WHAT SOFT CLUSTERING OBJECTIVES COMPUTE, WHEN THEY ARE TIGHT, AND WHAT SOFTNESS BUYS: THE CASE OF SOFT STRUCTURAL ENTROPY

Anonymous authors
Paper under double-blind review

ABSTRACT

Differentiable relaxations of discrete clustering objectives are standard in graph learning: DMoN, MinCutPool, Neuromap and structural-entropy (SE) methods (LSEnet, DeSE) relax modularity, normalized cut, the map equation and SE by substituting soft volumes and internal weights into the discrete formula. The resulting *bilinear plug-in* is read as the discrete objective. We ask what it computes, when it is tight and what the softness buys, with two-level SE as the running case. For a row-stochastic assignment S we compare the plug-in H_{bil}, the multilinear extension $H_{\mathrm{mul}}(S) = \mathbb{E}[H_2(Z)]$ under independent rounding, and a memoryless stochastic-encoder code length H_{code}. All three agree at hard assignments, and $H_{\mathrm{code}}(S) = H_{\mathrm{bil}}(S) + \frac{1}{2m}\sum_j d_j\,\mathcal{H}(S_j)$ exactly: the deployed objective is a bits-back net rate. The multilinear extension is an *exact* relaxation (its minimum is a vertex and equals the discrete minimum), whereas the plug-in is *provably loose*: at the uniform assignment it equals $H_1 - \log_2(K)/K$ on every graph, so a strictly soft S beats every partition whenever the discrete optimum lies above that floor, and local overlap makes it loose even when it does not (5-node bowtie: $\log_2 3$ against $\frac{7}{6} + \frac{1}{3}\log_2 3$ bits). Beyond SE, the multilinear extension is exact for every partition objective; Neuromap's plug-in obeys the same bits-back identity but has no floor; pairwise objectives differ from their expectation by a degree-weighted Gini refund, which makes Markov stability at even times exactly tight and bounds modularity's looseness by $(1 - \frac{1}{K})\sum_i d_i^2/(2m)^2$. In an exhaustive census plug-ins deployed for SE, DMoN, MinCutPool and Dasgupta's cost beat every partition on 12, 14, 10 and 9 of 20 instances; the map equation's never did by more than 10^{-3} bits. On the sparse graphs we tested the looseness does not bind: on Cora, CiteSeer and planted SBMs the plug-in optima are near-hard, the multilinear objective trained with its unbiased gradient finds no better partitions, and the soft-over-hard advantage of our encoder control largely vanishes against a Gumbel-softmax hard arm. Preregistered gates are reported verbatim, including the failures (a planted-overlap null control fails at 5 and at 20 seeds).

1 INTRODUCTION

Graph-learning methods routinely learn a clustering by gradient descent on a relaxation of a discrete objective: DMoN (Tsitsulin et al., 2023) relaxes modularity, MinCutPool (Bianchi et al., 2020) normalized cut and Neuromap (Blöcker et al., 2024) the map equation, each by substituting soft volumes and internal weights of a row-stochastic assignment S into the discrete formula. These losses pool nodes in graph neural networks and cluster learned embeddings, and the relaxed value is read as a measurement of the discrete objective: a low loss, or a margin over a degree-matched null, is taken as evidence of cluster structure. Agreement at one-hot assignments, the usual justification, says nothing about the interior where gradient descent operates; if a strictly soft S scores below every partition, a low value may reflect the relaxation rather than the data. A relaxed objective therefore owes (i) a precise code for its value, (ii) a reduction to the discrete objective, and (iii) controls isolating the objective from "a better encoder". We supply all three and find one pattern across eight objectives (§4.3): the multilinear extension (the expected hard objective under independent rounding) is exact for every partition objective, while the deployed plug-in (a bits-back net rate for code lengths,

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

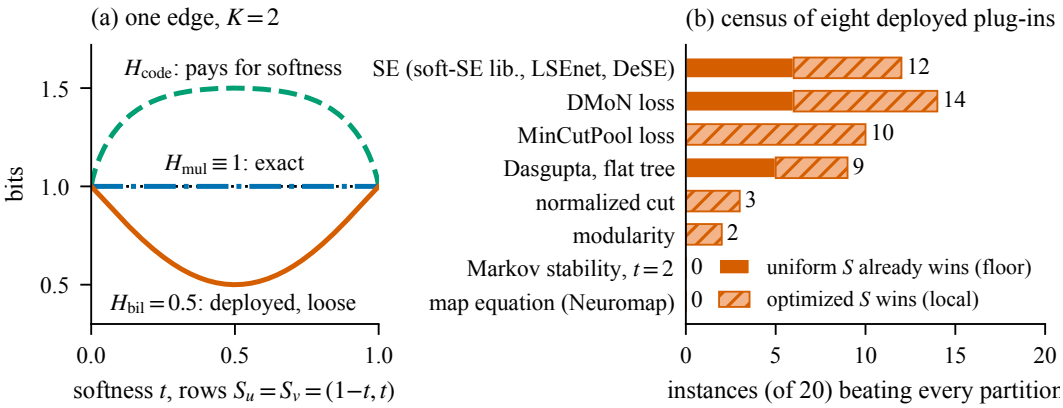

Figure 1: **The problem at a glance.** (a) One edge, $K = 2$, rows $S_u = S_v = (1-t, t)$; every hard partition costs 1 bit. The deployed plug-in *descends* to 0.5 at uniform S, beating every partition; the multilinear extension stays at the discrete optimum; the stochastic-encoder code *rises* to 1.5, the gap being the assignment entropy (Theorem 2). (b) On how many of the 20 census instances (§5) each deployed plug-in beats every partition: already at uniform S (floor, Theorem 4 and analogues) or only after optimization. Values as in Table 1.

a refunded expectation for pairwise objectives) beats every partition on some census instances for six of the eight (Figure 1b).

We develop the theory on structural entropy (SE; Li and Pan, 2016), where it is sharpest. SE measures how well a hierarchical partition compresses a random walk; its encoding tree promises a hierarchy from an objective, which has made it attractive to representation learning and to state abstraction and exploration in reinforcement learning (Zeng et al., 2023; 2024). SE assumes hard partitions, so differentiable SE methods (LSEnet, DeSE, open-source SE libraries) optimize the same plug-in.

This paper is a theory of the relaxation, not another encoder. Our contributions:

1. **Three definitions, one point of agreement** (§3). The plug-in H_{bil}, the multilinear extension H_{mul} and the memoryless stochastic-encoder code H_{code} all equal H_2 at one-hot assignments (Theorem 1), yet on one edge with $K{=}2$ uniform S gives 0.5, 1.0 and 1.5 bits.
2. **A bits-back identity** (Theorem 2): $H_{\text{code}} = H_{\text{bil}} + \frac{1}{2m}\sum_j d_j \mathcal{H}(S_j)$ exactly, so H_{bil} is the net rate of a stochastic two-level encoder whose assignment randomness is refunded.
3. **The tightness verdict** (§4). H_{mul} is an exact relaxation, $\min_S H_{\text{mul}} = \min_z H_2$ (Theorem 3), and its coordinate descent *is* the classical label-propagation local search. H_{bil} is loose: its value at uniform is the graph-independent floor $H_1 - \log_2(K)/K$ (Theorem 4), strictly below the discrete optimum on explicit families (Corollary 2), and looseness also arises at planted-overlap nodes when the floor does not bind (Example 2). A rounding bound (Proposition 2) controls the gap near hard solutions; $\tau \to \infty$ drives H_{bil} to the floor (Appendix E), and the preregistered low-τ check *failed* (optimization sticks).
4. **Eight objectives** (§4.3): the plug-in versus the expectation for the map equation (the same bits-back identity, but no floor), pairwise objectives (modularity: looseness at most $(1 - \frac{1}{K})\sum_i d_i^2/(2m)^2$; Markov stability at even times: exact) and Dasgupta's cost. The plug-ins deployed for SE, DMoN, MinCutPool and Dasgupta's cost beat every partition on 12, 14, 10 and 9 of 20 census instances, the map-equation plug-in never by over 10^{-3} bits.
5. **Fixtures, case study and controls** (§5): the bowtie; a planted-overlap fixture with a degree-preserving null; the census, drawn on four graphs (Figure 2); and encoder controls on Cora and CiteSeer, where the looseness does not bind and exactness buys no better partitions.

The verdict in one sentence. A deployed plug-in is not the objective it is named after; the multilinear extension is faithful to what a value *means*, though on the sparse real graphs tested not a better optimizer, and a reported plug-in value should be calibrated against its uniform-S value.

2 PRELIMINARIES

Let G be undirected and weighted with symmetric adjacency $A \in \mathbb{R}^{N\times N}_{\geq 0}$, no self-loops ($A_{ii} = 0$), degrees $d_i = \sum_j A_{ij}$, total volume $2m = \sum_i d_i > 0$. A hard partition into at most K modules is a map $z : [N] \to [K]$; module k has volume $V_k = \sum_{z_i=k} d_i$, internal weight $w_k = \sum_{ij} A_{ij}\mathbf{1}[z_i = k]\mathbf{1}[z_j = k]$ (each internal edge counted twice) and cut $g_k = V_k - w_k$. The one-level SE is $H_1 = -\sum_i \frac{d_i}{2m}\log_2\frac{d_i}{2m}$, and the canonical two-level SE (Li and Pan, 2016) is

$$H_2(z) \;=\; -\sum_k \sum_{i:z_i=k} \frac{d_i}{2m}\log_2\frac{d_i}{V_k} \;-\; \sum_k \frac{g_k}{2m}\log_2\frac{V_k}{2m}. \tag{1}$$

Lemma 1 (internal-weight form). $H_2(z) = H_1 + \dfrac{1}{2m}\sum_k w_k \log_2 \dfrac{V_k}{2m}$, *with the convention* $0\log 0 = 0$.

Proof. The leaf term of (1) equals $-\sum_i \frac{d_i}{2m}\log_2\frac{d_i}{2m} + \sum_k \frac{V_k}{2m}\log_2\frac{V_k}{2m}$, and $-g_k = w_k - V_k$; summing cancels the $\sum_k \frac{V_k}{2m}\log_2\frac{V_k}{2m}$ terms. □

A *soft assignment* is a row-stochastic $S \in \mathbb{R}^{N\times K}$, $S_i \in \Delta_{K-1}$; hard assignments are the vertices of the product Δ^N_{K-1}. Soft volumes and internal weights are $V_k(S) = \sum_i d_i S_{ik} = (d^\top S)_k$ and $w_k(S) = S_{\cdot k}^\top A S_{\cdot k}$.

Nothing below requires a combinatorial graph: every statement holds for any symmetric nonnegative A with zero diagonal, e.g. a kernel affinity on points (Appendix C).

3 THREE DEFINITIONS OF SOFT SE, AND WHAT CODE EACH ONE IS

Definition 1 (bilinear plug-in). $H_{\mathrm{bil}}(S) = H_1 + \dfrac{1}{2m}\sum_k w_k(S)\,\log_2\dfrac{V_k(S)}{2m}$.

This is algebraically identical to the soft-SE objective of a widely deployed open-source SE library (which is stated via soft cuts $g_k(S) = V_k(S) - w_k(S)$; verified numerically at random soft S to float32, max deviation $5\cdot 10^{-7}$), to the level-wise differentiable structural information of LSEnet (Sun et al., 2024) restricted to two levels and to DeSE's soft structural-entropy term (Zhang et al., 2025) (these two from their published definitions, not re-run), and it is the exact SE analogue of how DMoN (Tsitsulin et al., 2023) and MinCutPool (Bianchi et al., 2020) relax modularity and normalized cut: plug $S^\top AS$ and $d^\top S$ into the discrete formula.

Definition 2 (multilinear extension). $H_{\mathrm{mul}}(S) = \mathbb{E}_Z[H_2(Z)]$, *with independent* $Z_i \sim \mathrm{Cat}(S_i)$.

Coding interpretation: sample the partition once (shared randomness between encoder and decoder), then use the canonical two-level codebook of the realized partition; H_{mul} is the expected per-step code length. This is the same construction as the multilinear extension central to submodular optimization (Calinescu et al., 2011).

Definition 3 (memoryless stochastic-encoder code). *At each random-walk step, the encoder resamples module labels $z_i \sim S_i$, $z_j \sim S_j$ for the current and next node and uses one fixed codebook adapted to S: module codewords of length $-\log_2(V_k(S)/2m)$, and within-module node codewords of length $-\log_2\big(d_j S_{jk}/V_k(S)\big)$ (these are valid prefix codes: both sets of probabilities sum to one). $H_{\mathrm{code}}(S)$ is the expected code length per step of the stationary edge process $\Pr(i\to j) = A_{ij}/2m$: a step $i\to j$ costs the within-module codeword of j in z_j, plus the module codeword of z_j if $z_j \neq z_i$.*

Theorem 1 (reduction / consistency). *At every one-hot S (hard z), $H_{\mathrm{bil}}(S) = H_{\mathrm{mul}}(S) = H_{\mathrm{code}}(S) = H_2(z)$.*

Proof. H_{bil}: Lemma 1. H_{mul}: the product measure is a point mass at z. H_{code}: $S_{jk} \in \{0,1\}$ makes the within-codeword $-\log_2(d_j/V_{z_j})$ and the switch term the cut term of (1). □

Theorem 2 (bits-back identity)**.** *For every row-stochastic S,*

$$H_{\mathrm{code}}(S) \;=\; H_{\mathrm{bil}}(S) \;+\; \frac{1}{2m}\sum_j d_j\,\mathcal{H}(S_j), \qquad \mathcal{H}(S_j) = -\sum_k S_{jk}\log_2 S_{jk}.$$

Proof. Write $H_{\mathrm{code}} = T_{\mathrm{leaf}} + T_{\mathrm{switch}}$ with $T_{\mathrm{leaf}} = \sum_j \frac{d_j}{2m}\sum_k S_{jk}\big(-\log_2 \frac{d_j S_{jk}}{V_k}\big)$ (using $\sum_i A_{ij} = d_j$) and $T_{\mathrm{switch}} = \sum_{ij}\frac{A_{ij}}{2m}\sum_k S_{jk}(1-S_{ik})\big(-\log_2\frac{V_k}{2m}\big)$. Expanding the logarithm in T_{leaf} and using $\sum_j d_j S_{jk} = V_k$ gives $T_{\mathrm{leaf}} = H_1 + \sum_k \frac{V_k}{2m}\log_2\frac{V_k}{2m} + \frac{1}{2m}\sum_j d_j\mathcal{H}(S_j)$; and $T_{\mathrm{switch}} = -\frac{1}{2m}\sum_k (V_k - w_k)\log_2\frac{V_k}{2m}$ since $\sum_{ij} A_{ij}S_{jk} = V_k$ and $\sum_{ij} A_{ij}S_{ik}S_{jk} = w_k$. Summing yields $H_1 + \frac{1}{2m}\sum_k w_k \log_2\frac{V_k}{2m}$ plus the entropy term. □

Reading the identity. H_{code} is an operational expected code length for every S. The deployed objective H_{bil} equals that code length minus the (degree-weighted) entropy of the assignment itself — exactly a *bits-back* net rate (Wallace, 1990; Hinton and van Camp, 1993): the randomness invested in choosing a module can be recovered by the decoder, so only the net rate is charged. Two consequences: (i) H_{bil} has a precise coding semantics after all — but of a *stochastic-membership* code, not of any partition; (ii) $H_{\mathrm{bil}}(S) \geq H_{\mathrm{code}}(S) - \log_2 K$, so if H_{code} never undercuts the discrete optimum (Conjecture 1, supported at all census samples), the looseness of the deployed relaxation is at most $\log_2 K$ bits.

Example 1 (one edge, $K = 2$; Figure 1a)**.** *Both hard partitions of K_2 have $H_2 = H_1 = 1$; at uniform S, $H_{\mathrm{bil}} = \frac{1}{2}$, $H_{\mathrm{mul}} = 1$, $H_{\mathrm{code}} = \frac{3}{2}$: the plug-in rewards softness, the code pays for it.*

4 TIGHTNESS

4.1 THE MULTILINEAR EXTENSION IS EXACT

Theorem 3. *H_{mul} is affine in each row S_i (others fixed). Consequently $\min_{S\in\Delta_{K-1}^N} H_{\mathrm{mul}}(S) = \min_z H_2(z)$; every minimizer is a product distribution supported on hard minimizers; and $\partial H_{\mathrm{mul}}/\partial S_{ik} = \mathbb{E}_{Z_{-i}}[H_2(Z_{-i}, z_i{=}k)]$. (Proof: Appendix B.)*

Corollary 1. *Exact coordinate descent on H_{mul} (row-wise, each step solvable at a vertex) is precisely the greedy label local search used by discrete SE optimizers (e.g. the local-move phase of standard discrete SE optimizers); and $\partial H_{\mathrm{mul}}/\partial S_{ik}$ admits an unbiased one-sample estimator, giving a REINFORCE-style stochastic gradient for discrete SE with no relaxation bias.*

4.2 THE BILINEAR PLUG-IN IS LOOSE

Theorem 4 (graph-independent floor)**.** *For every graph and every $K \geq 2$, the uniform assignment $S_{ik} = 1/K$ gives $H_{\mathrm{bil}}(S) = H_1 - \log_2(K)/K$.*

Proof. $w_k = \mathbf{1}^\top A\mathbf{1}/K^2 = 2m/K^2$ and $V_k = 2m/K$, so $\frac{1}{2m}\sum_k w_k \log_2\frac{V_k}{2m} = K\cdot\frac{1}{2m}\cdot\frac{2m}{K^2}\log_2\frac{1}{K} = -\log_2(K)/K$. □

Corollary 2 (strict looseness)**.** *Whenever $H_2^* := \min_z H_2(z) > H_1 - \log_2(K)/K$, we get $\min_S H_{\mathrm{bil}}(S) < H_2^*$: a strictly soft assignment beats every hard partition. This holds e.g. for K_2 (1 vs. 0.5), all complete graphs in our census (K_6: 2.185 vs. 2.085; K_8: 2.571 vs. 2.500), and stars (S_9, $K{=}2$: 2.292 vs. 2.000).*

The floor is not an artifact of the two-level case. Deployed systems compose the relaxation level-wise, and the floor survives composition in closed form.

Theorem 5 (multi-level floor)**.** *Compose the plug-in with row-stochastic factors $S_1 \in \mathbb{R}^{N\times k_1}, S_2 \in \mathbb{R}^{k_1\times k_2}, \ldots, S_{L-1} \in \mathbb{R}^{k_{L-2}\times k_{L-1}}$, cumulative assignments $P_l = S_1\cdots S_l$, level volumes $V_l = P_l^\top d$, cuts $g_l = V_l - \mathrm{diag}(P_l^\top A P_l)$ and parent volumes $\tilde{V}_l = S_{l+1}V_{l+1}$ with $\tilde{V}_{L-1} = 2m\mathbf{1}$. At the uniform assignment on every level,*

$$H_{\mathrm{bil}}^{(L)} = H_1 - \log_2 k_1 - \sum_{l=1}^{L-2}\Big(1-\tfrac{1}{k_l}\Big)\log_2\tfrac{k_{l+1}}{k_l} + \Big(1-\tfrac{1}{k_{L-1}}\Big)\log_2 k_{L-1} \tag{2}$$

for every graph. *At $L = 2$ this is Theorem 4.*

Corollary 3 (depth buys headroom on every graph)**.** *Let $R = H_1 - H_{\text{bil}}^{(L)}$ be the headroom at the uninformative point; by* (2) *it depends only on the widths, and $R(k_1, k_2) > \max\{R(k_1), R(k_2)\}$ for all* 1,953 *pairs $2 \le k_2 < k_1 \le 64$ (checked numerically from the closed form); for instance $R(16) = 0.25$, $R(4) = 0.5$, $R(16, 4) = 0.625$ and $R(16, 4, 2) = 0.875$ bits. A deeper soft tree can therefore descend further below H_1 at the* maximally uninformative *assignment, on any graph whatsoever: a lower multi-level value is not by itself evidence of hierarchical structure, and depths must be compared against* (2)*, not zero.*

The floor mechanism is global (weak structure at the requested K). A second, *local* mechanism produces strictly soft optima even when the floor does not bind:

Example 2 (bowtie: planted overlap, exact values)**.** *Let G be two triangles sharing node* 2 *(N=5, $2m$=12), K=2. Exhaustive enumeration gives $H_2^* = \frac{7}{6} + \frac{1}{3}\log_2 3 \approx 1.6950$ (split off one triangle), and here the floor $H_1 - \frac{1}{2} = \frac{1}{6} + \log_2 3 \approx 1.7516$ lies* above *H_2^*. Yet the strictly soft $S^* = (e_1, e_1, (\frac{1}{2}, \frac{1}{2}), e_2, e_2)$ gives*

$$H_{\text{bil}}(S^*) \;=\; H_1 - \tfrac{2}{3} \;=\; \log_2 3 \;\approx\; 1.5850 \;<\; H_2^*,$$

and gradient descent from random initializations converges to it (recovered shared-node membership $(0.5000001, 0.4999999)$; ten seeds). The softness lands exactly on the planted overlap: the multilinear extension at S^* equals H_2^* to $7 \cdot 10^{-6}$, as Theorem 3 predicts for a mixture of the two hard optima. In the census, cycle and path graphs at K=3 show the same local mechanism (loose by 0.031–0.033 *bits with floors far above the hard optimum).*

Conjecture 1 (sandwich)**.** *For all S: $H_{\text{code}}(S) \ge H_{\text{mul}}(S) \ge H_{\text{bil}}(S)$, and $\min_S H_{\text{code}} = H_2^*$ (attained at vertices). Status: not proved. Empirics: zero violations of either inequality at all* 40,000 *Dirichlet samples of the census (20 instances × 2,000), where $H_{\text{mul}} - H_2^* \ge 0.0096$ and $H_{\text{code}} - H_2^* \ge 0.279$ bits; optimized H_{code} solutions were hard (GATE A3). Why a proof is delicate is discussed in Appendix C.*

4.3 Beyond structural entropy

Nothing in Definition 2 or in the proof of Theorem 3 uses the form of H_2: for any objective f on partitions, $\mathbb{E}f(Z)$ is affine in each row, exact ($\min_S \mathbb{E}f = \min_z f$) and has the same per-row derivative. What depends on f is how its plug-in f_{bil} (substitute $V_k(S)$, $w_k(S)$ and $n_k(S) = \sum_i S_{ik}$ into the formula) relates to $\mathbb{E}f$. All objectives are minimized.

Proposition 1 (plug-ins of other objectives)**.** *(a)* Row curvature. *If f_{bil} is concave in each row S_i, then $\min_S f_{\text{bil}} = \min_z f$. (b)* Pairwise objectives $f(z) = -\sum_{ij} B_{ij}\mathbf{1}[z_i{=}z_j]$ *satisfy $\mathbb{E}f(Z) = f_{\text{bil}}(S) - \sum_i B_{ii}(1 - \|S_i\|_2^2)$. Markov stability at every even time t ($B = \Pi P^t - \pi\pi^\top$ (Delvenne et al., 2010)) has $B_{ii} \ge 0$, so its plug-in is exact; modularity ($B = (A - dd^\top/2m)/2m$ (Newman and Girvan, 2004), $B_{ii} = -d_i^2/(2m)^2$) has $\min f_{\text{bil}} \ge \min f - (1 - \frac{1}{K})\sum_i d_i^2/(2m)^2$. (c)* Map equation *(Rosvall and Bergstrom, 2008). For undirected flow $A/2m$, exit rates $q_k = (V_k - w_k)/2m$ and $p_k = V_k/2m$, Neuromap's plug-in L_{bil} and the memoryless stochastic two-level map code L_{code} satisfy $L_{\text{code}} = L_{\text{bil}} + \frac{1}{2m}\sum_j d_j \mathcal{H}(S_j)$, as in Theorem 2; but at the uniform assignment $L_{\text{bil}} = H_1 + \phi(K)$, $\phi(K) = -q\log_2 q - 2\log_2 K + (2 - \frac{1}{K})\log_2(2K-1) > 0$ with $q = 1 - \frac{1}{K}$, which lies* above *the one-module code length H_1: no floor. (d)* Dasgupta's cost *(Dasgupta, 2016) of the flat tree (root, K clusters, leaves): $\mathbb{E}f(Z) = f_{\text{bil}}(S) + \sum_{i<j} A_{ij}\sum_k S_{ik}S_{jk}(2 - S_{ik} - S_{jk}) \ge f_{\text{bil}}(S)$, and $f_{\text{bil}} = mN(1 - \frac{K-1}{K^2})$ at the uniform S on every graph.*

Table 1 runs the census on eight objectives, including the regularized losses that DMoN and MinCutPool actually train; its preregistered checks G1–G3 all passed (Table 2). Looseness is not an SE peculiarity (Figure 1b). DMoN's collapse term is smallest at balanced soft assignments, so its loss is floor-captured on 6 instances and loose on 14; MinCutPool's loss is loose on 10 and Dasgupta's plug-in on 9. Modularity is loose only on the cycle and path at K=3, within its bound, and Markov stability never, as proved. The map-equation plug-in carries SE's bits-back identity but not its floor, and never beat the partitions by more than $7 \cdot 10^{-4}$ bits. Tightness is decided by the plug-in's row curvature and by whether its uninformative point lies below the partitions, not by whether the objective is a code length.

Table 1: Plug-in relaxations of eight clustering objectives on the census of §5 (20 instances, exhaustive hard optima, 5 restarts; extension E1). "floor": the plug-in at uniform S already beats every partition; "loose": the optimized plug-in beats every partition by more than $10^{-3}\max(1, |f^*|)$ (SE: 10 at GATE A2's 0.01-bit threshold); last column: largest such margin, in the objective's units.

objective (deployed as)	plug-in vs. $\mathbb{E}f$	floor	loose	max margin
SE (soft-SE library, LSEnet, DeSE)	bits-back, Thm. 2	6	12	0.321 bits
map equation (Neuromap)	bits-back, Prop. 1(c)	0	0	$7 \cdot 10^{-4}$ bits
modularity (DMoN term)	Gini refund, (b)	0	2	0.010
Markov stability, $t = 2$	Gini refund, exact, (b)	0	0	0
Dasgupta, flat tree	pairwise refund, (d)	5	9	8.0 (rel. max 6.3%)
normalized cut	—	0	3	0.042
DMoN loss (with collapse term)	—	6	14	0.185
MinCutPool loss (with orth. term)	—	0	10	0.219

Table 2: Every preregistered gate and its verdict, grouped by experiment, with values as reported in §5 and Appendix D. $\pm$: pooled standard error over 5 seeds. "expl.": exploratory, no pass/fail rule. Failures are not reinterpreted.

gate	what it tests	result	verdict
Census, 20 instances (GATE A, C3; §5)			
A1	H_{mul} never below the hard minimum	0/40,000 points; min. margin $+0.0096$	**PASS**
A2	H_{bil} strictly loose (> 0.01 bits) somewhere	10/20 instances; max. gap 0.321 bits	**PASS**
A3	optimized H_{code} hard, never below H_2^*	softness $< 10^{-5}$; 0 violations	**PASS**
A4	sandwich ordering of Conj. 1	0 violations	expl.
C3	rounding bound at the optima	holds at every optimum	**PASS**
Soft-membership fixture and temperature (GATE B, C1, C2; extension E3)			
B1	H_{bil} membership error below hard	4/5 seeds; 16/20 in E3	**PASS**
B2	overlap selectivity above null 95th pct.	1/5 seeds; 4/20 in E3	**FAIL**
B3	optimized H_{code} near-vertex	softness $< 3 \cdot 10^{-6}$	expl.
C1	reduction at hard S	max. deviation 0	**PASS**
C2a	$\tau = 30$ reaches the floor	0.008 bits from the floor	**PASS**
C2b	$\tau = 0.01$ rounds to the hard baseline	4.398 vs. 4.330 bits	**FAIL**
Real graphs (GATE D; extension E2; §5.3)			
D1	soft beats hard ST, same encoder (Cora)	$\Delta\mathrm{NMI} = +0.136 \pm 0.028$	**SUPPORTED**
R1	H_{mul} reaches lower hard SE than H_{bil}	0/5 seeds, Cora and CiteSeer	**FAIL**
R3	D1 holds against Gumbel-softmax ST	$+0.025 \pm 0.022$ / $+0.006 \pm 0.009$	not robust
Other objectives, census (extension E1; §4.3)			
G1	identities of Prop. 1	max. deviation $1.4 \cdot 10^{-12}$	**PASS**
G2	$\mathbb{E}f$ never below the hard minimum	0/40,020 points per objective	**PASS**
G3	floor predictions from closed forms	all held (SE 6, map eq. 0, mod. 0)	**PASS**

5 Experiments

All experiments were preregistered before the decisive runs; every gate uses ≥ 5 seeds (two descriptive arms in Table 3 have 2); the planted-overlap experiment carries a degree-preserving rewiring null. Extensions E1–E3 were preregistered separately. Table 2 lists every gate with its verdict verbatim, failures included; the preregistrations, code, fixtures and machine-readable results are in the supplementary material.

5.1 Exhaustive census (GATE A)

20 instances (10 graphs $\times K \in \{2, 3\}$, $n \leq 9$, all K^n assignments enumerated), 2,000 Dirichlet probes each, 5-seed Adam optimization of H_{bil} and H_{code} (Table 2, first block). H_{mul} never fell below the exhaustive hard minimum, as the tightness theorem requires; H_{bil} is strictly loose on 10/20 instances, with gap up to 0.32 bits (star, K=3) and soft optima that are exactly the uniform floor on the complete graphs and the star; optimized H_{code} solutions were hard (softness, the

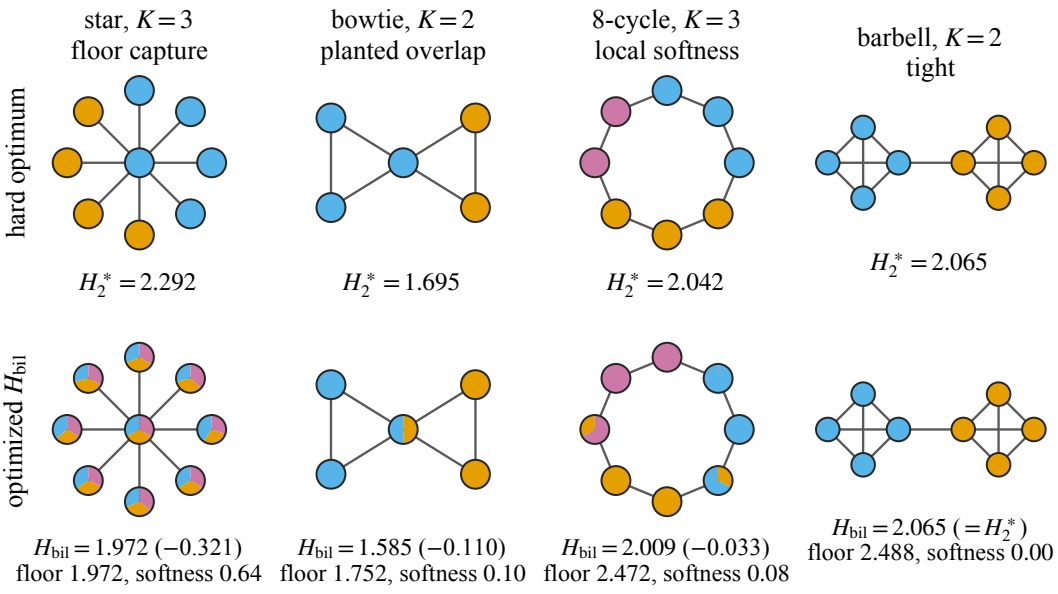

Figure 2: **Case study: where the deployed plug-in puts its softness.** Top: the exhaustive hard optimum (node colour = module). Bottom: the optimized H_{bil} assignment on the same layout, each node a pie of its row S_i (columns matched to the hard modules). Below each graph: H_2^*, the plug-in value and its margin below H_2^*, the floor $H_1 - \log_2(K)/K$ and the mean softness. Instances chosen by the rule in §5; best of the released 5 (bowtie: 10) optimizer seeds.

mean of $1 - \max_k S_{ik}$, is $< 10^{-5}$). Figure 4 plots realized looseness against floor headroom $H_2^* - (H_1 - \log_2(K)/K)$.

Case study. Figure 2 draws the optimized H_{bil} assignment beside the exhaustive hard optimum on four graphs chosen by a fixed rule: the floor-captured census instance with the largest gap (star), the loose instance with the largest gap whose floor lies above H_2^* (8-cycle), the first clustered $K{=}2$ instance in census order (barbell), and the bowtie. Floor capture spreads every node over all modules; local softness splits only boundary nodes (the bowtie's shared node exactly in half); on clear clusters the plug-in returns the hard optimum. Assignments come from re-running the released optimizer and seeds; values match the result files to 10^{-9}.

5.2 Soft-membership fixture (Gate B) and its honest reading

Deterministic core: the bowtie fixture (§4) passes all four checks (reduction $6.7 \cdot 10^{-16}$; overlap membership recovered to 10^{-7}; multilinear value $= H_2^*$ to $7 \cdot 10^{-6}$; temperature limits as predicted). Statistical version: 2 blocks $\times$ 16 core nodes + 4 overlap nodes at $(\frac{1}{2}, \frac{1}{2})$, $p_{\text{in}} = 0.55$, $p_{\text{out}} = 0.05$, 5 graph seeds. Any hard assignment has mean-L_1 membership error $\geq 4/36 = 0.111$ by construction. B1 (soft beats hard): PASS, 4/5 seeds (errors 0.086–0.111 vs. hard 0.111). B2 (overlap-selectivity beats a degree-preserving null's 95th percentile): **FAIL**, 1/5 seeds ($\Delta_{\text{real}} = 0.00$–$0.115$ vs. null95 0.072–0.117). The failure is informative: with strong blocks the H_{bil} optimum is near-hard and its overlap signal is weak relative to the softness the null graphs' high-degree nodes acquire. With 20 graph seeds (extension E3, identical pipeline; seeds 0–4 reproduce the values above exactly) B2 wins on 4/20, and on 4/20 again for a stronger fixture (core 8, $p_{\text{in}}{=}0.8$): above the 1/20 expected without selectivity (one-sided binomial $p = 0.016$), far below the 16/20 the gate requires. The selectivity is real but weak; the original failure was not only low power. An exploratory sweep (Figure 5; designed after a first test run) shows soft beating hard on mean error in all 16 cells (a seed majority in 15), with overlap-row softness up to 0.24 at core size 8, where degrees are comparable to module volumes, and ≤ 0.14 at core size 16. The statistical version therefore calibrates how weak the effect is; it is not a general win. A soft-SE margin over a degree-matched null is evidence about the bits-back rate H_{bil}, to be reported against the floor and a matched null, not as the SE of a partition. B3: the optimized H_{code} arm collapsed to near-vertex solutions as predicted (softness $< 3 \cdot 10^{-6}$), though often to poor

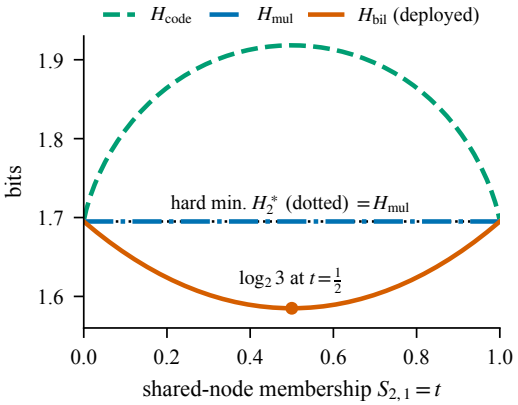

Figure 3: The three definitions along the bowtie's overlap coordinate (all rows hard except the shared node's, $S_2 = (t, 1-t)$). H_{mul} is affine with minimum at the vertices (exactness); H_{code} penalizes softness; H_{bil} dips 0.110 bits below every hard partition, with minimum exactly at the planted membership $t = \frac{1}{2}$.

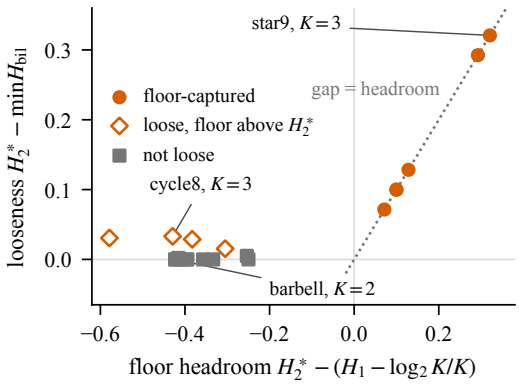

Figure 4: Census, one point per graph–K instance (x: floor headroom; y: realized looseness of the optimized H_{bil}). Points on the dotted diagonal are floor-captured; hollow points above zero at negative headroom are local softness. Labelled: the instances of Figure 2.

vertices (the entropy term hardens assignments before structure is found), so H_{code} does not express the planted overlap here; H_{mul} was not optimized on this fixture, and by Theorem 3 it can be soft only by mixing tied hard optima, never *rewarding* overlap.

5.3 Encoder controls on Cora and CiteSeer (gate D, extension E2)

The control demanded by requirement (iii) of §1: same 2-layer GCN, same graph, same budget (300 Adam iterations $\times$ 3 restarts kept by objective value, K=7, 5 seeds; labels used only for the final NMI). Arms: soft H_{bil} objective; *hard* objective through a straight-through (ST) argmax (the forward pass evaluates canonical H_2 exactly, by Theorem 1); soft objective with no encoder (free logits); and the library's Sinkhorn-head baseline (Table 3, first block). GATE D1 — soft-vs-hard difference under the same encoder — **SUPPORTED**: $\Delta\text{NMI} = +0.136$ with pooled standard error 0.028 (the soft arm wins on every seed, worst-case margin $+0.049$). Both GCN arms use only 3 of the 7 modules (4 on one hard-arm seed), while the free-logit arm and the Sinkhorn head use all 7, and the Sinkhorn head scores higher than either gate arm.

Extension E2 (preregistered before it ran) re-implements the protocol, adds CiteSeer, a Gumbel-softmax ST hard arm, and the multilinear objective, trained through the exact per-row derivative of Theorem 3 estimated from one sample per step (a local-expectation gradient (Titsias and Lázaro-Gredilla, 2015)); the re-implementation reproduces the soft and ST arms (Table 3, second block). Two of three outcomes went against our expectations. (1) R1 **failed**: the multilinear arm never reached a lower hard SE than the plug-in (0/5 seeds on both datasets; 1/5 each with free logits in place of the GCN) and NMI did not differ (-0.030 ± 0.025 on Cora). The plug-in optima are near-hard (softness ≤ 0.007 with the GCN, ≤ 0.014 with free logits) and their value is within 0.01 bits of the SE of their own rounding: on these sparse graphs the looseness never binds, so exactness buys no better partitions. The same holds on planted SBMs up to $N = 1024$, where no optimized plug-in of SE, the map equation, modularity or Dasgupta's cost beat the best hard partition found (0/36 cells each; Appendix D). (2) D1 is mostly an estimator effect: against the Gumbel-softmax hard arm the soft advantage is 0.025 ± 0.022 NMI on Cora and 0.006 ± 0.009 on CiteSeer (R3 not robust); only plain ST is worse, and no E2 contrast survives Holm correction over the seven NMI contrasts (Welch 95% CIs: D1 $[+0.07, +0.21]$, $p_{\text{Holm}} = 0.02$; ST on Cora $[+0.02, +0.16]$, $p_{\text{Holm}} = 0.12$). (3) Every GCN arm's mean hard SE is at least 0.68 bits above the hard SE of encoder-free local search on Cora (9.39 vs. 8.71), whose partitions have far lower NMI (0.09): lower SE is not better label agreement (free-logit, map-equation, modularity and PubMed arms: Appendix D).

Table 3: Encoder controls: NMI against labels (mean ± sample sd over 5 seeds unless noted; higher is better), hard SE in bits of the argmax partition (lower is better, but lower SE is not better label agreement, §5.3), and modules used, with K = number of classes. No entry is bold: the arms test a mechanism, not a leaderboard. n/r: not recorded in the original run; —: not run (CiteSeer was added in E2). GCN encoder unless noted.

	Cora (K=7)			CiteSeer (K=6)		
arm	NMI↑	hard SE↓	mod.	NMI↑	hard SE↓	mod.
GATE D, original run						
soft H_{bil}	$.411 \pm .027$	n/r	3	—	—	—
hard, ST	$.274 \pm .056$	n/r	3–4	—	—	—
soft H_{bil}, no encoder (2 seeds)	$.153 \pm .022$	n/r	7	—	—	—
Sinkhorn head (2 seeds)	$.468 \pm .021$	n/r	7	—	—	—
Extension E2, re-implementation						
soft H_{bil}	$.399 \pm .035$	9.39	3–4	$.246 \pm .012$	9.59	3
multilinear H_{mul}	$.369 \pm .042$	9.41	3	$.231 \pm .027$	9.60	3
hard, ST	$.308 \pm .057$	9.56	3–4	$.205 \pm .036$	9.80	2–3
hard, Gumbel-softmax ST	$.375 \pm .034$	9.40	3	$.240 \pm .016$	9.59	3
local search, no encoder	$.094 \pm .021$	8.71	7	$.022 \pm .004$	8.94	6

6 Related work

Structural entropy and encoding trees: Li and Pan (2016). Differentiable SE: LSEnet (Sun et al., 2024) gives level-wise soft assignments with an equivalence at binary assignments and an $O(\epsilon)$ near-hard bound; DeSE (Zhang et al., 2025) builds soft-assignment SE into deep clustering; SEP (Wu et al., 2022) pools with encoding trees. All use Definition 1-type plug-ins; none, to our knowledge, states the floor, the strict looseness, the bits-back identity or the exact multilinear alternative. HypCSE (Zeng et al., 2026) relaxes SE through a hyperbolic lowest-common-ancestor softmax, outside our analysis. DMoN, MinCutPool and Neuromap apply the plug-in pattern to modularity (Newman and Girvan, 2004), normalized cut and the map equation (Rosvall and Bergstrom, 2008); §4.3 settles their looseness, and that of Markov stability (Delvenne et al., 2010) and Dasgupta's cost (Dasgupta, 2016). Closest to our tightness results, neural combinatorial optimization trains on the expected cost under independent rounding (Karalias and Loukas, 2020) and certifies rounding for entry-wise concave relaxations (Wang et al., 2022); Theorem 3 and Proposition 1(a) instantiate these for partitions, and we add each plug-in's code, its floor and which deployed losses are loose. Our multilinear gradient is a local-expectation gradient (Titsias and Lázaro-Gredilla, 2015). Relaxation tools: multilinear extensions (Calinescu et al., 2011); temperature relaxations (Jang et al., 2017; Maddison et al., 2017); bits-back coding (Wallace, 1990; Hinton and van Camp, 1993). Overlapping memberships: MMSB (Airoldi et al., 2008).

7 Limitations

(1) Conjecture 1 is unproven; the census supports it at every sampled point (10 graphs, $n \leq 9$), but a proof or counterexample is the main open item. (2) The rounding bound's constant is loose and its v is path-dependent; a cleaner statement should exist. (3) The floor generalizes per level (Theorem 5), but the identity and the exactness verdict are stated for the two-level undirected case only, and Corollary 3 is checked over a width grid, not proved. (4) The real-graph controls use one encoder family, K fixed by the class count and 300 iterations; with this budget every GCN arm used only 2–4 modules and the map-equation arms collapsed to one, so they test the objectives under a weak optimizer; DeSE/LSEnet were not trained as baselines, and PubMed was run graph-only with free logits. (5) The multilinear extension is trained with a one-sample per-row estimator, which on our graphs did not beat the plug-in. The generality results (Proposition 1) cover flat two-level objectives; hierarchical Dasgupta relaxations and nested-SBM description lengths are not analysed. (6) B2 (also at 20 seeds) and C2b failed as preregistered: H_{bil}'s overlap selectivity at realistic block strengths is weak and cold-start optimization is brittle.

REPRODUCIBILITY STATEMENT

All formal claims are stated with their assumptions in the main text and proved either there or in the appendix. Every experiment reported here is a closed specification: the constructions, their parameters, the seed set, the null model and its draw count, and the acceptance criteria are given in the experimental sections, and the criteria were fixed before the decisive runs. Code sufficient to regenerate every number and figure, together with the raw logged outputs and the preregistration file recording each criterion and its verdict, is included as anonymous supplementary material.

AI USE STATEMENT

We used generative AI tools on the following tasks from ICLR's required-disclosure list: helping to develop the conceptual framework, helping to formulate mathematical claims, providing ingredients for and assisting in the writing of proofs, proposing and refining hypotheses, giving feedback on experimental design, implementing methods, and helping to interpret results. We did *not* use generative AI tools to generate synthetic data sets reported as measurements, to clean or reformat data sets, for translation, or for qualitative or thematic data analysis. From the recommended-disclosure list we additionally used generative AI tools to draft and edit parts of this paper, to create and edit figures, to identify and summarise related literature, to format references, and to suggest a structure for the paper.

We have reviewed all AI-assisted work. Every theorem statement and proof appearing in this paper was checked line by line by the authors; every numerical value reported was regenerated from the released code and compared against the logged experimental outputs; every preregistered criterion and its verdict was recorded before the corresponding run and is reported verbatim, including the failures. We take responsibility for the final content of this work, including all text, claims and artifacts produced with the aid of generative AI.

REFERENCES

A. Li and Y. Pan. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62(6):3290–3339, 2016.

L. Sun, Z. Huang, H. Peng, Y. Wang, C. Liu, and P. S. Yu. LSEnet: Lorentz structural entropy neural network for deep graph clustering. In *ICML*, PMLR 235, 2024. arXiv:2405.11801.

J. Zhang, H. Peng, L. Sun, G. Wu, C. Liu, and Z. Yu. Unsupervised graph clustering with deep structural entropy. In *KDD*, 2025. arXiv:2505.14040.

X. Zeng, H. Peng, A. Li, C. Liu, L. He, and P. S. Yu. Hierarchical state abstraction based on structural information principles. arXiv:2304.12000, 2023.

X. Zeng, H. Peng, and A. Li. Effective exploration based on the structural information principles. arXiv:2410.06621, 2024.

J. Wu, X. Chen, K. Xu, and S. Li. Structural entropy guided graph hierarchical pooling. In *ICML*, 2022.

A. Tsitsulin, J. Palowitch, B. Perozzi, and E. Müller. Graph clustering with graph neural networks. *JMLR*, 24(127):1–21, 2023.

F. M. Bianchi, D. Grattarola, and C. Alippi. Spectral clustering with graph neural networks for graph pooling. In *ICML*, 2020.

G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Computing*, 40(6):1740–1766, 2011.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with Gumbel-softmax. In *ICLR*, 2017.

C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: a continuous relaxation of discrete random variables. In *ICLR*, 2017.

C. S. Wallace. Classification by minimum-message-length inference. In *Advances in Computing and Information*, 1990.

G. E. Hinton and D. van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *COLT*, 1993.

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *JMLR*, 9:1981–2014, 2008.

C. Blöcker, C. Tan, and I. Scholtes. The map equation goes neural: Mapping network flows with graph neural networks. In *NeurIPS*, 2024. arXiv:2310.01144.

M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *PNAS*, 105(4):1118–1123, 2008.

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

J.-C. Delvenne, S. N. Yaliraki, and M. Barahona. Stability of graph communities across time scales. *PNAS*, 107:12755–12760, 2010.

S. Dasgupta. A cost function for similarity-based hierarchical clustering. In *STOC*, pp. 118–127, 2016.

M. K. Titsias and M. Lázaro-Gredilla. Local expectation gradients for black box variational inference. In *NeurIPS*, 2015.

N. Karalias and A. Loukas. Erdos goes neural: an unsupervised learning framework for combinatorial optimization on graphs. In *NeurIPS*, 2020. arXiv:2006.10643.

H. Wang, N. Wu, H. Yang, C. Hao, and P. Li. Unsupervised learning for combinatorial optimization with principled objective relaxation. In *NeurIPS*, 2022. arXiv:2207.05984.

G. Zeng, H. Peng, A. Li, L. Sun, C. Liu, S. Li, Y. Pan, and P. S. Yu. Hyperbolic continuous structural entropy for hierarchical clustering. In *AAAI*, 2026. arXiv:2512.00524.

A Additional exploratory figure

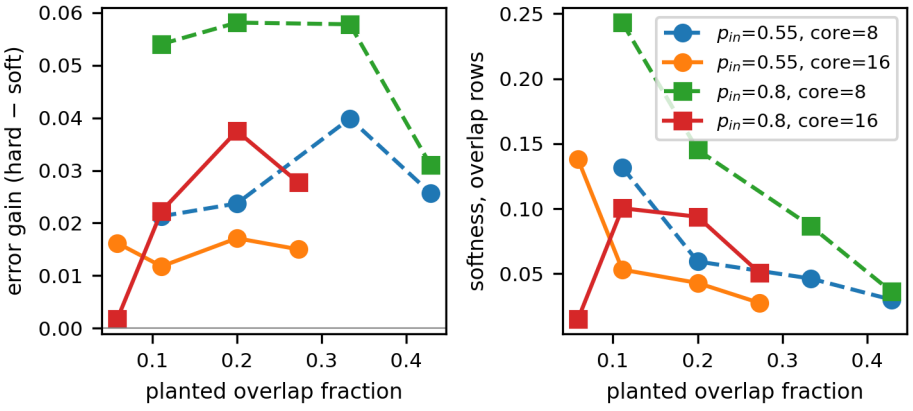

Figure 5: Exploratory regime sweep of the statistical fixture: 16 cells ($p_{\text{in}} \in \{0.55, 0.8\} \times$ core size $\{8, 16\} \times \{2, 4, 8, 12\}$ overlap nodes), 5 seeds per cell. Left: mean membership-error improvement of the H_{bil} solution over the hard baseline (positive = soft better). Right: mean softness $1 - \max_k S_{ik}$ of the planted-overlap rows.

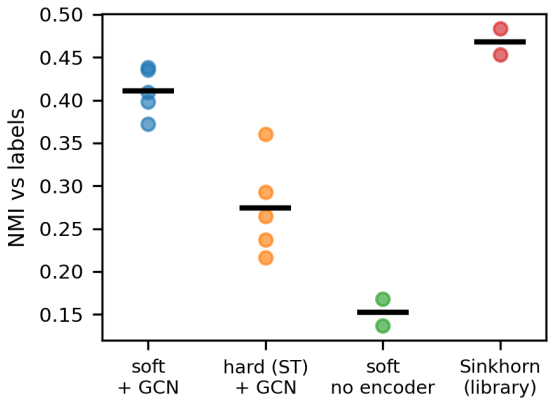

Figure 6: Cora encoder control: NMI against labels per seed (dots) and arm mean (bars); gate arms 5 seeds, auxiliary arms 2.

B PROOFS AND REMARKS MOVED FROM THE MAIN TEXT

Proof of Theorem 3. $H_{\mathrm{mul}}(S) = \sum_z \big(\prod_i S_{iz_i}\big) H_2(z)$ is multilinear in the rows. For the minimum: given any S, replacing row i by the vertex minimizing the affine restriction never increases H_{mul}; iterating over rows terminates at a vertex with value $\le H_{\mathrm{mul}}(S)$, and vertices have $H_{\mathrm{mul}} = H_2$ (Theorem 1). Conversely every vertex value is attainable. The support claim: if some z' in the support of the product measure had $H_2(z') > \min$, the expectation would exceed the minimum. The gradient is the standard multilinear-extension derivative. □

Proof of Theorem 5. At uniform assignments $(P_l)_{vj} = 1/k_l$, so $V_l = (2m/k_l)\mathbf{1}$, $\mathrm{diag}(P_l^\top A P_l) = (2m/k_l^2)\mathbf{1}$ and hence $g_l = (2m/k_l - 2m/k_l^2)\mathbf{1}$, while $\tilde{V}_l = (2m/k_{l+1})\mathbf{1}$ for $l < L-1$. The leaf term contributes $H_1 - \log_2 k_1$ and level l contributes $-(1-1/k_l)\log_2(k_{l+1}/k_l)$, with the top level contributing $+(1-1/k_{L-1})\log_2 k_{L-1}$. Summing gives (2); we verify it against direct evaluation of the composed objective to $< 10^{-15}$ on random graphs at $L \in \{2,3,4\}$. □

Proof sketch of Proposition 1. (a) As for Theorem 3: a concave function of one row attains its minimum at a vertex. (b) $\mathbb{E}\mathbf{1}[Z_i{=}Z_j] = S_i^\top S_j$ for $i \neq j$ and 1 for $i = j$, while the plug-in uses $\|S_i\|^2$ on the diagonal. For $t = 2s$, $(P^t)_{ii} = \sum_l \lambda_l^{2s} u_l(i)^2 \ge u_1(i)^2 = \pi_i$ with u_l the eigenvectors of $D^{-1/2}AD^{-1/2}$, so $B_{ii} \ge 0$ and the row restriction $-B_{ii}\|S_i\|^2 +$ affine is concave; for modularity $1 - \|S_i\|^2 \le 1 - \frac{1}{K}$. (c) The computation of Theorem 2 with exit, index and module codebooks whose probabilities are the realized use rates; $\phi(K) > 0$ since $2\log_2(2 - \frac{1}{K}) > \log_2(2K-1)/K$ for $K \ge 2$. (d) $\mathbb{E}\big[n_{Z_i}\mathbf{1}[Z_i{=}Z_j]\big] = \sum_k S_{ik}S_{jk}\big(2 + \sum_{l\neq i,j} S_{lk}\big)$. □

Depth and the floor in practice. This matters for practice because the level-wise composition in (2) is exactly the one used by deployed multi-level relaxations, and because it interacts with the discrete fact that minimum structural entropy is itself monotone in tree depth: neither the discrete objective nor the deployed relaxation can, on its own, prefer a shallower tree.

C MATERIAL MOVED FROM THE MAIN TEXT

Figure 7 summarizes the three definitions and what each computes.

Continuous data. Nothing below requires A to be a combinatorial graph: all three definitions, the identity, the floor, and the tightness results hold for any symmetric nonnegative A with zero diagonal. For points $X = \{x_i\} \subset \mathbb{R}^d$ with no pre-built graph, take a symmetric kernel affinity (e.g. Gaussian mutual-kNN, $A_{ij} = \exp(-\|x_i - x_j\|^2/\sigma^2)$ on mutual neighbors, symmetrized) and optimize S —

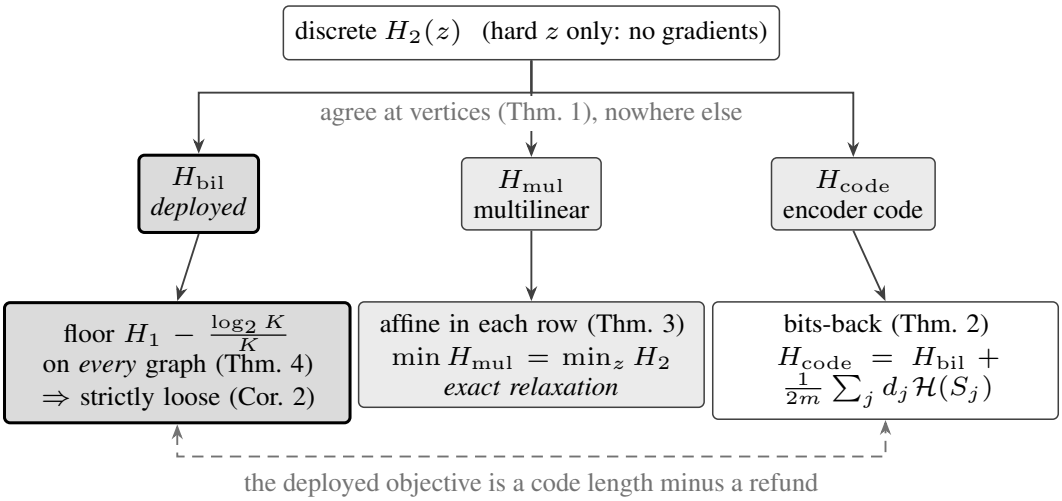

Figure 7: The three ways to extend a discrete clustering objective to the interior of the simplex product, and what each one turns out to compute. They coincide on hard assignments and differ everywhere else, so the choice is not an implementation detail. The multilinear extension is the faithful relaxation; the deployed plug-in is a bits-back net rate with a floor that does not depend on the graph, which is why a sufficiently soft assignment can beat every partition.

or an encoder producing S — end-to-end; no pre-clustering is required, so no discretization step precedes the objective. The kernel itself (k, σ) remains a modeling degree of freedom that this paper does not resolve; the floor calibration and the matched-null protocol below apply to it unchanged.

Why the sandwich conjecture is delicate. Writing W_k, V_k for the (random) internal weight and volume under independent rounding, $H_{\mathrm{mul}} - H_{\mathrm{bil}} = \frac{1}{2m}\sum_k \big[\mathrm{Cov}(W_k, \log_2 V_k) + \mathbb{E}[W_k](\mathbb{E}[\log_2 V_k] - \log_2 \mathbb{E}[V_k])\big]$, where the covariance is ≥ 0 by the Harris/FKG inequality (W_k and $\log_2 V_k$ are non-decreasing in the independent indicators $\mathbf{1}[z_i{=}k]$) while the Jensen term is ≤ 0; a proof must show the association term dominates. Per-row analysis shows H_{code} restricted to one row is a sum of terms $(q_k + 2r_k s_k)\log_2 \frac{c_k + d_i s_k}{2m} - \frac{d_i}{2m} s_k \log_2 s_k$ whose concavity can fail coordinate-wise, so vertex-attainment does not follow from the simple concavity argument; settling it is open.

C.1 Rounding

Proposition 2 (volume-restricted rounding bound)**.** *Let $\hat{z} = \arg\max$-rounding of S, $\varepsilon_i = 1 - \max_k S_{ik}$, and let $v > 0$ lower-bound all module volumes appearing along the row-by-row interpolation from S to the one-hot matrix of $\hat{z}$ (over modules with nonzero internal weight). Then*

$$\big|H_{\mathrm{bil}}(S) - H_2(\hat{z})\big| \;\leq\; \frac{2}{2m}\Big(\sum_i d_i \varepsilon_i\Big)\left(2\log_2 \frac{2m}{v} + \frac{2m}{v \ln 2}\right).$$

Proof sketch. Change rows one at a time. A single row change from s to t perturbs each w_k by at most $2d_i|s_k - t_k|$ (since $(AS)_{ik} \leq d_i$) and each V_k by $d_i|s_k - t_k|$; splitting the difference of $w_k \log_2(V_k/2m)$ terms accordingly and using $|\log_2 V' - \log_2 V| \leq |V' - V|/(v \ln 2)$, $\sum_k w_k \leq 2m$, $\|s - t\|_1 = 2\varepsilon_i$ for t the rounding vertex, and summing over rows gives the display (the stated constant is $2\times$ the tightest version of this argument, kept for the empirical check's path-sampled v). □

The bound is only useful near-hard (small $\sum_i d_i \varepsilon_i$ and volumes bounded away from 0): at the census's near-hard optima it holds everywhere (preregistered check C3, PASS), while for solutions captured by the floor it is vacuous — consistent with the fact, established above, that no rounding statement *can* rescue tightness there. This also delimits LSEnet's $O(\epsilon)$ approximation claim (Sun et al., 2024): such bounds are conditional on the optimizer staying near-hard, and Corollary 2 shows the unconstrained optimizer need not.

Seeds per Cora arm. The two gate arms are reported for all 5 seeds (seeds 2–4 computed with a functionally identical implementation, which reproduces the gate verdict on those seeds alone); the per-seed results are in the supplement. The two auxiliary arms have 2 seeds, are descriptive and enter no gate; the 5-seed free-logit arms of E2 supersede the no-encoder arm. Figure 6 shows the per-seed values.

D EXTENSION EXPERIMENTS E1–E3: DETAILS

All three were preregistered before the first run; the preregistration, a run log with every command, seed set and wall time, and the list of amendments are in the supplementary material. The amendments include one post hoc arm set (free logits on Cora and CiteSeer), which we label as such. The supplementary code implements each extension, evaluates the closed-form bounds, self-tests the identities, move tables and estimator unbiasedness, and prints every number below.

E1: objectives and the map code. All minimized; the plug-in substitutes $V_k(S)$, $w_k(S)$, $n_k(S)$. SE and the map equation are in bits; modularity and Markov stability ($t = 2$) are negated; normalized cut is $\sum_{k\le K} \mathrm{cut}_k/V_k$ with an empty module contributing 1 (the plug-in's limit as a column empties); the DMoN loss is $-Q_{\mathrm{bil}} + \frac{\sqrt{K}}{N}\|\sum_i S_i\|_2 - 1$; the MinCutPool loss is $-\mathrm{tr}(S^\top AS)/\mathrm{tr}(S^\top DS) + \|S^\top S/\|S^\top S\|_F - I_K/\sqrt{K}\|_F$; Dasgupta's cost of the flat tree is $\sum_{i<j} A_{ij}$ (n_{z_i} if $z_i{=}z_j$ else N). The memoryless map code of Proposition 1(c) resamples $k \sim S_i$, $l \sim S_j$ at each step $i \to j$; if $k \neq l$ it emits the exit codeword of k (probability $q_k/(q_k + p_k)$ in module k's book) and the entry codeword of l ($q_l/\sum_{k'} q_{k'}$ in the index book), then node j in module l's book (probability $d_j S_{jl}/(2m(q_l + p_l))$). The identity was checked against this first-principles computation (deviation $\le 1.4 \cdot 10^{-12}$ over 200 random S per instance). For the map equation the multilinear value fell below the plug-in at 108 of 40,000 probes, so the analogue of the sandwich's middle inequality fails there; for Markov stability it lies below the plug-in everywhere, as (b) requires. The largest map-equation margin, $7 \cdot 10^{-4}$ bits (path, $K{=}3$, softness 0.027), is below the preregistered tolerance. A warm start at a hard optimum (reported separately) changed no count; its largest map-equation margin was $8 \cdot 10^{-4}$ bits.

E1b: scale. Planted 4-block SBMs with mean degree 10, $N \in \{128, \dots, 1024\}$, mixing $\mu \in \{0.2, 0.4, 0.6\}$, 3 seeds (36 cells), $K = 4$. For SE, the map equation, modularity and Dasgupta's cost, the optimized plug-in was near-hard (softness ≤ 0.004) and never beat the best hard partition found (every margin negative, the largest $-1.4 \cdot 10^{-4}$), and the uniform value never fell below it (SE: at least 0.53 bits above). The preregistered prediction that the SE gap would stay bounded away from zero at high μ was wrong: at this density the floor stays far above the hard optimum even at $\mu = 0.6$.

E2: free logits, map equation, modularity, PubMed. The original no-encoder arm's low loss and low NMI are not an exploitation of looseness: re-run with free logits the plug-in is near-hard (softness 0.003) and reaches hard SE 8.68 bits on Cora, below local search. Under the GCN every map-equation arm on both datasets collapsed to a single module ($L = H_1$); with free logits all arms used all K modules and the plug-in's rounding (9.49 bits on Cora) was below local search (9.55). For modularity the multilinear objective (exact in closed form) reached higher modularity than the plug-in with the GCN on Cora ($5/5$ seeds, 0.602 vs. 0.575) but not on CiteSeer ($1/5$), nor with free logits ($2/5$, $1/5$); the excess bound of (b) is $8.9 \cdot 10^{-4}$ on Cora and $6.3 \cdot 10^{-4}$ on CiteSeer, so the two objectives nearly coincide there. PubMed (19,717 nodes; the offline copy has no features, so free logits, $K = 3$): the multilinear arm reached lower hard SE ($4/5$ seeds; 11.960 vs. 11.971 bits) and map-equation length ($4/5$) than the plug-in, but local search was lower than both (SE 11.918), and every arm's NMI was below 0.04.

E3: B2 with 20 seeds. Primary fixture (core 16, $p_{\mathrm{in}}{=}0.55$): B1 $16/20$, B2 $4/20$; secondary (core 8, $p_{\mathrm{in}}{=}0.8$): B1 $14/20$, B2 $4/20$; one-sided binomial $p = 0.016$ against a 0.05 per-seed rate in both. By the preregistered rule: weak but real selectivity; the original failure was partly power.

E TEMPERATURE

Parameterize $S = \mathrm{softmax}(\theta/\tau)$ with logits bounded, $\|\theta\|_\infty \le B$.

Proposition 3. *(i) As $\tau \to 0$, $S \to$ one-hot at $\arg\max_k \theta_{ik}$ (generic θ) and all three objectives converge to $H_2(\hat{z})$. (ii) As $\tau \to \infty$, $S \to$ uniform and $H_{\text{bil}} \to H_1 - \log_2(K)/K$ for every graph: the high-temperature limit of the deployed objective is structure-blind, and its gradient signal about the graph vanishes.*

Empirically (Figure 8; the statistical soft-membership fixture of §5, graph seed 0, logits clipped to ± 5): at $\tau \in [0.1, 1]$ the optimized solutions round to the hard baseline exactly; at $\tau = 30$ the optimized value sits 0.008 bits from the floor (preregistered check C2a, PASS); but at $\tau = 0.01$ optimization enters hard and sticks in a worse local optimum (4.398 vs. baseline 4.330; check C2b, **FAIL**, reported verbatim). Our reading, not tested with an annealing schedule, is an *annealing window*: too cold never explores, too hot cannot see the graph at all — Theorem 4 is why annealing must end well below the floor-capture regime. This mirrors the temperature lore of concrete/Gumbel relaxations (Jang et al., 2017; Maddison et al., 2017), but here the high-temperature pathology has a closed form.

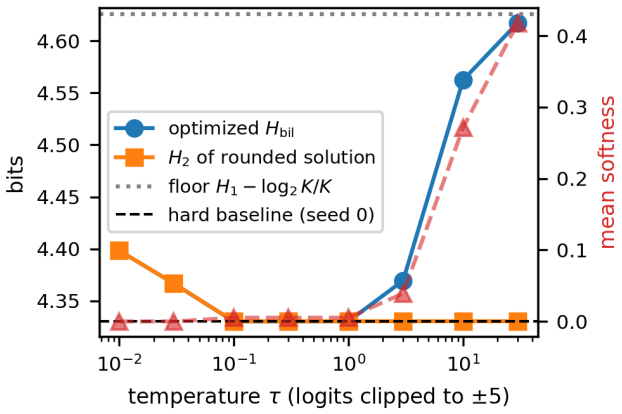

Figure 8: Temperature protocol on the statistical soft-membership fixture (graph seed 0; logits clipped to ± 5). High τ: floor capture (closed form, graph-independent). Low τ: premature hardening (preregistered check C2b failed). Window $\tau \in [0.1, 1]$: rounds to the hard baseline.