-Supplementary Material-

A. Complementary Results

A.1. Imbalance analysis of inferred K on non-contrastive SSL (BYOL/SimSiam)

In the following, we present the effect of class imbalance on the inferred K using non-contrastive SSL embeddings.

Method	**Inferred K**		
	CIFAR-10$^{\text{imb}}$	CIFAR-20$^{\text{imb}}$	ImageNet-10$^{\text{imb}}$
DPmeans [20]	16.20 ± 1.17	21.80 ± 1.47	45.40 ± 1.36
PDC-DP-Means [9]	15.33 ± 0.47	23.00 ± 1.41	62.67 ± 0.47
moVB [15]	8.60 ± 1.14	18.40 ± 4.04	9.80 ± 2.28
DeepDPM [27]	4.40 ± 0.89	8.00 ± 4.00	9.40 ± 1.14
DBSCAN [10]	9.00 ± 0.00	24.00 ± 0.00	21.00 ± 0.00
DP-TGMM [31]	1.00 ± 0.00	11.25 ± 0.50	1.75 ± 0.50
DP-vMF-means [30]	11.40 ± 0.89	**20.80 ± 0.84**	11.20 ± 0.45
DeepDP-TGMM (ours)	**10.20 ± 0.45**	12.40 ± 0.55	**10.00 ± 0.71**

Table 7. Comparisons of Euclidean and spherical clustering methods on non-contrastive SSL embeddings (BYOL/SimSiam). Inferred number of clusters (K) on imbalanced CIFAR-10, CIFAR-20, and ImageNet-10. Best results are in bold and second best are underlined. For K, best and second best are defined by closeness to the ground-truth K.

A.2. Imbalance analysis of inferred K on contrastive SSL (MoCo v1/v3)

In the following, we present the impact of class imbalance on the inferred K using contrastive SSL embeddings.

Method	**Inferred K**		
	CIFAR-10$^{\text{imb}}$	CIFAR-20$^{\text{imb}}$	ImageNet-10$^{\text{imb}}$
DPmeans [20]	9.00 ± 0.00	12.60 ± 0.49	**10.00 ± 0.00**
PDC-DP-Means [9]	15.33 ± 0.47	**23.00 ± 1.41**	14.33 ± 0.47
moVB [15]	3.00 ± 0.00	2.60 ± 0.55	9.60 ± 0.89
DeepDPM [27]	8.75 ± 0.50	8.00 ± 4.00	8.00 ± 0.00
DBSCAN [10]	22.00 ± 0.00	24.00 ± 0.00	8.00 ± 0.00
DP-TGMM [31]	**9.25 ± 0.96**	11.25 ± 0.50	8.75 ± 1.26
DP-vMF-means [30]	5.60 ± 0.55	11.80 ± 0.84	12.60 ± 0.55
DeepDP-TGMM (ours)	11.00 ± 0.00	12.40 ± 0.55	8.80 ± 0.45

Table 8. Comparisons of Euclidean and spherical clustering methods on contrastive SSL embeddings (MoCo v1/v3). Inferred number of clusters (K) on imbalanced CIFAR-10, CIFAR-20, and ImageNet-10. Best results are in bold and second best are underlined. For K, best and second best are defined by closeness to the ground-truth value.

A.3. Ablation of model components (BYOL–CIFAR-10)

In addition to main paper Table 6, we report complementary ablations on BYOL–CIFAR-10, including NMI, ARI, and inferred K across different initializations.

B. Dataset Statistics and Evaluation Metrics

B.1. Evaluation Metrics

We assess clustering quality using three complementary measures that are widely adopted in recent benchmarks [1, 27, 33]. The *Adjusted Rand Index (ARI)* measures the agreement between two partitions while correcting for similarities that may arise by chance [14]. In contrast, the *Normalized Mutual Information (NMI)* offers an information-theoretic perspective, quantifying the dependence between cluster assignments and ground-truth labels in a scale-invariant manner [32].

In addition to these two partition-based metrics, we report *Clustering Accuracy (ACC)*, which measures the proportion of correctly assigned samples once the predicted clusters have been aligned with the ground-truth classes. Because cluster

	NMI			ARI			Inferred K		
	$K_{\text{init}} = 3$	$K_{\text{init}} = 10$	$K_{\text{init}} = 30$	$K_{\text{init}} = 3$	$K_{\text{init}} = 10$	$K_{\text{init}} = 30$	$K_{\text{init}} = 3$	$K_{\text{init}} = 10$	$K_{\text{init}} = 30$
No splits/merges	0.50±0.02	**0.82±0.01**	0.76 ± 0.01	0.27±0.03	0.78 ± 0.02	0.62±0.02	3.00±0.00	**10.00±0.00**	18.80±0.84
No splits	0.49±0.01	**0.82±0.01**	**0.80±0.01**	0.26±0.02	**0.79±0.02**	0.71 ± 0.01	3.00±0.00	**10.00±0.00**	12.60±0.55
No merges	**0.81±0.00**	0.80±0.02	0.76 ± 0.01	**0.76±0.01**	0.73±0.04	0.64±0.02	11.80±0.84	12.80±0.84	19.40±1.82
No priors in M-step	0.65±0.00	0.58±0.00	0.55±0.00	0.30±0.01	0.14±0.00	0.09±0.00	46.20±0.84	126.20±1.30	205.50±4.12
Spherical 2-means (instead of fsub)	0.78±0.01	0.81 ± 0.01	**0.80±0.01**	0.69±0.02	0.77±0.03	0.71 ± 0.01	8.50 ± 0.58	**10.00±0.00**	**11.75±0.96**
DeepDP - TGMM (full method)	0.80 ± 0.01	0.80±0.01	**0.80±0.01**	0.75 ± 0.01	0.73±0.03	**0.73±0.01**	**11.40±0.55**	12.50 ± 0.58	12.50 ± 0.58

Table 9. NMI, ARI, and inferred K under different initial cluster counts K_{init} on BYOL–CIFAR-10. Best results are in bold and second best are underlined. For K, best and second best are defined by closeness to the ground-truth value.

indices are arbitrary, computing ACC requires solving an assignment problem. Formally,

$$\text{ACC} = \max_{m} \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(y_i = m(z_i)) \,, \tag{11}$$

where y_i denotes the true label of sample i and z_i its cluster index. The mapping m represents the permutation of cluster labels that maximizes agreement, which we compute using the *Hungarian algorithm*. This procedure ensures that ACC remains a meaningful measure even in nonparametric scenarios where the number of inferred clusters may not match the number of ground-truth categories.

B.2. Dataset Statistics

We present in Tab. 10 the statistics of the benchmark datasets used in this paper. For ImageNet-100, we employ a randomly selected subset of 100 classes from ImageNet-1K [28] . To ensure reproducibility, the exact synset list is provided in Tab. 11.

Dataset	Train samples	Val samples	Data Dimension	GT
CIFAR-10 [18]	50,000	10,000	$32 \times 32 \times 3$	10
CIFAR-20 [19]	50,000	10,000	$32 \times 32 \times 3$	20
ImageNet-10 [28]	13,000	–	$224 \times 224 \times 3$	10
ImageNet-100 [28]	130,000	5,000	$224 \times 224 \times 3$	100

Table 10. Overview of tested datasets.

Table 11. WordNet IDs of the ImageNet-100 classes used in our experiments.

ImageNet-100 Class Synsets									
n01440764	n01443537	n01484850	n01491361	n01494475	n01496331	n01498041	n01514668	n01514859	n01531178
n01537544	n01560419	n01582220	n01592084	n01601694	n01608432	n01614925	n01622779	n01630670	n01632458
n01632777	n01644900	n01664065	n01665541	n01667114	n01667778	n01675722	n01677366	n01685808	n01687978
n01693334	n01695060	n01698640	n01728572	n01729322	n01729977	n01734418	n01735189	n01739381	n01740131
n01742172	n01749939	n01751748	n01753488	n01755581	n01756291	n01770081	n01770393	n01773157	n01773549
n01773797	n01774384	n01774750	n01775062	n01776313	n01795545	n01796340	n01798484	n01806143	n01818515
n01819313	n01820546	n01824575	n01828909	n01829413	n01833805	n01843065	n01843383	n01847000	n01855032
n01855672	n01860187	n01871265	n01872401	n01873310	n01877812	n01882714	n01883070	n01887896	n01889328
n01889520	n01891239	n01893091	n01895032	n01896444	n01910747	n01914609	n01915811	n01917289	n01922303
n01924916	n01930112	n01943899	n01944390	n01945685	n01950731	n01955084	n01968897	n01978287	n01978455

B.3. Imbalanced Dataset Creation

Following prior work [27], we construct imbalanced variants by randomly selecting the classes to be under-sampled and applying fixed per-class retention ratios. In CIFAR-10, classes 1, 3, 4, and 8 were retained at 21%, 47%, 44%, and 55%, respectively. In CIFAR-20, classes 1, 3, 6, 8, 9, 12, 15, 16, and 18 were retained at 32%, 28%, 31%, 13%, 26%, 28%, 27%, 17%, and 11%. Similarly, in ImageNet-10, classes 1, 5, 8, and 9 were retained at 45%, 27%, 38%, and 53%.
For ImageNet-100, following [27], we sample class weights from a symmetric Dirichlet distribution over the 100-dimensional

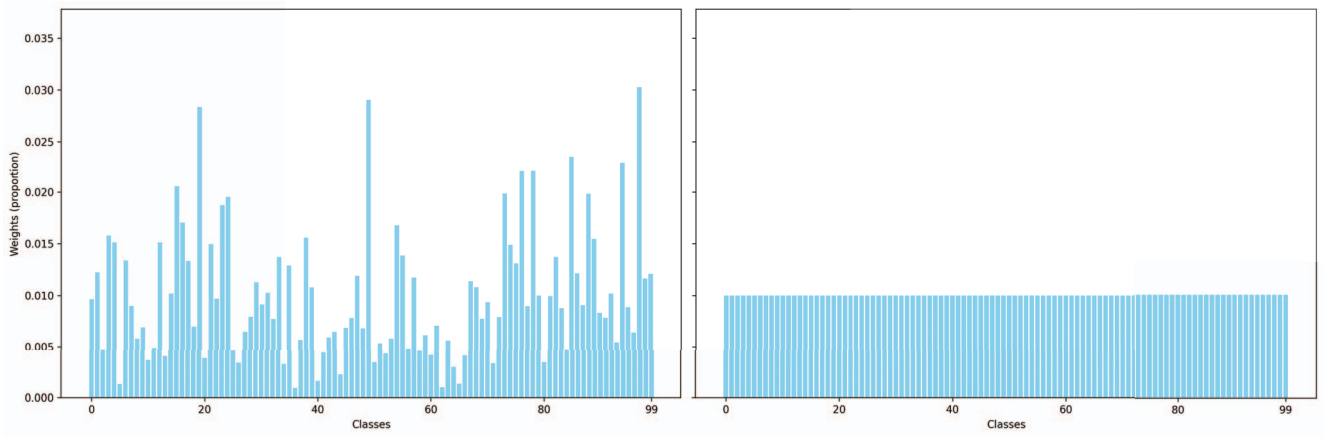

Figure 4. Class distributions for ImageNet-100. The right panel shows the balanced dataset, while the left panel illustrates an imbalanced version obtained by sampling class weights from a symmetric Dirichlet distribution over the 100-dimensional probability simplex.

probability simplex to obtain a random non-uniform histogram. As illustrated in Fig. 4, the right panel shows the balanced ImageNet-100 distribution, while the left panel displays the Dirichlet-sampled distribution adopted in our imbalanced setting.

C. Supplementary Experimental Setup

Here we provide supplementary implementation details for our proposed DeepDP–TGMM model, together with the SSL training configurations, clustering baselines, and hyperparameter settings that complement the description in the main paper.

C.1. SSL Configurations

For BYOL [11], SimSiam [5], and MoCo-v1 [12], we used the implementation provided in ProPos [13], while MoCo-v3 was employed following the official release described in [7]. All SSL models were trained for 1000 epochs. BYOL and SimSiam embeddings were trained on $4\times$ NVIDIA RTX A5000 GPUs, while MoCo-v1 and MoCo-v3 were trained on $8\times$ A5000 GPUs. We followed the recommended image transformation strategies described in the respective papers for all datasets. Apart from the details specified below, all hyperparameters were set as recommended in the original publications.

- **BYOL** [11]: on CIFAR-10 we used a ResNet-34 backbone with batch size 256 and embedding dimension 128, while on ImageNet-100 the setup employed a ResNet-50 with batch size 64 and dimension 256.
- **SimSiam** [5]: the CIFAR-20 configuration relied on ResNet-34 with batch size 256 and dimension 128, whereas ImageNet-10 was trained with the same backbone but batch size 64 and dimension 128.
- **MoCo-v1** [12]: CIFAR-10 experiments used ResNet-34 with batch size 512 and dimension 64, and CIFAR-20 used the same backbone with batch size 512 and dimension 128.
- **MoCo-v3** [7]: ImageNet-10 experiments were run with ResNet-34, batch size 512, and dimension 128.

C.2. Clustering Methods Implementation

We benchmarked our approach against a diverse set of clustering baselines, spanning both Euclidean and hyperspherical methods, as detailed below.

- **PDC-Means** [9]: adopted from the reference implementation provided by the authors.
- **moVB** [15]: integrated from the `bnpy` library without modification.
- **DeepDPM** [27]: employed using the official PyTorch release.
- **DBSCAN** [10]: used from the `scikit-learn` implementation with cosine distance as metric.
- **DP-TGMM** [31]: evaluated using the original code release.
- **DP-vMF-Means** [30]: adapted from the official release, running on hyperspherical embeddings.
- **DP-Means** [20]: reimplemented in PyTorch for this work, strictly following the original formulation and update rules described in the paper.

C.3. Data splits and K initialization.

Unless stated otherwise, we train all methods on the official *train* split and report all metrics on the *validation* split. For ImageNet-10 (subset), which has no public test labels, evaluation is performed on its training set. For all methods that require an initial number of clusters, we use $K_{\text{init}} = 3$ for all datasets except ImageNet-100, where we set $K_{\text{init}} = 10$.

C.4. Clustering Hyperparameters

DeepDP–TGMM is trained under the following settings:

- **Optimization:** Adam optimizer with learning rate 2×10^{-3}.
- **Training schedule:**
  - Batch size of 256 (128 for MoCo v3 on ImageNet-10(/IMB)).
  - 200 epochs in general, extended to 400 epochs for SimSiam on ImageNet-10(/IMB).
- **Prior:** isotropic inverse-Wishart distribution with dataset-dependent scaling Δ:
  - BYOL/SimSiam on CIFAR-10/20: $\Delta = 0.01$.
  - SimSiam on ImageNet-10: $\Delta = 0.005$, and on ImageNet-10-IMB: $\Delta = 0.001$.
  - MoCo v1 on CIFAR-10: $\Delta = 15$, and on CIFAR-10-IMB: $\Delta = 10$.
  - MoCo v1 on CIFAR-20(/IMB): $\Delta = 0.005$.
  - MoCo v3 on ImageNet-10(/IMB): $\Delta = 0.3$.
- **Degrees of freedom of NIW prior:**
  - $\nu = 129$ for CIFAR-10 and ImageNet-10(/IMB).
  - $\nu = 140$ for CIFAR-20(/IMB).
  - $\nu = 65$ for MoCo v1 on CIFAR-10(/IMB).
- **Refinement:** performed every 20 epochs to adapt the partition structure.

D. MAP Updates for Cluster Parameters (Supplemental Derivations)

We recall here, for completeness and to make the derivations self-contained, the Bayesian assumptions introduced in the main paper: data $\{x_i\}_{i=1}^N \subset \mathcal{S}^{D-1}$ carry soft responsibilities r_{ik} with $\sum_{k=1}^K r_{ik} = 1$; mixing proportions follow a stick-breaking prior $\pi \sim \mathrm{GEM}(\alpha)$ which, under a finite truncation to K components, reduces to a symmetric Dirichlet prior $\pi \sim \mathrm{Dir}(\alpha/K, \ldots, \alpha/K)$; cluster centers are distributed $\mu_k \sim \mathrm{Unif}(\mathcal{S}^{D-1})$; and covariances follow an inverse–Wishart prior $\Sigma_k \sim \mathrm{IW}(\Delta, \nu)$, with Σ_k defined on the $(D-1)$-dimensional tangent space at μ_k.

Prior on the center. Because we will reuse it both below and later in the split–merge acceptance ratios, we make explicit the normalized uniform density on the sphere:

$$p(\mu_k) = \frac{1}{S_{D-1}}, \qquad S_{D-1} = \frac{2\pi^{D/2}}{\Gamma(D/2)}, \quad \mu_k \in \mathcal{S}^{D-1}. \tag{12}$$

Here D denotes the ambient Euclidean dimension (so the manifold is $\mathcal{S}^{D-1} \subset \mathbb{R}^D$), and $\Gamma(\cdot)$ is the Euler gamma function. This factor is constant in μ_k and therefore cancels out in MAP optimization of the center, yet it contributes nontrivially in Metropolis–Hastings ratios whenever the number of clusters changes (see Sec. 3.3).

Cluster mean (MAP derivation). Unlike the covariance and the mixture weights, the MAP estimate of the *center* $\mu_k \in \mathcal{S}^{D-1}$ does not admit a closed form: even with a uniform prior, the likelihood depends on μ_k through the nonlinear logarithm map $\mathrm{Log}_{\mu_k}(\cdot)$, and the covariance Σ_k itself is defined in the tangent space anchored at μ_k, so conjugacy does not apply and

the posterior has no analytic mode. Starting from Bayes' rule and discarding terms independent of μ_k, we obtain

$$\begin{aligned} p(\mu_k \mid \Sigma_k, \{x_i\}, \{r_{ik}\}) &= \frac{p(\mu_k)\, p(\{x_i\} \mid \mu_k, \Sigma_k, \{r_{ik}\})}{p(\{x_i\} \mid \Sigma_k, \{r_{ik}\})} \\ &= \frac{1}{Z}\, p(\mu_k) \prod_{i=1}^{N} \mathcal{N}\big(\operatorname{Log}_{\mu_k}(x_i);\, 0, \Sigma_k\big)^{r_{ik}} \\ &\propto p(\mu_k)\, \exp\left(-\tfrac{1}{2} \sum_{i=1}^{N} r_{ik}\, \operatorname{Log}_{\mu_k}(x_i)^\top \Sigma_k^{-1} \operatorname{Log}_{\mu_k}(x_i)\right) \\ &\propto p(\mu_k)\, \exp\left(-\tfrac{1}{2} \sum_{i=1}^{N} r_{ik}\, \left\|\Sigma_k^{-1/2} \operatorname{Log}_{\mu_k}(x_i)\right\|^2\right), \end{aligned} \tag{13}$$

where $Z = p(\{x_i\} \mid \Sigma_k, \{r_{ik}\})$ does not depend on μ_k. Using (12), maximizing the posterior is equivalent to minimizing the Mahalanobis-weighted Karcher objective

$$\hat{\mu}_k = \arg\min_{\mu \in \mathcal{S}^{D-1}} \sum_{i=1}^{N} r_{ik} \left\|\Sigma_k^{-1/2} \operatorname{Log}_{\mu}(x_i)\right\|^2, \tag{14}$$

which reduces to the classical Karcher mean in the isotropic case $\Sigma_k = \sigma^2 I$.

Covariance. Writing $v_{ik} = \operatorname{Log}_{\mu_k}(x_i)$, we form the responsibility-weighted scatter

$$S_k = \sum_{i=1}^{N} r_{ik}\, v_{ik} v_{ik}^\top \in \mathbb{R}^{(D-1)\times(D-1)}. \tag{15}$$

The conjugate posterior is

$$\Sigma_k \mid \{x_i, r_{ik}\} \sim \mathrm{IW}(\Delta + S_k, \nu + N_k), \tag{16}$$

with effective count $N_k = \sum_{i=1}^{N} r_{ik}$. Following standard properties of the Inverse–Wishart distribution [**?**], the MAP estimate (mode) under our parameterization is

$$\hat{\Sigma}_k = \frac{\Delta + S_k}{\nu + N_k + (D-1) + 1}. \tag{17}$$

Mixing weights. Under $\pi \sim \mathrm{GEM}(\alpha)$ truncated to K, the posterior over the truncated vector is Dirichlet with parameters $\alpha/K + N_k$; as in the main paper, we use the posterior mean [**?**],

$$\hat{\pi}_k = \frac{\alpha/K + N_k}{\alpha + N}, \tag{18}$$

rather than the mode, in order to avoid degeneracy.

Hard assignments as a special case. We note that by setting $r_{ik} = \mathbf{1}[z_i = k]$, all expressions above reduce to their hard-assignment forms; in particular, (14), (15), (16), and (17) specialize by replacing weighted sums with sums over indices i such that $z_i = k$, and (18) becomes the usual count-based average.

E. Mahalanobis-Weighted Karcher Mean Algorithm

As mentioned in the main text, the classical Karcher mean [17, 24] extends the Euclidean mean to Riemannian manifolds by minimizing the sum of squared geodesic distances. On the hypersphere $\mathbb{S}^{D-1}$, it can be computed through an iterative procedure alternating between logarithmic and exponential maps. The Mahalanobis-weighted variant proposed here generalizes this approach by replacing the standard Euclidean metric in the tangent space with a Mahalanobis norm defined by a positive-definite matrix Σ, enabling the algorithm to account for directional variability in the data.

The fixed-point formulation introduced by Pennec [24] is adopted, where each update step moves in the average log-map direction in the tangent space. This strategy is also presented in DP-TGMM [31] in the context of spherical optimization.

Incorporating the fixed Mahalanobis metric throughout the optimization yields an intrinsic mean that minimizes a weighted sum of squared Mahalanobis distances in the tangent space. The resulting estimate corresponds to the Maximum A Posteriori (MAP) solution under a tangent-space Gaussian prior with covariance Σ. Setting $\Sigma = I$ recovers the standard Karcher mean.

Algorithm (fixed-point iteration)

We solve the following minimization problem:

$$\min_{p \in \mathbb{S}^{D-1}} \sum_{i=1}^{N} w_i \, \mathrm{Log}_p(x_i)^\top \Sigma^{-1} \, \mathrm{Log}_p(x_i),$$

where $x_i \in \mathbb{S}^{D-1}$, $w_i \geq 0$, $\sum_i w_i = 1$, and Σ is a symmetric positive-definite matrix.

1. **Weight normalization**
$$w_i = \begin{cases} \tilde{w}_i / \sum_j \tilde{w}_j, & \text{if provided,} \\ 1/N, & \text{otherwise.} \end{cases}$$
2. **Covariance transform setup**
Compute the Cholesky decomposition $\Sigma = LL^\top$ with lower-triangular L [**?**]. Define $A = L^{-\top} = \Sigma^{-1/2}$ so that $\|Av\|^2 = v^\top \Sigma^{-1} v$. In the iteration, the averaged, whitened vector is mapped back to the original tangent space by $g = A^\top \bar{v}'$, ensuring that $g = \Sigma^{-1} \sum_i w_i v_i$.
3. **Initialization**
$$p^{(0)} = \frac{\sum_i w_i x_i}{\| \sum_i w_i x_i \|}.$$
4. **Iterate** until convergence ($\|g\| < \text{tol}$) or maximum iterations:
   (a) Log map to tangent space: $v_i = \mathrm{Log}_{p^{(t)}}(x_i)$.
   (b) Whiten: $v_i' = A \, v_i$ so that $\|v_i'\|^2 = v_i^\top \Sigma^{-1} v_i$.
   (c) Weighted average in whitened space: $\bar{v}' = \sum_i w_i v_i'$.
   (d) **Mahalanobis-weighted direction (map back with $A^\top$):** $g = A^\top \bar{v}'$ (note $g = \Sigma^{-1} \sum_i w_i v_i$).
   (e) Update on the sphere: $p^{(t+1)} = \mathrm{Exp}_{p^{(t)}}(\eta \, g)$ with default step size $\eta = 1$.
5. **Return final** $p = p^{(t+1)}$.

F. Randomized Merge Ratio

As DP–TGMM explains [31], following the construction of Chang and Fisher [3], deterministic merges are almost never accepted in practice. To maintain reversibility, we therefore adopt the **randomized merge** strategy.

Proposal. Given two clusters b and c to be merged into a parent a, the assignments are deterministically combined as $\hat{\mathbf{z}}_{I_b \cup I_c} = \mathrm{merge}_{bc}(\mathbf{z})$. The parent parameters are then proposed according to $\mu_a \sim q(\mu_a \mid \mathbf{x}, \mathbf{z})$ and $\Sigma_a \sim p(\Sigma_a \mid \mathbf{x}, \mathbf{z}, \mu_a)$, as in [3, 31]. Although the reassignment step is deterministic, the move is termed *randomized* because it is the reverse of a random split and its acceptance ratio depends on the corresponding random proposal probabilities.

Acceptance ratio. The move is accepted with probability $\min(1, H^{\text{rand}}_{\text{merge}})$, where

$$H^{\text{rand}}_{\text{merge}} = \frac{\Gamma(\alpha)\,\Gamma(\hat{N}_a)\,\Gamma(\frac{\alpha}{2} + N_b)\,\Gamma(\frac{\alpha}{2} + N_c)}{\alpha\,\Gamma(\frac{\alpha}{2})^2\,\Gamma(\alpha + \hat{N}_a)\,\Gamma(N_b)\,\Gamma(N_c)} \cdot \frac{p(\mathbf{x} \mid \hat{\mathbf{z}}, \hat{\mu}_a)}{p(\mathbf{x} \mid \mathbf{z}, \mu_b)\,p(\mathbf{x} \mid \mathbf{z}, \mu_c)\,p(\mu_b)} \cdot \frac{q(\mu_b \mid \mathbf{x}, \mathbf{z})\,q(\mu_c \mid \mathbf{x}, \mathbf{z})}{q(\hat{\mu}_a \mid \mathbf{x}, \hat{\mathbf{z}})}. \tag{19}$$

Here $\Gamma(\cdot)$ denotes the gamma function, $\hat{N}_a = N_b + N_c$ the total size of the merged cluster, $p(\mathbf{x} \mid \hat{\mathbf{z}}, \hat{\mu}_a)$ the likelihood under the merged cluster, $p(\mathbf{x} \mid \mathbf{z}, \mu_b)\,p(\mathbf{x} \mid \mathbf{z}, \mu_c)$ the likelihood under the two clusters prior to merging, $p(\mu_b)$ the uniform spherical prior on a proposed center, and $q(\cdot \mid \cdot)$ the auxiliary proposal densities used in the reverse move.

G. DeepDP-TGMM Hyperparameters: Empirical Guidelines

The performance of DeepDP-TGMM is influenced by five factors: the training horizon (number of epochs), the initialization of the number of clusters K, the Dirichlet concentration parameter α, and the inverse-Wishart parameters (Δ, ν). Below we summarize how these were chosen in practice and the empirical effects we observed.

Number of epochs. The number of epochs was determined by the point at which the inferred number of clusters K converged, i.e., when no additional split–merge proposals were accepted. We selected this empirically by monitoring training runs and recording the average epoch at which K stabilized. The maximal number of epochs was then set slightly beyond this point to provide a margin of safety.

Initial number of clusters K_{init}. DeepDP-TGMM was found to be robust to the choice of K_{init}, with final clustering performance remaining stable across a wide range of initializations. Nevertheless, providing a good prior estimate of the number of clusters accelerates convergence by reducing the number of split–merge moves required.

Concentration parameter α. The parameter α controls cluster granularity. Larger values encourage additional clusters, whereas smaller values promote fewer, broader clusters. In all experiments, we fixed $\alpha = 10$, and this value yielded stable results across all datasets and SSL embeddings tested. This demonstrates that $\alpha = 10$ serves as a reliable default choice for DeepDP-TGMM.

Inverse-Wishart scale Δ. We parameterized the scale matrix as $\Delta = sI$, where $I \in \mathbb{R}^{d\times d}$ is the identity matrix. The scalar s controls the dispersion of tangent-space Gaussians. Small values of s produce compact clusters, leading to frequent splits and almost no merges, whereas large values produce broad clusters, suppressing splits and hindering refinement. Empirically, we found that s should be set such that both split and merge proposals are accepted during the first few hundred epochs.

Inverse-Wishart degrees of freedom ν. The degrees of freedom ν must satisfy $\nu > D - 1$ for the prior to be proper. Larger values enforce near-spherical covariances and slow refinement, whereas smaller values allow more anisotropy and can lead to frequent splits. From our experiments, a default setting of $\nu = D + 1$ provided stable results across datasets, with slightly larger values sometimes beneficial to reduce excessive anisotropy and stabilize covariance estimation.