DPLC: Dirichlet Process Guided Long-tail Clustering

Letian Cui[2], **Xulun Ye**[2], **Kun Zhou**[1,3†]
[1] School of Architecture & Urban Planning, Shenzhen University, Shenzhen, China
[2] School of Artificial Intelligence, Ningbo University, Ningbo, China
[3]Intelligent Computing Center, Shenzhen University
goujie114514@gmail.com, yexulun@nbu.edu.cn, zhoukun@szu.edu.cn

Abstract

The main challenge of long-tailed deep clustering is that imbalanced class frequencies cause head classes to dominate the partitioning of the representation space, which in turn submerges or distorts the cluster structures of tail classes. Existing methods typically alleviate this issue through heuristic head-tail partitioning or fixed auxiliary clustering. However, such approaches rely on manually specified discrete granularities and therefore struggle to capture the continuous latent structure of real-world long-tailed data. To address this limitation, we propose DPLC, a Dirichlet Process guided Long-tailed Clustering method that adaptively models the latent sub-cluster structure from a nonparametric Bayesian perspective. Specifically, DPLC periodically extracts soft latent sub-clusters from self-supervised embeddings and leverages them as data-driven rebalancing signals to mitigate the clustering bias induced by head-class dominance. By requiring neither a predefined number of auxiliary clusters nor explicit binary head-tail partitioning, DPLC enables more flexible modeling of complex long-tailed distributions. Extensive experiments show that DPLC consistently outperforms existing methods across multiple long-tailed clustering benchmarks and a wide range of imbalance ratios. The code is available at https://github.com/V2tPkDuck/DPLC.

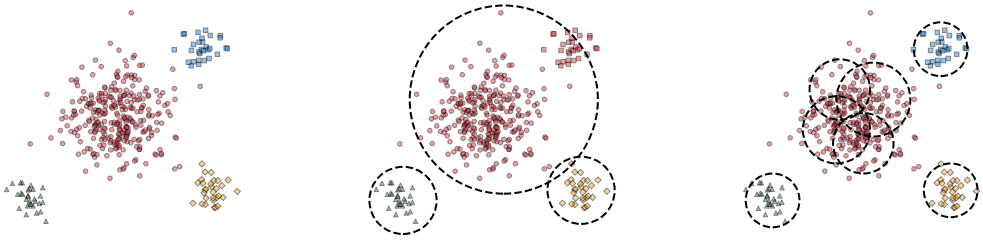
1 Introduction

Deep clustering is an essential task in unsupervised representation learning that trains deep neural networks to partition data points into coherent, semantically consistent clusters without relying on manual annotations[Xie et al., 2016, Caron et al., 2018, Zhou et al., 2024, Ren et al., 2024]. Existing deep clustering algorithms generally operate under the implicit assumption that category distributions are relatively balanced or that different categories will receive approximately equal optimization opportunities during training[Van Gansbeke et al., 2020, Li and Jia, 2025, Zhang et al., 2024, Jiang et al., 2021]. In real-world scenarios, a few head classes dominate the sample count while many tail classes contain only a handful of instances. These long-tailed distributions pose a significant challenge in unsupervised representation learning. Head classes tend to dominate the organization of the representation space and dictate the formation of clustering boundaries. Consequently, the representation space is skewed and the semantic boundaries of minority categories are marginalized, leading to a significant performance drop[Kang et al., 2019, Zhang et al., 2023].

Existing unsupervised long-tail methods generally rely on post-hoc heuristics or fixed auxiliary clusters with manually chosen granularity[Qian, 2023, Li and Jia, 2025, Zhang et al., 2024, Li et al., 2025b]. For instance, early deep clustering approaches such as DeepCluster [Caron et al., 2018] adopt a fixed number of K-means clusters to generate pseudo-labels; when applied to long-tailed data, this design still requires manual tuning of the cluster count and decision thresholds. If the granularity is too coarse, tail samples are forced into head clusters; if too fine, head classes are artificially fragmented.

†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Long-Tail Distribution (b) Traditional Methods(Biased) (c) DPLC: Latent Sub-Clusters

Figure 1: Illustration for our method(red: head class, other colors: tail classes; points shown are random samples generated by Matplotlib for illustrative purposes only). Moving from the problem setting to a coarse-grained baseline, and finally to our method, we show how DPLC breaks the head/tail binary by discovering rich latent sub-clusters via a Dirichlet Process on unsupervised embeddings. These sub-clusters provide a non-parametric signal for dynamic class reweighting (co-occurrence counts) and similarity regularization, enabling balanced clustering even under severe imbalance.

These limitations largely stem from predefined discrete structural assumptions. Real-world long-tailed data rarely conforms to a simple binary head/tail split; instead, it exhibits a continuous, unknown spectrum of latent structures. Solely relying on explicit cutoffs or fixed granularity, therefore, may not adequately capture this complexity, making it challenging to effectively mitigate head-dominated clustering bias.

Motivated by these observations, we explore a more data-driven modeling approach that aims to adaptively discover latent structures directly within the representation space, utilizing these structures to provide dynamic rebalancing signals for clustering optimization. To move beyond such rigid discrete assumptions and adaptively discover latent structures, we introduce DPLC (Dirichlet Process Guided Long-tail Clustering), a nonparametric Bayesian framework that models the generative distribution of self-supervised embeddings via a Dirichlet Process prior[Ferguson, 1973, Neal, 2000, Blei and Jordan, 2006, Teh et al., 2004]. As illustrated in Figure 1, DPLC periodically extracts a rich set of soft latent sub-clusters from the embedding space without needing a predefined number of clusters. These sub-clusters serve as a non-parametric, data-driven signal for rebalancing. The posterior assignments are transformed into surrogate gradient signals that dynamically reweight the clustering loss and enforce pairwise similarity regularization. By starting directly from the underlying distribution rather than relying solely on discrete class counts, DPLC aims to mitigate clustering bias at its root and preserve structural integrity across all classes, enabling distribution-level re-weighting in a fully unsupervised manner.

In summary, the main contributions of this work are as follows:

- We propose **DPLC**, an unsupervised long-tailed clustering framework that directly models the underlying data distribution via a Dirichlet process prior, avoiding heuristic head/tail thresholding and enabling principled nonparametric inference.
- We develop a differentiable learning scheme that transforms the nonparametric Bayesian posterior into gradient signals by integrating score-function estimation, inverse-propensity reweighting, and similarity alignment, enabling end-to-end optimization without supervision.
- Extensive experiments on standard long-tailed benchmarks (CIFAR, STL-10, Tiny-ImageNet, ImageNet-50, ImageNet-100) show that DPLC achieves state-of-the-art performance, validating the effectiveness of label-free, distribution-level reweighting.

2 Related Works

Deep Clustering aims to train deep neural networks to group unlabeled data into coherent clusters without any manual annotations [Zhou et al., 2024]. Compared to supervised learning, it eliminates the costly process of label annotation. Existing deep clustering methods can be broadly divided into two categories: representation-learning-based approaches[Tao et al., 2021, Huang et al., 2022, 2014]

and clustering-head-based approaches[Van Gansbeke et al., 2020, Li et al., 2021, Niu et al., 2022, Li et al., 2022, Qian, 2023]. Representation-learning methods first learn meaningful embeddings using autoencoders[Huang et al., 2014, Xie et al., 2016, Guo et al., 2017] or contrastive objectives [Huang et al., 2022, Li et al., 2023, Tsai et al., 2020], then apply conventional clustering algorithms such as K-means[Hartigan and Wong, 1979] or spectral clustering [Von Luxburg, 2007]. In contrast, clustering-head-based methods attach a linear head after the encoder and directly optimize pseudo-labels, making the overall architecture resemble supervised classification [Van Gansbeke et al., 2020, Li et al., 2021, Qian, 2023, Tian et al., 2017, Jia et al., 2024, Li et al., 2025a]. Most of these techniques, however, are designed and evaluated under the assumption of balanced class distributions. When confronted with the long-tailed distributions prevalent in real-world data, their performance degrades substantially [Li and Jia, 2025].

Long-tailed Learning. Real-world datasets frequently exhibit long-tailed class distributions, where a few head classes dominate while many tail classes contain only a handful of samples. Models trained on such data tend to favor head classes and perform poorly on tail classes. In the supervised setting, this bias is commonly mitigated through three families of techniques: resampling that over-samples tail classes or under-samples head classes [Chawla et al., 2002, Wang et al., 2019, Zhang and Pfister, 2021]; re-weighting that assigns class-specific weights in the loss function according to label frequencies [Cui et al., 2019, Cao et al., 2019, Tan et al., 2020]; logit adjustment that modifies the model's output logits to equalize prediction difficulty across classes [Tan et al., 2020, Menon et al., 2020, Zhang et al., 2021]. All of these strategies rely on explicit label information and therefore cannot be directly transferred to unsupervised deep clustering. While the supervised long-tail learning strategies have been thoroughly explored, unsupervised long-tailed learning remains largely underexplored. A few recent works focus on representation learning under long-tailed distributions[Jiang et al., 2021, Zhou et al., 2022], while others treat long-tailed clustering as a downstream task[Zhang et al., 2024]. Recent advances have also proposed self-supervised contrastive frameworks specifically designed to learn robust representations from long-tailed data without any label supervision[Hoang et al., 2025, Yi et al., 2025]. Furthermore, Bayesian nonparametric techniques have begun to appear in unsupervised clustering pipelines to better capture the continuous spectrum of class frequencies[Wang et al., 2021, Bing et al., 2023]. The most relevant approach to our work is MiniClustering [Li et al., 2025b], which uses fixed K-means mini-clusters to guide reweighting but requires manually specifying the number of mini-clusters M and a hard threshold δ. In contrast, DPLC eliminates both hyperparameters by using a Dirichlet Process (CRP-Gibbs sampler) that automatically adapts the number of sub-clusters to the data geometry. To the best of our knowledge, training a clustering head for long-tailed deep clustering by directly modeling the underlying probability distribution through a nonparametric Bayesian framework and using the resulting posterior for reweighting has not been explored before.

3 Preliminaries

Dirichlet Processes. A Dirichlet Process $G \sim \mathrm{DP}(\alpha, H)$ is a distribution over probability measures on a space Θ: for any measurable partition $A_1, \ldots, A_m$, the vector $(G(A_1), \ldots, G(A_m))$ follows a Dirichlet distribution with parameters $(\alpha H(A_1), \ldots, \alpha H(A_m))$[Ferguson, 1973, Gershman and Blei, 2012]. A draw $G = \sum_k \pi_k \delta_{\phi_k}$ is almost surely discrete, with atoms $\phi_k \sim H$ and stick-breaking weights π_k, and the concentration α controls how quickly new atoms appear. Marginalizing G out and conditioning on n draws yields the **Chinese Restaurant Process**: customer $n{+}1$ joins an existing table k with probability $n_k/(n+\alpha)$ and opens a new table with probability $\alpha/(n+\alpha)$, where n_k is the current occupancy. By exchangeability, the full conditional of one assignment given the rest is exactly the cavity conditional in Eq. (1), which is what the Gibbs sampler iteratively samples.

4 Methods

This section presents the DPLC framework.

Problem formulation. Given an unlabeled dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ with a long-tailed class distribution, the goal is to learn an encoder $f_\theta : \mathbf{x}_i \mapsto \mathbf{z}_i$ and a clustering head $h_\omega : \mathbf{z}_i \mapsto \{1, \ldots, K\}$ that partitions embeddings into K balanced clusters, despite severe sample imbalance between head and tail classes, without any label supervision.

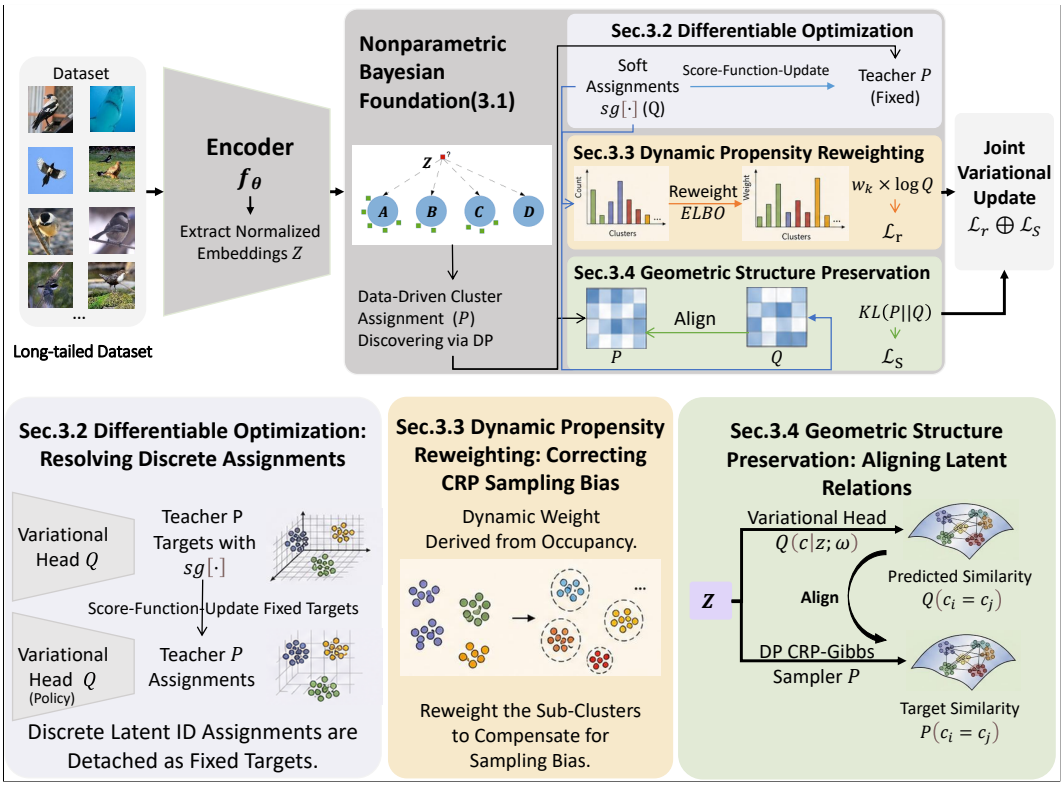


Figure 2: **DPLC pipeline.** The model alternates between nonparametric DP sub-cluster discovery and end-to-end rebalanced clustering. P is the CRP-Gibbs posterior on embeddings, providing assignments and cluster weights as a nonparametric teacher. Q is the clustering head's predicted soft assignment. Matching Q to reweighted P balances clusters without manual cutoffs. The mathematical justification for matching Q to the reweighted P via Eq. 8 is established in Appendix. A.

We organize the method into the following components. For each component, we first state its objective, then describe the approach, and finally provide implementation details. We first establish a **Nonparametric Bayesian Foundation** in Sec. 4.1. The key motivation is to move beyond rigid binary head/tail splits and fixed cluster numbers: by placing a Chinese Restaurant Process (CRP) prior over the embeddings, the model automatically infers a soft, continuous spectrum of sub-clusters directly from the data geometry. From this posterior, we extract three quantities that are otherwise unavailable without labels: soft cluster assignments, cluster sizes, and pairwise co-occurrence similarities. However, transforming this posterior into a fully differentiable training signal for the clustering head introduces three key challenges. Discrete latent assignments prevent direct gradient backpropagation, which we resolve in Sec. 4.2 via a **Differentiable Optimization** using a score-function estimator augmented with stop-gradient operations. Inherent sampling bias in the CRP prior exacerbates long-tail skew by favoring dominant clusters; we correct this in Sec. 4.3 through **Dynamic Propensity Reweighting**. Loss of global structural consistency occurs when optimizing point-wise assignments in isolation; we enforce **Geometric Structure Preservation** in Sec. 4.4 via a pairwise similarity regularization. These components are not independent but jointly shape a balanced clustering objective. We detail each mechanism below and ultimately unify them into the joint loss in Eq. (10). Finally, Figure 2 illustrates the overall pipeline of DPLC.

4.1 Nonparametric Bayesian Foundation: The CRP Generative Model

This subsection aims to define the generative process under a CRP prior that supplies the nonparametric Bayesian foundation for all subsequent reweighting and alignment steps. Let $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$ be the D-dimensional representation obtained from a backbone encoder parameterized by θ. We model the generative distribution of these embeddings via a CRP prior (Sec. 3)[Gershman and Blei, 2012, Ferguson, 1973, Neal, 2000]. Concretely, a Dirichlet Process $G \sim \mathrm{DP}(\alpha, H)$ places a prior over distributions whose draws are almost surely discrete; marginalizing G out yields the Chinese

Restaurant Process, in which a sample joins an existing cluster with probability proportional to its current occupancy and opens a new cluster with probability proportional to α. The Gibbs sampler therefore infers the posterior of the latent cluster assignment c_i by iteratively sampling the cavity conditional:

$$P(c_i = k \mid \mathbf{c}_{\setminus i}, \mathbf{z}, \alpha, \phi) \propto \begin{cases} n_{k,\setminus i} F(\mathbf{z}_i \mid \phi_k), & k \in \{1, 2, \ldots, K\} \\ \alpha \int F(\mathbf{z}_i \mid \phi) H(\phi) d\phi, & k = K + 1 \end{cases} \tag{1}$$

In this formulation, $\mathbf{c}_{\setminus i}$ represents the assignments of all samples excluding i. The count $n_{k,\setminus i}$ denotes the number of samples currently assigned to cluster k. The parameter α is the concentration parameter. $F(\mathbf{z}_i \mid \phi_k)$ denotes the likelihood of the embedding, modeled here as a von Mises-Fisher (vMF) density[Banerjee et al., 2005] to account for the normalized embedding space. Let $\mathbf{z}_i \in \mathbb{R}^D$ be the L_2-normalized representation, so that $F(\mathbf{z}_i \mid \phi_k) = C_D(\kappa) \exp(\kappa \boldsymbol{\mu}_k^\top \mathbf{z}_i)$. The vMF distribution is the standard model for directional data on the unit hypersphere, which matches the normalized nature of self-supervised embeddings produced by contrastive objectives like BYOL[Grill et al., 2020]. $H(\phi)$ is the conjugate prior for the cluster parameters: $\phi_k = (\boldsymbol{\mu}_k, \kappa)$ collects the mean direction and the concentration of the vMF density of cluster k, with a vMF prior on $\boldsymbol{\mu}_k$ and a Gamma prior on κ. K denotes the current number of non-empty clusters.

4.2 Differentiable Optimization: Resolving Discrete Assignments via Score-Function

This subsection aims to resolve the non-differentiability of discrete latent assignments. The Gibbs sampler outputs hard cluster IDs, which block gradient flow. We circumvent this by treating those IDs as fixed "targets" from a teacher distribution and letting the neural head mimic them via a policy-gradient-style update, stabilized by detaching the teacher's own scores. We introduce a variational distribution $Q(c \mid \mathbf{z}; \omega)$, parameterized by a clustering head $h_\omega : \mathbb{R}^D \to \mathbb{R}^K$, which is a categorical distribution over the K clusters:

$$Q(c = k \mid \mathbf{z}; \omega) = \frac{\exp\big([h_\omega(\mathbf{z})]_k\big)}{\sum_{j=1}^{K} \exp\big([h_\omega(\mathbf{z})]_j\big)}, \qquad k = 1, \ldots, K. \tag{2}$$

To bridge the discrete CRP posterior with a differentiable clustering head, we treat the nonparametric posterior as a stochastic teacher and train the neural head via a score-function estimator[Williams, 1992, Mnih and Gregor, 2014, Schulman et al., 2015] with stop-gradient operations; a derivation and intuitive explanation are provided in Appendix. B.

$$\mathcal{L}_{\mathrm{ELBO}}(\theta, \omega) = \mathbb{E}_{Q(c \mid \mathbf{z}; \omega)} \big[\log P(\mathbf{z} \mid c, \theta) + \log P(c) \big] + \mathcal{H}\big(Q(c \mid \mathbf{z}; \omega)\big) \tag{3}$$

Here, $P(\mathbf{z} \mid c, \theta) = F(\mathbf{z} \mid \phi_c)$ is the vMF likelihood of Eq. (1), whose dependence on θ enters only through $\mathbf{z} = f_\theta(\mathbf{x})$; $P(c)$ is the CRP prior induced by the same cavity conditional; and $\mathcal{H}(Q) = -\mathbb{E}_Q[\log Q(c \mid \mathbf{z}; \omega)]$ is the entropy of the variational head.

$$\nabla_\theta \mathcal{L}_{ELBO} \approx \frac{1}{M} \sum_{m=1}^{M} \nabla_\theta \log P(\mathbf{z} \mid \hat{c}^{(m)}, \theta) \tag{4}$$

Since $\mathcal{H}(Q)$ and $\log P(c)$ do not depend on θ, only the likelihood term contributes to Eq. (3), which we differentiate pathwise through $\mathbf{z} = f_\theta(\mathbf{x})$. Here $\hat{c}^{(m)}$ is drawn from the CRP posterior $P(c \mid \mathbf{z})$, which is refreshed every 15 epochs and held fixed in between (Sec. 5.1), so the inner expectation is always taken over a fixed target within one round. For the clustering head ω, we optimize a surrogate objective whose gradient is the score-function estimator with a stop-gradient baseline:

$$\mathcal{L}_{\mathrm{surr}}(\omega) = -\mathbb{E}_{\hat{c} \sim P(c \mid \mathbf{z})} \Big[\big(\log p(\mathbf{z}, \hat{c} \mid \theta) - \mathrm{sg}\big[\log Q(\hat{c} \mid \mathbf{z}; \omega) \big] \big) \log Q(\hat{c} \mid \mathbf{z}; \omega) \Big]. \tag{5}$$

Since $\nabla_\omega Q = Q \, \nabla_\omega \log Q$, differentiating Eq. (5) recovers exactly the score-function gradient, with the stop-gradient baseline now placed inside the loss rather than inside a gradient expression. The

stop-gradient operator $\mathrm{sg}[\cdot]$ detaches the head's own log-probability from the "reward", preventing the estimator from becoming self-reinforcing. Sampling from the fixed posterior P instead of the evolving Q turns the update into **variational distillation**: the head ω is trained to mimic the statistically robust assignments of the Bayesian sampler. This procedure is variational EM with a nonparametric Bayesian prior[Dempster et al., 1977, Minka, 2001]: the Gibbs sampler performs the E-step, and the gradient update on ω performs the M-step, whose score-function form is standard in Monte Carlo and black-box variational EM. A complete derivation and an intuitive reinforcement-learning interpretation are provided in Appendix B.

4.3 Dynamic Propensity Reweighting: Correcting CRP Sampling Bias

This subsection aims to correct the inherent sampling bias of the CRP prior, which amplifies head dominance under long-tailed distributions. The CRP's rich-get-richer dynamic causes head sub-clusters to dominate the gradient signal. We counterbalance this by up-weighting gradients from rarer sub-clusters, similar to inverse-frequency reweighting but now driven by the nonparametric posterior occupancy. We introduce dynamic inverse-propensity weights w_k[Rosenbaum and Rubin, 1983, Swaminathan and Joachims, 2015] that explicitly compensate for the prior's skew. The cavity conditional in Eq. (1) already fixes the propensity of cluster k for sample i,

$$\pi_k = \frac{n_{k,\setminus i} + \alpha/K}{N - 1 + \alpha}, \tag{6}$$

i.e., the probability that i is assigned to k given all other assignments. By the Horvitz–Thompson principle, a quantity computed from cluster k is estimated without bias by inverse-weighting each draw with π_k, which gives

$$w_k = \pi_k^{-1} = \left[\frac{n_{k,\setminus i} + \alpha/K}{N - 1 + \alpha}\right]^{-1}. \tag{7}$$

Equation (7) is therefore not a heuristic choice among many: it is the weighting that yields an unbiased M-step gradient under the CRP model, and, unlike thresholding schemes, it needs no head/tail cutoff and varies continuously with n_k; its only free parameter is the prior concentration α (Sec. 5.3). Here, $n_{k,\setminus i}$ is the cluster occupation count excluding sample i, N is the total number of samples, K is the number of active clusters, and α is the CRP concentration parameter. The term α/K is a small smoothing correction that avoids zero weights when $n_{k,\setminus i} = 0$; it does not affect the asymptotic behavior of the CRP prior. Rarer sub-clusters (small n_k) thus receive larger weights, which corrects the long-tail bias of the CRP prior by amplifying the gradients of under-represented sub-clusters. Rather than maximizing the ELBO, which inherits the rich-get-richer bias, we optimize a rebalanced surrogate whose gradient equals the inverse-propensity-weighted score function, as shown in Appendix A:

$$\mathcal{L}_r = -\sum_{i=1}^{N} \mathbb{E}_{\hat{c}_i \sim P(c|\mathbf{z}_i)} \left[w_{\hat{c}_i} \log Q(\hat{c}_i \mid \mathbf{z}_i; \omega) \right], \tag{8}$$

where $w_{\hat{c}_i}$ is the weight of Eq. (7) evaluated at the sampled cluster index $\hat{c}_i$. Substituting the score-function gradient into a first-order Taylor approximation yields an effective loss whose gradient with respect to the predicted soft assignment $Q_{i,j} := Q(c = j \mid \mathbf{z}_i; \omega)$ is proportional to $w_j \cdot \mathbb{I}(\hat{c}_i = j)/Q_{i,j}$. Minimizing the weighted cross-entropy in Eq. (8) exactly reproduces this gradient, thereby grounding the inverse-propensity weights w_j in the ELBO objective.

4.4 Geometric Structure Preservation: Aligning Latent Relations

This subsection aims to preserve global cluster geometry while locally rebalancing by enforcing consistency between the pairwise similarity matrices of the variational head's predictions and the DP posterior's soft assignments. Point-wise reweighting alone can drift cluster boundaries and fracture the manifold. We explicitly enforce that two samples are similar in the neural head's view if and only if they are similar under the DP posterior, thereby anchoring the local updates to a stable global structure. To anchor the local neural updates to the stable global structure discovered by the

Dirichlet Process, we introduce a similarity alignment loss $\mathcal{L}_s$. The core idea is to distill the relational knowledge embedded in the nonparametric posterior. To construct the target, we first compute the empirical co-occurrence frequency $P(c_i = c_j \mid \mathbf{z}_i, \mathbf{z}_j)$ from multiple Gibbs sampling sweeps, which serves as a soft, probabilistic measure of pairwise affinity. The alignment term then minimizes the Kullback-Leibler divergence between this target affinity matrix and the predicted similarity matrix from the clustering head, acting as a soft structural regularizer[Dwibedi et al., 2021, Van Gansbeke et al., 2020].

$$\mathcal{L}_s = \sum_{i,j} P(c_i = c_j | z_i, z_j) \log \frac{P(c_i = c_j | z_i, z_j)}{Q(c_i = c_j | z_i, z_j)} \tag{9}$$

In this pairwise formulation, $P(c_i = c_j | z_i, z_j)$ represents the elements of the target similarity matrix, and $Q(c_i = c_j | z_i, z_j)$ represents the elements of the predicted similarity matrix. In practice, $P(c_i = c_j \mid \mathbf{z}_i, \mathbf{z}_j)$ is estimated as the empirical co-occurrence frequency of i and j being assigned to the same cluster by the Gibbs sampler (averaged over multiple samples). We implement $\mathcal{L}_s$ as a mean squared error between the row-normalized similarity matrices of P and Q, which approximates this KL objective near convergence while offering smoother gradients (see Appendix. C). $\mathcal{L}_s$ acts as a structural constraint, ensuring that the local updates of $\mathcal{L}_r$ do not violate the global cluster geometry. The likelihood in Eq. (1) is point-wise: once the per-sample assignments are reweighted independently, repeated updates can still drift the neighborhood structure, which is why the relational anchor is needed (Sec. 5.3). In practice $\mathcal{L}_s$ is evaluated on mini-batches, so its cost is $\mathcal{O}(B^2)$ per step, while the global target matrix P is computed once per DP refresh and reused; the term therefore adds no $\mathcal{O}(N^2)$ overhead (Sec. **??**). The final joint objective is

$$\mathcal{L} = \mathcal{L}_r + \beta \mathcal{L}_s \tag{10}$$

where β is a scaling multiplier balancing point-wise assignment and structural alignment.

5 Experiments

5.1 Experimental Settings

Datasets. Following the standard protocols in long-tailed deep clustering [Li and Jia, 2025, Li et al., 2025b, Van Gansbeke et al., 2020], we evaluate DPLC on six benchmark datasets with varying degrees of imbalance: CIFAR-10[Krizhevsky et al., 2009], CIFAR-20[Krizhevsky et al., 2009] and STL-10[Coates et al., 2011], at imbalance ratios (imb_ratio) of 5 and 10; Tiny-ImageNet[Le et al., 2015], ImageNet-100 and ImageNet-50[Liu et al., 2019]. ImageNet-100 and ImageNet-50 are subsets of ImageNet[Deng et al., 2009], containing 100 and 50 randomly selected classes, respectively. CIFAR-20 is CIFAR-100 using 20 super-classes as the cluster number. For CIFAR-10, CIFAR-20, STL-10, Tiny-ImageNet, ImageNet-100, and ImageNet-50, we construct long-tailed training sets by sampling the original data with exponentially decaying class sizes, controlled by an imbalance ratio $\rho = \frac{\max_k N_k}{\min_k N_k}$ (Zhou et al. [2020], Cao et al. [2019]). On STL-10 we combine the official train and test sets for unsupervised training, following[Li et al., 2025b]. Unless otherwise noted, only the training split is used for representation pre-training and clustering; the test split serves for evaluation and periodic DP inference.

Implementation details. The encoder is pre-trained with BYOL for 1000 epochs (SGD, cosine schedule). We then freeze the encoder, initialize the target head weights with normalized K-means centers, and cache all training features. The head is trained for 300 epochs with Adam. Every 15 epochs we run the CRP-Gibbs sampler ($\alpha = 6.0$, 60 sweeps, 3 restarts) on the cached training embeddings (PCA to 64 dim, L2-normalized) to obtain a global soft posterior over sub-clusters. In each batch, two augmented views are encoded; the head produces logits ℓ^1 and, with stop-gradient, ℓ^2[Pearson, 1901, Abdi and Williams, 2010]. A confidence mask ($\tau = 0.95$) filters low-confidence samples. The loss is $\mathcal{L} = \mathcal{L}_r + \beta \mathcal{L}_s$, where $\mathcal{L}_r$ is the masked weighted cross-entropy (Eq 8) using dynamic inverse-propensity weights derived from the DP posterior, and $\mathcal{L}_s$ aligns the pairwise similarity of the head's predictions with that of the DP soft assignments (Eq. 9, implemented as MSE; $\beta = 0.7$). All hyperparameters are fixed across datasets. We also test SwAV and DINOv3 backbones in the ablations (Table 4); MoCo is omitted due to significantly lower accuracy[Caron et al., 2020, Siméoni et al., 2025, Chen et al., 2021]. For more details, read the Appendix. C.

All experiments were conducted on a single NVIDIA H100 GPU. Since the encoder is frozen after pre-training, we cache the features once and reuse them across all epochs; with this optimization the complete DPLC pipeline takes about 1.2 hours on CIFAR-10, of which 7 minutes are spent on feature caching, 25 minutes on all CRP-Gibbs sweeps, and 40 minutes on head training. Training times for larger benchmarks scale roughly linearly with dataset size and remain within a few hours. Despite the periodic CRP-Gibbs sampling, the amortized computational cost stays comparable to that of contemporary deep clustering baselines; a stage-wise time breakdown and the memory footprint are given in Appendix D.

Method	CIFAR-10				CIFAR-20				STL-10			
	ACC	CAA	NMI	ARI	ACC	CAA	NMI	ARI	ACC	CAA	NMI	ARI
SCAN (ECCV2020)	55.6	60.3	54.0	39.3	38.8	39.7	42.6	25.1	48.8	49.8	46.2	34.9
SDCLR (ICML2021)	44.1	50.5	43.4	38.0	40.2	38.7	40.3	23.8	35.8	37.4	34.4	19.3
CC (AAAI2021)	25.2	20.9	18.3	1.47	16.0	12.0	18.0	0.00	41.5	30.0	46.2	22.4
IDFD (ICLR2021)	56.7	63.0	51.8	37.1	31.2	30.7	30.5	16.2	41.6	41.2	39.6	24.3
CoNR (NeurIPS2023)	41.4	43.5	34.9	23.3	23.8	23.4	21.6	12.1	32.4	31.3	29.0	18.2
ProPos (TPAMI2022)	51.4	59.2	52.2	34.1	40.7	39.3	42.6	26.4	37.2	39.3	34.5	21.8
DMICC (AAAI2023)	40.6	42.5	36.9	25.4	25.2	23.1	20.6	7.5	45.9	45.6	45.3	35.5
SeCu (ICCV2023)	56.1	59.0	54.5	40.3	34.8	33.7	35.8	19.9	51.6	50.8	48.2	39.1
LFSS (ICML2025)	58.1	63.1	59.7	43.3	43.5	42.2	46.8	28.6	46.6	43.7	46.0	34.0
ConMix (ICLR2025)	61.6	65.4	59.8	45.6	42.8	41.0	43.9	27.7	48.9	49.7	48.8	35.9
MiniClustering (ICLR2026)	62.3	67.4	**65.6**	50.4	44.2	43.6	47.2	31.3	49.8	51.1	48.4	34.9
DPLC (ours)	**65.7**	**68.6**	65.5	**51.8**	**45.0**	**45.0**	**48.7**	**32.1**	**50.1**	**52.0**	**49.1**	**35.2**

Table 1: Clustering performance in percentage (%) on three benchmark datasets (imbalance ratio = 5) across various methods.

Method	CIFAR-10				CIFAR-20				STL-10			
	ACC	CAA	NMI	ARI	ACC	CAA	NMI	ARI	ACC	CAA	NMI	ARI
SCAN (ECCV2020)	52.0	50.1	59.4	43.8	37.0	32.4	37.8	23.1	47.1	45.4	47.5	34.2
SDCLR (ICML2021)	38.9	44.3	42.5	26.5	37.8	35.9	39.6	22.9	34.6	37.9	32.2	17.3
CC (AAAI2021)	40.6	27.5	43.9	18.8	19.9	14.3	21.9	1.1	43.0	35.3	44.7	25.4
IDFD (ICLR2021)	47.5	54.9	48.4	33.1	28.7	27.2	28.6	15.1	38.6	34.7	36.8	22.8
CoNR (NeurIPS2023)	31.4	44.3	29.2	17.8	20.3	17.9	17.3	8.0	34.8	32.5	30.7	21.0
ProPos (TPAMI2022)	46.1	49.3	52.5	34.2	36.8	33.6	40.1	22.5	35.6	38.2	37.2	23.9
DMICC (AAAI2023)	36.6	39.5	36.8	25.9	24.7	21.9	20.7	10.1	41.3	41.9	38.7	30.3
SeCu (ICCV2023)	51.5	57.0	54.8	39.6	32.0	30.7	33.3	17.6	48.2	49.6	50.0	36.8
LFSS (ICML2025)	56.3	59.7	57.9	43.0	41.1	**40.0**	**46.8**	**28.4**	43.3	42.3	45.4	30.8
ConMix (ICLR2025)	53.3	58.2	57.1	40.8	**41.7**	39.3	43.6	27.0	47.4	48.7	48.2	33.9
MiniClustering (ICLR2026)	59.9	58.2	60.3	**52.5**	35.8	35.8	41.2	23.6	48.0	42.1	44.2	33.0
DPLC (ours)	**61.4**	**67.3**	**61.9**	51.7	37.8	36.6	43.4	24.3	**54.0**	**46.2**	**47.4**	**40.1**

Table 2: Clustering performance in percentage (%) on three benchmark datasets (imbalance ratio = 10) across various methods.

5.2 Main Results

Table 1 and Table 2 report clustering performance on CIFAR-10, CIFAR-20, and STL-10 at imbalance ratios 5 and 10, respectively; Table 3 further extends the comparison to Tiny-ImageNet, ImageNet-100, and ImageNet-50. We compare against a comprehensive set of state-of-the-art long-tailed clustering methods: SCAN[Van Gansbeke et al., 2020], SDCLR[Jiang et al., 2021], CC[Li et al., 2021], IDFD[Tao et al., 2021], CoNR[Yu et al., 2023], ProPos[Huang et al., 2022], DMICC[Li et al., 2023], SeCu[Qian, 2023], LFSS[Li et al., 2025c], ConMix[Li and Jia, 2025], and MiniClustering[Li et al., 2025b]. The BYOL encoder followed by fixed K-means ($K = 200$) serves as the label-free baseline and is reported as "w/o DP" in Table 4; the gap between this baseline and the full model therefore isolates the contribution of the DP posterior. Most results are cited from ConMix; SCAN, SeCu, and LFSS are reproduced under identical long-tailed configurations. Performance is measured by clustering accuracy (ACC), class-averaged accuracy (CAA), normalized mutual information (NMI), and adjusted Rand index (ARI), with the best in each column highlighted in bold.

Results on CIFAR-10, CIFAR-20 and STL-10. Table 1 (imb_ratio = 5) and Table 2 (imb_ratio = 10) show that DPLC consistently achieves new state-of-the-art across almost all metrics and datasets. On CIFAR-10 (imb_ratio = 5), DPLC reaches 65.7% ACC, 68.6% CAA, and 65.5% NMI, clearly surpassing all prior works. Under the more severe imbalance (imb_ratio = 10), DPLC attains 61.4% ACC, 67.3% CAA, and 61.9% NMI; the CAA gain over the strongest existing method is 9.1 points, showing that the gain concentrates on the minority classes rather than inflating the majority ones. On STL-10 and CIFAR-20, DPLC also delivers highly competitive performance: on STL-10 it establishes new state-of-the-art results (54.0% ACC / 47.4% NMI), while on CIFAR-20 it remains competitive despite the dataset's high intra-class ambiguity (37.8% ACC / 36.6% CAA / 43.4% NMI at imb_ratio = 10); the reasons behind this variation are analyzed below. ARI and CAA values are competitive and, in most settings, either match or exceed the best previously reported figures. These gains are achieved without manually tuning the number of auxiliary clusters or head/tail thresholds—the built-in CRP-Gibbs sampler automatically adapts the sub-cluster granularity to the data geometry, making DPLC both highly effective and essentially tuning-free for long-tailed clustering.

Where the gains come from. The benefit of DPLC scales with the richness of the latent geometry, since the DP posterior can only reweight structure that the embedding already exposes: it is largest on the big, well-structured benchmarks and smallest on CIFAR-20, whose super-class construction raises intra-class ambiguity, yet DPLC remains competitive there. The correction also acts on the intended classes: per-class accuracies on CIFAR-10 show that tail classes are preserved rather than sacrificed, and the inferred weights decrease with sub-cluster size (see Appendix D).

Additional datasets. We also evaluate DPLC on larger-scale benchmarks (Tiny-ImageNet, ImageNet-100, ImageNet-50), where it consistently improves over other competitors. We use DINOv3 weights[Siméoni et al., 2025], a standard fallback for large-scale datasets where BYOL underperforms. Together, these results confirm that directly modeling the latent data distribution via a Dirichlet Process posterior and transforming it into rebalancing gradients provides a robust, tuning-free solution for long-tailed deep clustering.

Methods	**Tiny-ImageNet-LT**				**ImageNet-100-LT**				**ImageNet-50-LT**			
	ACC	CAA	NMI	ARI	ACC	CAA	NMI	ARI	ACC	CAA	NMI	ARI
LFSS (ICML2025)	22.9	18.4	28.7	14.0	54.3	52.4	76.2	43.1	29.6	26.2	52.3	18.4
ConMix (ICLR2025)	16.5	14.5	34.9	8.3	37.3	35.3	64.7	26.6	29.1	25.9	50.4	16.2
MiniClustering (ICLR2026)	25.5	19.4	40.6	14.9	52.6	45.8	74.4	41.9	33.9	26.7	53.7	22.6
DPLC (ours)	**30.9**	**24.7**	**47.8**	**18.2**	**59.4**	**56.3**	**79.9**	**46.6**	**36.5**	**30.1**	**57.0**	**26.3**

Table 3: Performance comparison on larger datasets with an imb_ratio = 10. Best results are in bold.

5.3 Ablation Studies

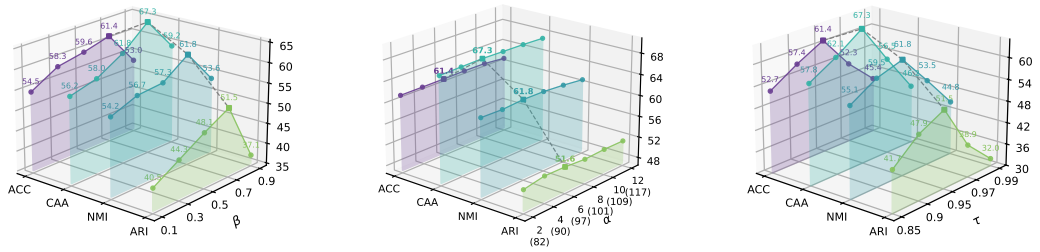
We ablate each design of DPLC on CIFAR-10 under the severe long-tailed setting (imb_ratio=10). Removing the Dirichlet Process prior—that is, the BYOL + K-means baseline with a fixed number of 200 clusters—drops ACC from 61.4% to 57.0%, confirming that adaptive, data-driven sub-cluster inference is essential[Hartigan and Wong, 1979]. Disabling dynamic propensity reweighting (uniform cluster weights) artificially inflates overall ACC to 64.4% but collapses CAA by 8 points — a clear symptom of head-class overfitting that validates the importance of the distribution-level rebalancing signal. Eliminating the stop-gradient operation destabilizes the score-function estimator, lowering ACC to 54.6%. Removing the similarity alignment loss $\mathcal{L}_s$ reduces ACC by 2.6% with a marked NMI decline, demonstrating that structural regularization is necessary to prevent point-wise reweighting from drifting cluster boundaries.

Hyper-parameter Sensitivity. We further examine the three key hyper-parameters: the CRP concentration α, the loss-balancing coefficient β, and the confidence threshold τ. **Concentration α.** Figure 3b varying α from 2 to 12 changes the inferred number of sub-clusters smoothly from 82 to 117, yet all metrics remain remarkably stable: ACC stays within [61.12, 61.52] and CAA within [67.02, 67.34], both peaking around $\alpha = 6$–8. The number of occupied sub-clusters remains stable throughout training (Appendix D). **Loss-balancing β.** Figure 3a shows that DPLC performs best at $\beta = 0.7$ (ACC 61.4 %, CAA 67.3 %); within $\beta \in [0.3, 0.7]$ accuracy stays above 58 %, indicating robust mixing of rebalancing and geometric alignment. **Confidence threshold τ.** Figure 3c shows that $\tau = 0.95$ gives the best effect. An overly permissive mask ($\tau = 0.85$) admits noisy pseudo-labels,

while an excessively strict one ($\tau \geq 0.97$) discards informative tail samples, causing ACC to drop sharply.

Variant	ACC	CAA	NMI	ARI
Full DPLC	61.4	67.3	61.9	51.7
w/o DP (K-means)	57.0	62.8	59.2	43.9
w/o grad-guidance	64.4	59.3	58.2	51.8
w/o stop-gradient targets	54.6	61.7	60.1	44.5
w/o L_s	58.8	64.2	60.9	46.7
SwAV backbone	50.6	45.4	40.7	34.7
DINOv3 backbone	64.5	55.6	66.4	58.8

Table 4: Ablation studies on CIFAR-10 (imb_ratio=10). "w/o DP" replaces CRP-DP with K-means, i.e., the BYOL + K-means baseline; "w/o grad-guidance" sets uniform weights $w_k = 1$ in Eq. (7), disabling the dynamic propensity reweighting module; "w/o stop-gradient targets" disables the stop-gradient operation on the Gibbs-sampled targets in Eq. (5); "w/o $\mathcal{L}_s$" removes the similarity alignment loss in Eq. (9).



(a) β sensitivity (b) α sensitivity (c) τ sensitivity

Figure 3: Hyper-parameter sensitivity analysis on CIFAR-10 (imb_ratio=10). Figure a Effect of the loss-balancing coefficient β. Figure b Effect of the CRP concentration α. Figure c Effect of the confidence threshold τ. Numbers in parentheses beside α denote the number of occupied sub-clusters (N_{occ}).

6 Conclusions

We presented DPLC, a nonparametric Bayesian framework that directly models latent sub-cluster structures via a Dirichlet Process prior to address long-tailed deep clustering bias. Extensive experiments across six benchmarks and varying imbalance ratios demonstrate consistent state-of-the-art performance, with notable gains in class-averaged accuracy under severe imbalance. The adaptive nature of the CRP prior makes DPLC both effective and essentially tuning-free, opening a new direction for fully unsupervised learning on real-world imbalanced data.

Limitations. A current limitation is the computational overhead of periodic full-batch CRP-Gibbs sampling, which may hinder its application to web-scale streams. Moreover, the framework inherits the quality assumptions of the upstream self-supervised representations; if pre-training produces weak features, the subsequent DP modeling cannot fully compensate. In future work, we plan to investigate online or amortized inference for the DP prior and to extend DPLC to tasks such as open-set recognition and anomaly detection, where long-tailed distributions naturally arise.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62606327, 62471266, 62006131, 62071260; the National Natural Science Foundation of Zhejiang Province under Grant LQ24F020001; the Ningbo Major Research and Development Plan Project (No.2023Z224); and the Intelligent Computing Center of Shenzhen University.

References

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9), 2005.

Zhenshan Bing, Yuan Meng, Yuqi Yun, Hang Su, Xiaojie Su, Kai Huang, and Alois Knoll. Diva: A dirichlet process mixtures based incremental deep clustering algorithm via variational auto-encoder. *arXiv preprint arXiv:2305.14067*, 2023.

David M Blei and Michael I Jordan. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9640–9649, 2021.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22, 1977.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9588–9597, 2021.

Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

Samuel J Gershman and David M Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *Ijcai*, volume 17, pages 1753–1759, 2017.

John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

Cuong Manh Hoang, Yeejin Lee, and Byeongkeun Kang. Unsupervised contrastive learning using out-of-distribution data for long-tailed dataset. *Neurocomputing*, 649:130779, 2025.

Peihao Huang, Yan Huang, Wei Wang, and Liang Wang. Deep embedding network for clustering. In *2014 22nd International conference on pattern recognition*, pages 1532–1537. IEEE, 2014.

Zhizhong Huang, Jie Chen, Junping Zhang, and Hongming Shan. Learning representation for clustering via prototype scattering and positive sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7509–7524, 2022.

Yuheng Jia, Jianhong Cheng, Hui Liu, and Junhui Hou. Towards calibrated deep clustering network. *arXiv preprint arXiv:2403.02998*, 2024.

Ziyu Jiang, Tianlong Chen, Bobak J Mortazavi, and Zhangyang Wang. Self-damaging contrastive learning. In *International conference on machine learning*, pages 4927–4939. PMLR, 2021.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009.

Yann Le, Xuan Yang, et al. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Hanyang Li, Yuheng Jia, Hui Liu, and Junhui Hou. You can trust your clustering model: A parameter-free self-boosting plug-in for deep clustering. *arXiv preprint arXiv:2511.21193*, 2025a.

Hongyu Li, Lefei Zhang, and Kehua Su. Dual mutual information constraints for discriminative clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 8571–8579, 2023.

Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8547–8555, 2021.

Yunfan Li, Mouxing Yang, Dezhong Peng, Taihao Li, Jiantao Huang, and Xi Peng. Twin contrastive learning for online clustering. *International Journal of Computer Vision*, 130(9):2205–2221, 2022.

Zhixin Li and Yuheng Jia. Conmix: Contrastive mixup at representation level for long-tailed deep clustering. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zhixin Li, Yuheng Jia, Guanliang Chen, Hui Liu, and Junhui Hou. Mini-cluster guided long-tailed deep clustering. In *The Fourteenth International Conference on Learning Representations*, 2025b.

Zhixin Li, Yuheng Jia, Junhui Hou, et al. Learning from sample stability for deep clustering. In *Forty-second International Conference on Machine Learning*, 2025c.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546, 2019.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Thomas Peter Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799. PMLR, 2014.

Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

Chuang Niu, Hongming Shan, and Ge Wang. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278, 2022.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

Qi Qian. Stable cluster discrimination for deep clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16645–16654, 2023.

Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, Philip S Yu, and Lifang He. Deep clustering: A comprehensive survey. *IEEE transactions on neural networks and learning systems*, 36(4):5858–5878, 2024.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. *Advances in neural information processing systems*, 28, 2015.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research*, 16(1):1731–1755, 2015.

Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020.

Yaling Tao, Kentaro Takagi, and Kouta Nakata. Clustering-friendly representation learning via instance discrimination and feature decorrelation. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=e12NDM7wkEY`.

Yee Teh, Michael Jordan, Matthew Beal, and David Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. *Advances in neural information processing systems*, 17, 2004.

Kai Tian, Shuigeng Zhou, and Jihong Guan. Deepcluster: A general clustering framework based on deep learning. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 809–825. Springer, 2017.

Tsung Wei Tsai, Chongxuan Li, and Jun Zhu. Mice: Mixture of contrastive experts for unsupervised image clustering. In *International conference on learning representations*, 2020.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5017–5026, 2019.

Zeya Wang, Yang Ni, Baoyu Jing, Deqing Wang, Hao Zhang, and Eric Xing. Dnb: A joint learning framework for deep bayesian nonparametric clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):7610–7620, 2021.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

Lingjie Yi, Jiachen Yao, Weimin Lyu, Haibin Ling, Raphael Douady, and Chao Chen. Geometry of long-tailed representation learning: Rebalancing features for skewed distributions. In *The Thirteenth International Conference on Learning Representations*, 2025.

Chunlin Yu, Ye Shi, and Jingya Wang. Contextually affinitive neighborhood refinery for deep clustering. *Advances in Neural Information Processing Systems*, 36:5778–5790, 2023.

Chuyu Zhang, Hui Ren, and Xuming He. P^2ot: Progressive partial optimal transport for deep imbalanced clustering. *arXiv preprint arXiv:2401.09266*, 2024.

Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10795–10816, 2023.

Zizhao Zhang and Tomas Pfister. Learning fast sample re-weighting without reward data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 725–734, 2021.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020.

Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Zhao Li, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, and Martin Ester. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *ACM Computing Surveys*, 57(3):1–38, 2024.

Zhihan Zhou, Jiangchao Yao, Yan-Feng Wang, Bo Han, and Ya Zhang. Contrastive learning with boosted memorization. In *International Conference on Machine Learning*, pages 27367–27377. PMLR, 2022.

Appendix

A Equivalence Between Weighted Cross-Entropy and the Score-Function Surrogate

This appendix proves that minimizing the weighted cross-entropy $\mathcal{L}_r$ (Eq. (8)) implements the score-function estimator for the rebalanced surrogate $\mathcal{J}(\omega) = \mathbb{E}_{c\sim P(c|\mathbf{z})}\big[w_c \log Q(c \mid \mathbf{z}; \omega)\big]$, which replaces the original ELBO reward during the M-step to correct the CRP prior's sampling bias. Let w_c be a non-negative weight associated with cluster c under the fixed teacher distribution $P(c \mid \mathbf{z})$. The score-function gradient is

$$\nabla_\omega \mathcal{J} = \mathbb{E}_{c\sim P}\left[w_c \nabla_\omega \log Q(c \mid \mathbf{z}; \omega)\right]. \tag{11}$$

Now consider the weighted cross-entropy loss defined in Eq. 8:

$$\mathcal{L}_r = -\sum_i \mathbb{E}_{c\sim P}[w_c \log Q(c \mid \mathbf{z}_i; \omega)]. \tag{12}$$

Computing its gradient with respect to $Q_{i,j}$ yields

$$\frac{\partial \mathcal{L}_r}{\partial Q_{i,j}} = -\frac{w_j}{Q_{i,j}}. \tag{13}$$

By the chain rule this gives $\nabla_\omega \mathcal{L}_r = -\nabla_\omega \mathcal{J}$: minimizing the weighted cross-entropy is exactly gradient ascent on the weighted log-likelihood $\mathcal{J}$. In other words, the weighted cross-entropy loss of Eq. 8 exactly realizes the score-function estimator for a reward w_c that we shape via inverse-propensity scores (Eq. 7). This establishes $\mathcal{L}_r$ as a principled, rebalanced surrogate for clustering, without requiring the original ELBO reward. The weights are chosen as $w_k \propto 1/(n_k + \alpha/K)$ to make gradient contributions approximately uniform across clusters, steering the head toward balance.

B Derivation and Intuition of the Score-Function Estimator

B.1 Formal Derivation.

Starting from the ELBO in Eq. 3, the gradient with respect to ω is:

$$\nabla_\omega \mathcal{L}_{\text{ELBO}} = -\nabla_\omega \mathbb{E}_{Q(c|z;\omega)}[\log Q(c|z;\omega) - \log P(z, c)]. \tag{14}$$

Applying the identity $\nabla_\omega Q(c) = Q(c)\nabla_\omega \log Q(c)$, the gradient of an expectation decomposes as $\nabla_\omega \mathbb{E}_Q[f] = \mathbb{E}_Q[\nabla_\omega f] + \mathbb{E}_Q[f\, \nabla_\omega \log Q]$. For the ELBO, $f(c) = \log P(\mathbf{z}, c) - \log Q(c \mid \mathbf{z}; \omega)$, and $\mathbb{E}_Q[\nabla_\omega f] = -\mathbb{E}_Q[\nabla_\omega \log Q] = -\nabla_\omega \sum_c Q(c) = 0$, so the first term vanishes and only the score-function form remains:

$$\nabla_\omega \mathbb{E}_{Q(c|z;\omega)}[f(c)] = \mathbb{E}_{Q(c|z;\omega)}[f(c)\nabla_\omega \log Q(c|z;\omega)]. \tag{15}$$

To stabilize the estimator, we switch the expectation to the fixed CRP posterior P—that is, we perform variational distillation—and introduce a stop-gradient baseline:

$$\nabla_\omega \mathcal{L}_{\text{ELBO}} \approx \mathbb{E}_{c\sim P}\left[(\log P(\mathbf{z}, c) - \text{sg}[\log Q(c \mid \mathbf{z}; \omega)])\, \nabla_\omega \log Q(c \mid \mathbf{z}; \omega)\right]. \tag{B.1}$$

Sampling from the stationary teacher P instead of the evolving Q corresponds to variational distillation; the stop-gradient baseline reduces variance without changing the expected update direction. Equivalently, Eq. (B.1) is the gradient of the surrogate loss in Eq. (5), where the baseline $\text{sg}[\log Q]$ is detached inside the loss rather than inside a gradient expression.

Intuition as Reinforcement Learning. This estimator can be understood through a reinforcement learning lens: the head Q acts as a *policy* that proposes cluster assignments, and the log-likelihood $\log P(z, \hat{c})$ serves as a *reward* measuring how well the assignment explains the data. The baseline $\text{sg}[\log Q]$ stabilizes training by centering the reward, increasing the probability of assignments that perform better than expected while decreasing those that perform worse. This yields the final form in Eq. 5 of the main paper.

C Detailed Experimental Settings

Datasets and Long-tailed Construction We evaluate on six benchmarks: CIFAR-10, CIFAR-20, STL-10, Tiny-ImageNet, ImageNet-100, and ImageNet-50. Long-tailed training sets are built by sub-sampling the original balanced data with an exponentially decaying class cardinality, controlled by the imbalance ratio ρ (defined in Section 5.1). We experiment with $\rho \in \{5, 10\}$ Zhou et al. [2020], Cao et al. [2019]. For STL-10, we combine the official training and test splits for unsupervised pretraining, following prior work. Following prior work on long-tailed clustering[Li and Jia, 2025, Li et al., 2025c,b], the test splits are truncated to reflect the same long-tailed distribution as the training set. This ensures that evaluation faithfully measures clustering quality under class imbalance and matches the exact protocol of our baselines.

Encoder Pre-training and Backbone Variants. The encoder is pre-trained with BYOL for 1000 epochs using SGD with a cosine learning rate schedule[Grill et al., 2020]. After pretraining, the backbone is frozen, and all training features are L2-normalized and cached. For larger-scale datasets (Tiny-ImageNet, ImageNet-100, ImageNet-50), we replace the BYOL backbone with DINOv3 pretrained weights, which provide stronger initial representations in these regimes[Le et al., 2015, Liu et al., 2019]. We additionally validate SwAV as an alternative backbone in the ablation studies[Caron et al., 2020, Siméoni et al., 2025].

Clustering Head and DP Inference Schedule. Two augmented views are encoded per batch; the head produces logits ℓ^1 and, with stop-gradient, ℓ^2. A confidence mask with threshold $\tau = 0.95$ filters low-confidence predictions. Every 15 epochs, we run CRP-Gibbs sampling on the PCA-projected (64-dim) and L2-normalized cached embeddings[Neal, 2000]. PCA is re-fitted on the current cached training embeddings at each DP inference cycle, which allows the low-dimensional projection to adapt to the evolving feature distribution[Pearson, 1901, Abdi and Williams, 2010]. The resulting soft posterior supplies dynamic inverse-propensity weights w_k (Eq. 7) and the co-occurrence similarity matrix for the geometric preservation term. In practice, we implement the similarity alignment as a mean squared error loss between the row-normalized pairwise similarity matrices of the head predictions and the DP soft assignments, rather than the KL divergence described in Eq. (9). Both formulations enforce the same structural consistency; MSE is chosen for its computational simplicity and stable gradients.

Loss Balancing and Hyper-parameters. The joint objective is $\mathcal{L} = \mathcal{L}_r + \beta\mathcal{L}_s$, where $\mathcal{L}_r$ is the masked weighted cross-entropy (Eq. 8) and $\mathcal{L}_s$ is the pairwise similarity alignment (Eq. 9). The balancing coefficient is fixed to $\beta = 0.7$ across all datasets and imbalance ratios. All hyper-parameters remain constant across experiments, requiring no per-dataset tuning. The clustering head is initialized once with the normalized K-means centers of the cached features—K-means is used only for this initialization and plays no role during training—and is trained for 300 epochs using the Adam optimizer with a learning rate of 0.05, as outlined in Sec. 5.1. We use a batch size of 256 for all datasets, with two augmented views per sample encoded per batch. The concentration parameter is set to $\alpha = 6.0$, with 60 sweeps and 3 independent restarts.

Implementation of $\mathcal{L}_s$. In Eq. 9 we define $\mathcal{L}_s$ as a KL divergence between the pairwise similarity matrices. In practice, we implement it as a mean squared error (MSE) between the row-normalized similarity matrices. This choice is motivated by the fact that near convergence the two distributions are close, making MSE a first-order approximation of KL; moreover, MSE yields smoother gradients and avoids numerical issues of log. Empirical comparisons (not shown) confirm that MSE and KL produce nearly identical clustering performance. Therefore, the MSE implementation faithfully preserves the structural alignment objective described in Sec. 4.4.

D Mechanism Analysis and Additional Visualizations

This section collects the mechanism-level evidence referenced in Sec. 5.3 and Sec. 5.1; unless stated otherwise, all results are on CIFAR-10 with imb_ratio $= 10$. The DP posterior recovers sub-clusters that align with class boundaries. Figure 4 shows the same subsampled training points embedded with t-SNE and colored by (a) ground-truth class, (b) DP sub-cluster assignment, and (c) the final DPLC prediction: the DP discovers fine-grained substructure within the long-tailed geometry, and DPLC reassembles it into balanced clusters without losing semantic coherence.

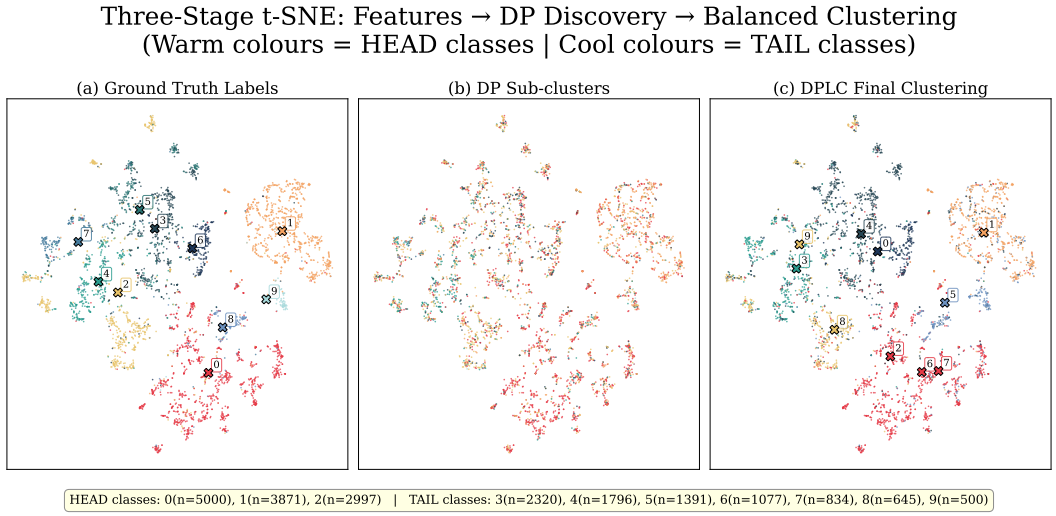


Figure 4: Three-stage t-SNE on CIFAR-10 (imb_ratio $= 10$), colored by (a) ground-truth labels, (b) DP sub-cluster assignments, and (c) final DPLC predictions (warm colours: head classes, cold colours: tail classes).

Rebalancing acts on the minority classes. Figure 5 reports per-class accuracy with classes sorted by training size: tail-class accuracy is preserved rather than sacrificed, confirming that the correction does not inflate majority scores.

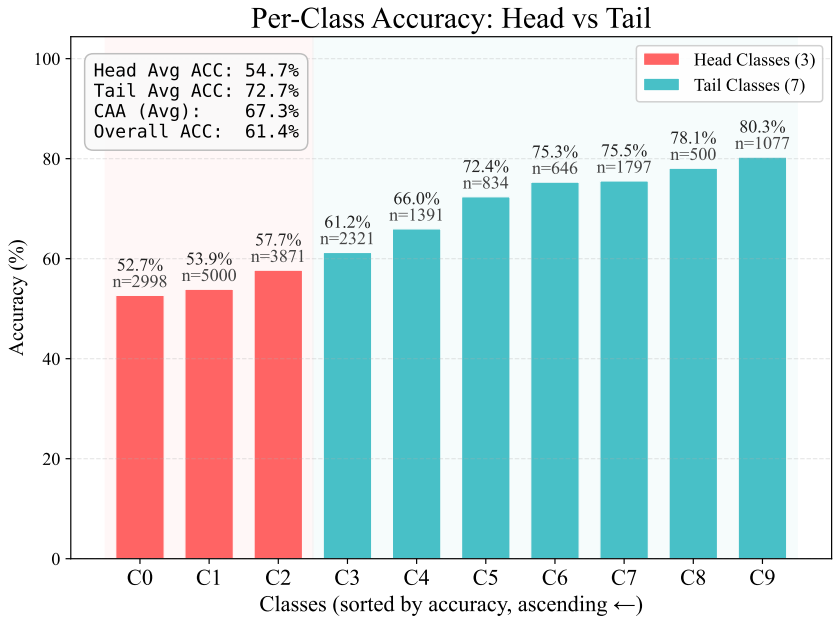


Figure 5: Per-class accuracy on CIFAR-10 (imb_ratio $= 10$), sorted by ascending accuracy. The top-3 most frequent classes are denoted as head classes (red), and the bottom-3 as tail classes (cyan). The DP granularity is stable across training. Figure 6 tracks the number of occupied sub-clusters over all 300 epochs (Gibbs refresh every 15 epochs) and Table 5 summarizes it: N_{occ} stays in a narrow band, so the periodic refresh does not destabilize the posterior signal on which the head trains.

Min	Max	Mean	Std	Range
90	103	94.50	2.85	90–103

Table 5: Statistics of N_{occ} across 300 epochs (CRP-Gibbs refreshed every 15 epochs).

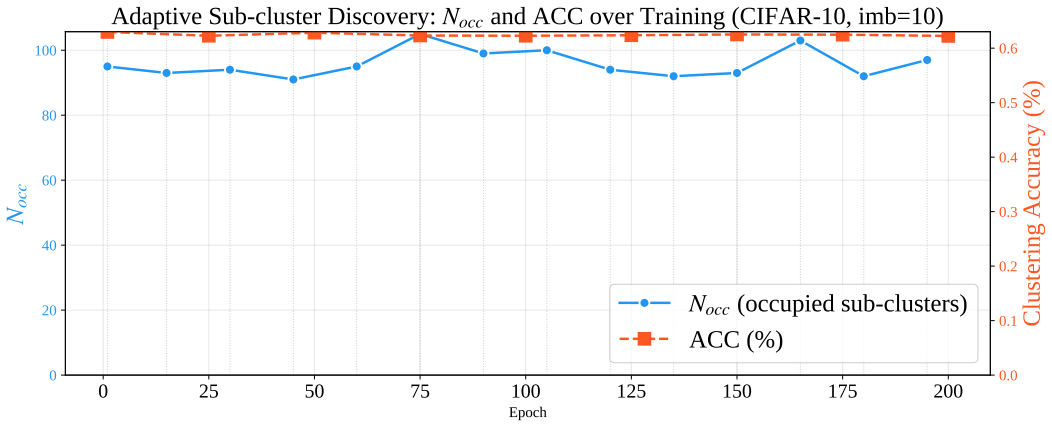


Figure 6: Number of occupied sub-clusters N_{occ} over the full training trajectory on CIFAR-10 (imb_ratio $= 10$).

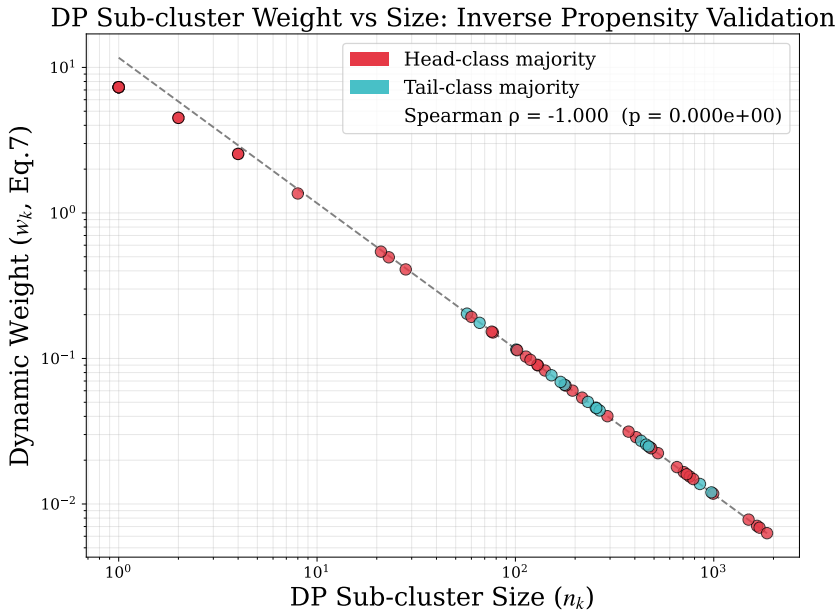


Figure 7: Weight–size correlation of the DP sub-clusters (red: head, blue: tail). Each point is one sub-cluster; the Spearman correlation between the inverse-propensity weight and the sub-cluster size is -1.000.

The IPS weights track the imbalance without labels. Figure 7 plots each DP sub-cluster's weight against its occupancy, colored by the majority ground-truth class; the Spearman correlation between weight and size is -1.000, i.e., tail sub-clusters systematically receive the largest weights, exactly as Eq. (7) prescribes.

Finally, the compute budget is modest: on CIFAR-10 the full pipeline takes about 1.2 hours on one NVIDIA H100 GPU, split into head training (0.68 h), CRP-Gibbs sampling (0.41 h over 21 calls), and feature caching (0.12 h). At the batch level, the encoder forward (4.44 ms) dominates, while the reweighted cross-entropy $\mathcal{L}_r$ (0.48 ms), the similarity alignment $\mathcal{L}_s$ (0.28 ms), and the backward pass (2.09 ms) add little overhead. The peak GPU memory footprint is 2270 MB, attained during the periodic Gibbs sweeps on the PCA-reduced cached embeddings, which keeps DPLC within the budget of a single consumer GPU.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the motivation, proposed method, and main findings. They accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations through experiments and conclusions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Appendix A provides a complete step-by-step derivation showing that the gradient of the weighted cross-entropy loss exactly reproduces the score-function policy gradient. All necessary assumptions are stated within that derivation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All implementation details, training settings, and evaluation protocols are provided in the main paper and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have released the official implementation of DPLC at (`https://github.com/V2tPkDuck/DPLC`). The repository contains the complete training and evaluation pipeline, including scripts for long-tailed dataset generation, the Dirichlet Process Gibbs sampler, and the reweighted clustering head. Detailed instructions for reproducing all reported results are provided in the README.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The paper provides detailed descriptions of experimental settings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Although full multi-run error bars are computationally prohibitive, we report extensive hyper-parameter sensitivity analyses (e.g., ACC varies only $\pm 0.2\%$ across $\alpha \in [2, 12]$, and metrics remain stable under $\beta \in [0.3, 0.7]$ and $\tau \in [0.90, 0.95]$), which jointly demonstrate that our results are statistically robust and not due to chance parameter selections.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides details on the compute resources used for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The work does not involve data or models with a high risk for misuse, so no specific safeguards are necessary.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets (e.g., code, datasets) used in the paper are properly credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The current submission does not include any released code, datasets, or pre-trained models. Code will soon be made publicly available after acceptance; the BYOL encoder is a standard self-supervised model trained by ourselves and its training scripts will be released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The paper uses LLMs only for writing, editing, and formatting purposes and does not affect the core methodology or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.