IDENTIFIABLE HIERARCHY DEPTH: AN MDL TREE COST FOR STRUCTURAL ENTROPY

Anonymous authors
Paper under double-blind review

ABSTRACT

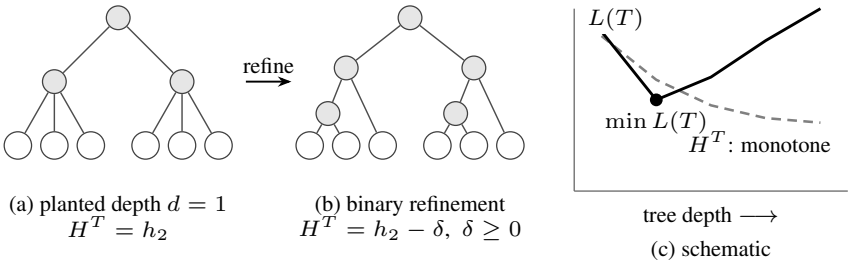
Hierarchical clustering, community detection and world models need the number of levels, which their objectives do not provide. Dasgupta's cost, hierarchical modularity and structural entropy (SE) never get worse when a level is added, so their optimizers add levels even to graphs with no hierarchy. Depth becomes identifiable with two ingredients: a per-level null-model test around any hierarchy builder and score, and, for SE, a completed code whose minimizer has finite depth. For SE the failure is exact: any multiway encoding tree can be refined to a binary tree without increasing $H^T(G)$. We give SE a two-part minimum-description-length (MDL) objective $L(T) = \mathrm{vol} \cdot H^T(G) + U(T) + C(T)$, whose terms are all prefix-code lengths: $\mathrm{vol} \cdot H^T$ is exactly the cost of coding each directed edge endpoint by its Li–Pan tree codeword, U is the up-addressing cost SE leaves unpaid, and C is an Elias-gamma size-list code for the tree topology. We derive the *exact* condition under which a coarse k-way split strictly beats its binary refinement—collapse improves L iff $2E_{\mathrm{int}}(a)\log_2(V_p/V_a) < g_a + \ell_\gamma(s_a)$—and prove that no L-optimal tree contains a null refinement, so depth is well defined. We then prove a boundary result: because the SE code fixes its probabilities by volumes, its gains are *linear* in cut fluctuations, so on sparse random null graphs a spurious level survives any topology cost of scale $o(n\sqrt{d})$. No such tree cost makes nulls flat; the null test, part of the method, has false-positive rate $\leq 1/(k{+}1)$ per level under exchangeability, which swap-chain resamples satisfy only approximately. On planted hierarchical SBMs with 160 vertices at hierarchy strength 6, the calibrated depth recovers $d = 1, 2, 3$ in every seed, where pure SE returns binary dendrograms of depth 6–9 and neither multilevel Infomap nor the nested SBM recovers all three; it calls 34 of 40 degree-matched nulls flat (both rivals call all 40 flat) and misses a fourth level of 10-vertex blocks, which it recovers at 640 vertices, where it instead keeps a spurious sub-level on flat two-block graphs. Around Dasgupta's cost, hierarchical modularity, the map equation and plain SE, the test keeps 35–40 of 40 nulls flat where the raw rules of all but the map equation call none flat, and with sparsest-cut bisection it recovers $d = 1$ to 4 in at least 7 of 10 seeds each; builders that merge planted levels, and the map equation, which prefers shallower trees, are not rescued. Code, raw results and preregistrations are in the supplement.

1 INTRODUCTION

How many levels does a hierarchy have? Hierarchical clustering, hierarchical graph pooling (Wu et al., 2022) and hierarchical world models all need this number, and recent models fix it by hand: SUNTA evaluates up to two levels and leaves deeper hierarchies to future work (Iiyama et al., 2026). The objectives that score a hierarchy do not supply it. Dasgupta's cost (Dasgupta, 2016), hierarchical modularity (Newman and Girvan, 2004; Traag et al., 2019) and structural entropy (Li and Pan, 2016) never score a tree worse for an added level; on degree-matched random graphs, which have no hierarchy, the raw depth rule of each reports structure on all 40 (Table 3). Multilevel Infomap (Rosvall and Bergstrom, 2011) and the nested SBM (Peixoto, 2014) call these graphs flat, but neither recovers a planted third level at hierarchy strength 6 (Section 6).

This paper makes depth an inference: each level must beat a null model that preserves the levels above it (Section 4.2). The test wraps any builder and score but can only remove levels, so it needs an objective whose optimum is neither a full dendrogram nor too shallow. We build one for structural

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



(a) planted depth $d = 1$ $H^T = h_2$ (b) binary refinement $H^T = h_2 - \delta,\ \delta \geq 0$ (c) schematic

Figure 1: Structural entropy cannot choose the number of levels. Refining any internal node never increases H^T (Theorem 1), so from (a) to (b) the objective can only fall and its minimum is always attained by a full binary dendrogram: the level count of the argument-minimum tracks the vertex count, not the data. Completing the code adds the two terms the walk code never pays, and the completed objective $L(T)$ acquires an interior minimum (c, schematic); Section 4.2 shows that this minimum can sit one fluctuation level too deep, which the null calibration removes.

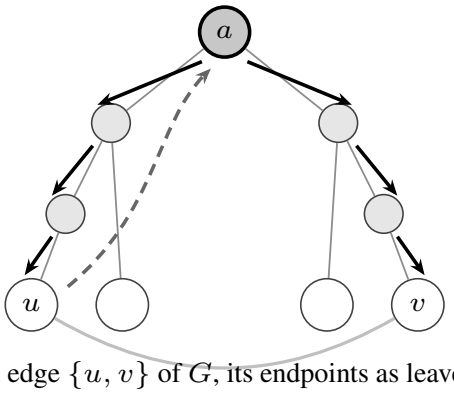
entropy (SE) (Li and Pan, 2016; Li, 2024), the codelength of a random walk along an encoding tree and our main instance. Refining a node never increases $H^T(G)$ (Theorem 1) because the code never pays to say where a codeword starts (Figure 2); charging for it gives a two-part code whose minimizer has finite depth. Our contributions:

1. **Null-calibrated depth for any hierarchy method** (Section 4.2). A per-level test against resamples that preserve the accepted levels, with false-positive rate $\leq 1/(k{+}1)$ under exchangeability. Wrapped around Dasgupta's cost, hierarchical modularity, the map equation and plain SE, it controls false levels on nulls and recovers depth with sparsest-cut bisection (Table 3).
2. **An honest two-part code for SE** (Section 3). $\mathrm{vol} \cdot H^T(G)$ is exactly the codelength of every directed edge endpoint under the Li–Pan tree code (Lemma 1); we add the up-addressing bits $U(T)$ it omits and a prefix code $C(T)$ for the topology. Every term is a code length, not a tuned penalty.
3. **Exact identifiability theory** (Section 4). Theorem 2 gives the exact collapse condition, so L-optimal trees contain no null refinements; Theorem 1 gives the SE gain of any refinement exactly, so "coarse beats binary refinement iff the SE gain is below the tree cost" is an identity, not a bound.
4. **A boundary theorem** (Proposition 1). The volume code's gains are linear in cut fluctuations, so random *null graphs* (proved for sparse Erdős–Rényi graphs, observed on degree-matched nulls) support one spurious level against any topology cost of scale $o(n\sqrt{d})$, the SE analogue of modularity's fluctuation overfitting (Guimerà et al., 2004) and the map equation's sparse-regime overfitting (Smiljanić et al., 2020). This is why choosing depth cannot be separated from comparing against a null model.
5. **Algorithm, experiments and released code** (Sections 5, 6). Collapse and pair-insert moves with closed-form ΔL, verified against full recomputation on every accepted move; planted hierarchical SBMs ($d \in \{1, 2, 3, 4\}$, four strengths), degree-matched nulls, six rival depth rules, $n = 640$ and branching-3 checks, and six real graphs as null controls; code, raw results and preregistrations are in the supplementary material.

2 Preliminaries

Let $G = (V, E, w)$ be an undirected weighted graph, $n = |V|$, $d_u = \sum_v w_{uv}$ (self-loops counted twice), $\mathrm{vol} = \sum_u d_u$. An *encoding tree* T is a rooted tree whose leaves are the vertices of G. For a node a, V_a is the sum of leaf degrees below a and g_a the total weight of edges with exactly one endpoint below a. The structural entropy of G under T (Li and Pan, 2016) is

$$H^T(G) \;=\; -\sum_{a \in T,\ a \neq \mathrm{root}} \frac{g_a}{\mathrm{vol}} \log_2 \frac{V_a}{V_{\mathrm{parent}(a)}}. \tag{1}$$



charged (Lemma 1):
descend $a \to u$ and $a \to v$,
$\log_2 \frac{V_a}{d_u} + \log_2 \frac{V_a}{d_v}$

never charged:
which ancestor a
the descent starts from

an edge $\{u, v\}$ of G, its endpoints as leaves

$$L(T) \;=\; \underbrace{\mathrm{vol} \cdot H^T(G)}_{\text{descent}} + \quad \underbrace{U(T)}_{\text{the missing up-address}} \quad + \quad \underbrace{C(T)}_{\text{topology, Elias-}\gamma}$$

Figure 2: Which bits the code omits. Lemma 1 rewrites $\mathrm{vol} \cdot H^T(G)$ edge by edge: for an edge $\{u, v\}$ with lowest common ancestor a, the code pays exactly to *descend* from a to each endpoint (solid). It never pays to say *which* ancestor the descent starts from (dashed), so concatenated codewords are not decodable and a deeper tree obtains the missing addressing for free. Charging it, plus a universal code for the topology, gives an objective in which every term is a genuine code length.

We write s_a for the number of leaves below a, k_a for its number of children, and, for an internal node a with children $c_1, \ldots, c_{k_a}$, $E_{\mathrm{int}}(a) = \frac{1}{2}\big(\sum_i g_{c_i} - g_a\big)$: the weight of edges joining distinct children of a.

Lemma 1 (Edge-LCA code identity)**.** *For every encoding tree T,*

$$\mathrm{vol} \cdot H^T(G) = \sum_{(u,v)} w_{uv} \Big[\log_2 \frac{V_{\mathrm{lca}(u,v)}}{d_u} + \log_2 \frac{V_{\mathrm{lca}(u,v)}}{d_v} \Big] = \sum_{\substack{\text{directed} \\ (u \to v)}} - w_{uv} \log_2 \prod_{a \in P(u,v)} \frac{V_a}{V_{\mathrm{parent}(a)}}, \tag{2}$$

where $P(u, v)$ is the path from $\mathrm{lca}(u, v)$ down to the leaf v (self-loops contribute zero). Hence $\mathrm{vol} \cdot H^T(G)$ is exactly the number of bits needed to transmit the head of every directed edge, given its tail and given that the decoder knows where the down-path starts, using the codebook $p(c \mid a) = V_c / V_a$ at each node (a valid prefix code: $\sum_c V_c / V_a = 1$).

Proof in Appendix A.

Lemma 1 makes precise what H^T does and does not pay for: it pays for descending to the head, but not for telling the decoder how far *up* from the tail the descent starts. Deeper trees get that information for free. This is the coding-theoretic root of the depth unidentifiability.

3 The two-part objective

Definition 1 (MDL tree objective)**.**

$$L(T) \;=\; \underbrace{\mathrm{vol} \cdot H^T(G)}_{\textit{down-paths}} + \quad \underbrace{U(T)}_{\textit{up-addressing}} \quad + \underbrace{C(T)}_{\textit{topology}}, \tag{3}$$

with

$$U(T) = \sum_{\textit{directed } (u \to v)} w_{uv} \big(\mathrm{depth}(u) - \mathrm{depth}(\mathrm{lca}(u, v))\big) \qquad \textit{(unary up-code, 1 bit per level)}, \tag{4}$$

$$C(T) = \sum_{\textit{internal } a} \; \sum_{c \in \mathrm{children}(a)} \ell_\gamma(s_c), \qquad \ell_\gamma(x) = 2\lfloor \log_2 x \rfloor + 1 \quad \textit{(Elias-gamma size lists)}. \tag{5}$$

The ablation L_{noU} sets $U \equiv 0$ (pure "SE + topology cost").

Both added terms are genuine codes (Appendix B): C is decodable top-down because child sizes sum to s_a, and its leaf-assignment part telescopes to the constant $\log_2 n!$, so each internal node a costs exactly $\ell_\gamma(s_a)$ bits; U is what makes the concatenated per-edge codewords of Lemma 1 decodable, the analogue of the map equation's exit codebooks (Rosvall and Bergstrom, 2011).

4 THEORY

Theorem 1 (Exact refinement gain; monotonicity of pure SE)**.** *Let T' be obtained from T by inserting a new node a' under an internal node a, adopting a subset S of a's children ($2 \le |S| < k_a$). Then*

$$\mathrm{vol}\big(H^{T'}(G) - H^{T}(G)\big) \;=\; -2\,E_{\mathrm{int}}(S)\,\log_2 \frac{V_a}{V_{a'}} \;\le\; 0, \tag{6}$$

where $E_{\mathrm{int}}(S)$ is the weight of edges joining distinct members of S and $V_{a'} = \sum_{c\in S} V_c$. In particular refinement never increases H^T, strictly decreases it iff $E_{\mathrm{int}}(S) > 0$ and $V_{a'} < V_a$, and pure SE minimization is therefore always weakly optimized by full binary dendrograms: depth is unidentifiable from H^T alone.

Proof in Appendix A.

Theorem 2 (Exact collapse condition)**.** *Let a be a non-root internal node of T with parent p, and let T' be T with a collapsed into p (children of a become children of p). Then*

$$L(T') - L(T) \;=\; 2\,E_{\mathrm{int}}(a)\,\log_2 \frac{V_p}{V_a} \;-\; g_a \;-\; \ell_\gamma(s_a), \tag{7}$$

and for the ablation, $L_{\mathrm{noU}}(T') - L_{\mathrm{noU}}(T) = 2E_{\mathrm{int}}(a)\log_2(V_p/V_a) - \ell_\gamma(s_a)$. Hence the coarse tree strictly beats keeping the refinement a iff its SE gain is below the tree cost of a:

$$2\,E_{\mathrm{int}}(a)\,\log_2 \frac{V_p}{V_a} \;<\; g_a + \ell_\gamma(s_a) \qquad (\textit{resp. } \ell_\gamma(s_a)). \tag{8}$$

Proof in Appendix A.

Corollary 1 (Identifiable depth; invariance to null refinements)**.** *Call an internal non-root node a a* null refinement *if $E_{\mathrm{int}}(a)\log_2(V_p/V_a) = 0$ (no edges join its children, or it duplicates its parent's volume). Every null refinement satisfies $L(T') - L(T) = -g_a - \ell_\gamma(s_a) < 0$ (and $-\ell_\gamma(s_a) < 0$ for L_{noU}), so no L-optimal tree contains one. Consequently: (i) all L-optimal trees are fixed points of collapse; (ii) null refinements never change the argmin, so the optimal level count cannot be inflated by them and is a function of G when the minimiser is unique; and (iii) a k-way node beats* every *binary refinement of itself whenever each inserted node a_i of the refinement has $2E_{\mathrm{int}}(a_i)\log_2(V_{p_i}/V_{a_i}) < g_{a_i} + \ell_\gamma(s_{a_i})$, the deltas being additive along the insertion sequence.*

Proof. Theorem 2 applied to each node, and for (iii) to each inserted node of the refinement. □

We report depth as (modal leaf depth $-$ 1): the number of module levels strictly between root and leaves. The flat tree has depth 0 (depth 1 if root-to-leaf edges are counted).

4.1 THE BOUNDARY: NULL GRAPHS ARE NOT NULL REFINEMENTS

Proposition 1 (Linear-gain overfitting; no topology cost suffices)**.** *Let T_2 be the two-level tree over a partition $\{B_1, B_2\}$ with volumes V_i and cut g (edge count* cut*). Then, exactly,*

$$L(T_2) - L(T_{\mathrm{flat}}) = -2\sum_i E_{\mathrm{in}}(B_i)\log_2 \frac{\mathrm{vol}}{V_i} \;+\; 2\,\mathrm{cut} \;+\; \ell_\gamma(|B_1|) + \ell_\gamma(|B_2|), \tag{9}$$

which for balanced volumes is $-2m + 4\,\mathrm{cut} + O(\log n)$: every bit of within-block edge mass pays off linearly*, regardless of whether the imbalance is structural or fluctuational. For $G \sim G(n, d/n)$ with $d \ge d_0$ fixed (d_0 a universal constant), with high probability there is a balanced bisection with*

$\mathrm{cut} \le m/2 - c\,n\sqrt{d}$ *for any fixed* $0 < c < \mathrm{P}^*/2$ *(Dembo et al., 2017), hence* $\min_T L(T) < L(T_{\mathrm{flat}})$ *and the L-optimal depth is at least* 1 *on null graphs w.h.p. Moreover the same conclusion holds for any modified objective* $\mathrm{vol} \cdot H^T + U + C'$ *whose cost of a single two-block level is* $o(n\sqrt{d})$*: no topology-only prefix cost of that scale can make null graphs flat.*

Proof in Appendix A.

The mechanism is quantitative, not just asymptotic: on the degree-matched nulls of Section 6 the optimizer's tree still gains an upper median of 643–809 bits over the flat code (Table 4, Figure 3a), as modularity is large on random graphs (Newman and Girvan, 2004; Guimerà et al., 2004).

The other two ways of setting the code's probabilities do not escape this (Remark 1, Appendix C): null-implied codes never gain, and fitted codes need an n-scale partition cost that a singleton-leaf hierarchy code cannot have. Within "SE + tree cost", therefore, null *refinements* are provably removable (Corollary 1) while null *graphs* provably are not (Proposition 1): depth identifiability against null graphs requires an explicit null comparison: choosing depth cannot be separated from testing against a null model.

4.2 Null-calibrated depth

Proposition 1 moreover applies *recursively*: within any sufficiently large, dense true block, within-block cut fluctuations also pay off linearly, so the raw L-optimal tree can carry, when blocks are large or dense (Table 4), the true hierarchy *plus one* bottom fluctuation level (the grid shows this: raw depth $d{+}1$ on every $d{=}1$ graph at $r \ge 4$, whose blocks hold 80 vertices, and raw depth ≥ 1 on all 100 nulls). The decoder must therefore test every level against a null that preserves the levels above it.

Definition 2 (Per-level calibrated depth)**.** *Let* $\hat{T}(G)$ *be the L-optimizer's tree with level partitions* $P_1 \succeq P_2 \succeq \cdots$*, and let* $T_{P_{1..\ell}}$ *denote the tree truncated below level* ℓ*. Level* 1 *is accepted iff* $L(T_{\mathrm{flat}}) - L(\hat{T}(G))$ *exceeds the maximum of the same statistic over* k *independent degree-preserving resamples of* G *(double edge swaps). Level* $\ell{+}1$ *is accepted iff the beyond-level-*ℓ *statistic* $S_\ell(X) = L(T_{P_{1..\ell}}(X)) - L(\hat{T}(X))$ *satisfies* $S_\ell(G) > \max_i S_\ell(N_i^{(\ell)})$*, where each* $N_i^{(\ell)}$ *is a degree-preserving resample of* G *that additionally preserves every block-pair edge class of* P_ℓ *(swaps restricted within classes), and* $\hat{T}(N_i^{(\ell)})$ *is the optimizer's tree on the resample. The calibrated depth* $\hat{d}_{\mathrm{cal}}$ *is the number of consecutively accepted levels, capped at the raw modal depth. Under exchangeability of* G *with its (conditional) resamples at a truly absent level, each test has false-positive rate* $\le 1/(k{+}1)$*, so* $\Pr[\hat{d}_{\mathrm{cal}} > d_{\mathrm{true}}] \le 1/(k{+}1)$*; we use* $k = 7$*.*

This is the model-free analogue of nested model selection in the hierarchical SBM (Peixoto, 2014): each level must justify itself against a null conditioned on the coarser levels. It is a wrapper, not a change of objective: the tree, its levels and all codelengths remain those of Definition 1.

5 Algorithm

We extend an open-source SE library's encoding-tree machinery with an exact best-improvement descent on L: **collapse** (Theorem 2) and **pair-insert** (group two children of a node; the dual, with closed-form $\Delta L = -2w_{12}\log_2(V_a/V_{12}) + (g_1{+}g_2{-}2w_{12}) + \ell_\gamma(s_{12})$). Closed-form deltas select the move, and every accepted move is re-verified by a full recomputation of L (tolerance 10^{-6}); independent improving moves are applied in verified batched sweeps, then one at a time. Multi-start over three initializers (flat; recursive-SE multiway; SE-agglomerative dendrogram) returns the lowest L; the output is a collapse fixed point, free of null refinements (Corollary 1). Run times: Appendix D.

6 Experiments

Setup. Balanced hierarchical SBMs with known module depth $d \in \{1,2,3,4\}$, branching $b = 2$, $n = 160$, nominal expected degree 16 (realised 9–15); a pair whose planted LCA sits at level j connects with probability $\propto r^j$, $r \in \{2,4,6,8\}$ the hierarchy strength; 10 seeds per cell at $r = 6$ (seeds 0–9, the preregistered gate) and 5 (seeds 0–4, exploratory) at $r \in \{2,4,8\}$. Every planted graph is probed model-free before any estimator runs: the realized per-level densities must be

Table 1: The preregistered gate at $r = 6$ on the decisive seeds 0–9 (10 planted graphs per d, and the 30 degree-matched evaluation nulls of the $d \leq 3$ graphs), fixed before these seeds ran and reported verbatim. G2a was frozen-predicted to fail (Proposition 1). Counts: graphs of 10 per d, or nulls of 30; $10/10$ has a 95% Clopper–Pearson lower bound of 0.69.

	criterion	required	observed ($d = 1, 2, 3$)	verdict
G1	calibrated depth $= d$	$\geq 7/10$ each	10, 10, 10	pass
	raw depth $\in \{d, d{+}1\}$	$\geq 8/10$ each	10, 10, 10	
G2a	raw depth 0 on nulls	$\geq 24/30$	$0/30$ (1 level: 26; 2: 4)	fail, as predicted
G2b	calibrated depth 0 on nulls	$\geq 22/30$	$25/30$	pass
G3	pure SE depth $= d$	$\leq 2/10$ each	0, 0, 0	pass

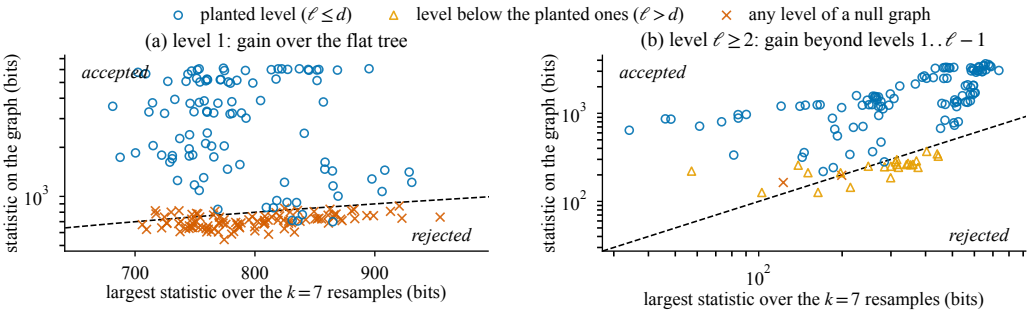


Figure 3: Why the calibration is needed and what it removes: every per-level test of Definition 2 run on the grid (all r and d; 100 planted graphs and their 100 degree-matched evaluation nulls). A level is accepted iff its statistic on the graph exceeds the largest of the same statistic over $k = 7$ resamples (above the diagonal). (a) Nulls have no planted structure, yet their trees save 543–873 bits over the flat code, so the raw depth is never 0 (Proposition 1); their resamples save as much, and 89 of 100 are rejected. Planted first levels are accepted on 96 of 100 graphs (the four misses at $r = 2$). (b) Deeper planted levels are accepted in 106 of 107 tests; fluctuation levels below the planted ones in 6 of 24 (four of them the $r = 4$, $d = 1$ sub-levels of Table 2).

strictly increasing (they are in all cells). Each graph gets a degree-matched evaluation null (double edge swap (Maslov and Sneppen, 2002; Fosdick et al., 2018), $30m$ swaps) and $k = 7$ independent calibration resamples per level. The decisive gate at $r^* = 6$ (Table 1) was fixed before the decisive seeds were run. The cells at $r \in \{2, 4, 8\}$, the Infomap and nested-SBM runs and the $C \equiv 0$ ablation came later and are exploratory; the extension experiments (Table 3, $n = 640$, branching 3, real graphs, a false-positive follow-up) came last, each with thresholds preregistered and timestamped before it ran (Appendix E).

Gate verdict (all preregistered, reported verbatim). On the decisive seeds (0–9, not used while the protocol was designed) G1, G2b and G3 pass and G2a fails exactly as frozen-predicted, the one-fluctuation-level signature of Proposition 1 (Table 1; Figure 3 shows every test of the grid). The five calibrated false positives, all on nulls of $d = 2$ graphs, exceed the $1/8$ design rate (Limitations (5)).

Recovery over the whole grid. Table 2 reports every cell we ran (per-seed detail at $r = 6$: Figure 5). The calibrated depth recovers $d = 1, 2, 3$ in every seed at $r = 6$, and $d = 2, 3$ at $r = 4$. The shape is recovered with the depth: the per-level cluster counts of the raw tree equal the planted $(2, 4)$ and $(2, 4, 8)$ in $8/10$ and $9/10$ seeds at $r = 6$. Raw MDL depth is exactly $d{+}1$ at $d = 1$ for $r \geq 4$ (blocks of 80 support a fluctuation sub-level) and exactly d at $d = 2, 3$ for $r = 4, 6$, as Proposition 1 predicts; the extra level on $d{=}1$ graphs consists mostly of single-vertex clusters (22 of 28 on average at $r = 6$). The per-level calibration removes it at $r = 6$ and $r = 8$; at $r = 4$ it keeps the sub-level on 4 of 5 $d{=}1$ graphs. At $r = 8$ the raw depth overshoots at $d = 2$ (denser blocks re-admit fluctuation sub-levels; raw exact recovery 0.4) and calibration raises it to 0.8, one seed staying a level too deep; at $d = 3$ one seed stops a level short (raw and calibrated 0.8).

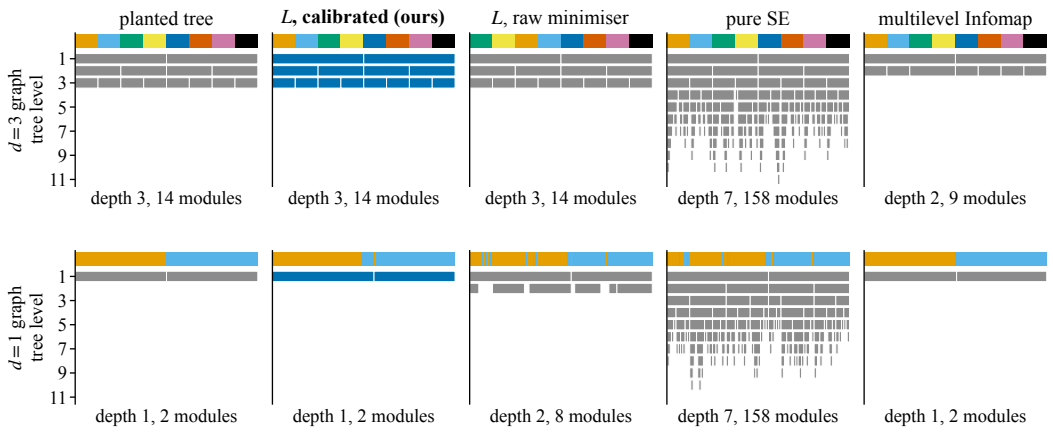


Figure 4: Case study: the trees behind the counts of Table 2 on two graphs chosen by rule (the first decisive seed at $r = 6$; top $d = 3$, bottom $d = 1$). Row k of each icicle shows the tree's modules at level k as bars over the vertices, ordered depth-first by that tree; the strip on top colours every vertex by its planted bottom block, so a one-colour bar is a pure module. Under each panel: the reported depth (modal leaf depth -1) and the number of modules. Every tree was recomputed with the released code and matches the logged depths, codelengths and level tests.

Case study. In Figure 4, on the $d = 3$ graph pure SE finds the planted blocks in its top three levels, then keeps bisecting (158 modules for 160 vertices); Infomap merges two bottom blocks and stops at two levels; L returns the planted tree. On the $d = 1$ graph the calibration removes the raw sub-level of six modules and 26 singletons, but its split misplaces 13 of 160 vertices (Infomap's, one): the depth is right, the partition is not exact.

Depth rules and description-length rivals. Pure SE returned depth 5–9 (6–9 at $r = 6$) with mean branching factor 2.00 in every cell, the motivating pathology reproduced. Sweeping the tree height for minimum SE cannot work in principle: by Theorem 1 H^T is weakly decreasing in height, so where the sweep is right ($d = 3$ at $r \leq 4$, $6/10$ at $r = 6$) it has merely stopped in the right place, and it is never right at $d \neq 3$. At $r = 6$ the 5%-elbow rule on the same sweep returns two levels whatever d is (Figure 5), so it is right exactly when $d = 2$ (at $r \leq 4$ it also returns one level on some $d = 1$ graphs); Paris (Bonald et al., 2018) has no depth rule (we report its modal depth). Multilevel Infomap (Rosvall and Bergstrom, 2011) (10 trials) and the nested SBM (Peixoto, 2014) do select depth from a description length. Infomap stops at two levels on every $d = 3$ graph at $r = 6$, recovers $d = 3$ on 1 of 25 graphs over the grid and $d = 4$ on none (its 0.9 at $d = 2$, $r = 6$ rests on a tie; Table 2, †). The nested SBM recovers $d = 2$ on $1/10$ graphs at $r = 6$ and $d \geq 3$ on none at any strength; with or without degree correction, five restarts and zero-temperature sweeps reproduce this exactly, so it is not a configuration artefact. At $r = 6$ only the calibrated MDL depth recovers $d = 1, 2, 3$ in every seed. On nulls the rivals are better: both call all 40 nulls at $r = 6$ flat (and all 100 over the grid), against $34/40$ for the calibrated MDL depth (two-sided Fisher exact $p = 0.026$).

Is the calibration specific to SE? Definition 2 needs only a hierarchy builder and a score. We wrapped it, unchanged ($k = 7$, same resamplers), around five builder–objective pairs (Table 3): Dasgupta's cost (Dasgupta, 2016) with recursive spectral sparsest-cut bisection and with Paris; hierarchical modularity with recursive Leiden (Traag et al., 2019); the multilevel map equation with multilevel Infomap (our codelength matches Infomap's to 10^{-6}); and pure SE with recursive SE partitioning. The raw rules of the four refinement-monotone objectives call structure on all 40 nulls; calibrated, all five keep 35–40 of 40 nulls flat and 87–100 of 100 over the grid. Recovery transfers to one builder: calibrated bisection recovers $d = 1, 2, 3, 4$ in $9, 7, 8$ and 10 of 10 seeds, including the fourth level that L misses. It does not transfer to recursive Leiden or recursive SE, whose first split already merges planted levels (hierarchical modularity scores the planted tree above Leiden's on all 20 graphs with $d = 2, 3$, so the loss is the builder's), nor to Paris, whose calibrated depth overshoots on 4 of 10 $d = 1$ and 5 of 10 $d = 3$ graphs. The map equation itself scores Infomap's shallower tree above the planted one on all 20 graphs with $d = 2, 3$, which a calibration that can only remove levels

Table 2: Depth recovered on hierarchical SBMs, over the whole grid of hierarchy strength r (the factor by which a pair's edge probability rises per shared level) and planted depth d. Ours are above the first rule: MDL cal, the null-calibrated depth of the L-optimal tree (Definition 2), and MDL raw, the same minimiser uncalibrated. Rows 1–8: fraction of seeds recovering d (10 seeds at $r = 6$, 5 elsewhere); **bold**, the column's largest non-zero value, ties included; '·', none recovered. Baselines (Section 6): multilevel Infomap, the nested SBM, pure SE, its min-H height sweep and a 5% elbow on it, and Paris (modal depth of its full dendrogram). Last row: fraction of degree-matched nulls the calibrated depth calls flat (Infomap and the nested SBM call all nulls flat at every strength). Nested SBM: $r = 6$ one run, other strengths degree-corrected, best of five (reproduces $r = 6$ exactly; Appendix E). †Nine of ten trees put half the vertices one level higher, a tie broken toward the deeper level; under the shallower-level rule of every other row this entry would be 0. ↑: higher is better. Gains and raw depths: Table 4.

fraction exact ↑	$r=2$				$r=4$				$r=6$				$r=8$			
d	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
MDL cal (ours)	·	·	·	·	0.2	**1.0**	**1.0**	·	**1.0**	**1.0**	**1.0**	·	**1.0**	0.8	**0.8**	·
MDL raw (ours)	**0.8**	0.2	·	·	·	**1.0**	**1.0**	·	·	**1.0**	**1.0**	·	·	0.4	**0.8**	·
Infomap	·	·	·	·	0.4	·	·	·	**1.0**	0.9†	·	·	**1.0**	0.8	0.2	·
nested SBM	·	·	·	·	**1.0**	·	·	·	**1.0**	0.1	·	·	**1.0**	**1.0**	·	·
pure SE	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
min-H sweep	·	·	**1.0**	·	·	·	**1.0**	·	·	·	0.6	·	·	·	·	·
elbow on sweep	**0.8**	**0.6**	·	·	0.6	**1.0**	·	·	·	**1.0**	·	·	·	**1.0**	·	·
Paris	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
nulls flat (cal) ↑	0.6	0.8	1.0	1.0	1.0	1.0	0.8	1.0	1.0	0.5	1.0	0.9	1.0	0.8	1.0	1.0

cannot repair. A binary builder has a price: at branching 3 ($n = 486$) calibrated bisection is exact on 0 of 15 graphs and calibrated L on 11. On six real graphs, which have no known depth and serve only as null controls, every calibrated method calls at least 25 of 30 degree-preserving nulls flat (Table 8). With heterogeneous degrees (degrees 1–159) calibrated L is exact on 10, 10 and 9 of 10 graphs at $d = 1, 2, 3$ and calls $28/30$ nulls flat; calibrated bisection, whose Dasgupta cost ignores degrees, is exact on 2 of 30 (Table 10).

Where recovery stops. At $d = 4$ the bottom blocks hold 10 vertices and neither L nor any rival in Table 2 recovers the fourth level; at $r \geq 4$ MDL returns three levels, whose block counts do not match the planted ones. This is a limit of the L minimiser at $n = 160$, not of the graphs: calibrated bisection recovers it (Table 3), and at $n = 640$ (blocks of 40) calibrated L is exact on $5/5$ graphs with the planted block counts $(2, 4, 8, 16)$, and on $5/5$ at $d = 2, 3$. Scale also enlarges the fluctuation level: the calibration keeps a mostly-singleton sub-level on $5/5$ flat graphs at $n = 640$ (blocks of 320) and on $4/5$ flat three-block graphs at $n = 486$, whose deeper levels (branching 3) it recovers on $5/5$ each; it calls 32 of these 35 nulls flat. At $r = 2$ the calibrated depth is below d on 19 of 20 graphs and a level too deep on the twentieth (d=1): on 14 the raw minimiser is already too shallow, and on 5 the deepest level fails its null; the raw minimiser is right on $4/5$ d=1 graphs.

Which term identifies depth. Ablating on three d=2 seeds at $r = 6$: with $U \equiv 0$ the raw minimiser has depth 6 on all three (96–98 internal nodes, against 12–15 for L), with $C \equiv 0$ depth $2, 3, 3$; the calibrated depth is 2 in all six runs. The up-addressing term carries most of the raw signal; the calibration makes the depth exact in both ablations.

7 Related work

Li and Pan (2016) define structural entropy (see also Li, 2024); Dasgupta's cost (Dasgupta, 2016) scores hierarchies but is likewise weakly minimized by full binary trees. HCSE (Pan et al., 2021) selects a height for k-dimensional SE by a stretch/compress heuristic with a sparsest-level criterion, a useful constructor but not a code-length account of depth; we compare against height sweeps of its family. The multilevel map equation (Rosvall and Bergstrom, 2011) selects depth through exit-codebook terms inside its data term—the closest precedent to our $U(T)$—and its regularized

Table 3: The per-level null calibration (Definition 2) wrapped around other hierarchy objectives, each with its native builder and its own score as the test statistic ($r = 6$, the graphs of Table 2; $k = 7$). Nulls flat: degree-matched nulls called depth 0, raw (the builder's own tree) and calibrated, of 40; over the whole grid of 100. Exact: calibrated depth equals d, seeds of 10. Transfers: the preregistered rule (yes: $\geq 32/40$ nulls flat, $\geq 80\%$ over the grid, and $\geq 7/10$ at each $d \leq 3$; builder: null control holds, and every miss has a raw depth already below d, which calibration cannot raise; no: otherwise). The first row is Table 2's method. Full grid in Appendix E.

		nulls flat ↑			exact ↑ at $d =$				
objective	builder	raw	cal	grid	1	2	3	4	transfers
L (ours)	collapse/pair-insert	0	34	89/100	10	10	10	0	–
Dasgupta cost	sparsest-cut bisection	0	37	93/100	9	7	8	10	yes
Dasgupta cost	Paris	0	40	91/100	6	8	5	10	no
hier. modularity	recursive Leiden	0	36	87/100	8	0	1	0	no
map equation	multilevel Infomap	40	40	100/100	10	0	0	0	builder
$\mathrm{vol} \cdot H^T$ (pure SE)	recursive SE	0	35	88/100	8	1	0	0	no

Bayesian variant (Smiljanić et al., 2020) shows that flow codes overfit sparse networks. Peixoto's nested SBM (Peixoto, 2014) is the strongest rival for depth selection: its likelihood-based description length gains quadratically in density deviations and pays partition costs, at the price of abandoning the random-walk coding objective that defines SE; our Remark 1 makes the comparison exact. Statistical tests of hierarchy in block models decide whether to split further, by recursive bipartitioning with a non-backtracking stopping rule (Li et al., 2022) or by spectral detection of hierarchical equitable partitions (Schaub et al., 2023); our test instead wraps any builder and score. Two-part MDL follows Rissanen (1978); Grünwald (2007).

8 Limitations

(1) Proposition 1's random-graph statement is proved for Erdős–Rényi ensembles via Dembo et al. (2017); for the exact double-edge-swap (uniform simple graph with given degrees) null we verify the mechanism empirically but do not prove the constant. (2) The unary up-code is one valid convention; any prefix code on climb heights yields the same theorems with g_a replaced by its cost. (3) The optimizer is a local search with multi-start; depth claims inherit its optima, though every reported tree is a verified collapse fixed point, and the calibration costs one optimizer run per resample and level. (4) All results are for undirected graphs with hard hierarchies; directed, soft and temporal variants are out of scope. (5) Calibration false positives are not uniform: the calibrated depth was flat on $10/10$, $5/10$, $10/10$ and $9/10$ of the nulls derived from $d = 1, 2, 3, 4$ graphs ($34/40$ pooled; $89/100$ over the grid; $25/30$ on the decisive seeds); five or more in ten has probability 0.0045 at the design rate $1/8$. A preregistered follow-up (Table 6, Appendix E) does not support swap-chain mixing as the cause; fresh resamples of the published nulls concentrate their false positives on the same five $d = 2$ nulls (3–4 of 5 refreshes each). The guarantee is marginal over the null draw, and the bound ignores that P_ℓ is estimated from G. (6) Scope: synthetic graphs up to $n = 640$ with balanced blocks (one degree-corrected check); real graphs, whose depth is unknown, serve only as null controls (Table 8).

9 Conclusion

Structural entropy cannot choose the number of levels because its code omits bits that a decodable code must pay. The edge–LCA identity locates the omission, L charges it, and the collapse condition removes every zero-gain level. Because a volume code's gains stay linear in cut fluctuations, no topology cost of scale $o(n\sqrt{d})$ makes null graphs flat, so the per-level null comparison is part of the method. At strength 6 the calibrated depth recovers depths 1 to 3 in every seed, which no tested rival does; it keeps a fluctuation level on flat graphs with large blocks ($r = 4$, or $n \geq 486$), and the rivals are flatter on nulls. The calibration is not tied to SE: it controls false levels for Dasgupta's cost, hierarchical modularity and the map equation, and with bisection recovers depths 1 to 4.

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 analysis code that implements each pass/fail criterion and prints its verdict, is included as anonymous supplementary material. The cells at $r \in \{2, 4, 8\}$ and the rival and $C \equiv 0$ runs were added after the gate and are marked as exploratory. The extension experiments of Appendix E have their own timestamped preregistration, run log, per-unit raw outputs and the code that evaluates every preregistered rule, all in the supplementary material.

AI USE STATEMENT

We used a generative AI assistant (Claude, Anthropic), under the authors' direction, 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 (drafting the optimizer, analysis and plotting code), 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 the assistant to draft and edit the text of this paper, to create and edit figures and tables, 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 by the authors 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

T. Bonald, B. Charpentier, A. Galland, and A. Hollocou. Hierarchical graph clustering using node pair sampling. In *KDD Workshop on Mining and Learning with Graphs*, 2018. arXiv:1806.01664.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006.

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

A. Dembo, A. Montanari, and S. Sen. Extremal cuts of sparse random graphs. *Annals of Probability*, 45(2):1190–1217, 2017. doi:10.1214/15-AOP1084.

P. Elias. Universal codeword sets and representations of the integers. *IEEE Transactions on Information Theory*, 21(2):194–203, 1975.

B. K. Fosdick, D. B. Larremore, J. Nishimura, and J. Ugander. Configuring random graph models with fixed degree sequences. *SIAM Review*, 60(2):315–355, 2018. doi:10.1137/16M1087175.

P. D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70:025101, 2004.

T. Iiyama, M. Suzuki, and Y. Matsuo. SUNTA: Hierarchical video prediction with surprise-based chunking. arXiv:2607.02087, 2026.

B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83:016107, 2011.

A. Li. *Science of Artificial Intelligence: Mathematical Principles of Intelligence* (in Chinese). Science Press, Beijing, 2024.

T. Li, L. Lei, S. Bhattacharyya, K. Van den Berge, P. Sarkar, P. J. Bickel, and E. Levina. Hierarchical community detection by recursive partitioning. *Journal of the American Statistical Association*, 117(538):951–968, 2022. doi:10.1080/01621459.2020.1833888.

A. Li and Y. Pan. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62(6):3290–3339, 2016. doi:10.1109/TIT.2016.2555904.

S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296:910–913, 2002.

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

Y. Pan, F. Zheng, and B. Fan. An information-theoretic perspective of hierarchical clustering. arXiv:2108.06036, 2021.

T. P. Peixoto. Hierarchical block structures and high-resolution model selection in large networks. *Physical Review X*, 4:011047, 2014.

T. P. Peixoto. Bayesian stochastic blockmodeling. In *Advances in Network Clustering and Blockmodeling*, pages 289–332. Wiley, 2019.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

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

M. Rosvall and C. T. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PLoS ONE*, 6(4):e18209, 2011.

M. T. Schaub, J. Li, and L. Peel. Hierarchical community structure in networks. *Physical Review E*, 107(5):054305, 2023. doi:10.1103/PhysRevE.107.054305.

J. Smiljanić, D. Edler, and M. Rosvall. Mapping flows on sparse networks with missing links. *Physical Review E*, 102:012302, 2020.

V. A. Traag, L. Waltman, and N. J. van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9:5233, 2019. doi:10.1038/s41598-019-41695-z.

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

A PROOFS

Proof of Lemma 1. An edge $\{u, v\}$ crosses node a (contributes to g_a) iff a contains exactly one endpoint, i.e. iff a lies on the path from $\mathrm{lca}(u, v)$ (exclusive) to u or to v. Grouping the sum defining H^T by edges and telescoping $\sum_{a \in P(u,v)} \log_2(V_a/V_{\mathrm{parent}(a)}) = \log_2(d_v/V_{\mathrm{lca}})$ gives the identity. (The identity is also verified numerically to 10^{-9} in the released code.) □

Proof of Theorem 1. Only the terms of the children in S change (their parent volume drops from V_a to $V_{a'}$), and the new node adds $-(g_{a'}/\mathrm{vol}) \log_2(V_{a'}/V_a)$. Using $g_{a'} = \sum_{c \in S} g_c - 2E_{\mathrm{int}}(S)$ and collecting, $\Delta H = (1/\mathrm{vol}) \log_2(V_{a'}/V_a)\big(\sum_{c \in S} g_c - g_{a'}\big) = (2E_{\mathrm{int}}(S)/\mathrm{vol}) \log_2(V_{a'}/V_a)$. □

Proof of Theorem 2. The H term is Theorem 1 reversed. $\Delta C = -\ell_\gamma(s_a)$: in T, p's size list contains s_a and a's list contains the child sizes; in T', p's list contains the child sizes directly, so exactly the codeword $\ell_\gamma(s_a)$ disappears (the telescoped assignment constant is unchanged). $\Delta U = -g_a$: collapsing a lowers by one the depth of every leaf below a; a directed edge changes its up-distance iff its tail is below a and its lca is not (i.e. it leaves a), and there are g_a such units of weight. Both identities are asserted against full recomputation on every accepted optimizer move in the released code. □

Proof of Proposition 1. The displayed identity follows from Theorems 1 and 2 applied twice. On the fluctuation side, Dembo et al. (2017) determine the sharp constant: the minimum bisection of $G(n, d/n)$ is $m/2 - n\big(\mathrm{P}^*/2 + o_d(1) + o_n(1)\big)\sqrt{d}$ with $\mathrm{P}^* \approx 0.7632$ the Parisi constant, so any fixed $0 < c < \mathrm{P}^*/2$ works for $d \geq d_0$. Substituting a bisection with cut $\leq m/2 - cn\sqrt{d}$ gives $\Delta L \leq -4cn\sqrt{d} + O(\log n) < 0$ for large n, and replacing $\ell_\gamma(|B_1|) + \ell_\gamma(|B_2|)$ by any $o(n\sqrt{d})$ topology cost leaves the sign unchanged. □

B THE TWO ADDED TERMS ARE CODES

Why C is a code, and what cancels. Decode top-down. At each internal node a the receiver knows s_a (the root has $s = n$); the child leaf-counts arrive as Elias-gamma integers (Elias, 1975), self-terminating because they sum to s_a; any block of size ≥ 2 is necessarily internal (leaves are single vertices), so no leaf/internal flags are needed. Given all sizes, the assignment of vertices to blocks is a multinomial code whose total over *any* encoding tree with singleton leaves is, by the chain rule of multinomial coefficients, exactly $\log_2 n!$—a constant we drop. Trees are thus compared only on their arity/size structure, and each internal node a costs exactly $\ell_\gamma(s_a)$ bits (Theorem 2). This constancy is a structural fact about singleton-leaf hierarchies that Remark 1 returns to.

Why U is charged. Without U, the concatenation of the per-edge codewords of Lemma 1 is not decodable: the decoder cannot know at which ancestor of the tail the down-path begins. The unary convention (one bit per level climbed; $\sum_j 2^{-j} \leq 1$, a valid prefix code (Cover and Thomas, 2006)) is the direct analogue of the exit-codebook bits in the multilevel map equation (Rosvall and Bergstrom, 2008; 2011). Self-loops climb zero levels and pay zero. For the flat tree $U = \mathrm{vol} - 2\cdot(\text{self-loop mass})$: every step pays one bit to say "leave the leaf".

C WHY SE-TYPE CODES CANNOT SELF-CALIBRATE

Remark 1 (Why SE-type codes cannot self-calibrate)**.** *Three code families exhaust the design space for the addressing/descent probabilities. (i)* Null-implied *probabilities (all codeword lengths taken from the degree/volume null) make every tree code a refactorization of the flat code: null-flat, but structure-blind—trees can never gain. (ii)* Fixed non-null *probabilities—SE's volume ratios—gain linearly in first-order deviations (Proposition 1); fluctuations of scale $\sqrt{mn}$ therefore always buy a level. (iii)* Fitted *probabilities (empirical block densities (Karrer and Newman, 2011)) gain only quadratically in deviations, which is why nested-SBM description lengths (Peixoto, 2014; 2019) do not detect blocks in random graphs—but they are protected by an n-scale partition cost that a singleton-leaf hierarchy code cannot have, since its leaf-assignment cost is the constant $\log_2 n!$ for every tree (Section 3), and their data term is no longer H^T. Within "SE + tree cost", therefore, null* refinements *are provably removable (Corollary 1) while null* graphs *provably are not (Proposition 1): depth identifiability against null graphs requires an explicit null comparison. This is the precise sense in which choosing depth cannot be separated from testing against a null model.*

D DESCRIPTIVE RESULTS ON THE DEPTH GRID

Table 4: What each minimiser returns on the grid of Table 2: depth range over seeds of the raw L-optimal tree and of pure SE, the mean branching factor of pure SE, and the (upper) median gain of L over the flat code (bits) on the planted graphs and on their degree-matched nulls; depths are ranges over seeds. All descriptive.

	$r=2$				$r=4$				$r=6$				$r=8$			
d	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
MDL raw depth	1–2	1–2	1–2	1–2	2	2	3	3	2	2	3	3	2	2–3	2–3	3
pure SE depth	6–7	7–8	6–7	6–7	6–7	6–7	6–8	5–6	6–7	6–9	6–7	6–7	7–8	6–8	6–7	6–7
pure SE branching	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
L gain, planted	713	917	1159	1433	1252	2389	3856	5242	1747	3251	5080	6057	2039	3749	5768	5961
L gain, null	715	802	767	809	688	701	704	713	671	747	673	734	648	660	643	730

Run time. On $n = 160$ graphs one optimization takes $\sim$0.3 CPU-seconds (median over the 40 graphs at $r = 6$) and a full calibrated run (one optimization per resample and level, $k = 7$) $\sim$8–11 CPU-seconds on four seed-0 graphs; at $n = 640$ a calibrated run took 4–11 minutes on one core.

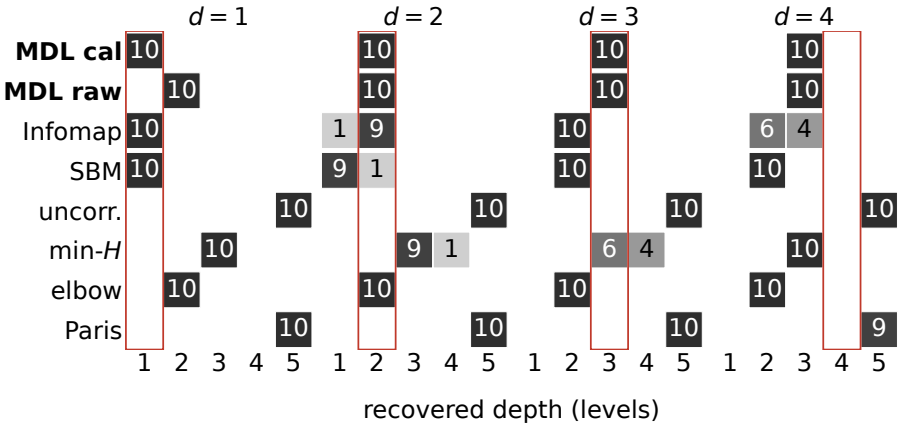


Figure 5: Recovered depth of every estimator at $r = 6$. Rows: the eight estimators of Table 2, ours in bold (SBM: the nested block model; uncorr.: pure SE; min-H and elbow: the height sweep and its elbow rule). Columns: recovered depth in levels, the last pooling 5 and above. Cell: seeds out of ten, shaded by count; outlined: the planted depth. Paris failed on one d=4 graph (9 seeds).

E Extension experiments

All experiments below were added after the gate; each has its question, design, thresholds and frozen predictions in a preregistration (supplementary material), timestamped before any of its units ran, and the outcomes are reported against them.

E1: the calibration around other objectives (Tables 3, 5). Same graphs and evaluation nulls as Table 2 (checked to be identical). Each method is a native builder plus its own score, used unchanged as the statistic of Definition 2: level 1 compares the score gain of the builder's tree over the flat tree with the same gain on $k = 7$ degree-preserving resamples; level $\ell{+}1$ compares the gain of rebuilding, with the same builder, the subtree inside every block of P_ℓ against resamples that preserve every block-pair edge count of P_ℓ. Builders: recursive spectral sparsest-cut bisection (Fiedler-vector sweep minimising $\mathrm{cut}(A, B)/(|A||B|)$) and Paris (Dasgupta cost $\sum_{uv\in E} |\mathrm{leaves}(\mathrm{lca}(u, v))|$); recursive Leiden, splitting a block while Leiden returns more than one community of positive subgraph modularity (score: minus the sum, over internal nodes, of the modularity excess of the node's split within its own subgraph); multilevel Infomap, 10 trials (score: the multilevel map equation for undirected flow, our implementation, equal to Infomap's codelength on Infomap's own tree); recursive two-level SE partitioning (score $\mathrm{vol} \cdot H^T$). Depth is the modal leaf depth -1 with ties broken toward the shallower level for every method. Preregistered outcomes: E1-a (raw rules over-call nulls) and E1-b (calibrated null flatness $\geq 32/40$ and $\geq 80\%$ over the grid) hold for all five; E1-c ($\geq 7/10$ exact at each $d \leq 3$) holds only for bisection. The frozen predictions held for bisection, the map equation, recursive Leiden and recursive SE; for Paris E1-c was predicted to hold at $d = 1, 2$ and failed at $d = 1$ (6/10). A Paris bug found during the run (a linkage whose rows are sorted by height) was fixed and all Paris units were re-run; only the re-run is reported. Calibrated depths agree across the two machines used on 349 of 352 comparisons; the three differences are Infomap runs.

E3: the false-positive follow-up (Table 6). The level-1 test of the calibrated L depth (a null has calibrated depth ≥ 1 iff it passes) on the 40 nulls at $r = 6$. C0 reproduces the published $34/40$ with the same six nulls. Preregistered rules: the mixing hypothesis needed C2 below C1 (one-sided Fisher $p < 0.05$); observed $29/200$ against $24/200$, $p = 0.81$, so it is not supported. C1 has $18/50$ false positives on $d = 2$ nulls, more than the 10 that would have made the published $5/10$ a resample-draw effect: it is a property of those null graphs. C3 is valid (interval $[0.058, 0.144]$ contains $1/8$). The

fraction of null edges inside planted bottom blocks is 0.23 in every condition (planted graphs: 0.80), so no planted structure survives the swaps.

E2, E4–E7. E6 (Table 9) runs the nested SBM (graph-tool's nested minimiser) over the whole grid, degree-corrected and not, five runs each followed by 100 zero-temperature multiflip sweeps, keeping the lowest description length; both variants give the published $r = 6$ row, recover $d = 2$ at $r = 8$, never $d \geq 3$, and call all 100 nulls flat. E7 (Table 10, added before it ran) met its preregistered criterion ($\geq 7/10$ at each d, $\geq 24/30$ nulls flat). E2 runs multilevel Infomap over the whole grid (Table 2); with the locally available Infomap version the $r = 6$ row reproduces the published depths except one $d = 4$ seed (2 instead of 3; both wrong), and Infomap calls all 100 nulls flat; the frozen prediction that it never recovers $d \geq 3$ failed on one $r = 8$ graph. E4 (Table 7) predicted calibrated L exact on $\geq 3/5$ at $d = 4$ and $\geq 4/5$ at each $d \leq 3$ for $n = 640$, and $\geq 4/5$ at each d for $n = 486$, $b = 3$: it holds except at $d = 1$ ($0/5$ and $1/5$), where a fluctuation sub-level survives; calibrated null flatness was $18/20$ (predicted ≥ 16), and bisection, as predicted, never matches a branching-3 depth. E5 (Table 8) required every method to call $\geq 80\%$ of the real-graph nulls flat; all six do.

Table 5: Objective-agnostic calibration over the whole grid (E1). For each method: fraction of seeds whose calibrated depth equals d (cal), the same for the builder's raw depth (raw), and the fraction of degree-matched nulls the calibrated depth calls flat (nulls). 10 seeds at $r = 6$, 5 elsewhere; '·': zero.

		$r=2$				$r=4$				$r=6$				$r=8$			
method		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
sparsest-cut bisection	cal	·	0.4	0.2	·	0.8	0.8	0.4	1.0	0.9	0.7	0.8	1.0	0.6	0.8	1.0	0.6
	raw	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
	nulls	0.8	1.0	1.0	0.8	1.0	1.0	1.0	0.8	0.9	0.9	0.9	1.0	0.8	1.0	1.0	1.0
Paris	cal	0.2	·	·	0.4	0.4	0.4	0.6	0.6	0.6	0.8	0.5	1.0	1.0	0.8	1.0	0.8
	raw	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
	nulls	0.8	0.6	0.8	1.0	0.8	0.8	0.8	1.0	1.0	1.0	1.0	1.0	1.0	0.6	1.0	1.0
recursive Leiden	cal	·	0.2	·	·	1.0	·	·	·	0.8	·	0.1	·	1.0	·	·	·
	raw	·	0.2	1.0	·	·	·	0.6	·	·	·	0.5	·	·	·	0.8	·
	nulls	1.0	1.0	0.6	1.0	1.0	1.0	0.8	0.6	1.0	0.9	0.9	0.8	0.8	0.8	0.8	0.8
multilevel Infomap	cal	·	·	·	·	·	·	·	·	1.0	·	·	·	1.0	·	·	·
	raw	·	·	·	·	·	·	·	·	1.0	·	·	·	1.0	·	·	·
	nulls	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
recursive SE	cal	0.2	0.2	·	·	0.4	·	·	·	0.8	0.1	·	·	1.0	0.2	·	·
	raw	·	·	1.0	·	·	·	1.0	·	·	·	0.6	·	·	·	·	·
	nulls	0.8	0.8	0.8	1.0	0.8	1.0	0.8	0.8	0.8	0.9	0.9	0.9	1.0	1.0	0.8	1.0

Table 6: Level-1 false positives of the calibrated L depth on the 40 nulls at $r = 6$ (E3). FP: tests in which a null passes level 1 (so its calibrated depth is ≥ 1); 95% Clopper–Pearson interval; per d of the graph the null was drawn from; within: mean fraction of null edges inside planted bottom blocks (planted graphs: 0.80).

	condition	FP	rate	95% CI	d=1	d=2	d=3	d=4
C0	published null and resamples ($30m$ swaps)	6/40	0.150	[0.057, 0.298]	0/10	5/10	0/10	1/10
C1	published null, 5 fresh resample sets	24/200	0.120	[0.078, 0.173]	0/50	18/50	1/50	5/50
C2	fresh null and resamples, $300m$ swaps, 5 sets	29/200	0.145	[0.099, 0.202]	13/50	5/50	4/50	7/50
C3	null and resamples all drawn from G, 5 sets	19/200	0.095	[0.058, 0.144]	6/50	5/50	6/50	2/50

Table 7: Scale and branching (E4; $r = 6$, 5 seeds per cell). Calibrated exact recovery (raw in parentheses) and nulls called flat by the calibrated depth, of 5. Infomap: the convention of Table 2, raw only.

	$n = 640, b = 2$				$n = 486, b = 3$		
method	d=1	2	3	4	1	2	3
L (ours)	0 (0)	5 (5)	5 (1)	5 (5)	1 (1)	5 (5)	5 (5)
nulls flat	4	4	5	5	5	5	4
sparsest-cut bisection	4 (0)	4 (0)	4 (0)	5 (0)	0 (0)	0 (0)	0 (0)
nulls flat	5	3	3	3	5	4	3
Paris	3 (0)	0 (0)	1 (0)	3 (0)	0 (0)	0 (0)	0 (0)
nulls flat	2	5	4	4	4	5	4
multilevel Infomap (map eq.)	5 (5)	0 (0)	0 (0)	0 (0)	5 (5)	1 (1)	0 (0)
nulls flat	5	5	5	5	5	5	5
Infomap, Table 2 convention	5	5	0	0	5	0	0
nulls flat	5	5	5	5	5	5	5

Table 8: Real graphs (E5). Calibrated depth on the graph (raw in parentheses), and nulls called flat by the calibrated depth, of 5 degree-preserving nulls per graph. No ground-truth depth is claimed; the endpoint is null flatness.

graph	L	bisect.	Paris	Leiden	Infomap	rec. SE
karate	1 (1)	2 (6)	3 (4)	1 (2)	1 (1)	0 (2)
nulls flat	5/5	4/5	4/5	4/5	4/5	5/5
dolphins	2 (2)	5 (7)	3 (5)	2 (2)	1 (1)	0 (2)
nulls flat	4/5	5/5	5/5	5/5	5/5	4/5
Les Mis.	2 (2)	8 (10)	4 (4)	2 (2)	1 (1)	1 (3)
nulls flat	4/5	4/5	3/5	5/5	5/5	5/5
pol. books	1 (1)	4 (4)	6 (6)	3 (3)	1 (1)	2 (3)
nulls flat	4/5	3/5	5/5	5/5	4/5	5/5
football	1 (1)	6 (7)	5 (5)	2 (2)	1 (1)	1 (3)
nulls flat	4/5	4/5	5/5	4/5	5/5	4/5
email-Eu-core	1 (1)	8 (48)	8 (13)	3 (4)	1 (1)	0 (4)
nulls flat	4/5	5/5	5/5	4/5	5/5	4/5
pooled nulls flat	25/30	25/30	27/30	27/30	28/30	27/30

Table 9: Nested SBM over the whole grid (E6): fraction of seeds recovering d and nulls called flat, degree-corrected (DC) and not (NDC), best of 5 minimisations each.

	$r = 2$				$r = 4$				$r = 6$				$r = 8$			
d	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
DC exact	·	·	·	·	1.0	·	·	·	1.0	0.1	·	·	1.0	1.0	·	·
DC nulls	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
NDC exact	·	·	·	·	1.0	·	·	·	1.0	0.1	·	·	1.0	1.0	·	·
NDC nulls	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 10: Degree-corrected hierarchical SBM (E7; $n = 160$, $r = 6$, Pareto degree propensities with tail exponent 2.5, realised degrees 1–159; 10 seeds per d). Calibrated exact recovery (raw in parentheses) and nulls called flat by the calibrated depth.

method	d=1	d=2	d=3	nulls flat
L (ours)	10/10 (10)	10/10 (10)	9/10 (9)	28/30
sparsest-cut bisection	0/10 (0)	1/10 (0)	1/10 (0)	25/30
multilevel Infomap (map eq.)	7/10 (7)	1/10 (1)	0/10 (0)	30/30