is the normalized fitting-set mean. For $K > 1$, spherical k-means uses ten restarts with at most 100 assignment/update iterations; the selected restart minimizes fitting-set distortion.

Evaluation distortion is the mean of $1 - \max_k \mathbf{f}^\top \mathbf{m}_k$ over the evaluation features. We average errors across classes within each frequency group and repeat the fit/evaluation split with three random seeds, keeping the checkpoint fixed. Error bars are sample standard deviations of the three split-level means. The overall relative reduction from $K = 1$ to $K = 3$ is computed from the corresponding all-class mean errors. Here, held-out refers to samples excluded from center fitting and reserved for evaluation. The post-hoc centers are distinct from the online training memory, and the plotted errors measure geometric representation rather than OOD detection performance.

A.2 More details about Datasets and Data Usage

Long-tailed ID data. CIFAR-10-LT and CIFAR-100-LT are constructed from the official CIFAR training sets. CIFAR-10-LT contains 12,406 training images, with class sizes ranging from 5,000 to 50, while CIFAR-100-LT contains 10,847 training images, ranging from 500 to 5 samples per class. Classes are indexed in descending order of training count, so that class $c = 1$ is the most frequent, and the retained count decays exponentially as $N_c = \lfloor N_{\max} \rho^{(c-1)/(C-1)} \rfloor$, with $N_{\max} = 5000$ and 500, respectively. Within each class, the implementation retains the first N_c samples in the original dataset order. Thus, training membership is fixed across random seeds; the seeds change initialization, augmentation, and mini-batch order. The official balanced CIFAR test sets are retained for ID evaluation. ImageNet-LT uses the benchmark training split without applying this re-subsampling rule. ImageNet-LT benchmark contains 115,846 training images from 1,000 classes with class sizes ranging from 1,280 to 5 images (imbalance factor 256). Table 6 summarizes the ID training sets.

Table 6: ID training data at the main experimental settings.

Dataset	Classes	Training images	Minimum/class	Maximum/class
CIFAR-10-LT	10	12,406	50	5,000
CIFAR-100-LT	100	10,847	5	500
ImageNet-LT	1,000	115,846	5	1,280

Semantic OOD labels. For CIFAR, we follow the SC-OOD protocol (Yang et al., 2021). The loader reads each external image's `sc_label`: negative values identify truly OOD samples, while nonnegative values identify samples that semantically coincide with an ID class; the scores of these samples are appended to the ID test-score pool. Membership in an external image collection alone therefore does not determine whether a sample is OOD, and each CIFAR benchmark is evaluated separately on its truly OOD subset. For ImageNet-LT, we instead follow the large-scale near-OOD protocol (Huang et al., 2021), evaluating against iNaturalist, SUN, Places365, and DTD as entire external OOD collections.

Data use by stage. The encoder is trained exclusively on long-tailed ID training images. On CIFAR, post-hoc FC uses features from class-balanced sampling of the ID training set and auxiliary TinyImages. With the encoder and all FC parameters fixed, we select the fusion weight α using ID training data and auxiliary OOD data. The resulting configuration is fixed for all subsequent benchmark evaluations.

Automatic fusion calibration. The model and any feature-calibration state are fixed before selecting the fusion weight. Calibration scores use the same energy and geometric-confidence definitions as evaluation. On CIFAR, FC draws class-balanced samples from retained ID training images.

Fusion calibration then scores all retained ID training images and an equally sized prefix of the auxiliary TinyImages pool, or the full pool if smaller, using deterministic evaluation preprocessing. The candidate set is $\{0, 0.25, 0.5, 1, 2, 3, 4, 6, 8, 10, 12, 16, 24, 32, 48, 64\}$. We select the candidate with the highest ID-versus-auxiliary-OOD AUROC, weighting ID images inversely to their class counts so each class contributes equally. Ties favor the smaller weight. Selection is performed separately for each trained checkpoint, and the chosen weight is shared across all OOD evaluation datasets. Neither ID test images nor benchmark OOD scores are used for calibration. On ImageNet-LT, the same protocol applies without post-hoc FC: we score the full ID training set and an auxiliary ImageNet-O folder (Hendrycks et al., 2021) with the deterministic evaluation preprocessing, form the fused score for each candidate in the same candidate set, and select the candidate maximizing the class-balanced ID-versus-auxiliary-OOD AUROC. The auxiliary folder is verified to be disjoint from the four benchmark OOD sets. As on CIFAR, the chosen weight is fixed per checkpoint, shared across all OOD evaluation datasets, and neither ID test images nor benchmark OOD scores are used.

A.3 Preprocessing and Augmentation

For CIFAR, the anchor transform applies a 32×32 random crop without padding, followed by horizontal flip, the CIFAR AutoAugment policy, conversion to a tensor, and one Cutout region of side length 16. Each of the two positive views independently applies a random resized crop to 32×32, horizontal flip with probability 0.5, color jitter with probability 0.8, and grayscale conversion with probability 0.2. Color-jitter strengths are $(0.4, 0.4, 0.4, 0.1)$ for brightness, contrast, saturation, and hue. All three views use channel means $(0.4914, 0.4822, 0.4465)$ and standard deviations $(0.2023, 0.1994, 0.2010)$.

ID and SC-OOD test images are resized to 32×32 and use the same normalization, without stochastic augmentation. FC has a separate preprocessing path: a 32×32 random crop with padding 4, horizontal flip, and normalization with means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$. Training shuffles samples and drops the incomplete final mini-batch; ID and OOD evaluation retain all samples without shuffling.

A.4 Training Configuration

Table 7: Experimental configuration of AdaGeo. Shared settings apply to all three benchmarks.

Setting	CIFAR-10-LT / CIFAR-100-LT	ImageNet-LT
Backbone	ResNet-18	ResNet-50
Optimizer	Adam	SGD
Initial learning rate	10^{-3}	0.1
Weight decay	5×10^{-4}	5×10^{-5}
SGD momentum	—	0.995
Learning-rate schedule	Cosine annealing	$\times 0.1$ at epochs 60, 80
ID batch size	128	32
Imbalance ratio ρ	0.01	$1/256$
Auxiliary OOD for FC	TinyImages	None
Shared settings		
Hardware / software	Single RTX 4090 (24 GB), Python 3.11	
Epochs / seeds	100 / $\{0, 1, 2\}$	
Views per ID image	3 (one anchor, two positives)	
Maximum components K	3	
Initialization granularity k_{base}	2	
Shared temperature τ	0.5	
Geometry logit weight μ	0.5	
Loss weights $(\lambda_{\text{geo}}, \lambda_{\text{LA}})$	$(1.0, 0.5)$	
Center / radius EMA (β_m, β_r)	$(0.9, 0.995)$	
Initial radius / numerical ε	0.2 / 10^{-6}	
Pruning warm-up / strength	$E_{\text{warm}} = 20$ / $\lambda_{\text{BIC}} = 5$	

Table 7 lists the training settings. Batch sizes count ID images before augmentation. The CIFAR ResNet-18 produces 512-dimensional features, normalized for geometry modeling. Python,

NumPy, and PyTorch use seeds 0, 1, and 2; cuDNN benchmarking is enabled, so execution is not bitwise deterministic. The shared temperature τ controls sample reweighting, geometric logits, and inference-time cohesion.

A.5 Training Procedure and Implementation

Algorithm 1 outlines the training procedure of AdaGeo, it performs several stages such as initialization (Line 1), multi-view optimization (Lines 3-15), and adaptive pruning (Lines 18-26). Generally speaking, for each mini-batch it creates one anchor and two positive views, and uses a shared encoder to produce normalized features; the anchor features update memory and sample weights, while positive views receive geometry-aware supervision (Lines 3–6). As for each positive view, it updates centers, radii and counts, computes weights, combines classifier and geometric evidence with class priors, and finally uses the resulting logits and detached weights to form the view's loss (Lines 8–11). After handling all positive views, it averages positive-view losses, computes logit-adjustment losses, and updates parameters such as θ and φ by the optimizer step obtained based on the weighted sum (Lines 13-15). After warm-up, for each class it attempts to prune the components with the lowest contribution (Lines 18-24). Finally, it returns f_θ, φ, and the memory $\mathcal{M}$.

Algorithm 1 AdaGeo training procedure

Require: ID training set $\mathcal{D}_{\text{in}}$, component limit K, epochs E, training hyperparameters

1: Initialize f_θ, φ, and $\mathcal{M}$ using Eq. 2
2: **for** $e = 1, \dots, E$ **do**
3: **for** mini-batch $(\mathbf{x}, \mathbf{y})$ from $\mathcal{D}_{\text{in}}$ **do**
4: $\{\mathbf{z}^{(v)}\}_{v=1}^{V} \leftarrow f_\theta(\text{Augment}(\mathbf{x}, V))$
5: $\mathbf{f}^{(v)} \leftarrow \text{norm}(\mathbf{z}^{(v)})$ for each view v
6: $\mathbf{h} \leftarrow \text{stopgrad}(\mathbf{f}^{(1)})$
7: **for** $v = 2, \dots, V$ **do**
8: $\mathcal{M} \leftarrow \text{UpdateMemory}(\mathcal{M}, \mathbf{h}, \mathbf{y})$
9: $\mathbf{w} \leftarrow \text{stopgrad}(\text{ReliabilityWeights}(\mathbf{h}, \mathbf{y}, \mathcal{M}))$
10: $\boldsymbol{\ell}^{(v)} \leftarrow \text{CombinedLogits}(\mathbf{z}^{(v)}, \varphi, \mathcal{M})$
11: $L_v \leftarrow \text{WeightedCE}(\boldsymbol{\ell}^{(v)}, \mathbf{y}, \mathbf{w})$
12: **end for**
13: $\mathcal{L}_{\text{geo}} \leftarrow \text{Mean}(L_2, \dots, L_V)$
14: $\mathcal{L}_{\text{LA}} \leftarrow \text{LogitAdjustment}(\varphi(\mathbf{z}^{(1)}), \mathbf{y})$
15: Update θ, φ by an optimizer step on the joint loss
16: **end for**
17: **if** $e > E_{\text{warm}}$ **then**
18: **for** class $c \in \mathcal{Y}$ **do**
19: **while** $|\mathcal{K}_c| > 1$ **do**
20: $(k_{\min}, \gamma_{\min}) \leftarrow \text{WeakestComponent}(\mathcal{M}, c)$
21: **if** $\gamma_{\min} \geq \text{BICThreshold}(N_c)$ **then**
22: **break**
23: **end if**
24: $\mathcal{M} \leftarrow \text{RemoveComponent}(\mathcal{M}, c, k_{\min})$
25: **end while**
26: **end for**
27: **end if**
28: **end for**
29: **return** $f_\theta, \varphi, \mathcal{M}$

Numerical details in implementation. We apply safeguards at specific steps. In UpdateMemory$(\cdot)$, empty active slots are seeded from available class features with an initial count of one. Components with no assigned features skip the EMA updates, avoiding division by an empty assignment set. Feature vectors, means of assigned features, and centers are normalized with a denominator floored at 10^{-12}. In ReliabilityWeights $(\cdot)$, matched cosine similarities are clipped to $[-1, 1]$, radii are floored at 10^{-6} before division, and the batch mean of $1 - \rho_{\text{geo}}$ is floored at 10^{-6} before weight normalization. Geometric logits use counts floored at one before taking logarithms and

max-shift log-sum-exp evaluation. Inactive components are masked in assignment and geometric aggregation. These numerical operations preserve the gradient separation specified by $\mathrm{stopgrad}\,(\cdot)$.

A.6 Evaluation and Reporting

Checkpoint handling. The CIFAR training script saves the latest checkpoint after each epoch, and evaluation loads this checkpoint after the configured training horizon. It contains the model, optimizer, scheduler, epoch, and exported geometry state. Reproducing geometric scores requires restoring the centers, radii, assignment counts, active-component counts, and saved geometric settings together.

Score orientation and aggregation. Larger $S(x)$ indicates stronger OOD evidence. AUROC and AUPR-Out use OOD as the positive class; AUPR-In reverses both the labels and score sign. In the CIFAR evaluator, FPR95 is the ID false-positive rate at the empirical OOD recall closest to 95%. Metrics are first computed separately for each OOD dataset and then averaged equally across datasets, rather than pooling all OOD images into a single score distribution. Reported run-wise standard deviations describe variation over seeds 0, 1, and 2; the post-hoc geometric diagnostic instead uses repeated data splits of one fixed checkpoint. The two sources of variation are distinct.

Per-group protocol. For CIFAR-10-LT and CIFAR-100-LT, we use frequency-rank thresholds as implemented in our evaluation code. With class counts sorted in descending order and indexed from one, let $t_{\mathrm{many}} = N_{\lfloor 0.34C \rfloor + 1}$ and $t_{\mathrm{few}} = N_{\lfloor 0.67C \rfloor + 1}$. Many classes have $N_c > t_{\mathrm{many}}$, few classes have $N_c < t_{\mathrm{few}}$, and medium classes contain the remaining categories. For ImageNet-LT, we instead use the conventional count thresholds: many (> 100), medium (20–100), and few (< 20 training samples).

B Other Experimental Results and Analysis

B.1 Performance Results across Imbalance Ratios

Recall that the main results use the most severe imbalance ratio $\rho = 0.01$ (Table 2); it remains unclear how sensitive AdaGeo is to the imbalance severity. We therefore study the impact of ρ by sweeping it over $\{0.1, 0.05, 0.01\}$ on both CIFAR benchmarks, keeping all other settings unchanged (Table 8).

From the table, all reported metrics worsen as ρ decreases: from $\rho = 0.1$ to 0.01, ACC drops by 6.16 and 14.86 percentage points on CIFAR-10-LT and CIFAR-100-LT, respectively, AUROC drops by 3.15 and 4.09 points, and FPR95 rises by 10.10 and 7.56 points. Importantly, the degradation is gradual rather than abrupt: even at the most severe ratio, AdaGeo retains 86.11% ACC and 91.50 AUROC on CIFAR-10-LT, and 52.05% ACC and 76.73 AUROC on CIFAR-100-LT. This graceful degradation shows that the learned geometry remains usable across imbalance severities rather than collapsing at the extreme.

Two observations refine this picture. First, the AUROC declines are smaller in percentage points than the ACC declines, yet FPR95 rises substantially: a comparatively modest change in overall ID–OOD ranking can therefore accompany a marked deterioration at the 95% OOD-recall operating point. Second, the largest observed change in each metric occurs between $\rho = 0.05$ and 0.01. These comparisons also change the number of retained CIFAR training images as well as class imbalance, so they describe performance under increasingly scarce and skewed supervision rather than isolating the effect of imbalance alone. Overall, ranking-level ID–OOD discriminability is maintained as class imbalance increases. The main performance degradation occurs at the operating point and under the most extreme imbalance ratio. AdaGeo thus remains reliable across different imbalance severities and does not require severity-specific retuning.

B.2 Performance Results across Class-Frequency Groups

Recall that the main results and ablations report metrics averaged over all classes, whereas long-tailed supervision concentrates the difficulty in the tail classes. Such averages may therefore hide how the gains and remaining limitations are distributed across class-frequency groups. We thus study the frequency groups defined in Appendix A.6, asking (i) whether the improved tail-class accuracy

Table 8: Different imbalance ratios.

Dataset	ρ	ACC↑	AUROC↑	AUPR-In↑	AUPR-Out↑	FPR95↓
CIFAR-10-LT	0.1	92.27	94.65	94.75	93.36	20.45
	0.05	90.65	93.96	94.00	92.87	23.19
	0.01	86.11	91.50	91.55	89.63	30.55
CIFAR-100-LT	0.1	66.91	80.82	81.87	75.61	55.91
	0.05	62.51	79.59	80.70	73.97	58.40
	0.01	52.05	76.73	77.66	71.32	63.47

Table 9: ID accuracy by group on ImageNet-LT.

Method	Many↑	Medium↑	Few↑	Overall↑
MSP	62.01	35.90	11.91	42.69
PASCL	58.31	44.49	25.88	47.28
PATT	65.76	52.36	36.38	55.35
Ours	**66.96**	**52.78**	**37.00**	**56.09**

Table 10: Per-group performance on CIFAR-10-LT and CIFAR-100-LT.

Dataset	Group	Method	ACC↑	AUROC↑	FPR95↓
CIFAR-10-LT	Many	PATT	94.79	91.99	30.42
		Ours	93.49	94.55	21.04
	Medium	PATT	80.34	89.81	35.70
		Ours	80.34	89.91	34.80
	Few	PATT	80.39	91.40	30.83
		Ours	86.43	91.07	32.79
CIFAR-100-LT	Many	PATT	68.18	77.13	60.56
		Ours	67.93	80.49	57.16
	Medium	PATT	48.63	75.74	63.00
		Ours	47.89	79.18	59.11
	Few	PATT	30.62	70.07	73.14
		Ours	39.58	69.95	75.10

carries over to tail-group OOD detection, and (ii) where the OOD gains concentrate. Table 10 reports the per-group results.

Compared with PATT, AdaGeo raises few-group ACC by 6.04 and 8.96 percentage points on CIFAR-10-LT and CIFAR-100-LT, respectively. Yet few-group AUROC decreases by 0.33 and 0.12 points, and FPR95 increases by 1.96 points on both datasets. The additional correct tail-class predictions therefore do not translate into better tail-group OOD discrimination.

The detection gains occur instead in the many and medium groups. Many-group AUROC increases by 2.56 and 3.36 points, while FPR95 falls by 9.38 and 3.40 points on the two benchmarks. Medium-group detection also improves, although the changes on CIFAR-10-LT are small (0.10 points in AUROC and 0.90 points in FPR95). This group-wise contrast identifies tail-group OOD rejection as a remaining limitation despite the improved tail-class accuracy.

On ImageNet-LT, Table 9 shows a 0.74-point overall ACC gain over PATT, with increases of 1.20, 0.42, and 0.62 points in the many, medium, and few groups, respectively. The gains are therefore positive across all frequency groups, although the largest improvement remains in the many group. This table reports groupwise recognition only, so the ImageNet-LT results do not establish whether its tail-group OOD detection follows the CIFAR pattern.

Overall, tail-class accuracy improves consistently across groups on both benchmarks, while the CIFAR OOD gains concentrate in the well-supported many and medium groups; the tail-group OOD gap is thus bounded by scarce supervision rather than by the learned geometry.

B.3 Additional Ablation Study for Geometric Scoring

Recall Section 5.3, we have conducted ablation study. Nevertheless, the reader could be curious how about if we use different strategies for inference scoring. For this, we conduct additional ablation study. Table 11 presents the ablation results on the inference scoring, where the first row is the energy-only baseline (i.e., no-geometric-scoring configuration) and the last row is the full cohesion–separation configuration. From this table, we can see that adding cohesion or separation alone already raises AUROC by 0.44 and 0.39 points and lowers FPR95 by 0.77 and 1.00 points over the energy-only baseline. In addition, fusing them yields the best metrics (AUROC 76.73, FPR95 63.47). This set of experiments justifies that both elements contribute to the fused score. In addition, we observe that the gap between the two elements remains small, this actually explains why we use the mean-fusion design in our solution.

B.4 Results on Hyperparameter Sensitivity

Table 12 reports one-at-a-time hyperparameter sweeps on CIFAR-100-LT ($\rho = 0.01$), with reference values in bold. We study four hyperparameters. Specifically, we vary the component limit K, the temperature τ (shared by sample reweighting, geometric logits, and inference-time cohesion), the geometry-logit weight μ, and the BIC-inspired pruning strength λ_{BIC}.

Table 11: Inference scoring components on CIFAR-100-LT.

Scoring	AUROC↑	AUPR-In↑	AUPR-Out↑	FPR95↓
Energy only	76.12	76.84	70.94	64.82
Cohesion only	76.56	77.42	71.30	64.05
Separation only	76.51	77.60	70.92	63.82
Cohesion + Separation	**76.73**	**77.66**	**71.32**	**63.47**

Table 12: Hyper-parameter sensitivity.

Param	Value	ACC↑	AUROC↑	FPR95↓
K	1	51.70	75.07	66.26
	2	51.81	75.16	65.89
	3	52.05	76.73	63.47
	4	51.88	75.51	66.05
τ	0.1	51.64	76.48	64.03
	0.2	51.68	76.07	64.48
	0.5	52.05	76.73	63.47
	0.7	51.69	75.66	64.89
μ	0.0	50.13	74.40	68.41
	0.3	50.88	76.03	65.78
	0.5	52.05	76.73	63.47
	0.7	51.05	74.60	67.87
λ_{BIC}	1.0	51.44	75.57	65.31
	3.0	51.77	76.66	63.57
	5	52.05	76.73	63.47
	10.0	51.89	75.83	64.44

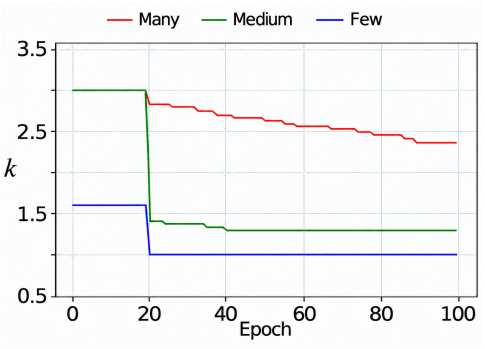


Figure 4: Mean active components per class by frequency group during training on CIFAR-100-LT.

The reference setting achieves the highest ACC and AUROC and the lowest FPR95 among the tested values. Among the four groups of sweeps, we can see that μ shows the largest variation. When removing geometric logits from the combined classifier ($\mu = 0$), it lowers ACC by 1.92 and AUROC by 2.33 percentage points and raises FPR95 by 4.94 points relative to reference point $\mu = 0.5$. Increasing μ to 0.7 also reduces AUROC by 2.13 points. This set of results indicate that the best result for μ in this grid comes from an intermediate geometry-logit weight.

We can see that the K sweep is likewise non-monotonic. Three components outperform both smaller budgets ($K = 1$, $K = 2$) and larger budgets ($K = 4$) on these main metrics. Thus, simply increasing the component limit does not necessarily improve performance.

By comparison, the largest AUROC drops across the tested τ and λ_{BIC} values are 1.07 and 1.16 points, respectively. Overall, these two hyperparameters exhibit low sensitivity.

B.5 In-depth Analysis of Adaptive Pruning

Table 13: BIC-inspired pruning statistics on CIFAR-100-LT.

Group	#Pruned	$\gamma_{c,k} \geq \gamma_{\text{BIC}}$	$\pi_{c,k} \geq 1/k_c$	Mean retained
Many	23	0	0	2.32
Medium	58	0	0	1.29
Few	19	0	0	1.00
All	100	0	0	—

Recall Section 5.3, we study the effectiveness of the BIC-inspired pruning rule. That ablation compares OOD detection performance with and without pruning. However, it remains unclear how pruning unfolds during training and whether the removed components satisfy the pruning rule. Therefore, we next address them by investigating: (i) the evolution of active-component counts across frequency groups; and (ii) the properties of the removed components.

Fig. 4 tracks the mean number k of active components per class during training on CIFAR-100-LT. From this figure we can see that the curves remain flat during the 20-epoch warm-up, and then

separate by class-frequency group. At the first pruning pass, the few-group mean number k falls to one (see the blue line), while the medium-group mean number k falls close to its final value and reaches a plateau around epoch 40 (see the green line); the many-group mean number k declines more gradually, and further removal events remain observable until approximately epoch 90 (see the red line). These curves describe changes in active-component counts; they however do not establish convergence of the centers or radii.

To address the above trouble, Table 13 summarizes the removal events over 100 epochs. Let γ_{BIC} denote the right-hand side of the pruning threshold in Eq. 7. For each removed component, the table records whether its contribution satisfied $\gamma_{c,k} \geq \gamma_{\mathrm{BIC}}$ or its assignment proportion satisfied $\pi_{c,k} \geq 1/k_c$, where k_c is the active-component count immediately before removal. "Mean retained" is the final mean active-component count per class. From this table, we can see that the final mean active-component counts are 2.32, 1.29, and 1.00 for many, medium, and few classes, respectively, with medium classes accounting for 58 of the 100 removals. Together, the counts show how frequency-based initialization and subsequent pruning allocate different capacities under this training run, and they further qualify the held-out diagnostic discussed in Appendix **??**. That is, multiple post-hoc centers can reduce distortion for few-shot classes, but that reduction does not determine how many online components should be retained under long-tailed supervision.

Among the 100 removed components, none met the BIC threshold, as expected from the pruning condition; and none had assignment support at or above the uniform share $1/k_c$. The first result checks that the recorded removals follow the implemented rule; the second shows that removed components also had below-uniform support in this run.

B.6 Comparison Results on Training Efficiency

Table 14 reports the mean end-to-end training time over three runs under the hardware, epoch count, and ID batch sizes listed in Table 7, excluding validation, post-hoc calibration, and testing. Times are formatted as hh:mm:ss; the speedup is the PATT time divided by the AdaGeo time, and the reduction is the time saved relative to PATT.

Table 14: Training-time comparison with PATT.

Dataset	PATT	AdaGeo	Speedup	Reduction
CIFAR-10-LT	00:25:58	00:15:33	1.67×	40.12%
CIFAR-100-LT	00:23:20	00:14:34	1.60×	37.57%
ImageNet-LT	37:37:29	20:11:56	1.86×	46.31%

We can see that AdaGeo is faster on all three benchmarks. The gain follows from restricting encoder optimization to ID images. In our solution, geometric modeling consists of lightweight EMA and assignment updates, which add no learnable parameters and do not enlarge backpropagation. This avoids the auxiliary OOD batches that other pipelines forward throughout optimization. Note that, we actually experimented with adding auxiliary OOD batches during optimization, but this uniformly degraded performance on CIFAR. Thus, we confine auxiliary OOD data to the one-shot post-hoc calibration, which is excluded from this table.

The speedup is largest on ImageNet-LT, where the omitted OOD batches are most expensive, and slightly smaller on CIFAR-100-LT. This is because its larger number of classes and components raises the relative cost of the geometry updates.

C Frequently Used Notations

Table 15 summarizes the frequently used symbols for ease of reference. These symbols are organized in three blocks: (i) data and features, (ii) local geometry, and (iii) learning and OOD scoring.

Table 15: Main notations used in the paper.

Symbol	Description
Data and Features	
$\mathcal{D}_{\text{in}}, \mathcal{Y}$	ID training set and class-label set.
$x, y, c, \hat{c}$	Input image, ground-truth label, class index, and predicted class.
$C, N, N_c, N_{\min}, N_{\max}$	Number of classes, total training images, training images in class c, minimum and maximum class training counts.
ρ	Imbalance ratio: smallest to largest class count.
π_c	Class prior N_c/N.
$t_{\text{many}}, t_{\text{few}}$	Frequency-rank thresholds separating many/medium/few groups.
f_θ, φ	Encoder with parameters θ and classifier.
$\mathbf{z}, \mathbf{f}$	Encoder output and its ℓ_2-normalized feature.
Local Geometry	
$K, k_c, \mathcal{K}_c$	Maximum components per class, active count, and active index set.
$k_c^{\text{init}}, k_{\text{base}}$	Initial active count and initialization granularity parameter.
k, k^*	Component index and matched active-component index.
$\mathbf{m}_{c,k}$	Center of component (c, k).
$\mathcal{F}_{c,k}, \bar{\mathbf{f}}_{c,k}$	Assigned feature set and its normalized mean.
$n_{c,k}, \pi_{c,k}$	Cumulative assignment count and within-class support proportion.
$r_{c,k}$	EMA compactness statistic based on cosine distance.
β_m, β_r	EMA momenta for centers and compactness.
$\gamma_{c,k}$	Component contribution $\pi_{c,k}(1 - r_{c,k})$.
$\lambda_{\text{BIC}}, \gamma_{\text{BIC}}$	Pruning-strength parameter and class-dependent pruning threshold.
E_{warm}	Warm-up epochs before pruning is activated.
Learning and OOD Scoring	
τ	Shared temperature for reweighting, geometric logits and cohesion.
$\rho_{\text{geo}}(\mathbf{f}, c)$	Local geometric reliability of a feature in class c.
$w(\mathbf{f}, y)$	Normalized reliability-aware sample weight.
$b_{c,k}$	Component log-count relative to the within-class maximum.
$\ell_c^{\text{geo}}, \ell_c, \boldsymbol{\ell}(\mathbf{z})$	Geometric class logit, combined prior-adjusted logit, and the vector of the latter over all classes.
μ	Weight of geometric evidence in the combined logit.
$\mathcal{L}, \mathcal{L}_{\text{geo}}, \mathcal{L}_{\text{LA}}$	Joint objective, geometry loss, and logit-adjustment loss.
$\mathcal{L}_{\text{geo}}^{(1\to v)}$	Geometry loss for positive view v, weighted by the anchor.
$\lambda_{\text{geo}}, \lambda_{\text{LA}}$	Weights of the two training losses.
$Q_{\text{C}}, Q_{\text{S}}, Q(\mathbf{f})$	Local cohesion, inter-class separation, and their mean confidence.
$\varphi_c(\mathbf{z}), E(x)$	Classifier logit used for OOD scoring and classifier energy.
$\alpha, S(x)$	Automatically calibrated fusion weight and final OOD score.
ε	Numerical safeguards for denominators.