Beyond Spectral Clustering: Probabilistic Cuts for Differentiable Graph Partitioning

Ayoub Ghriss
ayoub.ghriss@polytechnique.org
Department of Computer Science
University of Colorado, Boulder

Abstract

Probabilistic relaxations of graph cuts offer a differentiable alternative to spectral clustering, enabling end-to-end and online learning without eigendecompositions, yet prior work centered on RatioCut and lacked general guarantees and principled gradients. We present a unified probabilistic framework that covers a wide class of cuts, including Normalized Cut. Our framework provides tight analytic upper bounds on expected discrete cuts via integral representations and Gauss hypergeometric functions with closed-form forward and backward. Together, these results deliver a rigorous, numerically stable foundation[1] for scalable, differentiable graph partitioning covering a wide range of clustering and contrastive learning objectives.

1 INTRODUCTION

Self-Supervised Learning (SSL) has become the backbone of modern representation learning, closing the gap to supervised baselines across vision, speech, and language (Radford et al., 2021; Baevski et al., 2020). Successful objectives broadly fall into two families: *contrastive* methods that optimize pairwise relationships (Chen et al., 2020; van den Oord et al., 2018; He et al., 2020), and *non-contrastive* or *masked* approaches that rely on local reconstruction or invariance (Grill et al., 2020; Siméoni et al., 2025; He et al., 2021). While effective, clustering-based variants such as DeepCluster (Caron et al., 2018) or SwAV (Caron et al., 2020) often impose Voronoi tessellations on the latent space. These assumptions bias models toward convex, globular clusters, failing to capture the intrinsic *manifold structure* of complex data distributions.

The ideal alternative can be achieved through graph cut-based partitioning, more specifically through the *Normalized Cut* (NCut) (Shi and Malik, 2000). Unlike simple Ratio Cuts (RCut); which balance partitions based on node counts; NCut balances partitions based on *volume* (total edge weight). This defines distance not by Euclidean proximity, but by connectivity through *dense domains*, approximating geodesic distance on the underlying manifold. Despite this geometric superiority, NCut remains largely incompatible with modern deep learning: standard spectral relaxations require solving generalized eigenvalue problems, a process that is computationally expensive ($O(N^3)$ in dense setting) and notoriously unstable to differentiate end-to-end.

In this work, we bridge the divide between the geometric purity of graph cuts and the scalability of modern SSL. We present a *unified, differentiable probabilistic framework* that relaxes the discrete graph cut problem without resorting to spectral decomposition by extending prior work of Ghriss and Monteleoni (2025). By treating cluster assignments as probabilistic variables, we resolve the challenge of evaluating the expected volume-normalized cut; an expectation of a ratio using *Gauss hypergeometric functions*. This yields tight, analytic upper bounds that turn the discrete cut-based objectives into a smooth, stable surrogate.

Our contributions are as follows:

- We derive a probabilistic relaxation for a wide class of graph cuts, including Normalized Cut. Unlike prior approximations, our framework respects the volume constraints required for manifold-aware partitioning.
- We provide closed-form forward and backward passes using hypergeometric polynomials, estab-

[1] Github: https://github.com/ayghri/pgcuts

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

lishing a numerically stable foundation for scalable differentiation that avoids eigendecompositions entirely.

- We rigorously control the approximation error via two-sided AM–GM gap bounds and a zero-aware penalty, ensuring the surrogate faithfully minimizes the true expected cut.
- We connect this geometric view back to SSL, showing that widely used contrastive objectives (e.g., SimCLR, CLIP) emerge as special cases of our envelope when the graph is constructed from batch embeddings.

2 PRELIMINARIES

Let $\mathcal{G} = (V, E, \boldsymbol{W})$ be an undirected weighted graph on $n = |V|$ vertices with a symmetric, elementwise nonnegative adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{n\times n}$; assume $\boldsymbol{W}_{ii} = 0$. Define the degree of $i \in V$ by $d_i \overset{\text{def}}{=} \sum_{v_j \in V} \boldsymbol{W}_{ij}$ and the degree matrix $\boldsymbol{D} \overset{\text{def}}{=} \operatorname{diag}(d_1, \ldots, d_n)$.

For $A \subseteq V$, let $\overline{A} \overset{\text{def}}{=} V \setminus A$ and identify A by its indicator vector $\mathbf{1}_A \in \{0,1\}^n$. The cut associated with A is:

$$\operatorname{cut}(A) = \operatorname{cut}(\overline{A}) \overset{\text{def}}{=} \sum_{(v_i, v_j)\in A\times\overline{A}} \boldsymbol{W}_{ij} \;=\; \mathbf{1}_A^\top \boldsymbol{W}\, \mathbf{1}_{\overline{A}},$$

and the associated volume-normalized cut is:

$$\operatorname{VolCut}(A) \overset{\text{def}}{=} \frac{\operatorname{cut}(A)}{\operatorname{vol}(A)}, \tag{1}$$

where the *volume* is $\operatorname{vol}(A) \overset{\text{def}}{=} \sum_{v_i \in A} s(v_i)$ for a given vertex *size* function $s : V \to \mathbb{R}_{>0}$. For example, the ratio cut (**RCut**) uses $s(v_i) \equiv 1$ so $\operatorname{vol}(A) \equiv |A|$, whereas the normalized cut (**NCut**) uses $s(v_i) = d_i$. This difference can yield different partitioning that minimizes the volume-normalized cuts as shown in Figure 1.

We fix the size function s and write $\boldsymbol{s} \overset{\text{def}}{=} (s_1, \cdots, s_n)$ with $s_i \overset{\text{def}}{=} s(v_i)$. The goal is to find a k-way clustering $\mathcal{C}_k = \{\mathbb{C}_\ell\}_{\ell=1}^k$ of V that minimizes the volume-normalized graph cut:

$$\operatorname{GraphCut}(\mathcal{C}_k) \overset{\text{def}}{=} \frac{1}{2}\sum_{\ell=1}^{k} \operatorname{VolCut}(\mathbb{C}_\ell). \tag{2}$$

2.1 Probabilistic Relaxation

The Probabilistic Ratio-Cut (PRCut) (Ghriss and Monteleoni, 2025) adopts a probabilistic relaxation of k-way clustering. Let $\mathbf{a}_\ell \in \{0,1\}^n$ be the random indicator of the cluster $\mathbb{C}_\ell$. The clustering $\mathcal{C}_k$ is parameterized by a row-stochastic matrix $\boldsymbol{P} \in [0,1]^{n\times k}$ with $\sum_{\ell=1}^k \boldsymbol{P}_{i\ell} = 1$ and $\boldsymbol{P}_{i\ell} = \Pr(\mathrm{a}_{\ell,i} = 1) = \Pr(v_i \in \mathbb{C}_\ell)$.

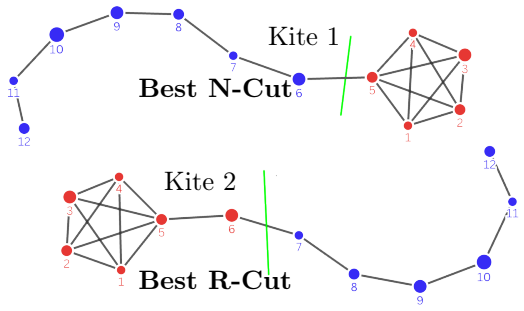


Figure 1: In the kite graph with $W_{i,j} = 1$ for connected nodes, RCut maximizes the average within-cluster edge weight, while NCut identifies clusters with densely connected nodes. Using Equation (1) on Kite 1 and Kite 2, NCut evaluates to, respectively, $\left(\frac{33}{260}, \frac{33}{242}\right)$, while RCut evaluates to $\left(\frac{12}{35}, \frac{12}{36}\right)$.

The expected graph cut is defined as:

$$\operatorname{GraphCut}(\boldsymbol{P}) \overset{\text{def}}{=} \frac{1}{2}\sum_{\ell=1}^{k} \mathbb{E}\left[\widehat{\operatorname{VolCut}}(\mathbf{a}_\ell)\right], \tag{3}$$

where

$$\widehat{\operatorname{VolCut}}(\mathbf{a}_\ell) \overset{\text{def}}{=} \frac{\mathbf{a}_\ell^\top \boldsymbol{W}\,(\mathbf{1} - \mathbf{a}_\ell)}{\boldsymbol{s}^\top \mathbf{a}_\ell}. \tag{4}$$

The following bound underpins the PRCut framework:

Proposition 1 (PRCut bound (Ghriss and Monteleoni, 2025)). *For the ratio cut ($s_i \equiv 1,\ \forall i \in \{1, \ldots, n\}$):*

$$\mathbb{E}\left[\widehat{\operatorname{VolCut}}_R(\mathbf{a}_\ell)\right] \le \frac{1}{n\,\overline{\boldsymbol{P}_{:,\ell}}} \sum_{i,j=1}^{n} W_{ij}\left(P_{i\ell} + P_{j\ell} - 2\,P_{i\ell}P_{j\ell}\right),$$

where $\overline{\boldsymbol{P}_{:,\ell}} \overset{def}{=} \frac{1}{n}\sum_{i=1}^n P_{i\ell}$ denotes the expected fraction of vertices assigned to $\mathbb{C}_\ell$.

In this paper, we derive tighter, more general bounds for an arbitrary vertex-weight function s, with concentration guarantees and gradients that are compatible with first-order optimization.

3 PROPOSED METHOD

By symmetry in Equation (3), it suffices to bound the expected VolCut for a single cluster. Fix a cluster $\mathbb{C}$ and drop the index ℓ. Let $\mathbf{a} \in \{0,1\}^n$ be its random indicator with independent coordinates $\mathrm{a}_i \sim \text{Bernoulli}(p_i)$, and let $\boldsymbol{p} = (p_1, \ldots, p_n)^\top \in [0,1]^n$.

$$\widehat{\operatorname{VolCut}}(\mathbf{a}) \;=\; \sum_{i,j=1}^{n} \boldsymbol{W}_{ij}\, \frac{\mathrm{a}_i(1 - \mathrm{a}_j)}{\sum_{l=1}^n s_l\, \mathrm{a}_l}. \tag{5}$$

Without loss of generality, let $(i, j) = (1, 2)$(note that $\boldsymbol{W}_{11} = \boldsymbol{W}_{22} = 0$) and write:

$$\mathbb{E}\left[\frac{\mathrm{a}_1(1-\mathrm{a}_2)}{\sum_{i=1}^n s_i\,\mathrm{a}_i}\right] \;=\; p_1(1-p_2)\,\mathbb{E}\left[\frac{1}{s_1+\sum_{i=3}^n s_i\,\mathrm{a}_i}\right].$$

Thus, we must evaluate expectations of the form $\mathbb{E}\,[1/(q+\mathrm{x})]$ with $q > 0$ and $\mathrm{x} = \sum_{i=3}^n s_i\,\mathrm{a}_i$. It turns out that x follows a generalized Poisson–Binomial distribution. The Poisson–Binomial distribution is well studied and has applications across seemingly unrelated areas (Chen and Liu, 1997; Cam, 1960). We use its generalized form:

Definition 1 (Generalized Poisson–Binomial (GPB)). *Let $\boldsymbol{\alpha} \in [0,1]^m$ and $\theta_i < \beta_i$ be real constants, and let $\mathrm{r}_i \sim \mathrm{Bernoulli}(\alpha_i)$ independently. The random variable $\mathrm{x} = \sum_{i=1}^m \big(\theta_i(1-\mathrm{r}_i) + \beta_i \mathrm{r}_i\big)$ follows a generalized Poisson–Binomial distribution (Zhang et al., 2017).*

In our setting, $m \overset{\text{def}}{=} n-2$, $\boldsymbol{\alpha} = (p_3, \ldots, p_n)$, and the weights are $(\theta_i, \beta_i) = (0, s_i)$, so $\mathrm{x} = \sum_{i=3}^n s_i\,\mathrm{r}_i$. We denote this special case by $\mathrm{GPB}(\boldsymbol{\alpha}, \boldsymbol{\beta})$, and compute its probability generating function (PGF) G_{x}:

$$G_{\mathrm{x}}(t) \overset{\text{def}}{=} \mathbb{E}\,[t^{\mathrm{x}}] = \prod_{i=1}^m \left(1 - \alpha_i + \alpha_i\, t^{\,\beta_i}\right),\; t \in [0,1]. \quad (6)$$

The target expectation can now be computed via the identity $x^{-1} = \int_0^1 t^{x-1}\,dt$ for $x > 0$ (see Appendix A.2):

Lemma 1 (Integral representation). *Define the integral $\mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta})$ as:*

$$\mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta}) \;\overset{def}{=}\; \int_0^1 t^{\,q-1} \prod_{i=1}^m \left(1 - \alpha_i + \alpha_i\, t^{\,\beta_i}\right) dt. \quad (7)$$

For any $q > 0$, we have:

$$\mathbb{E}\left[\frac{1}{q+\mathrm{x}}\right] \;=\; \mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta}) \quad (8)$$

For the ratio cut, $q = 1$ and $\beta_i \equiv 1$, and PRCut uses the bound $\mathbb{E}\left[\frac{1}{1+\mathrm{x}}\right] \le (\sum_i \alpha_i)^{-1}$. In this work, s need not be constant, so different tools are required.

We first consider the case $\beta_i \equiv \beta$ and recall Gauss's hypergeometric function ${}_2F_1$ (Chambers, 1992), defined for $|z| < 1$ by the absolutely convergent power series:

$${}_2F_1(a, b; c; z) \;=\; \sum_{k=0}^{\infty} \frac{(a)_k\,(b)_k}{(c)_k}\,\frac{z^k}{k!}, \quad (9)$$

where $(x)_k \overset{\text{def}}{=} x(x+1)\cdots(x+k-1)$ is the rising factorial, with $(x)_0 \overset{\text{def}}{=} 1$.

Lemma 2 (Euler's identity). *If $c > b > 0$ and $z \in [0,1]$, then ${}_2F_1(a, b; c; z)$ is equal to:*

$$\frac{\Gamma(c)}{\Gamma(b)\,\Gamma(c-b)} \int_0^1 t^{\,b-1}(1-t)^{\,c-b-1}\,(1-zt)^{-a}\,dt, \quad (10)$$

where Γ denotes the gamma function.

A useful property of ${}_2F_1$ is the derivative formula:

$$\frac{d}{dz}\,{}_2F_1(a, b; c; z) \;=\; \frac{ab}{c}\,{}_2F_1(a+1, b+1; c+1; z), \quad (11)$$

which, in particular, implies the following:

Lemma 3 (Properties of ${}_2F_1$). *Let $m \in \mathbb{N}$, $b > 0$, and $c > b$. On $[0,1]$, the function $f(z) \overset{def}{=} {}_2F_1(-m, b; c; z)$ is a degree-m polynomial that is decreasing, convex, and L-Lipschitz with $L = \frac{mb}{c}$.*

The integral in Lemma 1 admits a computable and differentiable upper bound (proof in Appendix A.3).

Theorem 1 (Hypergeometric bound). *Assume $\beta_i \equiv \beta > 0$. For any $q > 0$,*

$$\mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta}) \;\le\; \mathcal{H}_\beta(q; \bar{\alpha}, m) \quad (12)$$

where $\mathcal{H}_\beta(q; \bar{\alpha}, m) \overset{def}{=} \frac{1}{q}\;{}_2F_1\Big(-m,\,1;\,\frac{q}{\beta}+1;\,\bar{\alpha}\Big)$, and $\bar{\alpha} \overset{def}{=} \frac{1}{m}\sum_{i=1}^m \alpha_i$.

3.1 The Envelope Gap

To quantify the tightness of the bound from Theorem 1, we derive a pointwise Arithmetic Mean-Geometric Mean (**AM-GM**) gap.

Proposition 2 (Integrated AM–GM gap). *Let $\beta_i \equiv \beta > 0$ and $\boldsymbol{\alpha} \in [0,1]^m$. Define $h(t) = t^{q-1}\left(1 - \bar{\alpha} + \bar{\alpha}\,t^{\beta}\right)^m$ and:*

$$\underline{\Delta}(q, \boldsymbol{\alpha}) \overset{def}{=} \int_0^1 h(t)\Big(1 - e^{-\gamma(t)\mathrm{Var}(\boldsymbol{\alpha})}\Big)\,dt,$$

$$\overline{\Delta}(q, \boldsymbol{\alpha}) \overset{def}{=} \int_0^1 h(t)\Big(1 - e^{-\theta(t)\mathrm{Var}(\boldsymbol{\alpha})}\Big)\,dt,$$

with $\gamma(t) \overset{def}{=} \frac{m}{2}(1-t^{\beta})^2$ and $\theta(t) \overset{def}{=} \gamma(t)/t^{2\beta}$.

We have the following gap:

$$\underline{\Delta}(q, \boldsymbol{\alpha}) \le \mathcal{H}_\beta(q; \bar{\alpha}, m) \;-\; \mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta}) \le \overline{\Delta}(q, \boldsymbol{\alpha}), \quad (13)$$

with $\mathrm{Var}(\boldsymbol{\alpha})$ computed under uniform sampling of the graph nodes, and equality throughout iff $\mathrm{Var}(\boldsymbol{\alpha}) = 0$.

A convenient corollary gives an explicit upper bound.

Corollary 1 (Simple upper bound). *Under the conditions of Proposition 2, for any $q > 0$,*

$$\mathcal{H}_\beta(q; \bar{\alpha}, m) - \mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta}) \le \frac{m}{2}\mathrm{Var}(\boldsymbol{\alpha}) \int_0^1 h(t)\theta(t)\,dt.$$

See Appendix A.4 for the proofs of Proposition 2 and its corollary.

Zero-aware gap control. If we examine Equation (7), we see that coordinates with $\alpha_i = 0$ contribute the factor $(1 - \alpha_i + \alpha_i t^\beta) \equiv 1$ and thus have no influence on $\mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta})$. The AM–GM gap in Proposition 2 over-penalizes configurations with many inactive entries: zeros still inflate $\mathrm{Var}(\boldsymbol{\alpha})$ even though they do not affect the product inside the integral. We therefore replace the plain variance by a *zero-aware* weighted dispersion that vanishes at $\alpha_i = 0$. Let $\omega_0(x) \overset{\text{def}}{=} x$ (more generally, $\omega_0(x) = x^a, a \in [1, 2]$), and define:

$$\Omega \overset{\text{def}}{=} \sum_{i=1}^{m} \omega_0(\alpha_i), \qquad \bar{\alpha}^{\omega_0} \overset{\text{def}}{=} \frac{1}{\Omega} \sum_{i=1}^{m} \omega_0(\alpha_i)\, \alpha_i, \quad (14)$$

$$\mathrm{Var}^{\omega_0}(\boldsymbol{\alpha}) \overset{\text{def}}{=} \frac{1}{\Omega} \sum_{i=1}^{m} \omega_0(\alpha_i) \big(\alpha_i - \bar{\alpha}^{\omega_0}\big)^2 \quad (15)$$

Definition 2. *Let $\mathcal{H}_\beta(q; \bar{\alpha}, m)$ be the envelope from Theorem 1. Define the second forward β-difference:*

$$\Delta(q; \bar{\alpha}, \beta, m) \overset{def}{=} \sum_{r=0}^{2} \binom{2}{r} (-1)^r\, \mathcal{H}_\beta\big(q + r\beta; \bar{\alpha}, m\big). \quad (16)$$

and $\mathcal{A}(q; \bar{\alpha}, \beta, m) = \frac{\partial}{\partial \bar{\alpha}} \Delta(q; \bar{\alpha}, \beta, m)$.

Proposition 3 (Zero-aware AM-GM gap)**.** *Under the assumptions of Proposition 2 with common $\beta > 0$, a zero-aware replacement for the simple upper bound of Corollary 1 is:*

$$\widetilde{\mathcal{A}}(q, \boldsymbol{\alpha}, \beta, m) \overset{def}{=} \frac{m}{2}\, \mathrm{Var}^{\omega_0}(\boldsymbol{\alpha}) \mathcal{A}\big(q; \bar{\alpha}, \beta, m\big). \quad (17)$$

In particular, coordinates with $\alpha_i \equiv 0$ incur zero penalty, while $\alpha_i = 1$ retains full influence through $\mathrm{Var}^{\omega_0}(\boldsymbol{\alpha})$; when $\omega_0 \equiv 1$ we recover Corollary 1. More details are provided in Appendix A.5 about the derivation of various quantities.

The main takeaway from Equation (17) is that the AM-GM gap can be reduced by pushing the active entries towards a common non-zero value. That is particularly true for the case where $\alpha_i \in \{0, 1\}$ and validates the claim that PRCut (Ghriss and Monteleoni, 2025) bound is tight in the deterministic setting.

3.2 Concentration of batch-based estimators

Fix $m \in \mathbb{N}$, $q > 0$, $\beta > 0$. Let $c = \frac{q}{\beta}$ and $\boldsymbol{\alpha} \in [0, 1]^m$ with mean $\bar{\alpha}$ and population variance $\mathrm{Var}(\boldsymbol{\alpha})$. Form a random minibatch $S = (i_1, \ldots, i_B)$ of size B, sampled with replacement, and define the plug-in estimator:

$$\hat{\alpha}_S \overset{\text{def}}{=} \frac{1}{B} \sum_{r=1}^{B} \alpha_{i_r}, \qquad \hat{H}(S) \overset{\text{def}}{=} \mathcal{H}_\beta\big(q; \hat{\alpha}_S, m\big). \quad (18)$$

By Lemma 3, $\mathcal{H}_\beta(q; \cdot, m)$ is decreasing, convex and L-Lipschitz with:

$$L = \frac{m}{q\,(c+1)}. \quad (19)$$

Differentiating Equation (11) again gives:

$$\frac{d^2}{dz^2}\, \mathcal{H}_\beta(q; z, m) \le \underbrace{\frac{1}{q} \cdot \frac{2m(m-1)}{c(c+1)}}_{\overset{\text{def}}{=} K}, \quad (20)$$

and the Taylor bound from Equation (20) yields:

$$0 \le \mathbb{E}\left[\hat{H}(S)\right] - \mathcal{H}_\beta\big(q; \bar{\alpha}, m\big) \le \frac{K}{2}\, \frac{\sigma^2}{B}. \quad (21)$$

Changing one element of S changes $\hat{\alpha}_S$ by at most $1/B$, hence by Equation (19) the function $S \mapsto \hat{H}(S)$ changes by at most L/n. We obtain via McDiarmid's inequality for any $\varepsilon > 0$:

$$\Pr\Big(\big|\hat{H}(S) - \mathbb{E}\left[\hat{H}(S)\right]\big| \ge \varepsilon\Big) \le 2 \exp\left(\frac{-2B\,\varepsilon^2}{L^2}\right). \quad (22)$$

Combining Equations (21) and (22) with a triangle inequality yields the following finite-sample guarantee.

Proposition 4 (Concentration of the minibatch envelope)**.** *With probability at least $1 - \delta$,*

$$\big|\hat{H}(S) - \mathcal{H}_\beta(q; \bar{\alpha}, m)\big| \le L \sqrt{\frac{1}{2B} \log\frac{2}{\delta}} + \frac{K}{2}\, \frac{\sigma^2}{B}, \quad (23)$$

where L and K defined in Equations (19) and (20).

Proof. By McDiarmid, with probability $\ge 1 - \delta$, $|\hat{H}(S) - \mathbb{E}\left[\hat{H}(S)\right]| \le L\sqrt{\frac{1}{2B} \log(2/\delta)}$. Add and subtract $\mathcal{H}_\beta(q; \bar{\alpha}, m)$ and use $\mathbb{E}\left[\hat{H}(S)\right] - \mathcal{H}_\beta(q; \bar{\alpha}, m) \le \frac{K}{2}\mathrm{Var}(\hat{\alpha}_S)$ from Equation (21), with $\mathrm{Var}(\hat{\alpha}_S) = \sigma^2/B$. □

3.3 Heterogeneous degrees

When $(\beta_i)_i$ vary, directly using a single β loses heterogeneity. We partition indices into d disjoint bins $S_1, \ldots, S_d$ based on their β_i values. Let $m_j \overset{\text{def}}{=} |S_j|$, $\bar{\alpha}_j \overset{\text{def}}{=} m_j^{-1} \sum_{i \in S_j} \alpha_i$, and define the bin interval $B_j = [b_{j-1}, b_j]$ with representatives $\beta_j^\star \in B_j$ specified below.

For $t \in [0, 1]$ and $\alpha \in [0, 1]$, the map $\beta \mapsto (1 - \alpha + \alpha\, t^\beta)$ is nonincreasing. Hence, for any fixed bin S_j and any choice $\beta_j^\star \le \beta_i$ for all $i \in S_j$, we have:

$$\prod_{i \in S_j} \left(1 - \alpha_i + \alpha_i\, t^{\beta_i}\right) \le \prod_{i \in S_j} \left(1 - \alpha_i + \alpha_i\, t^{\beta_j^\star}\right). \quad (24)$$

Applying Jensen in α to the RHS (log is concave in α for fixed t^{β}) gives, for each j,

$$\prod_{i\in S_j}\left(1-\alpha_i+\alpha_i\,t^{\beta_j^\star}\right) \;\le\; \left(1-\bar\alpha_j+\bar\alpha_j\,t^{\beta_j^\star}\right)^{m_j}. \quad (25)$$

Recall the envelope $\mathcal{H}_\beta(q;\bar\alpha,m)$ from Theorem 1. We now control the heterogeneous case:

Theorem 2 (Binned Hölder bound)**.** *Let $q>0$ and partition $\{1,\dots,m\}$ into bins $S_1,\dots,S_d$. Choose representatives $\beta_j^\star\in B_j$ satisfying $\beta_j^\star\le\beta_i$ for every $i\in S_j$ (e.g., left endpoints). Then*

$$\mathcal{I}(q;\boldsymbol{\alpha},\boldsymbol{\beta}) \;\le\; \prod_{j=1}^{d}\Big[\,\mathcal{H}_{\beta_j^\star}(q;\bar\alpha_j,m)\,\Big]^{\frac{m_j}{m}}. \quad (26)$$

Hölder inequality is tight iff the functions $\{f_j\}_{j=1}^d$ are pairwise proportional (*colinear*) in L^{p_j}: there exist constants $\kappa_j>0$ and a common shape ϕ such that $f_j(t)=\kappa_j\,\phi(t)$ for almost every $t\in[0,1]$. In our construction,

$$f_j(t)\propto t^{\frac{q-1}{p_j}}\left(1-\bar\alpha_j+\bar\alpha_j\,t^{\beta_j^\star}\right)^{m}.$$

Hence near-tightness is promoted when, *across bins*, the curves $t\mapsto(1-\bar\alpha_j+\bar\alpha_j\,t^{\beta_j^\star})$ have similar shapes, and, *within bins*, replacing β_i by $\beta_j^\star$ induces minimal distortion.

3.4 Optimization objective

We now put everything together to define the optimization objective of our probabilistic graph cut framework. For cluster ℓ, the expected contribution of edge (i,j) is:

$$\frac{1}{s_i}\boldsymbol{W}_{ij}\,\boldsymbol{P}_{i\ell}(1-\boldsymbol{P}_{j\ell})\mathcal{I}\big(s_i;\boldsymbol{P}_{-\{i\ell\}},\boldsymbol{s}_{-\{il\}}\big)$$

Fix $\ell\in\{1,\dots,k\}$ and partition indices into d bins $S_{\ell1},\dots,S_{\ell d}$ by their exponents $\beta_u\equiv s_u$ (e.g., degree-based); let $m_{\ell j}\overset{\text{def}}{=}|S_{\ell j}|$, $m_\ell\overset{\text{def}}{=}\sum_j m_{\ell j}$, and

$$\bar p_{\ell j}\;\overset{\text{def}}{=}\;\frac{1}{m_{\ell j}}\sum_{u\in S_{\ell j}}\boldsymbol{P}_{u\ell},\qquad w_{\ell j}\;\overset{\text{def}}{=}\;\frac{m_{\ell j}}{m_\ell}.$$

Choose representatives $\beta_{\ell j}^\star\le s_u$ for all $u\in S_{\ell j}$ (e.g., the bin's left endpoint or in–bin minimum) so that the bound direction is preserved (Section 3.3).

For a fixed $q>0$, Theorem 2 yields the per–cluster integrand bound. Plugging $q=s_i$ for each source vertex i gives the *per-vertex* envelope:

$$\Phi_\ell(q)\;\overset{\text{def}}{=}\;\prod_{j=1}^{d}\Big[\,\mathcal{H}_{\beta_{\ell j}^\star}(q;\bar p_{\ell j},\,m_\ell)\,\Big]^{w_{\ell j}}.$$

Define the edge–aggregated source weights

$$M_{i\ell}(\boldsymbol{P})\;\overset{\text{def}}{=}\;\sum_{j=1}^{n}\boldsymbol{W}_{ij}\,\boldsymbol{P}_{i\ell}\,(1-\boldsymbol{P}_{j\ell}),$$

so that the total contribution of cluster ℓ is:

$$U_\ell(\boldsymbol{P})\overset{\text{def}}{=}\sum_{i=1}^{n}M_{i\ell}(\boldsymbol{P})\,\Phi_\ell\big(s_i\big), \quad (27)$$

and $U(\boldsymbol{P})\overset{\text{def}}{=}\sum_{\ell=1}^k U_\ell(\boldsymbol{P})$. By construction (linearity of expectation and Theorem 2), $I_{\text{true}}\le U$.

Within each bin, replacing $\{\boldsymbol{P}_{u\ell}\}_{u\in S_{\ell j}}$ by their mean $\bar p_{\ell j}$ induces an AM–GM gap controlled by Propositions 2 and 3. A conservative, separable upper bound for cluster ℓ is:

$$\Gamma_\ell(\boldsymbol{P})\overset{\text{def}}{=}\sum_{i=1}^{n}M_{i\ell}(\boldsymbol{P})\left[\sum_{j=1}^{d}w_{\ell j}\,\mathbb{A}(\beta_{\ell j}^\star,\boldsymbol{p}_{\ell j},m_\ell)\right], \quad (28)$$

and $\mathbb{A}$ is the zero–aware coefficient from Proposition 3. Summing over clusters, $\Gamma(\boldsymbol{P})\overset{\text{def}}{=}\sum_{\ell=1}^k\Gamma_\ell(\boldsymbol{P})$ satisfies

$$0\;\le\;U(\boldsymbol{P})-I_{\text{true}}(\boldsymbol{P})\;\le\;\Gamma(\boldsymbol{P}).$$

We minimize a penalized majorizer of the expected GraphCut:

$$\boxed{\min_{\boldsymbol{P}=\boldsymbol{P}(\boldsymbol{z})}\mathbb{J}_\rho(\boldsymbol{P})\overset{\text{def}}{=}U(\boldsymbol{P})\;+\;\rho\Gamma(\boldsymbol{P}),\quad\rho\ge0,} \quad (29)$$

where $\boldsymbol{z}$ can be the parameterization logits (via Softmax). Since $I_{\text{true}}\le U$ and $\Gamma\ge0$, we retain $I_{\text{true}}\le\mathbb{J}_\rho$ for all $\rho\ge0$ while explicitly shrinking the AM–GM gap.

We detail in Appendix G the forward-backward derivation and implementation for our final objective.

Time complexity. With minibatches of size B, we first construct the batch adjacency $\boldsymbol{W}_{\text{batch}}\in\mathbb{R}_+^{B\times B}$. For dense similarities this costs $O(B^2)$ time (and $O(B^2)$ memory); in sparse kNN settings we can replace B^2 by $\text{nnz}(\boldsymbol{W}_{\text{batch}})$. We then precompute the envelope terms $\mathcal{H}_{\beta_j^\star,\ell}$ for every (bin j, cluster ℓ). A straightforward implementation performs $O(d\,k\,m)$ work, where d is the number of bins, k the number of clusters, and m the polynomial degree in the ${}_2F_1(-m,\cdot;\cdot;z)$ evaluation. Because these computations factor across (j,ℓ), they are embarrassingly parallel; with $(d\times k)$ workers the wall-clock reduces to $O(m)$ (see Appendix G). Thus, each batch step practically takes $O(\text{nnz}(\boldsymbol{W}_{\text{batch}})\cdot k+m)$.

4 RELATED WORK

Let $\mathbf{a} \in \{0,1\}^n$ be a cluster indicator and $\boldsymbol{p} \in [0,1]^n$ with $p_i \overset{\text{def}}{=} \Pr[\mathbf{a}_i{=}1]$. Because $p_i^2 \le p_i$ on $[0,1]$, the Laplacian quadratic $\boldsymbol{p}^\top \boldsymbol{L}\boldsymbol{p}$ and the expected cut $\mathbb{E}\left[\mathbf{a}^\top \boldsymbol{L}\mathbf{a}\right]$ coincide *only* when $\boldsymbol{p}$ is binary: the degree term satisfies $\boldsymbol{p}^\top \boldsymbol{D}\boldsymbol{p} = \sum_i d_i p_i^2 \le \sum_i d_i p_i = \mathbb{E}\left[\mathbf{a}^\top \boldsymbol{D}\mathbf{a}\right]$, and the adjacency term $\mathbb{E}\left[\mathbf{a}^\top \boldsymbol{W}\mathbf{a}\right] = \boldsymbol{p}^\top \boldsymbol{W}\boldsymbol{p}$ requires the additional assumption $\mathbb{E}\left[\mathbf{a}_i\mathbf{a}_j\right] = p_i p_j$. Our formulation therefore keeps the expected (linear) degree term $\sum_i d_i p_i$ rather than the quadratic $\sum_i d_i p_i^2$.

Writing the soft cut for cluster ℓ with $\boldsymbol{p} = \boldsymbol{P}_\ell$:

$$\underset{\ell}{\text{cut}}(\boldsymbol{p}) = \sum_{ij} \boldsymbol{W}_{ij}\, p_i(1{-}p_j) = \underbrace{\boldsymbol{p}^\top \boldsymbol{L}\,\boldsymbol{p}}_{\text{convex}} + \underbrace{\textstyle\sum_i d_i\, p_i(1{-}p_i)}_{\text{concave}},$$

a Difference of Convex (DC) decomposition since $\boldsymbol{p}^\top \boldsymbol{L}\boldsymbol{p}$ is convex, while the concave *fuzziness* term $\sum_i d_i p_i(1{-}p_i)$ vanishes for hard assignments.

Spectral NCut (Ng et al., 2001) relaxes indicators to continuous vectors on the Stiefel manifold while our approach operates directly on the assignment polytope $\mathcal{U}_{n,K}$, where the row-stochastic constraint $\sum_\ell \boldsymbol{P}_{i\ell} = 1$ couples all K columns and precludes the per-column eigendecomposition.

XOR similarity and cross-entropy relaxation. The per-edge cut contribution $p_i(1{-}p_j)$ can be read as an *XOR similarity*: it is maximal when exactly one of p_i, p_j is 1 (a cross-partition edge) and zero when both agree. Symmetrizing gives the per-edge cut cost

$$\delta_{ij} \overset{\text{def}}{=} p_i(1{-}p_j) + p_j(1{-}p_i).$$

Since $1{-}p \le -\log p$ for $p \in (0,1]$, each term is bounded by its cross-entropy counterpart, yielding

$$\delta_{ij} \;\le\; -p_i \log p_j - p_j \log p_i \;=\; \mathrm{CE}(p_i \| p_j) + \mathrm{CE}(p_j \| p_i).$$

Weighting by $\boldsymbol{W}_{ij}$ and summing over edges upper-bounds the expected cut in Equation (3). It suffices to consider one direction:

$$\mathrm{CE}(p_i \| p_j) = H(p_i) + D_{\mathrm{KL}}(p_i \parallel p_j),$$

so minimizing the cross-entropy surrogate simultaneously encourages *neighbor agreement* (small D_{KL}) and *sharp assignments* (small entropy H).

Temperature annealing. Parameterizing assignments via a softmax with temperature, $\boldsymbol{P}_{i\ell} = \mathrm{softmax}(z_{i\ell}/\tau)$, provides a smooth interpolation between uniform ($\tau \to \infty$) and hard ($\tau \to 0$) assignments. As τ decreases, the probabilities are pushed toward $\{0,1\}$, which has two effects: (i) the fuzziness term $\sum_i d_i p_i(1{-}p_i)$ in the DC decomposition above vanishes, so $\boldsymbol{p}^\top \boldsymbol{L}\boldsymbol{p} \to \mathbb{E}\left[\mathbf{a}^\top \boldsymbol{L}\mathbf{a}\right]$; and (ii) the gap $1{-}p \le -\log p$ tightens, so the cross-entropy surrogate converges to the XOR cut cost. When the cluster index ℓ ranges over batch instances, $\mathrm{CE}(\boldsymbol{P}_i \| \boldsymbol{P}_j) = -\log \boldsymbol{P}_{j,i}$ recovers InfoNCE; when ℓ indexes prototypes, it enforces code consistency.

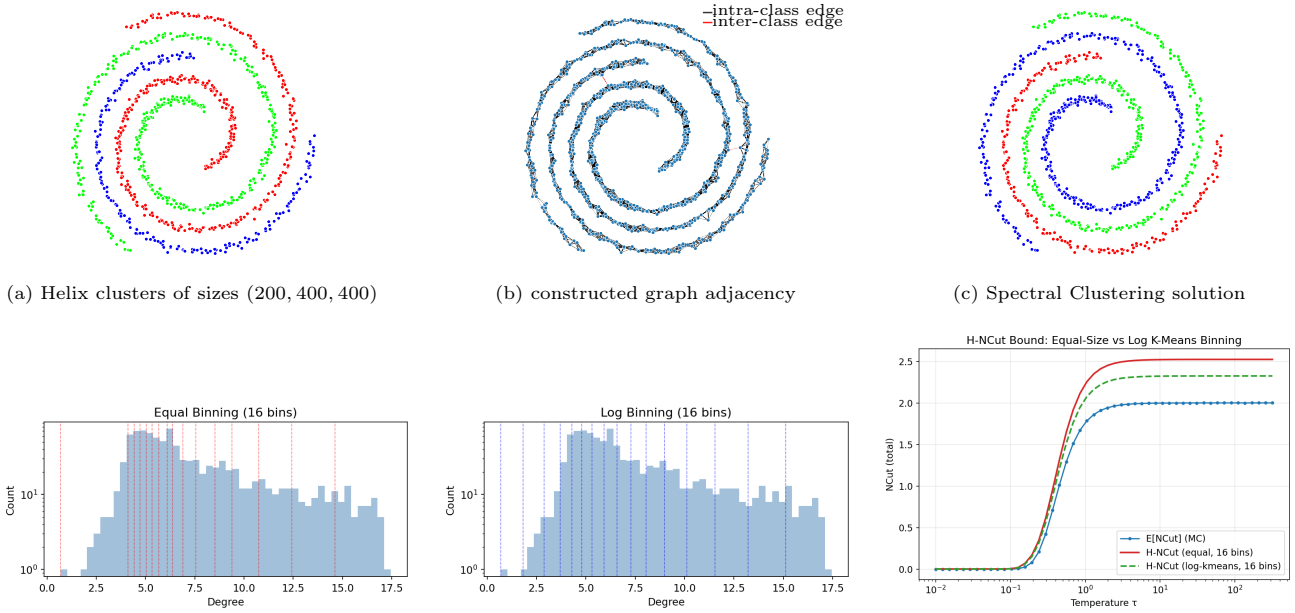
SimCLR as a probabilistic cut. SimCLR (Chen et al., 2020) builds representations via contrastive learning (Hadsell et al., 2006): for each image, two augmented views are embedded and trained with InfoNCE to attract positive pairs and repel negatives. This naturally defines a *view graph* with $\boldsymbol{W}_{ab} \overset{\text{def}}{=} \kappa(z_a, z_b)$ where $\kappa(u,v) = \exp(\langle u, v\rangle/\tau)$. With a single bin ($d{=}1$) and K equal to the number of latent classes, our envelope U_ℓ is modulated by $\mathcal{H}(s_i; \bar{p}_\ell, m_\ell)$, which is decreasing in $\bar{p}_\ell$ (Lemma 3): increasing same-class agreement monotonically decreases the bound. SimCLR's alignment and uniformity objectives reduce cross-edge mass in this view graph, so $\sum_\ell U_\ell$ decreases monotonically under InfoNCE updates. In the instance-discrimination limit (K equals batch size), minimizers of $\mathbb{J}_\rho$ coincide with those of SimCLR up to the reparameterization $\boldsymbol{P} = \mathrm{softmax}(\boldsymbol{z})$.

CLIP as a bipartite cut. CLIP (Radford et al., 2021) trains image and text encoders f_{θ_x}, g_{θ_t} with symmetric InfoNCE over both directions. Let $z_i = f_{\theta_x}(x_i)$, $u_j = g_{\theta_t}(t_j)$, and $\kappa(u,v) = \exp(\langle u, v\rangle/\tau)$. We construct a bipartite similarity graph $\mathcal{G} = (\mathcal{V}_x \cup \mathcal{V}_t, \mathcal{E})$ with no intra-modal edges:

$$\boldsymbol{W} = \begin{pmatrix} 0 & \boldsymbol{W}^{xt} \\ (\boldsymbol{W}^{xt})^\top & 0 \end{pmatrix}, \qquad \boldsymbol{W}^{xt}_{ij} \overset{\text{def}}{=} \kappa(z_i, u_j).$$

Each text node t_j defines a cluster $\ell{=}j$, and the model learns soft assignments $\boldsymbol{P}_{i\ell}$ of images to text clusters (and symmetrically, texts to image clusters). A matched image–text pair (x_i, t_j) with $y_i{=}j$ should have high $\boldsymbol{P}_{ij}$; minimizing the bipartite cut reduces the cross-edge mass $\sum_{(i,j):y_i \ne j} \boldsymbol{W}^{xt}_{ij}$, driving mismatched similarities down.

Because the two modalities live in different representation spaces, we use $d = 2$ Hölder bins with representatives $\beta_x^\star$, $\beta_t^\star$ (Section 3.3). The per-cluster envelope then factorizes as a product of the image-side and text-side envelopes. Taking the log converts this product into a sum over the two directions, recovering CLIP's symmetric image→text and text→image InfoNCE structure. In the paired-supervision limit (one text per class, one image per class), the minimizers coincide with those of CLIP up to the reparameterization $\boldsymbol{P} = \mathrm{softmax}(\boldsymbol{z})$.



(a) Helix clusters of sizes $(200, 400, 400)$ (b) constructed graph adjacency (c) Spectral Clustering solution

(d) Equal-frequency binning (d=16) (e) Log-adaptive binning (d=16) (f) H-NCut bound: equal vs. log-adaptive

Figure 2: **Synthetic helix simulation.** *Top:* three intertwined helices with unbalanced clusters and the constructed KNN graph. *Bottom:* degree distribution with bin boundaries for equal-frequency (d) and log-adaptive (e) binning. Log-adaptive binning concentrates boundaries where the degree distribution has mass, producing a tighter H-NCut bound across all temperatures (f). See Appendix D for per-cluster MC simulations.

5 EXPERIMENTS

We verify our bounds on a synthetic dataset of three intertwined helices with Gaussian noise and unbalanced clusters ($|C_1|$=200, $|C_2|$=$|C_3|$=400; Figure 2). A 50-NN RBF graph captures the manifold topology. We simulate soft assignments $\boldsymbol{P}_{i\ell} = \mathrm{softmax}(z_{i\ell}/\tau)$ at varying temperatures and estimate the expected cut via Monte Carlo (MC) sampling. Figure 2f shows that log-adaptive binning consistently produces a tighter H-NCut bound than equal-frequency binning. Additional MC simulations for RCut and NCut, per-cluster breakdowns, and the effect of the number of bins are reported in Appendix D.

Similarity Quality The performance of graph-based clustering depends critically on the quality of the constructed similarity graph. We introduce a normalized metric to quantify this:

Definition 3 (Graph Quality)**.** *Given a similarity graph $\boldsymbol{W}$ and ground-truth labels y, let $\boldsymbol{T} = \boldsymbol{D}^{-1}\boldsymbol{W}$ be the random-walk transition matrix. The* graph quality *is:*

$$Q = \frac{q - q_{chance}}{1 - q_{chance}}, \quad q = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{T}_{ij} \cdot 1[y_i = y_j], \tag{30}$$

where $n_k = |\{i : y_i = k\}|$ is the size of class k and $q_{chance} = \sum_k (n_k/n)^2$ is the probability of a same-class transition under a random labeling.

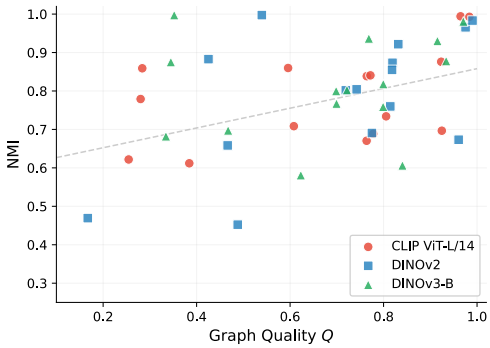


Figure 3: Graph quality Q vs. best NMI achieved by H-Cut across 15 datasets and three embedding models. The strong linear correlation confirms that Q is a reliable predictor of downstream clustering performance.

$Q = 0$ means the graph carries no class information beyond chance; $Q = 1$ means a single random walk step always lands in the same class. This metric accounts for the number of classes: $q = 0.1$ is excellent for $K = 100$ (since $q_{\text{chance}} \approx 0.01$) but poor for $K = 2$ (where $q_{\text{chance}} \approx 0.5$).

We observe a strong correlation between Q and downstream clustering accuracy across all datasets and embedding models (Tables 1 to 3), confirming that *the graph is the bottleneck*: when $Q > 0.8$, H-Cut consistently achieves high accuracy; when $Q < 0.4$, no

Table 1: **DINOv2 embeddings (Oquab et al., 2024).** Non-parametric Spectral Clustering (SC) vs. parametric probabilistic cut objectives on a 50-NN RBF graph. SC directly optimizes the graph Laplacian and achieves the lowest RCut/NCut, but H-Cut methods generalize better in ACC/NMI by leveraging the linear model as an implicit regularizer.

			Spectral Clustering				PRCut*				H-RCut				H-NCut			
Dataset	K	Q	ACC%	NMI%	RCut ↓	NCut ↓	ACC%	NMI%	RCut ↓	NCut ↓	ACC%	NMI%	RCut ↓	NCut ↓	ACC%	NMI%	RCut ↓	NCut ↓
CIFAR-10 (Krizhevsky, 2009)	10	0.98	78.5	90.2	**5.7**	**0.12**	93.2	91.4	13.1	0.29	98.5	96.5	8.9	0.20	**98.6**	**96.6**	8.8	0.20
CIFAR-100 (Krizhevsky, 2009)	100	0.82	67.4	81.6	**188**	**4.3**	81.7	86.3	601	12.6	80.6	86.0	562	11.7	**83.8**	**87.3**	543	11.7
STL-10 (Coates et al., 2011)	10	0.96	51.2	64.0	**3.9**	**0.09**	62.4	63.5	36.0	0.88	**68.1**	**67.3**	33.9	0.83	68.0	**67.3**	32.3	0.80
EuroSAT (Helber et al., 2019)	10	0.82	80.9	78.0	54	1.2	**93.2**	**85.5**	52.6	**1.1**	92.5	84.5	**52.1**	**1.1**	92.5	84.5	**52.1**	**1.1**
Imagenette (Howard, 2019)	10	0.99	**99.8**	**99.3**	**2.4**	**0.05**	99.2	98.1	3.5	0.08	99.2	98.3	3.3	0.07	99.2	98.3	3.2	0.07
Fashion-MNIST (Xiao et al., 2017)	10	0.81	71.2	75.4	**26**	**0.58**	**77.8**	75.8	38.4	0.83	77.3	**76.0**	38.0	0.83	77.1	75.9	38.3	0.83
MNIST (Lecun et al., 1998)	10	0.78	62.1	63.0	**60**	**1.3**	72.6	64.4	80.9	1.8	**77.0**	**69.1**	73.3	1.6	72.8	65.8	77.2	1.7
Pets (Parkhi et al., 2012)	37	0.83	87.0	91.6	**140**	**3.6**	**90.4**	**92.2**	176	4.4	89.9	91.9	176	4.4	90.0	91.8	172	4.3
Flowers-102 (Nilsback and Zisserman, 2008)	102	0.54	95.7	97.8	**1985**	**49.5**	**99.7**	**99.7**	2159	49.9	93.0	97.3	2386	52.3	94.3	97.2	2256	50.9
Food-101 (Bossard et al., 2014)	101	0.74	71.8	79.7	**367**	**8.0**	75.8	79.9	558	12.3	76.3	**80.4**	566	12.5	**76.6**	**80.4**	584	12.9
RESISC-45 (Cheng et al., 2017)	45	0.72	68.0	74.9	**314**	**7.3**	**79.2**	**80.2**	437	9.8	74.5	78.3	429	9.6	78.4	79.8	419	9.4
DTD (Cimpoi et al., 2014)	47	0.47	54.9	64.1	**575**	**15.0**	**57.9**	65.7	760	18.3	55.1	64.3	843	18.1	57.8	**65.8**	776	17.6
GTSRB (Stallkamp et al., 2012)	43	0.49	35.2	43.7	**526**	**12.9**	36.2	44.2	760	17.3	**37.9**	**45.2**	745	17.1	36.7	44.1	746	17.2
CUB-200 (Wah et al., 2011)	200	0.43	68.1	86.1	4029	**107**	**70.5**	**88.3**	**3784**	111	58.0	83.5	5244	118	58.7	83.8	5296	118
FGVC-Aircraft (Maji et al., 2013)	100	0.17	**21.9**	46.6	1494	**38.0**	21.4	**46.9**	**1453**	42.3	20.1	45.6	1665	41.4	20.1	45.7	1639	41.3

Table 2: **DINOv3-B embeddings (Siméoni et al., 2025).** DINOv3 uses register tokens and SigLIP-style training, producing higher-quality graphs on coarse-grained datasets (STL-10, GTSRB) but lower Q on fine-grained Flowers and CUB.

			Spectral Clustering				PRCut*				H-RCut				H-NCut			
Dataset	K	Q	ACC%	NMI%	RCut ↓	NCut ↓	ACC%	NMI%	RCut ↓	NCut ↓	ACC%	NMI%	RCut ↓	NCut ↓	ACC%	NMI%	RCut ↓	NCut ↓
CIFAR-10 (Krizhevsky, 2009)	10	0.93	87.5	**90.8**	**14**	**0.33**	86.6	83.7	34.9	0.83	**91.2**	87.8	31.5	0.75	85.6	83.3	46.1	1.1
CIFAR-100 (Krizhevsky, 2009)	100	0.70	63.1	76.2	**638**	**15.7**	**73.2**	**80.0**	1216	26.8	69.6	77.6	1306	28.9	70.7	78.9	1324	28.8
STL-10 (Coates et al., 2011)	10	0.92	88.6	**93.4**	**24**	**0.62**	87.2	86.4	44.0	1.1	86.5	88.2	40.6	1.0	**95.1**	93.0	31.9	0.83
EuroSAT (Helber et al., 2019)	10	0.80	82.7	79.3	**53**	**1.2**	89.1	81.1	68.6	1.6	**91.0**	**81.8**	65.8	1.5	86.2	79.4	69.8	1.6
Imagenette (Howard, 2019)	10	0.97	**99.4**	**98.3**	**8.8**	**0.20**	99.3	98.1	9.4	0.22	99.3	98.1	9.6	0.22	99.3	98.1	9.6	0.22
Fashion-MNIST (Xiao et al., 2017)	10	0.80	70.3	74.1	**22**	**0.51**	77.5	74.1	43.3	0.96	78.5	**75.9**	39.3	0.87	**78.6**	**75.9**	39.2	0.87
MNIST (Lecun et al., 1998)	10	0.84	**76.9**	**75.2**	**40**	**0.92**	64.1	58.5	77.2	1.8	66.5	60.6	74.1	1.7	66.5	59.3	74.4	1.7
Pets (Parkhi et al., 2012)	37	0.77	89.8	91.2	**256**	**6.7**	**93.7**	**93.7**	279	7.2	93.3	93.4	276	7.1	93.2	93.3	276	7.2
Flowers-102 (Nilsback and Zisserman, 2008)	102	0.35	94.1	97.3	**2392**	**66.8**	**99.7**	**99.7**	2451	66.5	86.5	92.3	2633	69.3	86.5	93.8	2638	69.0
Food-101 (Bossard et al., 2014)	101	0.72	73.5	**80.3**	**592**	**13.3**	**76.9**	**80.3**	876	20.4	75.9	79.8	857	19.6	74.5	79.0	908	20.8
RESISC-45 (Cheng et al., 2017)	45	0.70	66.7	74.6	**363**	**8.6**	**75.1**	**76.7**	497	11.6	70.7	74.6	529	12.3	70.6	74.4	527	12.2
DTD (Cimpoi et al., 2014)	47	0.47	61.0	68.3	**692**	**18.0**	**63.3**	**69.7**	832	20.5	60.0	67.5	868	20.9	59.7	67.6	870	21.0
GTSRB (Stallkamp et al., 2012)	43	0.62	**48.0**	**60.5**	**364**	**9.2**	46.1	55.6	619	15.1	45.2	55.9	649	15.7	46.1	58.1	627	15.1
CUB-200 (Wah et al., 2011)	200	0.34	73.7	**88.2**	**4761**	**125**	**74.2**	87.5	5061	128	62.3	83.2	5431	135	65.0	83.6	5298	133
FGVC-Aircraft (Maji et al., 2013)	100	0.33	**42.6**	**69.5**	**1722**	**43.4**	39.2	68.2	1908	48.0	41.0	67.6	1968	48.8	39.6	67.5	1945	48.0

graph-cut method can recover the class structure.

To apply the Binned Hölder Bound (Theorem 2) effectively, we must partition the vertices $\{1, \ldots, n\}$ into d disjoint sets $S_1, \ldots, S_d$ such that the representative exponent $\beta_j^\star = \min_{i \in S_j} \beta_i$ provides a tight lower bound for all β_i in that bin. We consider two strategies **Equal-Frequency Binning**, **Log-Adaptive (K-Means) Binning** (see Appendix C).

5.1 Evaluation and Baselines

We evaluate performance on standard benchmarks: CIFAR-10/100 (Krizhevsky, 2009), STL-10 (Coates et al., 2011), EuroSAT (Helber et al., 2019), MNIST (Lecun et al., 1998), FashionMNIST (Xiao et al., 2017), using both clustering metrics, Accuracy (ACC) and Normalized Mutual Information (NMI); and geometric graph metrics (Ratio Cut and Normalized Cut). We compare against: Logistic Regression (serving as a topological upper bound). K-Means, Spectral Clustering (Normalized Laplacian followed by K-Means), PRCut (Ghriss and Monteleoni, 2025), and our hypergeometric objective for RatioCut and NCut.

Experimental details, hyperparameters, and additional results are provided in Appendix B. We benchmark performance across 15 datasets using three foundation model embeddings: CLIP ViT-L/14 (Radford et al., 2021), DINOv2 (Oquab et al., 2024), and DINOv3-B (Siméoni et al., 2025) (Tables 1 and 2; CLIP in Table 3). We compare three probabilistic cut objectives: PRCut (Ghriss and Monteleoni, 2025), our hypergeometric RatioCut (H-RCut, Theorem 1), and hypergeometric NCut with Hölder binning (H-NCut, Theorem 2). All methods use the same linear model $\boldsymbol{P} = \mathrm{softmax}(\boldsymbol{W}_\theta \boldsymbol{x}/\tau)$ with edge-pair sampling and gradient mixing (Appendix E). We denote PRCut trained with our gradient mixing strategy as PRCut* since the original PRCut needs slower training schedule to avoid cluster collapse.

Key observations: Non-parametric SC generally achieves lower RCut/NCut values, but the parametric methods (PRCut*, H-RCut, H-NCut) consistently achieve higher ACC/NMI: the linear model acts as an implicit regularizer, preventing overfitting to noisy edges. Among the parametric methods, no single objective dominates: PRCut* excels on well-separated embeddings (Flowers, Pets), while H-NCut stands out on larger K where volume normalization matters (CIFAR-100, STL-10). The choice of embedding model is at least as important as the algorithm: DINOv2 achieves 98.6% on CIFAR-10 while CLIP reaches only 89.6%. (4) The graph quality metric Q (Section 5) strongly predicts downstream accuracy, confirming that the similarity graph construction is the primary bottleneck.

6 CONCLUSION

We presented a probabilistic framework for differentiable graph partitioning that provides tight upper bounds on the expected Normalized Cut via the Gauss hypergeometric function ${}_2F_1(-m, b; c; z)$. Our approach extends prior work (Ghriss and Monteleoni, 2025) in three ways: (1) we derive a tighter hypergeometric envelope for NCut and RatioCut, handling heterogeneous vertex degrees through a Hölder-product binning scheme; (2) we introduce a gradient mixing strategy that prevents cluster collapse without tuning loss weights; and (3) we provide full implementation of various methods with GPU-accelerated ${}_2F_1$ kernels (Triton and CUDA) with closed-form forward and backward passes, making the bound practical for first-order optimization and stochastic gradient.

Experiments on 15 datasets with three foundation model embeddings show that: the parametric H-Cut objectives consistently outperform non-parametric spectral clustering in clustering accuracy, while spectral clustering achieves lower raw cut values—the linear model acts as an implicit regularizer. The graph quality metric Q reliably predicts when graph-cut methods will succeed, confirming that the similarity graph is the primary bottleneck.

Future directions. Our framework takes a fixed similarity graph as input and optimizes cluster assignments. A natural extension is to *learn the similarity and clustering jointly*, closing the loop between graph construction and cut optimization. This would enable learning embeddings that are linearly separable by design. The parameterization from embeddings to cluster assignments (e.g., linear vs. nonlinear, shared vs. per-cluster) directly shapes the embedding space geometry. Such end-to-end training could connect differentiable cuts to metric learning and structured representation learning, where the graph encodes the desired invariances.

Limitations. Our analysis assumes independent Bernoulli assignments; modeling assignment dependencies (e.g., MRF couplings) remains open. The Hölder envelope is tightest when the degree distribution within each bin is concentrated; extremely heavy-tailed graphs may require adaptive binning. The ${}_2F_1$ evaluation is stable for small $a = -m$ (finite polynomial), but very large m might benefit from compensated summation.

References

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: a framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Bossard, L., Guillaumin, M., and Van Gool, L. (2014). Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, pages 446–461.

Cam, L. L. (1960). An approximation theorem for the Poisson binomial distribution. *Pacific Journal of Mathematics*, 10(4):1181 – 1197.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc.

Chambers, L. G. (1992). Hypergeometric functions and their applications, by james b. seaborn. pp 250. DM68. 1991. ISBN 3-540-97558-6 (springer). *The Mathematical Gazette*, 76(476):314–315.

Chen, S. X. and Liu, J. S. (1997). Statistical applications of the poisson-binomial and conditional bernoulli distributions. *Statistica Sinica*, 7(4):875–892.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations.

Cheng, G., Han, J., and Lu, X. (2017). Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. (2014). Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Coates, A., Lee, H., and Ng, A. Y. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 215–223.

Ghriss, A. and Monteleoni, C. (2025). Deep clustering via probabilistic ratio-cut optimization.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., kavukcuoglu, k., Munos, R., and Valko, M. (2020). Bootstrap your own latent - a new approach to self-supervised learning. In Larochelle, H., Ranzato, M.,

Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *Proceedings - 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006*, volume 2, pages 1735–1742.

He, K., Chen, X., Xie, S., Li, Y., Doll'ar, P., and Girshick, R. B. (2021). Masked autoencoders are scalable vision learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735.

Helber, P., Bischke, B., Dengel, A., and Borth, D. (2019). EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226.

Howard, J. (2019). Imagenette: A smaller subset of 10 easily classified classes from ImageNet. `https://github.com/fastai/imagenette`.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. (2013). Fine-grained visual classification of aircraft. Technical report.

Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In Dietterich, T., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press.

Nilsback, M.-E. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. (2024). Dinov2: Learning robust visual features without supervision.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. (2012). Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al. (2025). Dinov3. *arXiv preprint arXiv:2508.10104*.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332.

van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zhang, M., Hong, Y., and Balakrishnan, N. (2017). An algorithm for computing the distribution function of the generalized poisson-binomial distribution.

7 CHECKLIST

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Appendices

A Proofs

A.1 Generalized Poisson-Binomial

Lemma 4 (PGF of a weighted Bernoulli sum)**.** *Let $r_i \sim \text{Bernoulli}(\alpha_i)$ be independent and define $X \overset{def}{=} \sum_{i=1}^{m} \beta_i r_i$ with $\beta_i \in \mathbb{Z}_{\geq 0}$. Then the probability generating function $G_X(t) \overset{def}{=} \mathbb{E}[t^X]$ (for $|t| \leq 1$) is*

$$G_X(t) \;=\; \prod_{i=1}^{m} \big((1-\alpha_i) + \alpha_i\, t^{\beta_i}\big).$$

Proof. Since $X = \sum_i \beta_i r_i$ and $r_i \in \{0,1\}$, $t^X = \prod_{i=1}^{m} t^{\beta_i r_i}$. By independence,

$$G_X(t) \;=\; \mathbb{E}\left[\prod_{i=1}^{m} t^{\beta_i r_i}\right] = \prod_{i=1}^{m} \mathbb{E}\big[t^{\beta_i r_i}\big]\,.$$

For each i, $\mathbb{E}[t^{\beta_i r_i}] = (1-\alpha_i)t^0 + \alpha_i t^{\beta_i} = (1-\alpha_i) + \alpha_i t^{\beta_i}$. Multiplying the factors yields the claim. □

A.2 Integral Representation: Proof of Lemma 1

Lemma 1 (Integral representation)**.** *Define the integral $\mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta})$ as:*

$$\mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta}) \;\overset{def}{=}\; \int_0^1 t^{\,q-1} \prod_{i=1}^{m} \big(1 - \alpha_i + \alpha_i\, t^{\,\beta_i}\big)\, dt. \tag{7}$$

For any $q > 0$, we have:

$$\mathbb{E}\left[\frac{1}{q+\mathrm{x}}\right] \;=\; \mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{8}$$

Proof. The proof uses the integral representation of the reciprocal. For any $X > 0$, we have $\frac{1}{X} = \int_0^1 t^{X-1}\, dt$. Applying this with $X = q + \mathrm{x}$ (which is a.s. positive since $q > 0$ and $\mathrm{x} \geq 0$),

$$\frac{1}{q+\mathrm{x}} = \int_0^1 t^{\,q+\mathrm{x}-1}\, dt.$$

Taking expectations and using Tonelli's theorem (the integrand $t^{\,q+\mathrm{x}-1}$ is nonnegative on $[0,1]$),

$$\begin{aligned}\mathbb{E}\left[\frac{1}{q+\mathrm{x}}\right] &= \mathbb{E}\left[\int_0^1 t^{\,q+\mathrm{x}-1}\, dt\right] = \int_0^1 \mathbb{E}\big[t^{\,q+\mathrm{x}-1}\big]\, dt \\ &= \int_0^1 \mathbb{E}\big[t^{\,q-1}\, t^{\mathrm{x}}\big]\, dt = \int_0^1 t^{\,q-1}\, \mathbb{E}[t^{\mathrm{x}}]\, dt.\end{aligned}$$

Here $\mathbb{E}[t^{\mathrm{x}}]$ is the probability generating function (PGF) of x, denoted $G_{\mathrm{x}}(t)$. Substituting the PGF from Equation (6) gives

$$\mathbb{E}\left[\frac{1}{q+\mathrm{x}}\right] = \int_0^1 t^{\,q-1} G_{\mathrm{x}}(t)\, dt = \int_0^1 t^{\,q-1} \left[\prod_{i=1}^{m} \big(1 - \alpha_i + \alpha_i t^{\beta_i}\big)\right] dt.$$

□

A.3 Hypergeometric Bound: Proof of Theorem 1

Theorem 1 (Hypergeometric bound)**.** *Assume $\beta_i \equiv \beta > 0$. For any $q > 0$,*

$$\mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta}) \;\le\; \mathcal{H}_\beta(q; \bar{\alpha}, m) \tag{12}$$

where $\mathcal{H}_\beta(q; \bar{\alpha}, m) \overset{def}{=} \frac{1}{q}\, {}_2F_1\Big(-m,\, 1;\, \frac{q}{\beta} + 1;\, \bar{\alpha}\Big)$, and $\bar{\alpha} \overset{def}{=} \frac{1}{m}\sum_{i=1}^m \alpha_i$.

Proof. Assume $\beta_i \equiv \beta > 0$ and $q > 0$. Recall the definition of $\mathcal{I}(q, \boldsymbol{\alpha}, \beta)$:

$$\mathcal{I}(q, \boldsymbol{\alpha}, \beta) \overset{\text{def}}{=} \int_0^1 \left[\prod_{i=1}^m \big(1 - \alpha_i + \alpha_i t^\beta\big)\right] t^{q-1}\, dt.$$

For fixed $t \in [0, 1]$, the map $\alpha \mapsto \log\big(1 - \alpha + \alpha t^\beta\big)$ is concave (log of a positive affine function), hence by Jensen:

$$\sum_{i=1}^m \log\big(1 - \alpha_i + \alpha_i t^\beta\big) \;\le\; m \log\big(1 - \bar{\alpha} + \bar{\alpha}\, t^\beta\big), \quad \bar{\alpha} \overset{\text{def}}{=} \frac{1}{m}\sum_{i=1}^m \alpha_i.$$

Exponentiating and integrating gives:

$$I \;\le\; \int_0^1 \big(1 - \bar{\alpha} + \bar{\alpha}\, t^\beta\big)^m t^{q-1}\, dt \;=\; B.$$

Evaluate B via $u = t^\beta$ (so $dt = \frac{1}{\beta} u^{\frac{1}{\beta}-1} du$):

$$B = \frac{1}{\beta}\int_0^1 (1 - \bar{\alpha} + \bar{\alpha} u)^m\, u^{\frac{q}{\beta}-1}\, du = \frac{1}{\beta}\int_0^1 (1 - \bar{\alpha} v)^m (1 - v)^{\frac{q}{\beta}-1}\, dv,$$

with $v = 1 - u$. By Euler's integral for ${}_2F_1$ with $(a, b, c, z) = (-m, 1, \frac{q}{\beta} + 1, \bar{\alpha})$ (valid since $c > b > 0$),

$$B = \frac{1}{\beta} \cdot \frac{\Gamma(1)\Gamma(\frac{q}{\beta})}{\Gamma(\frac{q}{\beta} + 1)}\, {}_2F_1\Big(-m, 1; \tfrac{q}{\beta} + 1; \bar{\alpha}\Big) = \frac{1}{q}\, {}_2F_1\Big(-m, 1; \tfrac{q}{\beta} + 1; \bar{\alpha}\Big).$$

Therefore:

$$\boxed{I \;\le\; \frac{1}{q}\, {}_2F_1\Big(-m, 1; \tfrac{q}{\beta} + 1; \bar{\alpha}\Big)}.$$

□

A.4 AM-GM Gap: Proof of Proposition 2

Proposition 2 (Integrated AM–GM gap)**.** *Let $\beta_i \equiv \beta > 0$ and $\boldsymbol{\alpha} \in [0, 1]^m$. Define $h(t) = t^{q-1}\left(1 - \bar{\alpha} + \bar{\alpha}\, t^\beta\right)^m$ and:*

$$\underline{\Delta}(q, \boldsymbol{\alpha}) \overset{def}{=} \int_0^1 h(t)\Big(1 - e^{-\gamma(t)\mathrm{Var}(\boldsymbol{\alpha})}\Big)\, dt,$$
$$\overline{\Delta}(q, \boldsymbol{\alpha}) \overset{def}{=} \int_0^1 h(t)\Big(1 - e^{-\theta(t)\mathrm{Var}(\boldsymbol{\alpha})}\Big)\, dt,$$

with $\gamma(t) \overset{def}{=} \frac{m}{2}(1 - t^\beta)^2$ and $\theta(t) \overset{def}{=} \gamma(t)/t^{2\beta}$.

We have the following gap:

$$\underline{\Delta}(q, \boldsymbol{\alpha}) \le \mathcal{H}_\beta(q; \bar{\alpha}, m) \;-\; \mathcal{I}(q, \boldsymbol{\alpha}, \boldsymbol{\beta}) \le \overline{\Delta}(q, \boldsymbol{\alpha}), \tag{13}$$

with $\mathrm{Var}(\boldsymbol{\alpha})$ computed under uniform sampling of the graph nodes, and equality throughout iff $\mathrm{Var}(\boldsymbol{\alpha}) = 0$.

Proof. Fix $\tau \in [0,1]$ and set $c \stackrel{\text{def}}{=} 1-\tau$. Define $f_\tau(\alpha) \stackrel{\text{def}}{=} \log(1-\alpha+\alpha\tau) = \log(1-c\,\alpha)$, a concave function on $[0,1]$ with:

$$f''_\tau(\alpha) = -\frac{c^2}{(1-c\,\alpha)^2} \;\in\; \Big[-\frac{c^2}{\tau^2}, -c^2\Big].$$

Thus $-f_\tau$ is θ_τ-smooth with $\theta_\tau = c^2/\tau^2$ and γ_τ-strongly convex with $\gamma_\tau = c^2$ on $[0,1]$. By the standard Jensen two-sided bound for twice-differentiable concave functions:

$$\frac{\gamma_\tau}{2}\operatorname{Var}(\boldsymbol{\alpha}) \;\le\; f_\tau(\bar\alpha) - \frac{1}{m}\sum_{i=1}^m f_\tau(\alpha_i) \;\le\; \frac{\theta_\tau}{2}\operatorname{Var}(\boldsymbol{\alpha}). \tag{31}$$

Exponentiating Equation (31) yields the *pointwise* multiplicative AM–GM control (for any fixed $\tau \in [0,1]$):

$$\exp\Big(\tfrac{m}{2}\gamma_\tau \operatorname{Var}(\boldsymbol{\alpha})\Big) \;\le\; \frac{\big(1-\bar\alpha+\bar\alpha\,\tau\big)^m}{\prod_{i=1}^m(1-\alpha_i+\alpha_i\tau)} \;\le\; \exp\Big(\tfrac{m}{2}\theta_\tau \operatorname{Var}(\boldsymbol{\alpha})\Big), \tag{32}$$

with equality iff $\operatorname{Var}(\boldsymbol{\alpha}) = 0$ (or $\tau \in \{0,1\}$). Equivalently, the *additive* gap satisfies;

$$\big(1-\bar\alpha+\bar\alpha\tau\big)^m\Big(1-e^{-\frac{m}{2}\theta_\tau \operatorname{Var}(\boldsymbol{\alpha})}\Big) \ge \big(1-\bar\alpha+\bar\alpha\tau\big)^m - \prod_{i=1}^m(1-\alpha_i+\alpha_i\tau) \ge \big(1-\bar\alpha+\bar\alpha\tau\big)^m\Big(1-e^{-\frac{m}{2}\gamma_\tau \operatorname{Var}(\boldsymbol{\alpha})}\Big). \tag{33}$$

Set $\tau = t^\beta$ (the theorem's common exponent) and multiply Equation (33) by t^{q-1}, then integrate $t \in [0,1]$. Using:

$$I(q;\boldsymbol{\alpha},\beta) \;\stackrel{\text{def}}{=}\; \int_0^1 \Big[\prod_{i=1}^m\big(1-\alpha_i+\alpha_i t^\beta\big)\Big] t^{q-1}\,dt, \quad B_{\text{AMGM}}(q) \;\stackrel{\text{def}}{=}\; \int_0^1 \big(1-\bar\alpha+\bar\alpha\,t^\beta\big)^m t^{q-1}\,dt,$$

we obtain the deterministic two-sided bound:

$$\underline{\Delta}(q) \;\le\; B_{\text{AMGM}}(q) \;-\; I(q;\boldsymbol{\alpha},\beta) \;\le\; \overline{\Delta}(q), \tag{34}$$

with $\gamma(t) \stackrel{\text{def}}{=} (1-t^\beta)^2$, $\theta(t) \stackrel{\text{def}}{=} \gamma(t)/t^{2\beta}$, $B_{\text{AMGM}}(q) \stackrel{\text{def}}{=} \mathcal{H}_\beta(q,\bar\alpha,\beta)$ and:

$$\underline{\Delta}(q) \stackrel{\text{def}}{=} \int_0^1 t^{q-1}\big(1-\bar\alpha+\bar\alpha\,t^\beta\big)^m\Big(1-e^{-\frac{m}{2}\,\gamma(t)\operatorname{Var}(\boldsymbol{\alpha})}\Big)\,dt,$$
$$\overline{\Delta}(q) \stackrel{\text{def}}{=} \int_0^1 t^{q-1}\big(1-\bar\alpha+\bar\alpha\,t^\beta\big)^m\Big(1-e^{-\frac{m}{2}\,\theta(t)\operatorname{Var}(\boldsymbol{\alpha})}\Big)\,dt,$$

Using $1-e^{-x} \le x$ gives the simple upper bound:

$$0 \;\le\; B_{\text{AMGM}}(q) - I(q;\boldsymbol{\alpha},\beta) \;\le\; \frac{m}{2}\operatorname{Var}(\boldsymbol{\alpha})\int_0^1 t^{q-1}\big(1-\bar\alpha+\bar\alpha\,t^\beta\big)^m\theta(t)\,dt, \tag{35}$$

and $1-e^{-x} \ge \frac{x}{1+x}$ yields a corresponding explicit lower bound. The bounds in Equation (34) are tight iff $\operatorname{Var}(\boldsymbol{\alpha}) = 0$, in which case $B_{\text{AMGM}}(q) = I(q;\boldsymbol{\alpha},\beta)$. □

A.5 Proofs for zero-aware gap

Proposition 3 (Zero-aware AM-GM gap)**.** *Under the assumptions of Proposition 2 with common $\beta > 0$, a zero-aware replacement for the simple upper bound of Corollary 1 is:*

$$\widetilde{\mathcal{A}}(q,\boldsymbol{\alpha},\beta,m) \stackrel{\text{def}}{=} \frac{m}{2}\operatorname{Var}^{\omega_0}(\boldsymbol{\alpha})\mathcal{A}\big(q;\bar\alpha,\beta,m\big). \tag{17}$$

Proof. Fix $t \in [0,1]$ and write $\tau \overset{\text{def}}{=} t^\beta \in [0,1]$. Let $f_\tau(\alpha) \overset{\text{def}}{=} \log(1-\alpha+\alpha\tau)$, which is concave on $[0,1]$ with

$$-f''_\tau(\alpha) = \frac{(1-\tau)^2}{(1-(1-\tau)\alpha)^2} \in \Big[(1-\tau)^2,\, (1-\tau)^2/\tau^2\Big].$$

Set the zero-aware weights $\lambda_i \overset{\text{def}}{=} \omega_0(\alpha_i)/\Omega$ and denote the ω_0–weighted mean and variance by $\bar\alpha^{\omega_0} = \sum_i \lambda_i \alpha_i$ and $\mathrm{Var}^{\omega_0}(\boldsymbol{\alpha}) = \sum_i \lambda_i(\alpha_i - \bar\alpha^{\omega_0})^2$ (with the usual convention $\mathrm{Var}^{\omega_0} = 0$ if $\Omega = 0$).

By weighted Jensen for the concave f_τ:

$$\sum_{i=1}^m \lambda_i f_\tau(\alpha_i) \;\le\; f_\tau(\bar\alpha^{\omega_0}).$$

The standard second-order (weighted) Jensen gap bound gives:

$$f_\tau(\bar\alpha^{\omega_0}) - \sum_{i=1}^m \lambda_i f_\tau(\alpha_i) \;\le\; \frac{\theta_\tau}{2}\,\mathrm{Var}^{\omega_0}(\boldsymbol{\alpha}), \qquad \theta_\tau \overset{\text{def}}{=} \frac{(1-\tau)^2}{\tau^2}.$$

Multiplying by Ω and exponentiating yields the *pointwise* zero-aware AM–GM control:

$$0 \;\le\; \bar Y(t) - Y(t) \;\le\; \bar Y(t)\Big(1 - e^{-\frac{m}{2}\theta_\tau\,\mathrm{Var}^{\omega_0}(\boldsymbol{\alpha})}\Big),$$

where $Y(t) = \prod_i (1-\alpha_i+\alpha_i t^\beta)$ and $\bar Y(t) = (1-\bar\alpha+\bar\alpha t^\beta)^m$ (note: we keep the envelope centered at the *plain* mean $\bar\alpha$ as in Theorem 1).

Using $1 - e^{-x} \le x$ and integrating against t^{q-1} gives:

$$0 \;\le\; B_{\mathrm{AMGM}}(q) - \mathcal{I}(q,\boldsymbol{\alpha},\boldsymbol{\beta}) \;\le\; \frac{m}{2}\,\mathrm{Var}^{\omega_0}(\boldsymbol{\alpha}) \int_0^1 t^{q-1}\bar Y(t)\,\frac{(1-t^\beta)^2}{t^{2\beta}}\,dt.$$

By differentiating under the integral sign and the binomial identity $\sum_{r=0}^2 \binom{2}{r}(-1)^r t^{r\beta} = (1-t^\beta)^2$, one obtains the identity (derived in the main text):

$$\int_0^1 t^{q-1}\bar Y(t)\,\frac{(1-t^\beta)^2}{t^{2\beta}}\,dt \;=\; \frac{\partial}{\partial\bar\alpha}\sum_{r=0}^2 \binom{2}{r}(-1)^r\,\mathcal{H}_\beta\big(q+r\beta;\bar\alpha,m\big) \;=\; \widetilde{\mathcal{A}}\big(q;\bar\alpha,m\big),$$

valid for $q > 2\beta$ by Euler's integral (and for all $q > 0$ by analytic continuation). Combining the two displays yields

$$B_{\mathrm{AMGM}}(q) - \mathcal{I}(q,\boldsymbol{\alpha},\boldsymbol{\beta}) \;\le\; \frac{m}{2}\,\mathrm{Var}^{\omega_0}(\boldsymbol{\alpha})\,\widetilde{\mathcal{A}}\big(q;\bar\alpha,m\big) \;\overset{\text{def}}{=}\; \mathbb{A}(q,\boldsymbol{\alpha},m),$$

which is exactly Equation (17). The bound is zero-aware since $\omega_0(0) = 0$ removes inactive coordinates from Var^{ω_0}; it is tight when $\mathrm{Var}^{\omega_0}(\boldsymbol{\alpha}) = 0$ (i.e., the ω_0–weighted dispersion vanishes), in which case $\bar Y(t) \equiv Y(t)$ and equality holds. □

A.6 Hölder product bound for heterogeneous exponents

Let $\boldsymbol{\beta} = (\beta_i)_{i=1}^m$ and take d distinct values $\{b_1,\dots,b_d\} \subset (0,\infty)$, and partition indices by $S_j \overset{\text{def}}{=} \{i : \beta_i = b_j\}$ with sizes $m_j \overset{\text{def}}{=} |S_j|$ and $\sum_{j=1}^d m_j = m$. Define the group means:

$$\bar\alpha_j \;\overset{\text{def}}{=}\; \frac{1}{m_j}\sum_{i\in S_j}\alpha_i \in [0,1].$$

Recall the objective integral:

$$\mathcal{I}(q,\boldsymbol{\alpha},\boldsymbol{\beta}) \;\overset{\text{def}}{=}\; \int_0^1 \left[\prod_{i=1}^m \big(1-\alpha_i+\alpha_i t^{\beta_i}\big)\right] t^{q-1}\,dt, \qquad q > 0.$$

Lemma 5 (Hölder–binned envelope)**.** *With the notation above,*

$$\mathcal{I}(q,\boldsymbol{\alpha},\boldsymbol{\beta}) \;\le\; \prod_{j=1}^{d}\Big[\;\mathcal{H}_{b_j}\big(q;\bar{\alpha}_j,m\big)\;\Big]^{m_j/m},$$

where $\mathcal{H}_\beta(q;\bar{\alpha},m)=\frac{1}{q}\,{}_2F_1(-m,1;\frac{q}{\beta}+1;\bar{\alpha})$ *is the common–*β *envelope from Theorem 1.*

Proof. For each group S_j (fixed $t\in[0,1]$):

$$P_j(t) \;\overset{\text{def}}{=}\; \prod_{i\in S_j}\big(1-\alpha_i+\alpha_i t^{b_j}\big) \;\le\; \big(1-\bar{\alpha}_j+\bar{\alpha}_j t^{b_j}\big)^{m_j} \quad \text{by AM–GM.}$$

Multiplying over j gives:

$$\prod_{i=1}^{m}\big(1-\alpha_i+\alpha_i t^{\beta_i}\big) \;\le\; \prod_{j=1}^{d}\big(1-\bar{\alpha}_j+\bar{\alpha}_j t^{b_j}\big)^{m_j}.$$

Hence:

$$\mathcal{I}(q,\boldsymbol{\alpha},\boldsymbol{\beta}) \;\le\; \int_0^1 \prod_{j=1}^{d}\big(1-\bar{\alpha}_j+\bar{\alpha}_j t^{b_j}\big)^{m_j}\,t^{q-1}\,dt.$$

Let $w_j \overset{\text{def}}{=} m_j/m$ and split $t^{q-1}=\prod_{j=1}^{d} t^{(q-1)w_j}$. Set:

$$g_j(t) \;\overset{\text{def}}{=}\; \big(1-\bar{\alpha}_j+\bar{\alpha}_j t^{b_j}\big)^{m_j}\,t^{(q-1)w_j}.$$

Choose exponents $p_j \overset{\text{def}}{=} \frac{m}{m_j}>1$, so that $\sum_{j=1}^{d}\frac{1}{p_j}=\sum_j\frac{m_j}{m}=1$. By Hölder's inequality for products,

$$\int_0^1\prod_{j=1}^{d} g_j(t)\,dt \;\le\; \prod_{j=1}^{d}\Big(\int_0^1 |g_j(t)|^{p_j}\,dt\Big)^{1/p_j}.$$

But $g_j^{p_j}(t)=\big(1-\bar{\alpha}_j+\bar{\alpha}_j t^{b_j}\big)^{m}\,t^{q-1}$, hence:

$$\int_0^1\prod_{j=1}^{d} g_j(t)\,dt \;\le\; \prod_{j=1}^{d}\Big(\int_0^1\big(1-\bar{\alpha}_j+\bar{\alpha}_j t^{b_j}\big)^{m}\,t^{q-1}\,dt\Big)^{m_j/m}.$$

For each j, the inner integral equals:

$$\int_0^1\big(1-\bar{\alpha}_j+\bar{\alpha}_j t^{b_j}\big)^{m}\,t^{q-1}\,dt \;=\; \mathcal{H}_{b_j}\big(q;\bar{\alpha}_j,m\big) \;=\; \frac{1}{q}\,{}_2F_1\Big(-m,1;\tfrac{q}{b_j}+1;\bar{\alpha}_j\Big),$$

by the same change-of-variables/Euler-integral used in Theorem 1 (valid for $q>0$, $b_j>0$). Combining (i)–(iii) yields the stated bound. □

Remarks. If exponents are grouped into *bins* with ranges $[b_j^{\leftarrow},b_j^{\rightarrow}]$ rather than singletons, the same proof holds after replacing b_j by any representative $b_j^{\leftarrow}\le\beta_i$ for $i\in S_j$, preserving the upper-bound direction. The bound is a weighted geometric mean of d hypergeometric envelopes, with weights m_j/m, and avoids collapsing all exponents to a single conservative value.

Temperature–annealed probabilities tighten the zero-aware gap. Parameterize the assignment probabilities from logits Z at temperature $\tau>0$:

$$p_{i\ell}(\tau)=\operatorname{softmax}\Big(\frac{Z_{i\ell}}{\tau}\Big)\quad\text{(multiclass)}\qquad\text{or}\qquad p_{i\ell}(\tau)=\sigma\Big(\frac{Z_{i\ell}}{\tau}\Big)\quad\text{(binary).}$$

As $\tau\downarrow 0$, $p_{i\ell}(\tau)\to\{0,1\}$ elementwise. Our zero-aware gap in bin j uses weights $\omega(x)$ (e.g., $\omega(x)=x(1-x)$ or more generally $\omega(x)=x^a$, $a\in[1,2]$), the weighted mean $\mu=\bar{p}^{\omega}_{\ell j}$, and dispersion $V=\mathrm{Var}^{\omega}_{\ell j}(p)$ (Appendix G.5). Two facts hold:

1. **Vanishing weights at the extremes.** For the choices above, $\omega(0) = \omega(1) = 0$ and $0 \le \omega(x) \le \frac{1}{4}$, so for almost-hard assignments $p_{i\ell}(\tau) \in \{0, 1\}$ one has $\Omega_{\ell j}(\tau) = \sum_{r \in S_{\ell_j}} \omega\big(p_{r\ell}(\tau)\big) \xrightarrow[\tau \downarrow 0]{} 0$ and $V(\tau) \xrightarrow[\tau \downarrow 0]{} 0$.

2. **Zero-aware gap collapses.** The (per-bin) gap upper bound

$$\Gamma^{\mathrm{ewa}}_{\ell j}(q) = \frac{m_\ell}{2}\, w_{\ell j}\, V\, \widetilde{\mathcal{A}}_{b_j}\big(q; \bar{p}_{\ell j}, m_\ell\big)$$

   vanishes as $\tau \downarrow 0$ because $V(\tau) \to 0$ while $\widetilde{\mathcal{A}}_{b_j}$ stays bounded (finite polynomial in $\bar{p}$). Hence the total objective's slack from the AM–GM step is driven to zero by temperature annealing.

In contrast, the Hölder envelope terms depend on the *bin means* $\bar{p}_{\ell j}(\tau) = m_{\ell j}^{-1} \sum_{i \in S_{\ell j}} p_{i\ell}(\tau)$ and thus are insensitive to per-bin *dispersion*. Annealing shrinks only the gap (and any other dispersion-based penalties), tightening the overall upper bound without altering the envelope's functional form.

Combining the relaxed envelope and annealing gives the practical surrogate

$$\mathcal{U}_{\mathrm{relax}}(P; \tau) \;=\; \underbrace{\prod_{j=1}^{d} \left[\frac{1}{q}\, {}_2F_1\big(-m, 1; 2; \bar{p}_{\ell j}(\tau)\big)\right]^{w_{\ell j}}}_{\text{uniform-}c\text{ Hölder envelope}} \;+\; \rho \sum_j \underbrace{\frac{m_\ell}{2}\, w_{\ell j}\, V^{\omega}_{\ell j}(P(\tau))\, \widetilde{\mathcal{A}}_{b_j}\big(q; \bar{p}_{\ell j}(\tau), m_\ell\big)}_{\text{zero-aware gap}},$$

where $w_{\ell j} = m_{\ell j}/m_\ell$ and $\rho \ge 0$. As $\tau \downarrow 0$, $V^{\omega}_{\ell j}(P(\tau)) \to 0$ and the gap vanishes, while the envelope is upper-bounded uniformly by the simple $c = 2$ hypergeometric polynomial in the bin means.

B Experimental Framework

B.1 Graph Construction

Given a dataset $X \in \mathbb{R}^{N \times D}$ consisting of N samples with dimension D, we construct the affinity graph as follows:

1. **k-NN Graph:** We compute the k-nearest neighbors for each sample based on Euclidean distance, with $k = 50$.
2. **Adjacency Matrix:** A sparse adjacency matrix W is initialized such that $W_{ij} = 1$ if $j \in \mathcal{N}(i)$ or $i \in \mathcal{N}(j)$.
3. **Gaussian Kernel:** The binary edge weights are smoothed using a Gaussian kernel:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}\right), \tag{36}$$

   where σ_i is set to the average distances of neighbors of node i.
4. **Symmetrization:** We ensure the graph is undirected by updating $W \leftarrow (W + W^\top)/2$.

B.2 Probabilistic Assignments and Architecture

We employ a lightweight neural network f_θ (implemented as a single linear layer) to map input features x_i to cluster logits $z_i \in \mathbb{R}^K$, where K is the target number of clusters. Soft cluster assignments $P \in \mathbb{R}^{N \times K}$ are obtained via the softmax function:

$$p_{ik} = \frac{e^{z_{ik}}}{\sum_{j=1}^{K} e^{z_{ij}}}. \tag{37}$$

B.3 Optimization Objective

Our method **H-Cut** objective minimizes an objective that combines a pairwise similarity loss with a hypergeometric scaling term (to approximate RCut or NCut) and an entropy regularization term.

Pairwise Similarity Loss. For a batch of sampled edges $\mathcal{B} \subset \mathcal{E}$ with weights w_{ij}, we minimize the cross-entropy between connected nodes to encourage smoothness over the graph:

$$\mathcal{L}_{\text{sim}} = \frac{1}{\sum_{(i,j)\in\mathcal{B}} w_{ij}} \sum_{(i,j)\in\mathcal{B}} w_{ij} \sum_{k=1}^{K} -p_{ik} \log p_{jk}. \tag{38}$$

Hypergeometric Scaling. To enforce balanced partitions; acting as a differentiable proxy for the cut volume constraints; we scale the similarity loss using the Gauss hypergeometric function ${}_2F_1$. We maintain a moving average of the mean cluster probabilities, denoted α_k. The scaling factor S_k for cluster k is defined as:

$$S_k = {}_2F_1(-m, b, c, \alpha_k), \tag{39}$$

where we set $m = 512$, $b = 1$, and $c = 2$ for H-RCut. This term penalizes clusters that grow disproportionately large.

Entropy Regularization. To prevent trivial solutions (collapse to a single cluster), we maximize the entropy of the marginal cluster distribution $\bar{p}$, where $\bar{p}_k = \frac{1}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} p_{ik}$:

$$\mathcal{L}_{\text{bal}} = -\mathcal{H}(\bar{p}) = \sum_{k=1}^{K} \bar{p}_k \log \bar{p}_k. \tag{40}$$

Total Loss. The final objective balances the scaled similarity and the regularization:

$$\mathcal{L} = \sum_{k=1}^{K} (\mathcal{L}_{\text{sim},k} \cdot S_k) + \lambda \mathcal{L}_{\text{bal}}. \tag{41}$$

We utilize a gradient mixing strategy to balance the magnitude of gradients between the two terms for stable optimization.

B.4 Training Algorithm

The model is trained via Stochastic Gradient Descent (SGD) by sampling edges from the pre-computed sparse graph. The procedure is summarized below:

Algorithm 1 H-Cut Training Loop

Input: Features X, Clusters K, Pre-computed Edge List E
Initialize: Linear model f_θ, Moving average $\alpha \leftarrow \mathbf{1}/K$
while not converged **do**
  Sample batch of edges (i, j) from E
  Compute assignments $p_i = \text{softmax}(f_\theta(x_i))$, $p_j = \text{softmax}(f_\theta(x_j))$
  Compute pairwise similarity $\mathcal{L}_{\text{sim}}$
  Update α using batch mean assignments
  Compute scaling factors S_k
  Compute balance penalty $\mathcal{L}_{\text{bal}}$
  Update θ to minimize weighted $\mathcal{L}_{\text{sim}} + \mathcal{L}_{\text{bal}}$
Output: Hard assignments $C_i = \arg\max_k p_{ik}$

C Binning Strategies

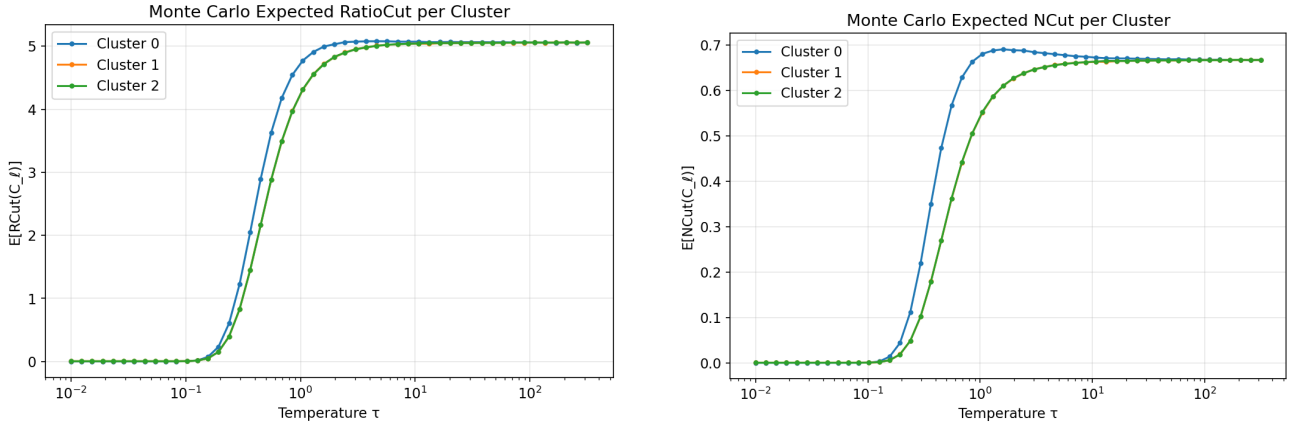
- **Equal-Frequency Binning:** We sort the vertices by their weights β_i and partition them into d bins of equal size $|S_j| \approx n/d$. This approach ensures that each term in the Hölder product contributes equally to the total envelope. However, for power-law degree distributions common in real-world graphs, the final bin often spans a massive range of values (e.g., covering both moderate and hub nodes). This high intra-bin variance causes the representative $\beta_j^\star$ to drastically underestimate the convexity for hub nodes, loosening the bound.
- **Log-Adaptive (K-Means) Binning:** To address the heavy-tailed nature of degree distributions, we apply K-Means clustering to the log-transformed weights $x_i = \log(\beta_i)$. This strategy groups vertices based on geometric proximity, ensuring that nodes are binned by their order of magnitude rather than population count. This aligns with the colinearity condition of the Hölder inequality: since the shape of the generating function t^β changes most rapidly for small β and stabilizes for large β, clustering in log-space minimizes the shape distortion between the true node functions and the bin-wise proxy, yielding a significantly tighter upper bound.

D Simulation Details

We verify our bounds on a synthetic three-helix dataset with unbalanced clusters ($|C_1|$=200, $|C_2|$=$|C_3|$=400) and a 50-NN RBF graph. Soft assignments are parameterized as $\boldsymbol{P}_{i\ell} = \text{softmax}(z_{i\ell}/\tau)$ with random logits, varying τ from 10^{-2} (hard) to 10^3 (uniform). The expected cut is estimated via Monte Carlo (MC) with 1000 samples per temperature.

E Gradient Mixing Strategy

A key challenge in optimizing probabilistic graph cuts is balancing the cut objective against a regularizer that prevents cluster collapse. Without regularization, the optimizer drives small clusters to zero volume, producing degenerate partitions.



(a) MC expected RCut per cluster

(b) MC expected NCut per cluster

Figure 4: Monte Carlo expected cut values per cluster as a function of temperature τ. The unbalanced cluster (C_0, 200 nodes) has higher expected RCut than the balanced ones (C_1, C_2, 400 nodes each), while NCut normalizes by volume and reduces this gap.

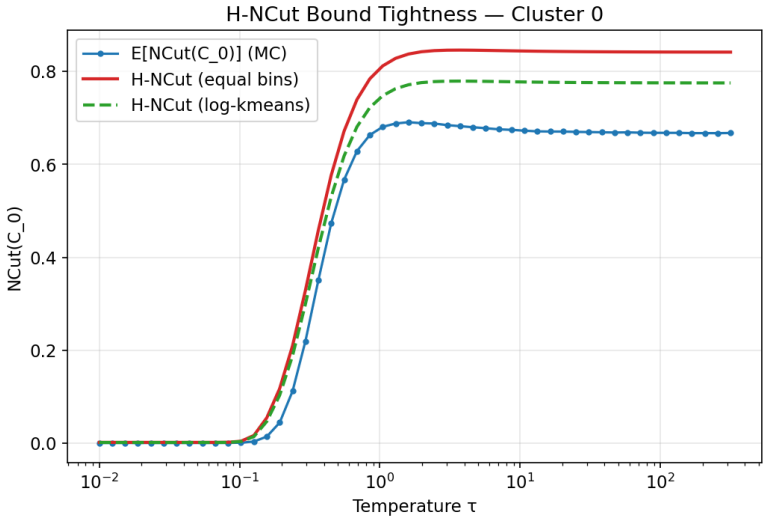


Figure 5: H-NCut bound tightness for the smallest cluster (C_0): comparison of equal-frequency and log-adaptive binning against the MC estimate. Log-adaptive binning tracks the MC curve more closely across all temperatures.

The original PRCut (Ghriss and Monteleoni, 2025) addresses this through its analytical gradient, which includes an implicit regularization via the $-\text{cut}_\ell/\bar{p}_\ell^2 n$ term. However, this term vanishes as clusters collapse ($\bar{p}_\ell \to 0$), making the original PRCut prone to empty clusters in practice; particularly on datasets with many classes ($K \geq 10$).

We introduce a *gradient mixing* strategy that decouples the cut and balance objectives:

$$\mathcal{L} = \mathcal{L}_{\text{cut}} + \mathcal{L}_{\text{balance}}, \quad \mathcal{L}_{\text{balance}} = -\sum_{\ell=1}^{K} \bar{p}_\ell \log \bar{p}_\ell, \tag{42}$$

where $\bar{p}_\ell = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} P_{i\ell}$ is the batch-average cluster proportion and $\mathcal{L}_{\text{balance}}$ is the negative entropy, encouraging uniform cluster sizes. Rather than adding the losses directly (which leads to one gradient dominating), we normalize each gradient independently before combining:

$$g_\theta = \frac{\nabla_\theta \mathcal{L}_{\text{cut}}}{\|\nabla_\theta \mathcal{L}_{\text{cut}}\|} + \frac{\nabla_\theta \mathcal{L}_{\text{balance}}}{\|\nabla_\theta \mathcal{L}_{\text{balance}}\|}. \tag{43}$$

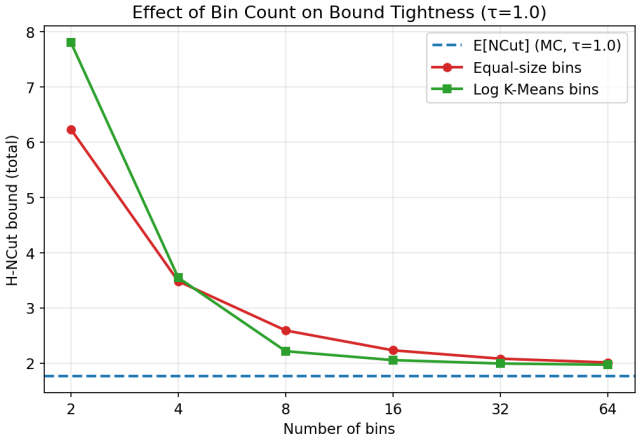


Figure 6: Effect of the number of bins d on H-NCut bound tightness at τ=1. Log-adaptive binning converges faster and achieves a tighter bound with fewer bins.

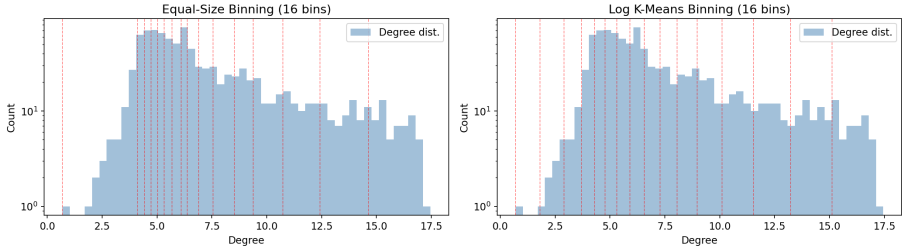


Figure 7: Degree distribution of the helix graph with bin boundaries (red dashed) for equal-frequency (left) and log-adaptive K-Means (right) binning with d=16. Log-adaptive binning places more boundaries in the dense region of the distribution.

This ensures both objectives contribute equally in gradient direction, regardless of their relative magnitudes. We refer to PRCut trained with this strategy as PRCut*.

F CLIP ViT-L/14 Results

Table 3: **CLIP ViT-L/14 embeddings (Radford et al., 2021).** CLIP's text-aligned representations yield high Q for coarse categories but struggle with fine-grained distinctions (Flowers, CUB: Q=0.28). SC = non-parametric Spectral Clustering. Best per row in **bold**.

			Spectral Clustering				PRCut*				H-RCut				H-NCut			
Dataset	K	Q	ACC%	NMI%	RCut ↓	NCut ↓	ACC%	NMI%	RCut ↓	NCut ↓	ACC%	NMI%	RCut ↓	NCut ↓	ACC%	NMI%	RCut ↓	NCut ↓
CIFAR-10 (Krizhevsky, 2009)	10	0.92	87.4	**89.8**	**18**	**0.40**	**89.6**	87.6	37.5	0.82	83.3	82.2	40.7	0.94	83.9	82.7	40.8	0.97
CIFAR-100 (Krizhevsky, 2009)	100	0.61	**61.9**	**71.8**	**995**	**22.6**	26.0	49.7	2307	73.2	57.6	70.8	1511	35.9	57.6	70.1	1566	36.9
STL-10 (Coates et al., 2011)	10	0.96	95.8	95.8	**9.0**	**0.22**	**99.8**	**99.4**	9.8	0.24	**99.8**	**99.4**	9.8	0.24	**99.8**	**99.4**	9.8	0.24
EuroSAT (Helber et al., 2019)	10	0.81	72.9	**73.8**	**43**	**1.0**	**78.3**	70.8	68.0	1.6	76.9	73.1	64.4	1.5	76.9	73.4	62.5	1.4
Imagenette (Howard, 2019)	10	0.98	99.7	99.0	**3.2**	**0.07**	99.7	99.2	3.6	0.08	**99.8**	**99.3**	3.5	0.08	**99.8**	**99.3**	3.4	0.08
Fashion-MNIST (Xiao et al., 2017)	10	0.76	**71.9**	**73.0**	**29**	**0.66**	65.6	63.8	55.1	1.2	63.7	61.9	66.6	1.5	69.4	67.0	60.8	1.4
MNIST (Lecun et al., 1998)	10	0.92	**80.9**	**78.3**	**19**	**0.42**	69.4	67.1	53.4	1.2	73.5	69.7	47.1	1.1	68.0	67.5	54.9	1.3
Pets (Parkhi et al., 2012)	37	0.60	74.5	81.2	**566**	**13.8**	81.4	85.3	567	13.9	81.3	85.4	604	14.4	**83.1**	**86.0**	603	14.4
Flowers-102 (Nilsback and Zisserman, 2008)	102	0.28	**81.3**	**91.0**	2728	**73.7**	60.0	84.0	**2526**	80.2	70.5	85.9	3229	77.4	69.4	84.3	3081	77.5
Food-101 (Bossard et al., 2014)	101	0.76	**80.6**	**85.5**	**611**	**13.6**	63.9	74.6	1259	37.8	78.3	83.9	908	20.9	75.5	83.2	957	21.8
RESISC-45 (Cheng et al., 2017)	45	0.77	76.2	82.9	**240**	**5.9**	**82.3**	**84.1**	303	7.4	77.8	82.1	324	8.0	76.8	81.9	336	8.3
DTD (Cimpoi et al., 2014)	47	0.38	**54.3**	60.9	**860**	**21.6**	52.7	61.1	938	25.5	54.1	61.1	964	24.5	53.9	**61.2**	995	24.2
GTSRB (Stallkamp et al., 2012)	43	0.78	**68.9**	**75.7**	**254**	**6.3**	64.4	68.9	469	12.3	62.5	67.9	532	12.8	62.0	67.8	525	12.6
CUB-200 (Wah et al., 2011)	200	0.28	**61.9**	**81.0**	5509	**140**	22.7	60.6	**4405**	179	52.3	77.5	6422	151	53.3	77.9	6404	150
FGVC-Aircraft (Maji et al., 2013)	100	0.25	37.4	58.8	**2270**	**56.4**	36.5	59.9	2609	64.3	37.9	61.8	2742	65.9	**39.0**	**62.2**	2748	64.4

Implementation Details. We implement the framework in PyTorch. Optimization is performed using AdamW with a learning rate of 10^{-4} and weight decay of 10^{-4}. We utilize a large batch size of 8,192 edges on a single NVIDIA GPU (A100). In our setup, the memory usage does not exceed 3GB of VRAM.

G Forward-Backward algorithms

Both envelopes (AM–GM/common–β and Hölder/binning) and the zero-aware gap are differentiable in the assignment parameters $\boldsymbol{\alpha}$ (hence in $\boldsymbol{P}$). The backward (pass) gradients were derived in §G.3, §G.4, and §G.5. Here we describe the *forward* computation and give robust, $O(m)$, numerically stable procedures for the truncated hypergeometric terms. Throughout we use:

$$\frac{d}{dz}{}_2F_1(a,b;c;z) = \frac{ab}{c}\,{}_2F_1(a+1,b+1;c+1;z),$$

and the fact that for $a=-m$ (or $-m+1$) the series *truncates* (finite polynomial).

G.1 Efficient computation of ${}_2F_1$

For $a=-m$ and $b=1$, the Gauss hypergeometric reduces to a degree-m polynomial:

$${}_2F_1(-m,1;c;z) = \sum_{k=0}^{m} \frac{(-m)_k(1)_k}{(c)_k}\frac{z^k}{k!} = \sum_{k=0}^{m}(-1)^k\binom{m}{k}\frac{z^k}{(c)_k}, \quad z\in[0,1].$$

Although the series is finite (exact after $k=m$), in practice many tails are negligible. We use an early-exit rule at index K when:

$$|t_{K+1}| < \varepsilon_{\text{rel}}\max\{|S_K|,\delta_{\text{abs}}\},$$

where S_K is the current partial sum, ε_{rel} a relative tolerance, and δ_{abs} a floor for tiny values (e.g., machine epsilon scaled). This is safe because the remaining $m-K$ terms are alternating and (empirically) rapidly shrinking for $z\in[0,1]$; for reproducibility one can cap $K\le m$.

At $z=0$ the value is 1; near $z=1$ we rely on Horner/compensation to manage cancellation. For large c (e.g., the $c=2$ relaxed envelope of §G.6), coefficients become very benign: ${}_2F_1(-m,1;2;z)=\sum_{k=0}^{m}(-1)^k\binom{m}{k}z^k/(k+1)!$.

Applying AM–GM *within* bins S_j of equal β (i.e., $\beta_i=b_j$) gives bin-wise polynomials after replacing m by m_j and $\bar\alpha$ by $\bar\alpha_j$; the product envelope's slack is then controlled by within-bin dispersions $\{\mathrm{Var}_j(\boldsymbol{\alpha})\}_j$.

G.2 The forward pass (Hölder envelope + derivatives)

We now give a concrete forward routine that returns the Hölder envelope $B_{\text{Hölder}}$ and the two hypergeometric building blocks needed for the backward pass (the ratio "H_j'/H_j" in the gradient). The algorithm evaluates:

$$B_{\text{Hölder}} = \prod_{j=1}^{d}\Big[\mathcal{H}_{b_j}(q;\bar\alpha_j,m)\Big]^{m_j/m}, \qquad \mathcal{H}_{b_j}(q;\bar\alpha_j,m) = \frac{1}{q}\,{}_2F_1\Big(-m,1;c_j;\bar\alpha_j\Big), \quad c_j=\frac{q}{b_j}+1.$$

We accumulate in the *log domain* to avoid underflow/overflow.

With $H_j={}_2F_1(-m,1;c_j;\bar\alpha_j)$ and $H_j'={}_2F_1(-m+1,2;c_j+1;\bar\alpha_j)$, the gradient w.r.t. an α_i in bin S_j (and zero otherwise) is:

$$\frac{\partial B_{\text{Hölder}}}{\partial\alpha_i} = -\frac{B_{\text{Hölder}}}{c_j}\frac{H_j'}{H_j}\cdot\frac{1}{m_j}, \qquad c_j=\frac{q}{b_j}+1,$$

as derived in §G.4. The same cached $\{H_j,H_j'\}$ also feed the zero-aware gap gradients in §G.5 via $\widetilde{\mathcal{A}}_{b_j}$ (finite sums of ${}_2F_1$ with shifted parameters).

Complexity and vectorization. The forward is $O\Big(\sum_j m\Big)=O(dm)$ scalar ops, embarrassingly parallel over bins. The backward reuses the same per-bin computations and adds only $O(dm)$ extra ops for H_j' and simple scalar multiplications.

Algorithm 2 HolderBound&Grad($\boldsymbol{\alpha}, \boldsymbol{\beta}, q$)

Bin indices by equal β: obtain $\{(b_j, S_j, m_j, \bar{\alpha}_j)\}_{j=1}^{d}$, with $m = \sum_j m_j$.
for $j = 1$ to d **do** ▷ ${}_2F_1(-m, 1; c_j; \bar{\alpha}_j)$ via term ratios or Horner
  $c_j \leftarrow q/b_j + 1$; $H_j \leftarrow 1$; $t \leftarrow 1$
  for $k = 1$ **to** m **do**
    $t \leftarrow t \cdot \frac{(-m+k-1)}{(c_j+k-1)} \cdot \bar{\alpha}_j$ ▷ alternates in sign
    $H_j \leftarrow H_j + t$ ▷ use Kahan/Neumaier compensation
    if $|t| < \varepsilon_{\text{rel}} \max(|H_j|, \delta_{\text{abs}})$ **then break**
  $B_j^* \leftarrow H_j/q$; $\ell_j \leftarrow \frac{m_j}{m} \log B_j^*$
$B \leftarrow \exp\Big(\sum_{j=1}^{d} \ell_j\Big)$ ▷ $B_{\text{Hölder}}$ in log-sum-exp form
for $j = 1$ to d **do** ▷ ${}_2F_1(-m+1, 2; c_j+1; \bar{\alpha}_j)$ for gradients
  $H'_j \leftarrow 1$; $t \leftarrow 1$
  for $k = 1$ **to** $m-1$ **do**
    $t \leftarrow t \cdot \frac{(-m+k)}{(c_j+1+k-1)} \cdot \frac{k+1}{k} \cdot \bar{\alpha}_j$
    $H'_j \leftarrow H'_j + t$ ▷ again sign-alternating; compensate
    if $|t| < \varepsilon_{\text{rel}} \max(|H'_j|, \delta_{\text{abs}})$ **then break**
return B and $\{H_j, H'_j\}_{j=1}^{d}$ ▷ used in the backward ratio H'_j/H_j

Forward objective with zero-aware gap. The training objective combines the Hölder (or common–β) envelope with the zero-aware gap penalty:

$$\mathcal{U}(P) = \underbrace{\prod_{j=1}^{d} \Big[\mathcal{H}_{b_j}(q; \bar{\alpha}_j, m)\Big]^{m_j/m}}_{\text{envelope}} + \rho \sum_j \underbrace{\frac{m}{2} \frac{m_j}{m} V_j^{\omega} \widetilde{\mathcal{A}}_{b_j}(q; \bar{\alpha}_j, m)}_{\text{zero-aware gap}},$$

where V_j^{ω} is the within-bin ω-weighted variance (Proposition 3). Both terms reuse the same forward hypergeometric blocks; the second depends only on bin means and the finite differences of the same envelopes.

G.3 Gradient of the envelope for common β

Recall the common–β envelope from Theorem 1:

$$\mathcal{H}_\beta(q; \bar{\alpha}, m) \;=\; \frac{1}{q}\, {}_2F_1\Big(-m,\, 1;\, \tfrac{q}{\beta} + 1;\, \bar{\alpha}\Big), \qquad q > 0,\ \beta > 0,\ \bar{\alpha} = \tfrac{1}{m} \sum_{i=1}^{m} \alpha_i.$$

Since $\mathcal{H}_\beta$ depends on $\boldsymbol{\alpha}$ only through $\bar{\alpha}$, the chain rule gives:

$$\frac{\partial \mathcal{H}_\beta}{\partial \alpha_i} \;=\; \frac{\partial \mathcal{H}_\beta}{\partial \bar{\alpha}} \cdot \frac{\partial \bar{\alpha}}{\partial \alpha_i} \;=\; \frac{1}{m} \frac{\partial \mathcal{H}_\beta}{\partial \bar{\alpha}}, \qquad i = 1, \ldots, m.$$

Using the standard derivative $\frac{d}{dz} {}_2F_1(a, b; c; z) = \frac{ab}{c}\, {}_2F_1(a+1, b+1; c+1; z)$ with $(a, b, c, z) = \big(-m, 1, \frac{q}{\beta} + 1, \bar{\alpha}\big)$, we obtain:

$$\frac{\partial \mathcal{H}_\beta}{\partial \bar{\alpha}} = \frac{1}{q} \cdot \frac{-m}{\frac{q}{\beta} + 1}\, {}_2F_1\Big(-m+1,\, 2;\, \tfrac{q}{\beta} + 2;\, \bar{\alpha}\Big),$$

and hence the per–coordinate gradient:

$$\boxed{\frac{\partial \mathcal{H}_\beta(q; \bar{\alpha}, m)}{\partial \alpha_i} = -\frac{1}{q\big(\frac{q}{\beta} + 1\big)}\, {}_2F_1\Big(-m+1,\, 2;\, \tfrac{q}{\beta} + 2;\, \bar{\alpha}\Big) = -\frac{\beta}{q(q+\beta)}\, {}_2F_1\Big(-m+1,\, 2;\, \tfrac{q}{\beta} + 2;\, \bar{\alpha}\Big), \quad i = 1, \ldots, m.}$$

The gradient is uniform across coordinates because the envelope depends on $\boldsymbol{\alpha}$ only via $\bar{\alpha}$. Since $-m$ is a nonpositive integer, ${}_2F_1(-m+1, 2; \cdot; \bar{\alpha})$ is a degree-$(m-1)$ polynomial in $\bar{\alpha}$, enabling stable evaluation via a finite sum or Horner's rule.

G.4 Gradient of the Hölder envelope for heterogeneous β

Recall the Hölder envelope (Appendix A.6): with distinct exponents $\{b_1, \dots, b_d\}$, groups $S_k \stackrel{\text{def}}{=} \{i : \beta_i = b_k\}$, sizes $m_k = |S_k|$, $m = \sum_k m_k$, and means $\bar{\alpha}_k = \frac{1}{m_k} \sum_{i \in S_k} \alpha_i$, we defined:

$$B_{\text{Holder}} \;=\; \prod_{k=1}^{d} \big(B_k^*\big)^{m_k/m}, \qquad B_k^* \;=\; \frac{1}{q}\, {}_2F_1\Big(-m,\, 1;\, c_k;\, \bar{\alpha}_k\Big), \quad c_k \stackrel{\text{def}}{=} \frac{q}{b_k} + 1,$$

with $q > 0$ and $b_k > 0$. We compute the gradient $\partial B_{\text{Holder}} / \partial \alpha_i$ for an index $i \in S_j$ (so $\beta_i = b_j$).

Since B_{Holder} depends on α_i only through $\bar{\alpha}_j$,

$$\frac{1}{B_{\text{Holder}}} \frac{\partial B_{\text{Holder}}}{\partial \alpha_i} \;=\; \frac{m_j}{m} \frac{1}{B_j^*} \frac{\partial B_j^*}{\partial \alpha_i}, \qquad \frac{\partial B_j^*}{\partial \alpha_i} \;=\; \frac{\partial B_j^*}{\partial \bar{\alpha}_j} \cdot \frac{\partial \bar{\alpha}_j}{\partial \alpha_i} \;=\; \frac{1}{m_j} \frac{\partial B_j^*}{\partial \bar{\alpha}_j}.$$

Thus:

$$\frac{1}{B_{\text{Holder}}} \frac{\partial B_{\text{Holder}}}{\partial \alpha_i} \;=\; \frac{1}{m} \frac{1}{B_j^*} \frac{\partial B_j^*}{\partial \bar{\alpha}_j}.$$

Differentiating the hypergeometric. Using $\frac{d}{dz} {}_2F_1(a, b; c; z) = \frac{ab}{c}\, {}_2F_1(a+1, b+1; c+1; z)$ with $(a, b, c, z) = \big(-m, 1, c_j, \bar{\alpha}_j\big)$,

$$\frac{\partial B_j^*}{\partial \bar{\alpha}_j} \;=\; \frac{1}{q} \cdot \frac{-m}{c_j}\, {}_2F_1\Big(-m+1,\, 2;\, c_j+1;\, \bar{\alpha}_j\Big).$$

Combining,

$$\frac{1}{B_{\text{Holder}}} \frac{\partial B_{\text{Holder}}}{\partial \alpha_i} \;=\; -\frac{1}{q\, c_j} \frac{{}_2F_1\big(-m+1, 2; c_j+1; \bar{\alpha}_j\big)}{B_j^*} \;=\; -\frac{1}{c_j} \frac{{}_2F_1\big(-m+1, 2; c_j+1; \bar{\alpha}_j\big)}{{}_2F_1\big(-m, 1; c_j; \bar{\alpha}_j\big)},$$

since $B_j^* = (1/q)\, {}_2F_1(-m, 1; c_j; \bar{\alpha}_j)$. Multiplying by B_{Holder} yields the per–coordinate gradient (identical for all $i \in S_j$, and 0 for $i \notin S_j$):

$$\boxed{\frac{\partial B_{\text{Holder}}}{\partial \alpha_i} \;=\; -\frac{B_{\text{Holder}}}{c_j} \frac{{}_2F_1\big(-m+1, 2;\, c_j+1;\, \bar{\alpha}_j\big)}{{}_2F_1\big(-m, 1;\, c_j;\, \bar{\alpha}_j\big)}, \qquad c_j = \frac{q}{b_j} + 1, \quad i \in S_j.}$$

Equivalent forms. Let $F_1(z) \stackrel{\text{def}}{=} {}_2F_1(-m, 1; c_j; z)$ and $F_2(z) \stackrel{\text{def}}{=} {}_2F_1(-m+1, 2; c_j+1; z)$. By the derivative identity, $F_2(z) = -\frac{c_j}{m} \frac{d}{dz} F_1(z)$, hence

$$\frac{1}{B_{\text{Holder}}} \frac{\partial B_{\text{Holder}}}{\partial \alpha_i} \;=\; -\frac{1}{c_j} \frac{F_2(\bar{\alpha}_j)}{F_1(\bar{\alpha}_j)} \;=\; \frac{1}{m} \frac{d}{dz} \log F_1(z)\Big|_{z=\bar{\alpha}_j}.$$

This gives two numerically equivalent implementations:

$$\text{(ratio form)} \quad \partial_{\alpha_i} \log B_{\text{Holder}} = -\frac{1}{c_j} \frac{F_2(\bar{\alpha}_j)}{F_1(\bar{\alpha}_j)},$$

$$\text{(log-derivative form)} \quad \partial_{\alpha_i} \log B_{\text{Holder}} = \frac{1}{m} \frac{d}{dz} \log\big[{}_2F_1(-m, 1; c_j; z)\big]\Big|_{z=\bar{\alpha}_j}.$$

Since $-m$ is a nonpositive integer, both hypergeometric terms truncate:

$${}_2F_1(-m, 1; c_j; z) = \sum_{k=0}^{m} \frac{(-m)_k (1)_k}{(c_j)_k} \frac{z^k}{k!} = \sum_{k=0}^{m} (-1)^k \binom{m}{k} \frac{z^k}{(c_j)_k},$$

$${}_2F_1(-m+1, 2; c_j+1; z) = \sum_{k=0}^{m-1} \frac{(-m+1)_k (2)_k}{(c_j+1)_k} \frac{z^k}{k!} = \sum_{k=0}^{m-1} (-1)^k \frac{(m-1)!}{(m-1-k)!} \frac{k+1}{(c_j+1)_k} z^k.$$

Thus the ratio in the boxed gradient can be evaluated via stable finite sums (Horner's rule).

Block structure of the gradient. For a fixed bin j, all coordinates $i \in S_j$ share the same partial derivative; for $i \notin S_j$ the derivative is zero:

$$\frac{\partial B_{\text{Holder}}}{\partial \alpha_i} = \begin{cases} -\dfrac{B_{\text{Holder}}}{c_j} \dfrac{{}_2F_1(-m+1, 2; c_j+1; \bar{\alpha}_j)}{{}_2F_1(-m, 1; c_j; \bar{\alpha}_j)}, & i \in S_j, \\ 0, & i \notin S_j. \end{cases}$$

The *log-derivative form* is preferred to avoid overflow/underflow when m is large. Note that both F_1 and F_2 are nonnegative on $z \in [0, 1]$; the gradient is non-positive (increasing any α_i weakly decreases the envelope), consistent with the envelope's monotonicity in $\bar{\alpha}_j$. Complexity is $O(d\, m)$ per gradient evaluation using the finite sums across bins; computation is easy to parallelize over j.

G.5 Gradients of the final objective

For cluster ℓ and bin index j, let $S_{\ell j}$ be the set of vertices assigned to bin j (with common exponent b_j), $m_{\ell j} \stackrel{\text{def}}{=} |S_{\ell j}|$, $m_\ell \stackrel{\text{def}}{=} \sum_j m_{\ell j}$, and:

$$\bar{p}_{\ell j} \stackrel{\text{def}}{=} \frac{1}{m_{\ell j}} \sum_{r \in S_{\ell j}} p_{r\ell}, \qquad w_{\ell j} \stackrel{\text{def}}{=} \frac{m_{\ell j}}{m_\ell} \quad \text{(Hölder weight)}.$$

The common–β (here $\beta = b_j$) envelope for cluster ℓ and bin j is:

$$\mathcal{H}_{b_j}\big(q; \bar{p}_{\ell j}, m_\ell\big) \;=\; \frac{1}{q}\, {}_2F_1\Big(-m_\ell, 1; \tfrac{q}{b_j} + 1; \bar{p}_{\ell j}\Big),$$

and the second forward β–difference and its $\bar{p}$–derivative are (Proposition 3):

$$\mathcal{A}_{b_j}(q; \bar{p}_{\ell j}, m_\ell) \stackrel{\text{def}}{=} \sum_{r=0}^{2} \binom{2}{r} (-1)^r\, \mathcal{H}_{b_j}\big(q + r b_j; \bar{p}_{\ell j}, m_\ell\big), \qquad \widetilde{\mathcal{A}}_{b_j}(q; \bar{p}_{\ell j}, m_\ell) \stackrel{\text{def}}{=} \frac{\partial}{\partial \bar{p}_{\ell j}} \mathcal{A}_{b_j}(q; \bar{p}_{\ell j}, m_\ell).$$

Zero-aware statistics in a bin. Fix $i \in S_{\ell j}$ and write

$$\omega_i \stackrel{\text{def}}{=} \omega(p_{i\ell}), \qquad \Omega_{\ell j} \stackrel{\text{def}}{=} \sum_{r \in S_{\ell j}} \omega(p_{r\ell}), \qquad S_2 \stackrel{\text{def}}{=} \sum_{r \in S_{\ell j}} \omega(p_{r\ell})\, p_{r\ell},$$

$$\mu \stackrel{\text{def}}{=} \bar{p}^{\omega}_{\ell j} \;=\; S_2/\Omega_{\ell j}, \qquad V \stackrel{\text{def}}{=} \mathrm{Var}^{\omega}_{\ell j}(p) \;=\; \frac{1}{\Omega_{\ell j}} \sum_{r \in S_{\ell j}} \omega(p_{r\ell}) \big(p_{r\ell} - \mu\big)^2,$$

with the convention $V = 0$ if $\Omega_{\ell j} = 0$. In the paper we take $\omega(x) = x(1-x)$ so that $\omega_i' \stackrel{\text{def}}{=} \frac{d}{dp}\omega(p_{i\ell}) = 1 - 2p_{i\ell}$ (zero-aware and symmetric). The derivatives of the weighted mean and variance are:

$$\frac{\partial \mu}{\partial p_{i\ell}} \;=\; \frac{\omega_i + \omega_i'\,(p_{i\ell} - \mu)}{\Omega_{\ell j}}, \qquad \frac{\partial V}{\partial p_{i\ell}} \;=\; \frac{1}{\Omega_{\ell j}} \Big[\, \omega_i'\,(p_{i\ell} - \mu)^2 + 2\omega_i\,(p_{i\ell} - \mu)\Big(1 - \frac{\partial \mu}{\partial p_{i\ell}}\Big)\,\Big] - \frac{V}{\Omega_{\ell j}}\omega_i'. \tag{44}$$

This is a standard quotient/chain-rule calculation using $\sum_{r \in S_{\ell j}} \omega(p_{r\ell})(p_{r\ell} - \mu) = 0$.

Hypergeometric derivatives needed. Let $q_r \stackrel{\text{def}}{=} q + r b_j$ and $c_r \stackrel{\text{def}}{=} \frac{q_r}{b_j} + 1$. Using $\frac{d}{dz}{}_2F_1(a, b; c; z) = \frac{ab}{c}{}_2F_1(a+1, b+1; c+1; z)$,

$$\frac{\partial}{\partial \bar{p}_{\ell j}} \mathcal{H}_{b_j}(q_r; \bar{p}_{\ell j}, m_\ell) = \frac{1}{q_r} \cdot \frac{-m_\ell}{c_r}\, {}_2F_1\Big(-m_\ell + 1,\, 2;\, c_r + 1;\, \bar{p}_{\ell j}\Big), \tag{45}$$

$$\frac{\partial^2}{\partial \bar{p}_{\ell j}^2} \mathcal{H}_{b_j}(q_r; \bar{p}_{\ell j}, m_\ell) = \frac{1}{q_r} \cdot \frac{-m_\ell}{c_r} \cdot \frac{(-m_\ell + 1) \cdot 2}{c_r + 1}\, {}_2F_1\Big(-m_\ell + 2,\, 3;\, c_r + 2;\, \bar{p}_{\ell j}\Big). \tag{46}$$

Therefore:

$$\widetilde{\mathcal{A}}_{b_j}(q;\bar{p}_{\ell j},m_\ell)=\sum_{r=0}^{2}\binom{2}{r}(-1)^r\,\frac{1}{q_r}\cdot\frac{-m_\ell}{c_r}\,{}_2F_1\Big(-m_\ell+1,2;c_r+1;\bar{p}_{\ell j}\Big), \tag{47}$$

$$\frac{\partial}{\partial\bar{p}_{\ell j}}\widetilde{\mathcal{A}}_{b_j}(q;\bar{p}_{\ell j},m_\ell)=\sum_{r=0}^{2}\binom{2}{r}(-1)^r\,\frac{1}{q_r}\cdot\frac{-m_\ell}{c_r}\cdot\frac{(-m_\ell+1)\cdot 2}{c_r+1}\,{}_2F_1\Big(-m_\ell+2,3;c_r+2;\bar{p}_{\ell j}\Big). \tag{48}$$

Zero-aware gap term and its gradient. As stated in the paper, our simple zero-aware upper bound for the AM–GM gap in bin j is:

$$\Gamma^{\text{ewa}}_{\ell j}(q)\ \stackrel{\text{def}}{=}\ \frac{m_\ell}{2}\,w_{\ell j}\,V\,\widetilde{\mathcal{A}}_{b_j}\big(q;\bar{p}_{\ell j},m_\ell\big),$$

so the per–coordinate gradient for $i\in S_{\ell j}$ is:

$$\frac{\partial\Gamma^{\text{ewa}}_{\ell j}}{\partial p_{i\ell}}=\frac{m_\ell}{2}\,w_{\ell j}\left[\frac{\partial V}{\partial p_{i\ell}}\,\widetilde{\mathcal{A}}_{b_j}\big(q;\bar{p}_{\ell j},m_\ell\big)\ +\ V\,\frac{\partial\widetilde{\mathcal{A}}_{b_j}}{\partial\bar{p}_{\ell j}}\cdot\frac{\partial\bar{p}_{\ell j}}{\partial p_{i\ell}}\right],\qquad\frac{\partial\bar{p}_{\ell j}}{\partial p_{i\ell}}=\frac{1}{m_{\ell j}}. \tag{49}$$

Here $\partial V/\partial p_{i\ell}$ and $\partial\mu/\partial p_{i\ell}$ are given by Equation (44), while $\widetilde{\mathcal{A}}_{b_j}$ and its derivative are Equation (47)–Equation (48). For $i\notin S_{\ell j}$, $\partial\Gamma^{\text{ewa}}_{\ell j}/\partial p_{i\ell}=0$.

Envelope term and stick–breaking backward. The binned Hölder envelope for cluster ℓ (Appendix A.6) is:

$$B_{\text{Holder},\ell}\ =\ \prod_k\Big(\mathcal{H}_{b_k}(q;\bar{p}_{\ell k},m_\ell)\Big)^{w_{\ell k}},$$

with per–coordinate gradient (for $i\in S_{\ell j}$):

$$\frac{\partial B_{\text{Holder},\ell}}{\partial p_{i\ell}}\ =\ -\frac{B_{\text{Holder},\ell}}{c_j}\,\frac{{}_2F_1\big(-m_\ell+1,2;c_j+1;\bar{p}_{\ell j}\big)}{{}_2F_1\big(-m_\ell,1;c_j;\bar{p}_{\ell j}\big)}\cdot\frac{1}{m_{\ell j}},\qquad c_j=\frac{q}{b_j}+1,$$

obtained by the same log–diff + chain rule used in Appendix G.4. (If the outer objective multiplies the envelope by additional factors; e.g., edge weights $M_{i\ell}(P)$ in the paper—apply product rule and chain through their own Jacobians.)

Let the final per–cluster contribution be:

$$\mathcal{L}_\ell(P)\ =\ U_\ell(P)\ +\ \rho\sum_j\Gamma^{\text{ewa}}_{\ell j}(q),$$

where U_ℓ uses the Hölder envelope (possibly multiplied by problem-specific weights), and $\rho\ge 0$ is the gap regularization. The gradient w.r.t. an entry $p_{i\ell}$ is:

$$\frac{\partial\mathcal{L}_\ell}{\partial p_{i\ell}}\ =\ \frac{\partial U_\ell}{\partial p_{i\ell}}\ +\ \rho\sum_{j:\,i\in S_{\ell j}}\frac{\partial\Gamma^{\text{ewa}}_{\ell j}}{\partial p_{i\ell}},$$

with the explicit pieces given in Equation (44)–Equation (49). These feed into the stick–breaking backward pass exactly as in the main text.

If $\Omega_{\ell j}=0$, set $\mu=0$, $V=0$, and $\partial\mu=\partial V=0$; the bin is inactive and contributes no gradient. Because $-m_\ell$ is a nonpositive integer, all ${}_2F_1$ terms truncate to finite polynomials in $\bar{p}_{\ell j}$, enabling stable Horner evaluation for both Equation (47) and Equation (48). For $\omega(x)=x^a$ with $a\in[1,2]$, replace ω_i' by $a\,p_{i\ell}^{a-1}$ in Equation (44); the rest of the derivation is unchanged.

G.6 A relaxed Hölder envelope

Setup. Recall the Hölder envelope (Appendix A.6) for heterogeneous exponents:

$$B_{\text{Holder}}\ =\ \prod_{j=1}^{d}\Big(\mathcal{H}_{b_j}(q;\bar{\alpha}_j,m)\Big)^{m_j/m},\qquad\mathcal{H}_{b_j}(q;\bar{\alpha}_j,m)=\frac{1}{q}\,{}_2F_1\Big(-m,1;\underbrace{c_j}_{=q/b_j+1};\bar{\alpha}_j\Big),$$

where $b_j>0$ is the exponent for bin j, $m_j=|S_j|$, $m=\sum_j m_j$, and $\bar{\alpha}_j=\frac{1}{m_j}\sum_{i\in S_j}\alpha_i$.

Monotonicity in c. For $m \in \mathbb{N}$ and $z \in [0,1]$, the truncated series:

$$ {}_2F_1(-m,1;c;z) = \sum_{k=0}^{m} \frac{(-m)_k (1)_k}{(c)_k} \frac{z^k}{k!} = \sum_{k=0}^{m} (-1)^k \binom{m}{k} \frac{z^k}{(c)_k} $$

has nonnegative terms in absolute value and *each* Pochhammer factor $(c)_k = c(c+1)\cdots(c+k-1)$ is strictly increasing in c. Hence the whole sum is *decreasing* in c:

$$ c_1 \le c_2 \implies {}_2F_1(-m,1;c_1;z) \ \ge \ {}_2F_1(-m,1;c_2;z). \tag{$\star$} $$

Within a bin j, choose a left–endpoint representative $b_j^{\leftarrow} \le \beta_i$ for $i \in S_j$ (as in Appendix A.6). Then $c_j^{\leftarrow} = q/b_j^{\leftarrow} + 1 \ge q/\beta_i + 1$ and, in particular, if $q \ge b_j^{\leftarrow}$ we have $c_j^{\leftarrow} \ge 2$. Combining the binwise AM–GM (intra-bin) and Hölder (across bins) steps with the monotonicity Equation ($\star$) yields the *relaxed* envelope:

$$ \mathcal{H}_{b_j}(q;\bar{\alpha}_j,m) \ = \ \frac{1}{q}\, {}_2F_1\Big(-m,1;c_j^{\leftarrow};\bar{\alpha}_j\Big) \ \le \ \frac{1}{q}\, {}_2F_1\Big(-m,1;2;\bar{\alpha}_j\Big), \quad \text{whenever } c_j^{\leftarrow} \ge 2. $$

Therefore:

$$ \boxed{ B_{\text{Holder}} \ \le \ \underbrace{\prod_{j=1}^{d} \left[\frac{1}{q}\, {}_2F_1\big(-m,1;2;\bar{\alpha}_j\big) \right]^{m_j/m}}_{\stackrel{\text{def}}{=} \ B_{\text{relax}(c=2)}} \qquad (\text{provided } q \ge b_j^{\leftarrow} \ \forall j). } $$

Intuitively, replacing c_j by the uniform lower value 2 (the "largest" case by Equation ($\star$)) gives a looser but simpler upper bound. It preserves bin structure through the $\bar{\alpha}_j$'s and weights m_j/m, but removes the explicit b_j–dependence from the hypergeometric parameter.

Practical simplifications for $c = 2$. Because $-m$ is a nonpositive integer, ${}_2F_1(-m,1;2;z)$ is a degree-m polynomial in z and can be evaluated stably by a finite sum (Horner's rule):

$$ {}_2F_1(-m,1;2;z) = \sum_{k=0}^{m} (-1)^k \binom{m}{k} \frac{z^k}{(2)_k} = \sum_{k=0}^{m} (-1)^k \binom{m}{k} \frac{z^k}{(k+1)!}. $$

Thus:

$$ B_{\text{relax}(c=2)} = \prod_{j=1}^{d} \left[\frac{1}{q} \sum_{k=0}^{m} (-1)^k \binom{m}{k} \frac{\bar{\alpha}_j^{\,k}}{(k+1)!} \right]^{m_j/m}. $$

This form is handy when one wants to precompute per-bin polynomials in $\bar{\alpha}_j$ independent of b_j.