Self-Directed Learning of Bounded-Cut Graph Labelings

Anna Zhou

Maryland, United States

Instructor

Jesse Geneson

S.-T. Yau High School Science Award (North America)
Computer Award
2026

Self-Directed Learning of Bounded-Cut Graph Labelings

Anna Zhou

Abstract

We study exhaustive prediction of an unknown categorical labeling on a known graph under a self-directed learning model. The learner chooses an unseen vertex, predicts its label, and then observes the truth; the realizable target has bounded total weight of edges whose endpoints have different labels. The feasible class is equivalently a sublevel set of categorical graph total variation. Our main geometric tool is an exchange theorem: relative to a minimum-cost completion of any history, every feasible completion decomposes into independently reversible blocks. These blocks generate feasible Boolean cubes and yield adaptive mistake lower bounds. We then characterize how prediction changes as the number of available labels grows. A graph-fragmentation parameter guarantees eventual stabilization of the minimax value and a single palette-blind optimal policy schema, yet every palette can be the first stabilization point. Value stability and policy stability behave strikingly differently: whenever two palette values agree, optimal policies transfer perfectly; once they separate, there are unweighted instances where every richer-palette optimum is suboptimal on the smaller class, with adjacent palettes admitting arbitrarily large value gaps that every richer-palette optimum pays in full on the smaller class. Restricting exploration to remain connected can also nearly double the optimal number of mistakes. Finally, removing realizability replaces version-space geometry by Hamming geometry. For every binary comparator class, deterministic regret is exactly its Chebyshev radius, so neither feedback nor adaptive order improves on one fixed prediction word. With three labels, feedback can instead provide a sharp factor-two advantage.

Keywords: self-directed learning; online graph prediction; bounded cuts; policy transfer; connected exploration; agnostic regret; computational complexity.

Contents

1 Introduction **3**

2 Exchange and graph geometry **8**
2.1 Exchange geometry and certificates 10
2.2 Exact benchmarks on elementary graph families 13

3 Palette saturation **18**
3.1 Two construction operations 21
3.2 Every stabilization threshold occurs 23
3.3 A weighted treewidth scale 25
3.4 Hardness of the mistake bound 26

4 Policy transfer **27**
4.1 Transfer spectra among upper-palette optima 29
4.2 A construction for nonadjacent palettes 30
4.3 The binary lower palette 32

5 The price of local access **33**

6 Deterministic agnostic regret **36**
6.1 Equality patterns and palette stabilization 38
6.2 Exact agnostic graph benchmarks 39
6.3 Computational complexity 41

7 Discussion **42**

8 Conclusion **43**

9 Acknowledgements **44**

A Technical verification for the palette constructions **46**
A.1 Role discovery and selector accounting 46
A.2 The difficult upper-palette child 47
A.3 The weighted treewidth construction 47

B Detailed transfer-gadget verifications **48**
B.1 Pooled-star scalar accounting 48
B.2 The two-role release used for the binary palette 50
B.3 Separated-palette construction accounting 51
B.4 Exact exhaustive computation for the universal binary construction 55

C The transfer-spectrum seed **56**

D Proof of the agnostic path-completion lemma **58**

1 Introduction

Imagine labeling the regions of an aerial image as "forest", "water", "road", or "building". Before each label is revealed, an automated assistant predicts it; a human reviewer then confirms or corrects the prediction. But the assistant also chooses which unlabeled region to inspect next. Because nearby regions often share a category, every revealed label does double duty: it informs both future predictions and where to inspect next. Machine-assisted image and corpus annotation provide familiar examples of this predict–review–correct workflow (Ringger et al., 2008; Andriluka et al., 2018). Classical online learning assumes that examples arrive in an externally determined order; in *self-directed learning*, the learner chooses that order itself. How much can this freedom reduce the number of inevitable mistakes?

This paper isolates the exact finite combinatorial problem behind that question. The input is a known, finite, simple, connected graph $G = (V, E, w)$ with nonnegative edge weights and a nonnegative cut budget γ. An unknown labeling $y : V \to [q]$ satisfies

$$C_G(y) = \sum_{uv \in E} w_{uv} \mathbf{1}\{y(u) \neq y(v)\} \leq \gamma,$$

where $C_G(y)$ is its *cut cost*. The primary model is unweighted, so this simply bounds the number of bichromatic edges; every result requiring nonuniform weights is identified explicitly. For a palette of size $q \geq 2$, write $[q] = \{1, \ldots, q\}$ and define the hard bounded-cut class

$$\mathcal{H}_\gamma^{(q)}(G) := \{h : V \to [q] : C_G(h) \leq \gamma\}. \tag{1}$$

We abbreviate this class to $\mathcal{H}_\gamma^{(q)}$.

Unlike ordinary online prediction, the learner controls which unlabeled vertex to predict next. In each round a deterministic policy chooses an unqueried vertex x, predicts a label $p \in [q]$, observes $h(x)$, and uses the entire history to choose its next action. Every vertex is eventually queried. If $M(A, h)$ is the number of mistakes made by policy A on target h, the free-order minimax value is

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, G) := \min_A \max_{h \in \mathcal{H}_\gamma^{(q)}(G)} M(A, h).$$

We also write $V_{\text{free}} = M_{sd}(\mathcal{H}_\gamma^{(q)}, G)$ when comparing protocols.

Why learner-chosen order matters. The simplest graph already separates learner-chosen from externally controlled order. In the externally presented binary model of Cesa-Bianchi et al. (2009), the class with exactly one bichromatic edge on an n-vertex path has deterministic minimax value $\Theta(\log n)$. In our at-most-one-cut class, which also contains the constant labelings, a self-directed learner starts at an endpoint and walks consecutively along the path, always predicting the label just revealed at the previous vertex. Only the initial prediction and the unique possible label change can cause mistakes, and Corollary 2.10 shows that the value is exactly 2 for $n \geq 2$. Thus choosing what to inspect next changes the combinatorics already on the simplest graph.

Even this simple formulation produces several sharp changes of behavior. The main geometric tool behind the realizable results is an exchange principle: one minimum-cost completion and one feasible completion can generate a Boolean cube of independently reversible alternatives. The minimax value always eventually stabilizes as labels are added, and a budgeted graph-fragmentation parameter controls when this must happen; yet every palette can be the first stabilization point. Equality of two palette values forces optimal policies to transfer perfectly, while after a strict

increase there are instances on which every richer-palette optimum pays the entire increase on the smaller class. Requiring exploration to remain connected can almost double the number of mistakes. Without realizability, version-space geometry gives way to Hamming geometry: the optimal regret for every binary comparator class is exactly its Chebyshev radius, whereas a third label can restore a sharp factor-two advantage from feedback.

We next record the history and local-access definitions needed to state the results precisely.

After a partial history (Q, y_Q), where $Q \subseteq V$ has been queried, the *version space* is

$$\mathcal{V}(Q, y_Q) = \{h \in \mathcal{H}_\gamma^{(q)}(G) : h|_Q = y_Q\}.$$

We consider only histories with nonempty version space. A completion of (Q, y_Q) is any labeling consistent with y_Q, whether or not it meets the budget. A *minimum-cost completion* minimizes C_G among all such labelings. Realizability implies that every minimum-cost completion is feasible.

The maximization is formally over a fixed target $h \in \mathcal{H}_\gamma^{(q)}$. For lower bounds we use the equivalent realizable-adversary view: after each query, an adversary may return any label consistent with at least one remaining target. As long as the version space stays nonempty, the complete response sequence ultimately defines one target in $\mathcal{H}_\gamma^{(q)}$.

We also compare free exploration with two local restrictions. A policy is *connected-prefix* if its first query is arbitrary and every later query is adjacent to some previously queried vertex. Equivalently, every realized queried prefix induces a connected subgraph; the next query need not be adjacent to the previous one. It is *neighbor-anchored connected-prefix* if, in addition, every noninitial prediction must equal the revealed label of some previously queried neighbor of the current vertex. Denote the corresponding initial minimax values by V_{conn} and V_{anch}. Then

$$V_{\mathrm{free}} \le V_{\mathrm{conn}} \le V_{\mathrm{anch}}.$$

The first prediction is unrestricted in all three games. More generally, $V_{\mathrm{free}}(H; G)$, $V_{\mathrm{conn}}(H; G)$, and $V_{\mathrm{anch}}(H; G)$ denote the three minimax values for an arbitrary target class $H \subseteq [q]^V$; the argument G records which adjacency restrictions are legal.

Main results

Two linked questions run through the realizable theory: what geometry forces mistakes, and, as labels are added, when do the minimax values stabilize—and do their optimal policies stabilize with them? We also quantify the cost of local exploration and ask what replaces version-space geometry once realizability is removed. The following results form the conceptual core.

Exchange geometry: one pair of completions generates a cube. Our main geometric tool begins at an arbitrary realizable history. Compare a minimum-cost completion $h^\star$ with any feasible completion g, and split their disagreement set into exchange blocks using the positive-weight edges that are monochromatic in at least one of the two labelings. Theorem 2.5 shows that reverting any collection of these blocks from g to $h^\star$ cannot increase the cut cost. The two completions therefore provide much more than two alternatives: t exchange blocks generate all 2^t vertices of a feasible Boolean cube and force t further mistakes. On a connected graph with positive edge weights, Proposition 2.7 identifies the largest empty-history certificate with the maximum number of components induced by a vertex set whose edge boundary fits the cut budget. The exchange theorem thus connects low-cut geometry to the cube lower bounds used throughout the realizable theory.

The palette eventually stops mattering, but not at a universal small palette. The values are nondecreasing in q and eventually stabilize. For a connected graph with positive edge weights, the budgeted fragmentation number $\beta_\gamma(G)$ is the largest number of connected pieces obtainable by deleting edges of total weight at most γ. Writing $b = \beta_\gamma(G)$, every feasible target uses at most b labels and $M_{sd}(\mathcal{H}_\gamma^{(q)}, G) \geq b$. Theorem 3.2 proves that the free-order and connected-prefix values stabilize once $q \geq b + 1$, while the neighbor-anchored value stabilizes already at $q \geq \max\{2, b\}$. In each case, one conservative palette-blind equality-pattern policy schema is optimal throughout the stable range. This is a universal mechanism, not a universal small threshold: for every $q_0 \geq 2$, Theorem 3.5 constructs a connected simple unweighted instance whose first free-order stabilization palette is exactly q_0.

Value stability and policy stability separate sharply. If allowing more labels does not change the minimax mistake bound, do optimal strategies transfer? For nested palettes $r < s$, equality $M_{sd}(\mathcal{H}_\gamma^{(r)}, G) = M_{sd}(\mathcal{H}_\gamma^{(s)}, G)$ gives a rigid affirmative answer: every s-optimal policy has optimal loss on the r-label subclass and can be legalized into an r-optimal policy. The opposite behavior is possible as soon as the values separate. For every $2 \leq r < s$, Theorem 4.6 gives a connected simple unweighted instance with a one-mistake value increase on which every s-optimal policy pays that extra mistake on r-label targets. For adjacent palettes and any $D \geq 1$, Theorem 4.1 makes the value gap D and forces every richer-palette optimum to pay the entire gap on the smaller class. Theorem 4.2 further realizes a complete spectrum of lower-palette losses among policies that are all optimal on the richer class. All results are analytic except one finite binary nonadjacent case, certified by the exact exhaustive Bellman computation reproduced in Appendix B.4 and independently reproducible from the archived source code (Zhou, 2026).

Connected exploration can almost double the cost. Theorem 5.2 constructs, for every integer $M \geq 2$, a connected simple unweighted graph such that, simultaneously for every palette $q \geq 2$,

$$V_{\text{free}} = M, \qquad V_{\text{conn}} = V_{\text{anch}} = 2M - 1.$$

The ratio approaches two. Consequently, if a universal multiplicative comparison with free play exists for either local protocol, its constant cannot be smaller than two.

Without realizability, Hamming geometry takes over. When the revealed labeling is arbitrary, the bounded-cut class serves only as a hindsight comparator. Regret is the learner's mistakes minus the fewest disagreements between the response labeling and a comparator. The binary theorem no longer depends on graphs at all: for every nonempty $\mathcal{H} \subseteq \{0,1\}^V$, Theorem 6.2 gives

$$R_{\text{sd}}^{\text{det}}(\mathcal{H}) = \text{rad}(\mathcal{H}),$$

where $R_{\text{sd}}^{\text{det}}$ denotes optimal deterministic regret and rad is the Chebyshev radius in Hamming space. An optimal learner can ignore both feedback and adaptive order, query in any order, and predict one fixed Chebyshev-center word. A third label changes the answer immediately: Theorem 6.3 gives a ternary two-vertex class with adaptive regret 1 while every fixed word has worst-case regret 2. Thus feedback recovers the sharp factor-two advantage.

Supporting benchmarks and computational consequences. The structural theory yields exact values for all three protocols on trees with positive edge weights and on unweighted complete graphs, cycles, and edge–triangle cacti. It also gives a sharp factor-two comparison between the

exchange certificate and the exact tree value. Exact realizable evaluation is NP-hard already for binary unweighted instances (Theorem 3.7). In the agnostic theory we also prove palette stabilization and exact unweighted clique, path, and cycle formulas. Exact agnostic evaluation is NP-hard on binary unweighted instances and, with positive integer weights, for every fixed palette (Corollary 6.9 and Theorem 6.10). All headline separation witnesses are connected, simple, and unweighted; every result using nonuniform weights states that scope explicitly.

Relation to prior work

Hard cut budgets in sequential prediction. The closest prior sequential/minimax formulation is due to Cesa-Bianchi et al. (2009). For a known unweighted tree T and $1 \le K < |V|$, they define a deterministic minimax value over binary labelings with between one and K bichromatic edges and characterize it up to constant factors. Thus, apart from their formal exclusion of zero-cut labelings, the structural promise is the binary specialization of (1). Their maximization also ranges over the vertex-presentation order, whereas our learner chooses that order; this is the source of the path separation highlighted above.

For trees with positive edge weights, Vitale et al. (2011) define $\xi(M)$ as the largest number of edges having total weight at most M, precisely our $\nu_\gamma(T)$, and study binary prediction under an unknown arbitrary order. A randomized feasible label assignment gives their $\xi(M)/2$ expected lower bound, and Shazoo has nearly matching tree-sensitive guarantees. With learner-chosen order, our deterministic value is exactly $\xi(\gamma) + 1$ for every palette and under both local protocols.

Cut-sensitive prediction under external order. In standard adversarial online prediction the environment supplies the instance order (Cesa-Bianchi and Lugosi, 2006). Graph predictors in this model commonly bound mistakes by properties of the realized cut rather than compute the minimax value of a fixed cut sublevel. Binary Perceptron bounds use cutsize and resistance diameter (Herbster and Pontil, 2006). The low-congestion-routing reduction of Fakcharoenphol and Kijsirikul (2008) gives a congestion-times-cutsize bound for arbitrary label sets, with one added initial prediction under our convention. For simple unweighted graphs, the charging argument of Brautbar (2009, Lemma 2.3.2 and Theorem 2.3) implies that binary neighborhood majority makes at most one plus the realized cutsize on a connected-prefix presentation. On K_n, the same rule makes at most $2r + 1$ mistakes when the minority class has size r, with a matching externally ordered deterministic lower bound when $n \ge 4r + 1$; Theorem 2.12 gives the exact hard-budget value for every n, budget, palette, and protocol.

For binary weighted graphs, random-spanning-tree methods give general lower bounds and, under their connectivity and weighted-cut conditions, expected-mistake upper bounds matching up to logarithmic factors in terms of the expected cutsize on the sampled tree (Cesa-Bianchi et al., 2013). The zero-temperature Ising method matches the known tree bounds up to constant factors and also gives per-cluster guarantees (Herbster et al., 2015). Restricting such realized-cut guarantees to a cut sublevel gives uniform bounds, but not the exact learner-ordered decision-tree value studied here.

Self-directed prediction on graphs. The finite-pool self-directed protocol was introduced for binary relations by Goldman et al. (1993), developed as a general learning model by Goldman and Sloan (1994), and given further binary combinatorial and general dimensional treatments by Ben-David and Eiron (1998); Devulapalli and Hanneke (2024). Recent work also studies self-directed linear classification (Diakonikolas et al., 2023). The closest work in the same generic graph protocol is that of Sokolov et al. (2025). Their principal classes have convex or near-convex clusters rather

than a cut-cost budget, but their TRAVERSE algorithm uses the same basic neighbor-copying traversal mechanism as our spanning-tree policy: it learns any multiclass labeling with at most $|\partial C_y| + 1$ mistakes, where ∂C_y consists of vertices incident to bichromatic edges. In our terminology, TRAVERSE is neighbor-anchored connected-prefix. For an unweighted target in (1), its guarantee is at most $2\lfloor\gamma\rfloor + 1$, whereas the sharper cut-edge accounting of our spanning-tree policy gives the uniform bound $1 + \min\{\lfloor\gamma\rfloor, |V| - 1\}$. We additionally optimize over the whole sublevel, obtain exact values on several graph families, and isolate palette phenomena.

Label queries and other learner-chosen information. Active learning ordinarily charges for labels that the learner requests and then uses to predict unqueried items (Cohn et al., 1994; Dasgupta, 2005). Several graph label-query models use the same small-cut assumption or parameter but charge for revealed labels rather than mistaken predictions. They include fixed-batch weighted-cut guarantees (Guillory and Bilmes, 2009), constant-factor optimal query placement for binary tree labelings with cutsize at most K (Cesa-Bianchi et al., 2010), and adaptive reconstruction under additional balance or cut-component assumptions (Afshani et al., 2007; Dasarathy et al., 2015). Mincut transduction instead selects a minimum-cut completion of a labeled subset (Blum and Chawla, 2001). Other nearby protocols combine learner-chosen training labels with a later externally ordered phase (Herbster et al., 2005), or reveal an initially unknown graph by letting the environment supply a neighbor of a learner-chosen anchor (Cesa-Bianchi et al., 2011). In our game every vertex is predicted before revelation, and the learner chooses the precise next vertex.

The bounded-cut class as hypothesis class and comparator. For the usual graph Laplacian L, a binary $\{-1, 1\}$ encoding satisfies $h^\top Lh = 4C_G(h)$, while the sum of the q one-hot coordinate energies is $2C_G(h)$. Hanneke (2006) bounds the cardinality of multiclass sublevels with at most c bichromatic edges and derives transductive error bounds, also treating rationally weighted cuts. Lever (2010) studies the binary class $\{h : h^\top Lh \le \phi\}$ directly and bounds its transductive Rademacher complexity using resistance-metric cluster structure.

In the agnostic setting, Rakhlin and Sridharan (2015) use hard budgets on violated graph-label constraints as comparator classes and derive semidefinite regret relaxations with evolving constraints. Rakhlin and Sridharan (2017) use the multiclass one-hot Laplacian sublevel—exactly the same categorical cut class after a factor-two reparameterization—as a comparator for randomized online prediction, and Zhou et al. (2023) develop scalable implementations of related graph-kernel online relaxations. Classical agnostic online learning also uses an externally supplied instance sequence (Ben-David et al., 2009); our agnostic problem instead asks for exact deterministic regret when the learner chooses the order. Our exact policy-transfer question likewise differs from online model-selection oracle inequalities, which include model-dependent adaptation terms (Foster et al., 2017).

Cut sublevels and graph energies. If the label at v is represented by the standard basis vector $z_v = e_{y(v)}$, then

$$C_G(y) = \frac{1}{2} \sum_{uv \in E} w_{uv} \|z_u - z_v\|_1.$$

Thus (1) is exactly a categorical restriction of a graph-total-variation sublevel; graph total variation is widely used to model piecewise-constant signals on networks (Jung et al., 2019; Madrid Padilla et al., 2018). If $W = \sum_{e \in E} w_e$, then

$$C_G(y) = W - \sum_{uv \in E} w_{uv} \mathbf{1}\{y(u) = y(v)\},$$

so, up to the additive constant W, cut cost is the zero-field ferromagnetic Potts energy (Wu, 1982). Its disagreement penalty is the uniform-metric pairwise term in metric labeling and a standard pairwise term in graphical-model energies used in vision (Kleinberg and Tardos, 2002; Boykov and Jolly, 2001; Boykov et al., 2001).

These identities relate the models, but energy minimization and prediction ask different questions. Minimum-cut closure representations characterize all optima of a fixed s–t cut instance (Picard and Queyranne, 1980); parametric maximum flow can certify part of a Potts optimum (Gridchyn and Kolmogorov, 2013); and fusion moves optimize over labelings that choose, coordinatewise, between two proposals (Lempitsky et al., 2010). Our learner must instead distinguish every labeling in a fixed energy sublevel through an adaptive decision tree.

Section 2 develops minimax tools, geometric certificates, and exact graph families. Sections 3 and 4 separate palette values from optimal-policy transfer, and Section 5 exhibits a near-factor-two separation for locally restricted policies. Section 6 develops the complementary agnostic theory, followed by a brief conclusion and open problems.

2 Exchange and graph geometry

We begin with the recursion for the number of additional mistakes at a game state. For a finite set X, a nonempty target class $K \subseteq [q]^X$, and a set $U \subseteq X$ of unqueried coordinates, let $v(K, U)$ denote the free-order minimax number of additional mistakes when the target is known to lie in K and exactly the coordinates in U remain to be queried. When $U = X$, write simply $v(K)$. For $x \in U$ and $a \in [q]$, let

$$K_{x=a} := \{h \in K : h(x) = a\}.$$

Then

Proposition 2.1. *For $U \neq \varnothing$,*

$$v(K, U) = \min_{x \in U} \min_{p \in [q]} \max_{a:\, K_{x=a} \neq \varnothing} \big(\mathbf{1}\{p \neq a\} + v(K_{x=a}, U \setminus \{x\})\big),$$

with $v(K, \varnothing) = 0$. In particular,

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, G) = v(\mathcal{H}_\gamma^{(q)}(G), V).$$

Proof. The learner first chooses (x, p). The realizable adversary may return exactly the labels a for which $K_{x=a}$ is nonempty. The displayed term is the immediate loss plus the value of the resulting subproblem. Induction on $|U|$ proves the formula. □

We will repeatedly use two elementary families of target classes as lower-bound certificates.

A *Boolean d-cube* is a target class with the following product structure. Its nonfixed vertices are partitioned into nonempty blocks $X_1, \dots, X_d$. On each X_i there are two distinct label patterns u_i^0, u_i^1, and every one of the 2^d independent choices $(u_1^{\varepsilon_1}, \dots, u_d^{\varepsilon_d})$, with $\varepsilon_i \in \{0, 1\}$, occurs in the class. Vertices outside these blocks, if any, have fixed labels. Thus a query in one block gives no information about any other block. Its free-order minimax value is d: the realizable adversary can defeat the first prediction that distinguishes each unresolved block, while one distinguishing query per block gives the matching upper bound.

A *punctured Boolean d-cube*, for $d \geq 1$, is obtained by deleting one corner of a full Boolean d-cube; its value is $d - 1$. To see this, encode the deleted corner by 0^d and query one distinguishing coordinate per block, predicting its bit-1 pattern until a response confirms that choice. Either a

prediction is correct, leaving at most $d-1$ mistakes in total, or the first $d-1$ responses are bit 0, forcing the last bit to be 1. Conversely, fixing one bit to 1 leaves a full $(d-1)$-cube. These elementary classes will be used repeatedly as mistake lower-bound certificates.

We will also use Hamming balls. For integers $0 \le h \le d$, a binary Hamming ball of radius h in d coordinates has value h: predicting the center coordinatewise gives the upper bound, while the realizable adversary can oppose the learner's first h predictions for the lower bound.

Next we establish a law of exact additivity for genuinely independent factors.

Lemma 2.2. *Let $X_1, \dots, X_t$ be pairwise disjoint, let $K_i \subseteq [q]^{X_i}$ be nonempty, and let $U_i \subseteq X_i$. Writing*

$$\prod_{i=1}^{t} K_i := \left\{ h \in [q]^{\bigsqcup_i X_i} : h|_{X_i} \in K_i \text{ for every } i \right\},$$

we have

$$v\left(\prod_{i=1}^{t} K_i, \bigsqcup_{i=1}^{t} U_i\right) = \sum_{i=1}^{t} v(K_i, U_i).$$

Queries may be interleaved arbitrarily among the factors.

Proof. Running optimal policies for the factors sequentially gives the upper bound.

For the reverse inequality, induct on $\sum_i |U_i|$, with the empty case immediate. Suppose the learner's first query x lies in U_j. After response a,

$$\left(\prod_{i=1}^{t} K_i\right)_{x=a} = (K_j)_{x=a} \times \prod_{i \ne j} K_i,$$

while the remaining coordinate set in factor j is $U_j \setminus \{x\}$. By the induction hypothesis, the continuation value is

$$v((K_j)_{x=a}, U_j \setminus \{x\}) + \sum_{i \ne j} v(K_i, U_i).$$

For the chosen query and prediction, the Bellman recursion supplies a response whose immediate loss plus the first term is at least $v(K_j, U_j)$. Hence every first action has value at least

$$\sum_{i=1}^{t} v(K_i, U_i),$$

which proves the lower bound. □

The product hypothesis is literal: the lemma applies when the target class itself factors as $\prod_i K_i$. Graph modules sharing a single cut budget generally admit additional mixed targets, so later constructions invoking product subclasses verify separately that those subclasses lie within the pooled budget.

We also consider shared policies across nested palettes. Identify $[r]$ with a subset of $[s]$ whenever $2 \le r < s$. Then $\mathcal{H}_\gamma^{(r)}(G) \subseteq \mathcal{H}_\gamma^{(s)}(G)$. For any graph and budget, write

$$m_q(G, \gamma) := M_{sd}(\mathcal{H}_\gamma^{(q)}, G).$$

When G and γ are fixed, abbreviate this quantity to m_q. For an s-label policy A, put

$$L_q(A) = \max_{h \in \mathcal{H}_\gamma^{(q)}(G)} M(A, h), \qquad \mathrm{Opt}_s = \{A : L_s(A) = m_s\}.$$

Here $L_r(A)$ means that the s-palette decision tree is run on the nested r-label target subclass; its predictions are still names in $[s]$. This convention measures literal policy transfer. A policy is *simultaneously minimax* for r and s, in this nested-class sense, if $L_r(A) = m_r$ and $L_s(A) = m_s$.

Definition 2.3. The best- and worst-choice downward transfer overheads are

$$\Delta^-_{r\leftarrow s} = \min_{A\in \mathrm{Opt}_s} \big(L_r(A) - m_r\big), \qquad \Delta^+_{r\leftarrow s} = \max_{A\in \mathrm{Opt}_s} \big(L_r(A) - m_r\big).$$

Thus $\Delta^- = 0$ precisely when a simultaneous optimum exists.

For completeness, every such transferred decision tree has a legal r-palette implementation. Run A internally, feed it each actual response, and whenever it predicts outside $[r]$, replace that prediction by a fixed name in $[r]$. Call the resulting policy $A^{\downarrow r}$. Its queries are exactly those of A, and an out-of-range prediction was certainly wrong, so

$$M(A^{\downarrow r}, h) \le M(A, h) \quad \text{for every } h \in \mathcal{H}^{(r)}_\gamma(G). \tag{2}$$

The unchanged queries preserve connected-prefix legality. In neighbor-anchored play, an out-of-range prediction can occur only on the unrestricted first round: every later legal prediction is a revealed neighbor label and hence belongs to $[r]$. Replacing that first prediction therefore preserves neighbor anchoring as well.

It follows that $m_r \le L_r(A)$ for every s-label policy A. Target-class inclusion also gives $L_r(A) \le L_s(A)$. Applying these observations to $A \in \mathrm{Opt}_s$ gives

$$m_r \le m_s, \qquad 0 \le \Delta^-_{r\leftarrow s} \le \Delta^+_{r\leftarrow s} \le m_s - m_r. \tag{3}$$

In particular, if $m_r = m_s$, every s-optimal policy has optimal loss on the r-label subclass, and its legalization $A^{\downarrow r}$ is an optimal legal r-palette policy. A map of label names can only merge label classes, so

$$C_G(\phi \circ h) \le C_G(h)$$

for every map ϕ; this elementary coarsening fact will be used repeatedly.

2.1 Exchange geometry and certificates

Definition 2.4. Fix a realizable partial history (Q, y_Q), a minimum-cost completion $h^\star$, and another arbitrary completion g. Let

$$D(h^\star, g) = \{v : h^\star(v) \ne g(v)\}.$$

On this vertex set, retain a positive-weight edge uv exactly when it is monochromatic in $h^\star$ or in g. The connected components of the retained graph are the *exchange blocks* of the pair $(h^\star, g)$. The definition is pair-specific: different feasible completions can induce different blocks.

For binary labels with fixed opposite terminals, minimum-cost completions reduce to minimum cuts, whose classical lattice structure is described by Picard and Queyranne (1980). For multiclass label spaces, the coordinatewise two-proposal space is the fusion space used in graphical-model optimization (Lempitsky et al., 2010). Our exchange statement has different quantifiers: one labeling is a minimum-cost completion, the other may be any completion of the same history, and all blockwise partial reversions are controlled simultaneously.

Theorem 2.5. *Let G be a finite graph with nonnegative real edge weights, and fix the target class $\{h : C_G(h) \le \gamma\}$. At any realizable history ρ, let $h^\star$ minimize C_G among all completions of ρ, and let g be any other completion of the same history. Let $B_1, \dots, B_t$ be the exchange blocks of $(h^\star, g)$. For $I \subseteq [t]$, let h_I use g on $\bigcup_{i \in I} B_i$ and use $h^\star$ elsewhere. Then*

$$I \subseteq J \quad \Longrightarrow \quad C_G(h_I) \le C_G(h_J).$$

Consequently, if g is feasible, every h_I is feasible. If g is also a minimum-cost completion, all 2^t partial reversions are minimum-cost completions.

Proof. Because $h^\star$ and g complete the same history, their disagreement set is disjoint from the queried vertices, so every h_I is a completion. Put $F(I) = C_G(h_I)$. Each graph edge meets at most two blocks, thus

$$F(I) = F(\varnothing) + \sum_{i \in I} a_i + \sum_{\{i,j\} \subseteq I} b_{ij}.$$

Minimality of $h^\star$ gives $a_i = F(\{i\}) - F(\varnothing) \ge 0$.

Only edges joining B_i and B_j contribute to b_{ij}. Orient one such positive-weight edge uv with $u \in B_i$ and $v \in B_j$. The edge was not retained, so it is bichromatic in both completion labelings. Its costs when neither block, only B_i, only B_j, or both blocks use g are

$$w_{uv}, \quad w_{uv}\mathbf{1}\{g(u) \ne h^\star(v)\}, \quad w_{uv}\mathbf{1}\{h^\star(u) \ne g(v)\}, \quad w_{uv},$$

respectively. Its contribution to b_{ij} is consequently

$$w_{uv}\big(2 - \mathbf{1}\{g(u) \ne h^\star(v)\} - \mathbf{1}\{h^\star(u) \ne g(v)\}\big) \ge 0.$$

Zero-weight edges contribute nothing. Hence every coefficient in the increment

$$F(I \cup \{i\}) - F(I) = a_i + \sum_{j \in I} b_{ij}$$

is nonnegative. This proves monotonicity.

If g is feasible, every partial reversion has cost at most $C_G(g) \le \gamma$. If g is also a minimum-cost completion, then

$$F(\varnothing) = F([t]).$$

For every $I \subseteq [t]$, monotonicity gives

$$F(\varnothing) \le F(I) \le F([t]),$$

so equality holds throughout. Hence every h_I is a minimum-cost completion. □

Remark. Unary data terms do not change the exchange argument. Consider the Potts objective with unary potentials (Lempitsky et al., 2010)

$$\Phi(h) = \sum_{v \in V} \theta_v(h(v)) + C_G(h),$$

where the θ_v are arbitrary real unary potentials. If $h^\star$ minimizes Φ among completions of the history and feasibility is defined by a sublevel of Φ, the same blockwise monotonicity holds: the unary potentials modify only the linear coefficients a_i, which remain nonnegative by minimality, while the interblock coefficients b_{ij} are unchanged. Thus the exchange theorem and its Boolean-cube certificate extend beyond the zero-unary Potts model. The later palette, transfer, and locality results, however, concern the cut-budget class (1) and do not automatically extend to general Φ-sublevels.

Corollary 2.6. *If $h^\star$ is a minimum-cost completion and g a feasible completion at history ρ, and the pair has t exchange blocks, then the residual version space contains a full Boolean t-cube of free-order value t. Consequently, the full residual game has value at least t under each of the three protocols.*

Proof. Theorem 2.5 supplies all 2^t mixed labelings obtained by choosing, independently on each exchange block, either the $h^\star$-pattern or the g-pattern. Coordinate i is precisely this binary choice on B_i; its two patterns disagree at every vertex of the nonempty block. The first query in each unresolved block can therefore be answered using the pattern whose label differs from the learner's prediction, and choices on distinct blocks remain independent. This forces one mistake per block. Conversely, one representative query from each block identifies its pattern, so the subclass has value exactly t. □

On a connected graph with positive edge weights, the empty-history exchange certificate has a simple graph-theoretic interpretation. For $F \subseteq E$, write $w(F) = \sum_{e \in F} w_e$, and for any graph H, let $c(H)$ denote its number of connected components. For $D \subseteq V$, let

$$\delta(D) := \{uv \in E : u \in D,\, v \notin D\}$$

denote its edge boundary. A labeling that differs from a constant labeling precisely on D spends at least $w(\delta(D))$ of the cut budget, while each component of the induced subgraph $G[D]$ contributes one exchange block. The binary labeling used below spends exactly this amount.

Proposition 2.7. *Let G be connected with positive edge weights, and let ξ_0 be the largest number of exchange blocks over all pairs $(h^\star, g)$ in which $h^\star$ is an empty-history minimum-cost completion and g is feasible. For every palette size $q \geq 2$,*

$$\xi_0 = \max_{\substack{\varnothing \neq D \subseteq V \\ w(\delta(D)) \leq \gamma}} c(G[D]). \tag{4}$$

Thus ξ_0 is palette-independent and $\xi_0 \leq M_{sd}(\mathcal{H}_\gamma^{(q)}, G)$.

Proof. On a connected positive-weight graph, the empty-history minimum-cost completions are exactly the constant labelings. Fix the constant label a. If g is feasible and $D = \{v : g(v) \neq a\}$, every edge of $G[D]$ is retained because it is monochromatic in $h^\star$. Hence the exchange blocks are the components of $G[D]$. Every edge of $\delta(D)$ is bichromatic under g, so its total weight is at most γ. This proves the upper bound. Conversely, give a displayed set D a second constant label $b \neq a$ and give its complement label a. The resulting binary labeling has cost exactly $w(\delta(D))$ and produces $c(G[D])$ exchange blocks. □

The root formula makes clear both the usefulness and the limitation of the certificate. In the binary-saturated regime it becomes a familiar graph parameter. Write $\mathrm{MaxCut}_w(G)$ for the maximum total weight of a cut in G, and $\alpha(G)$ for the largest size of an independent vertex set.

Corollary 2.8. *Let G be connected with positive edge weights. If $\gamma \geq \mathrm{MaxCut}_w(G)$, then for every $q \geq 2$,*

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, G) = |V(G)|, \qquad \xi_0 = \alpha(G),$$

where $\alpha(G)$ is the independence number. In particular, on K_n the root certificate is 1 while the minimax value is n.

Proof. Every binary labeling is feasible, so the class contains the full Boolean n-cube. The mistake bound is n for all policies. In (4), every set D is admissible. Selecting one vertex from each component of $G[D]$ gives an independent set, whereas a maximum independent set induces one singleton component per vertex. Hence the maximum is $\alpha(G)$. For K_n, this equals one. □

The underlying identity

$$\max_{D \subseteq V} c(G[D]) = \alpha(G)$$

is classical (Tittmann et al., 2011); here it identifies the root exchange certificate exactly.

2.2 Exact benchmarks on elementary graph families

A rooted-tree traversal gives a simple policy with an exact geometric accounting: predict the root arbitrarily, and thereafter predict the observed label of a vertex's parent. Each noninitial mistake identifies a distinct bichromatic tree edge; the policy is neighbor-anchored connected-prefix. Optimizing the tree produces a general upper bound.

For a spanning tree T of a graph with positive edge weights, define

$$\nu_\gamma(T) = \max\{|F| : F \subseteq E(T),\ w(F) \le \gamma\}, \qquad U_{\mathrm{tree}}(G,\gamma) = 1 + \min_{\substack{T \subseteq G \\ T \text{ a spanning tree}}} \nu_\gamma(T).$$

Proposition 2.9. *For every connected graph with positive edge weights and every $q \ge 2$,*

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, G) \le V_{\mathrm{conn}} \le V_{\mathrm{anch}} \le U_{\mathrm{tree}}(G,\gamma).$$

A spanning tree attaining the minimum is produced by Kruskal's algorithm with edges considered in decreasing order of weight.

Proof. Root T, query each child after its parent, and copy the parent's observed label. The policy is neighbor-anchored connected-prefix. Apart from a possible root mistake, its mistakes correspond to a set of bichromatic edges of T having total weight at most γ, so there are at most $1 + \nu_\gamma(T)$ of them.

For each threshold λ, descending Kruskal selects a maximal forest in the subgraph formed by edges of weight at least λ, and hence includes as many such edges as any spanning tree can. This is equivalent to coordinatewise dominance of the sorted tree-weight vector. Consequently, for every j the resulting tree maximizes the sum of its j lightest edges, minimizing the largest number of tree edges that can fit under the budget. □

Corollary 2.10. *Let T be a finite tree with positive edge weights. For every palette $q \ge 2$,*

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, T) = V_{\mathrm{conn}}(\mathcal{H}_\gamma^{(q)}; T) = V_{\mathrm{anch}}(\mathcal{H}_\gamma^{(q)}; T) = \nu_\gamma(T) + 1.$$

For an unweighted tree the value is

$$\min\{|V(T)|, \lfloor \gamma \rfloor + 1\}.$$

Proof. Proposition 2.9 gives the upper bound. If F attains $\nu_\gamma(T)$, deleting it leaves $\nu_\gamma(T) + 1$ components. Assign binary labels independently to these components. Every resulting target cuts only edges of F, so the class contains a full cube of the stated dimension, supplying the lower bound. □

In contrast to the factor-n separation possible on general graphs, the root exchange certificate is always at least half the tree value, and this bound is sharp.

Corollary 2.11. *For every finite tree T with positive edge weights and every palette $q \geq 2$,*

$$\left\lceil \frac{M_{sd}(\mathcal{H}_\gamma^{(q)}, T)}{2} \right\rceil \leq \xi_0 \leq \max\{1, M_{sd}(\mathcal{H}_\gamma^{(q)}, T) - 1\} \leq M_{sd}(\mathcal{H}_\gamma^{(q)}, T).$$

The factor two is sharp: for every integer $a \geq 1$, the unweighted path P_{2a} with budget $2a-1$ satisfies

$$\xi_0 = a, \qquad M_{sd}(\mathcal{H}_\gamma^{(q)}, P_{2a}) = 2a.$$

Proof. Let F attain $\nu_\gamma(T)$. Then $T - F$ has $\nu_\gamma(T) + 1 = M_{sd}(\mathcal{H}_\gamma^{(q)}, T)$ components. Contracting these components gives a tree. Take the larger side of its bipartition and let D be the union of the corresponding components of $T - F$. Since every quotient edge crosses the bipartition, $\delta_T(D) = F$, while the selected components remain distinct components of $T[D]$. Hence

$$c(T[D]) \geq \left\lceil \frac{\nu_\gamma(T) + 1}{2} \right\rceil = \left\lceil \frac{M_{sd}(\mathcal{H}_\gamma^{(q)}, T)}{2} \right\rceil,$$

and Proposition 2.7 gives the lower bound.

For the upper bound, let $\varnothing \neq D \subsetneq V(T)$ be admissible in the root formula. Since T is connected, every component of $T[D]$ has an incident edge in $\delta_T(D)$, and distinct components require distinct boundary edges. Therefore

$$c(T[D]) \leq |\delta_T(D)| \leq \nu_\gamma(T) = M_{sd}(\mathcal{H}_\gamma^{(q)}, T) - 1,$$

where the second inequality follows from $w(\delta_T(D)) \leq \gamma$. The remaining choice $D = V(T)$ contributes one component, proving the upper bound.

Finally, at budget $2a - 1$ on P_{2a} every edge may be deleted, so

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, P_{2a}) = 2a.$$

At this saturated budget every vertex set is admissible in the root formula, and hence

$$\xi_0 = \max_D c(P_{2a}[D]) = \alpha(P_{2a}) = a.$$

Thus the factor two is attained. □

Theorem 2.12. *Let K_n be the unweighted complete graph on $n \geq 1$ vertices, let $\gamma \geq 0$, and define*

$$t = \max\left\{ j \in \mathbb{Z} : 0 \leq j \leq \left\lfloor \frac{n}{2} \right\rfloor, j(n-j) \leq \gamma \right\}.$$

For every palette size $q \geq 2$,

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, K_n) = V_{\mathrm{conn}}(\mathcal{H}_\gamma^{(q)}; K_n) = V_{\mathrm{anch}}(\mathcal{H}_\gamma^{(q)}; K_n) = \min\{n, 2t+1\}.$$

In particular, all three values are independent of the palette size.

Proof. First suppose that $t < \lfloor n/2 \rfloor$. By maximality,

$$\gamma < (t+1)(n-t-1) \le \frac{n^2}{4}.$$

Every feasible labeling has a unique majority label. Indeed, if its label classes have sizes $a_1, a_2, \ldots$, all at most $n/2$, then

$$\sum_i a_i^2 \le \frac{n}{2} \sum_i a_i = \frac{n^2}{2},$$

and hence

$$C_{K_n}(h) = \frac{1}{2}\left(n^2 - \sum_i a_i^2\right) \ge \frac{n^2}{4} > \gamma,$$

a contradiction. If the majority class has size $n-r$, then $r < n/2$ and the edges between that class and all remaining vertices already contribute $r(n-r)$ to the cut. Since $r \mapsto r(n-r)$ is increasing on $[0, n/2]$, feasibility implies $r \le t$.

For the upper bound, query the vertices in any fixed order. Predict an arbitrary label on the first round and thereafter predict a label of maximum frequency among the responses observed so far, using deterministic tie breaking. This current plurality need not be the final majority. At most $r \le t$ mistakes occur outside the final majority class. Now consider a mistake on the jth occurrence of the final majority label. For $j = 1$, certainly $j \le t+1$. For $j \ge 2$, the predicted current plurality has already appeared at least $j-1$ times, because the final majority has appeared $j-1$ times. That predicted label is not the final majority and therefore appears at most $r \le t$ times in the entire target. Thus $j - 1 \le t$, so only the first $t+1$ majority occurrences can be mispredicted. The total loss is at most $2t+1$.

On a complete graph every fixed query order has connected prefixes, and every observed label is available at every subsequent vertex. The policy is therefore neighbor-anchored connected-prefix, giving

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, K_n) \le V_{\mathrm{conn}}(\mathcal{H}_\gamma^{(q)}; K_n) \le V_{\mathrm{anch}}(\mathcal{H}_\gamma^{(q)}; K_n) \le 2t+1.$$

For the matching lower bound, put $d = 2t+1 \le n$. Against an arbitrary free-order learner, answer the first d queries using only labels 1 and 2: answer 2 when the prediction is 1, and answer 1 otherwise. Every answer is wrong, including when the learner predicts outside $\{1,2\}$. Because d is odd, one response label occurs at most t times. Complete every unqueried vertex with the other label. If the minority class has size $s \le t$, the resulting binary target has cut cost

$$s(n-s) \le t(n-t) \le \gamma.$$

Thus even unrestricted play incurs at least $2t+1$ mistakes, proving equality in all three games.

Finally, suppose that $t = \lfloor n/2 \rfloor$. Every binary labeling cuts at most $t(n-t) \le \gamma$ edges, so the feasible class contains the full Boolean n-cube. It forces n mistakes even in the free-order game, while no policy can make more than n mistakes. Hence all three values equal $n = \min\{n, 2t+1\}$. □

Complete graphs are the dense counterpart to trees. Despite their very different geometry, both exact values are independent of palette size, whereas cycles provide the first natural palette-sensitive benchmark.

Lemma 2.13. *For every $r \ge 3$ and $q \ge 3$, against any prediction policy on C_r an adaptive adversary can contradict every prediction and complete the responses to a q-label labeling with a monochromatic edge; equivalently, with at most $r-1$ bichromatic edges.*

Proof. Give the first queried vertex a label a different from the prediction. At the second queried vertex, copy a if it is adjacent to the first vertex and a was not predicted; this immediately creates a monochromatic edge. Otherwise choose a label b different from both a and the prediction.

If no monochromatic edge has yet been created, choose an arc P between the two queried vertices with at least one internal vertex. Such an arc exists even when the queried vertices are adjacent. Its endpoints have distinct labels and all its internal vertices are unqueried. We maintain these properties until a monochromatic edge is produced.

A query that does not lie on P can be answered arbitrarily but incorrectly and will leave P unchanged. Now suppose a vertex of P is queried. If P has only one internal vertex, assign that vertex one of the two endpoint labels a or b different from the prediction; this creates a monochromatic edge.

Otherwise, if the queried vertex is adjacent to an endpoint of P, copy that endpoint's label unless it was predicted. If it was predicted, use a third label distinct from both endpoint labels and retain the subpath from the queried vertex to the opposite endpoint. In either case, either a monochromatic edge is created or the retained path has distinct labeled endpoints and fewer internal vertices.

Finally, suppose the queried vertex is an internal vertex of P and is nonadjacent to either endpoint. Answer with any label different from the prediction. The query splits P into two subpaths. Since the original endpoint labels are distinct, the new label cannot equal both of them; hence at least one subpath has endpoints with distinct labels. Retain such a subpath.

Thus whenever a vertex of the retained path is queried without producing a monochromatic edge, the number of its unqueried internal vertices strictly decreases. Since every vertex is eventually queried, a monochromatic edge must be produced. Thereafter answer every remaining query incorrectly; the already monochromatic edge persists. □

Theorem 2.14. *Let C_n be the unweighted cycle on $n \geq 3$ vertices, let $\gamma \geq 0$ be an integer, and let $q \geq 2$. Then all three protocols have the common value*

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, C_n) = \begin{cases} \min\left\{n, 2\min\left\{\left\lfloor \frac{\gamma}{2} \right\rfloor, \left\lfloor \frac{n}{2} \right\rfloor\right\} + 1\right\}, & q = 2, \\ 1, & q \geq 3 \text{ and } \gamma \leq 1, \\ \min\{n, \gamma + 1\}, & q \geq 3 \text{ and } \gamma \geq 2. \end{cases}$$

Proof. A binary labeling of a cycle has an even number of bichromatic edges, and every even count through $2\lfloor n/2 \rfloor$ occurs. With at least three labels, the attainable counts are

$$\{0, 2, 3, \ldots, n\}.$$

Indeed, for $2 \leq c \leq n$, partition the cycle into c nonempty consecutive runs and assign labels to the runs so that adjacent runs receive different labels; three labels suffice for $c \geq 3$, while $c = 2$ is immediate.

If $q \geq 3$ and $\gamma \leq 1$, the attainable-count description shows that every feasible labeling is constant. Query in cyclic order, predict arbitrarily on the first round, and thereafter copy the predecessor's revealed label. This neighbor-anchored connected-prefix policy makes at most one mistake. In every other case, query in cyclic order using a neighbor-anchored connected-prefix policy. Apart from the first query, a mistake can occur only when the traversed edge is bichromatic. The policy therefore makes at most $\min\{n, \gamma + 1\}$ mistakes. For $q = 2$, the number of bichromatic cycle edges is even and at most $2\lfloor n/2 \rfloor$, so the same traversal makes at most

$$\min\{1 + 2\min\{\lfloor \gamma/2 \rfloor, \lfloor n/2 \rfloor\}, n\}.$$

For the lower bound, first suppose $q \geq 3$. If $\gamma \leq 1$, a constant target forces one mistake. If $2 \leq \gamma < n$, partition the cycle into $\gamma + 1$ nonempty consecutive blocks and assign each block its label when the learner first queries a vertex in it. Apply Lemma 2.13 to the quotient cycle $C_{\gamma+1}$. The first query in every block is answered incorrectly, while the resulting monochromatic quotient edge ensures that at most γ boundaries between blocks are bichromatic. Thus the completed target is feasible and forces $\gamma + 1$ mistakes. If $\gamma \geq n$, every labeling is feasible and the realizable adversary can force n mistakes.

Now suppose $q = 2$, and put

$$c := 2\min\left\{\left\lfloor \frac{\gamma}{2} \right\rfloor, \left\lfloor \frac{n}{2} \right\rfloor\right\} \leq \gamma.$$

If $c = 0$, a constant target gives the lower bound, while if $c = n$ the full Boolean n-cube is feasible and forces n mistakes. Otherwise partition the cycle into $c + 1$ nonempty consecutive blocks. The realizable adversary can contradict the first prediction in each block. Every binary labeling of the odd quotient cycle has a monochromatic edge, so at most c boundaries between blocks are bichromatic. The resulting target is therefore feasible and forces $c + 1$ mistakes. □

A blockwise traversal is also exactly optimal on a natural class extending trees by allowing triangular blocks.

Theorem 2.15. *Let G be a connected unweighted cactus in which every cycle is a triangle. Let b_2 and b_3 be the numbers of bridges and triangles, respectively. For every real budget $\gamma \geq 0$ and every palette size $q \geq 2$, all three protocols have the common value*

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, G) = 1 + \max\{i + 2j : i, j \in \mathbb{Z}_{\geq 0},\ i \leq b_2,\ j \leq b_3,\ i + 2j \leq \gamma\}.$$

Moreover, one palette-blind neighbor-anchored connected-prefix policy is optimal for every $q \geq 2$ and every γ.

Proof. Root G at r and orient its block structure away from r. Every bridge or triangle then meets the part already closer to the root in a unique vertex, which we call its attachment vertex.

Consider the following traversal. Query r first and then process the blocks outward from r. At a bridge, query its new vertex and predict the observed label of its attachment. At a triangle, query its two new vertices, predicting the attachment label at both. Process a block only after its attachment has been queried. This policy is palette-blind, neighbor-anchored connected-prefix.

Fix a feasible target. Let i be the number of bridges whose new vertex differs from its attachment, and let j be the number of triangles in which at least one new vertex differs from the attachment. The traversal makes at most

$$1 + i + 2j$$

mistakes: at most one initially, one in each of the i bridges, and two in each of the j triangles. Each such bridge contributes one bichromatic edge. Each such triangle is nonconstant and therefore has at least two bichromatic edges. Since distinct blocks have disjoint edge sets,

$$i + 2j \leq C_G(h) \leq \gamma.$$

Thus

$$M_{sd}(\mathcal{H}_\gamma^{(q)}, G) \leq 1 + \max\{i + 2j : i, j \in \mathbb{Z}_{\geq 0},\ i \leq b_2,\ j \leq b_3,\ i + 2j \leq \gamma\}.$$

For the reverse inequality, choose i bridges and j triangles attaining the displayed maximum. We construct a Boolean cube of dimension $1 + i + 2j$ contained in $\mathcal{H}_\gamma^{(q)}$, using only two labels.

Choose the label of r freely between the two labels and propagate labels outward through the rooted cactus. Across every unchosen bridge, copy the attachment label to its new vertex; across every unchosen triangle, copy the attachment label to both new vertices. At each of the i chosen bridges, let the new vertex independently choose either binary label. At each of the j chosen triangles, let each of its two new vertices make this choice independently. Assign each vertex to the most recent independently labeled vertex on its outward block-tree route from r; in a selected triangle, the two new vertices begin two separate regions. These $1 + i + 2j$ regions are nonempty, pairwise disjoint, and cover V, and every vertex in a region copies its root coordinate. Hence a query reveals only that region's bit, while every binary assignment to the region bits is realized.

These choices give $1 + i + 2j$ independent Boolean coordinates. An unchosen block contributes no bichromatic edge. A chosen bridge contributes at most one, and a chosen triangle contributes at most two. Hence every labeling in the cube satisfies

$$C_G(h) \le i + 2j \le \gamma.$$

The feasible class therefore contains a full Boolean $(1 + i + 2j)$-cube, which forces at least $1 + i + 2j$ mistakes even for the unrestricted learner.

Since the same traversal was used independently of q and γ, it is simultaneously optimal for every palette and budget, and gives the same mistake bound for all three self-directed learning protocols. □

3 Palette saturation

The palette affects both the feasible target class and the shape of the game tree. Nevertheless, a target that can use only a bounded number of distinct labels cannot distinguish indefinitely many label names. We first turn this observation into a graph-independent lifting theorem, then identify the graph parameter that controls it for bounded-cut classes. Later constructions show that this eventual guarantee need not be the first point of stabilization.

For a labeling h of a finite set X, its *equality partition* is the partition of X into the nonempty fibers $h^{-1}(a)$. If Π is a family of partitions of X, let

$$K_q(\Pi) = \{h \in [q]^X : \text{the equality partition of } h \text{ lies in } \Pi\}.$$

An *equality-pattern policy schema* is a deterministic rule whose abstract state records the ordered queried vertices, the partition of revealed responses into blocks ordered by first appearance, and, for each earlier prediction, which response block then present it named, or that it requested a fresh name, together with whether that prediction was correct. An unsuccessful fresh prediction is not given a persistent name. The next abstract action chooses an unqueried vertex and predicts either a current response block or a *fresh* one. On $[q]$, a block action uses that block's concrete label, while a fresh action uses the first label not yet seen as a response in a fixed canonical order; it is allowed only while such a label exists. At the empty history this chooses a fixed canonical first name. The abstract rule is invariant under simultaneous renaming, although literal equivariance of a deterministic concrete prediction at the empty history is impossible. A schema is *conservative* if, after the first round, it predicts only a label that has appeared as a response, and *palette-blind* if its abstract rule does not depend on the palette size. The optimal stabilization schemas constructed below have both properties.

The next theorem strengthens ordinary palette stabilization by preserving both local access protocols.

Theorem 3.1. *Let G be a connected graph on X, and let Π be a nonempty family of partitions of X, each with at most r blocks, where $r \geq 2$. Let $s > r$.*

For every r-label free-order or connected-prefix policy A for $K_r(\Pi)$, there is one canonical policy schema $\widehat{A}$, runnable over every palette at least r, which obeys the same access restriction and satisfies

$$\max_{h \in K_r(\Pi)} M(\widehat{A}, h) \leq \max_{h \in K_r(\Pi)} M(A, h), \tag{5}$$

$$\max_{h \in K_s(\Pi)} M(\widehat{A}, h) \leq \max_{h \in K_r(\Pi)} M(A, h) + 1. \tag{6}$$

Thus, for either protocol, the two corresponding minimax values differ by at most one. If the difference is one, a canonical schema is optimal at both palettes. If the values are equal, every upper-palette optimum has optimal loss on the nested lower-palette subclass, and its downward legalization is a legal lower-palette optimum.

If A is conservative, the additive one in (6) disappears. In particular, neighbor-anchored connected-prefix values satisfy

$$V_{\mathrm{anch}}(K_s(\Pi); G) = V_{\mathrm{anch}}(K_r(\Pi); G) \qquad (s \geq r), \tag{7}$$

and one neighbor-anchored schema is optimal for every palette in this tail.

Proof. Simulate A while maintaining an injection from physical labels already observed to virtual labels in $[r]$. Feed the virtual history to A and make its query. If the virtual prediction already has a physical representative, predict that representative. Otherwise predict a canonically chosen unused physical name, choosing one in $[r]$ whenever possible.

An old physical response is translated through the current injection. A fresh response equal to the physical prediction receives the predicted virtual name. For a different fresh response, assign the canonically first unused virtual name different from the virtual prediction, whenever one exists. This includes the case of only one unused virtual name when the prediction is an old name. The physical and virtual mistake indicators then agree. There is only one exceptional configuration: the virtual prediction is the sole unused virtual name and a different fresh physical response arrives. It must be assigned the predicted virtual name, producing one physical mistake without a virtual mistake. This cannot recur, because all r virtual names have then appeared and a target partition has at most r blocks. On a genuine r-label target it cannot occur at all: after $r - 1$ physical labels have appeared, exactly one physical name remains unused, and the simulation predicts that name. The injection extends to every completion because its equality partition has at most r blocks; relabeling preserves membership in Π. Thus every simulated history is realizable in $K_r(\Pi)$, and the loss guarantee of A applies. This proves (5) and (6). The virtual history and injection are reconstructed from the abstract response blocks and prediction outcomes, so the construction is indeed one canonical schema.

The simulation makes exactly the queries chosen by A, so connected-prefix legality is preserved. If A is conservative, its noninitial prediction is already represented, while at the initial round at least two virtual names are unused. The exceptional case is therefore impossible and losses agree on every path. Under neighbor anchoring, the physical representative was observed at the same queried neighbor as the virtual prediction, so anchoring is preserved.

For free or connected-prefix play, class nesting and the lift give $v_r \leq v_s \leq v_r + 1$ for the two minimax values. If $v_s = v_r + 1$, the lift of a lower-palette optimum supplies legal optimal implementations at both palettes. If $v_s = v_r$, let B be an upper-palette optimum. Its loss on the nested lower class is at most v_r. The downward legalization in (2) applies verbatim to $K_r(\Pi)$: it leaves the query sequence unchanged, and in neighbor-anchored play only the unrestricted initial

prediction can be out of range. It therefore gives a legal lower-palette implementation with no larger loss. Lower-palette optimality forces equality. Thus the upper policy has optimal loss on the nested lower-palette subclass, and its downward legalization is a legal lower-palette optimum. Every neighbor-anchored policy is conservative after the first round, so the conservative lift of an optimal anchored policy, together with palette monotonicity, proves (7). □

Recall that $m_q = m_q(G, \gamma)$ for the fixed graph and budget. The relevant bound on label complexity is the *budgeted fragmentation number*

$$\beta_\gamma(G) := \max\{c(G - F) : F \subseteq E(G),\ w(F) \le \gamma\},$$

so $\beta_\gamma(G)$ is the largest number of components obtainable by deleting edges of total weight at most γ.

Theorem 3.2. *Let G be connected with positive edge weights, and put $b = \beta_\gamma(G)$. A feasible q-label target uses at most $\min\{q, b\}$ labels, and this bound is attained. Moreover, $m_q \ge b$ for every $q \ge 2$.*

The free-order and connected-prefix values are independent of q once $q \ge b + 1$; for each protocol, one conservative equality-pattern, palette-blind schema is optimal throughout this tail. The neighbor-anchored value is already independent of q for $q \ge \max\{2, b\}$, again with one palette-blind optimal schema.

At the boundary, whenever $s > r \ge \max\{2, b\}$, each free-order or connected-prefix value gap lies in $\{0, 1\}$. If the gap is one, a canonical schema is optimal at both palettes. If the values are equal, an upper-palette optimum is simultaneously minimax in the nested-class sense, and its downward legalization is a legal lower-palette optimum. The corresponding neighbor-anchored values are equal.

Proof. For a feasible labeling h, delete the set F_h of edges whose endpoints have different labels. Every component of $G - F_h$ has constant label, so h uses at most $c(G - F_h) \le b$ labels. Conversely, if F attains b, assign labels to the components of $G - F$ so as to use exactly $\min\{q, b\}$ names. Every edge whose endpoints receive different labels then belongs to F, proving the first claim.

Assign an independent binary label to each of the b components of $G - F$. The resulting targets form a Boolean b-cube. The adversary can force a mistake on the first queried vertex of each component, so this cube has value b and proves $m_q \ge b$.

It remains to justify stabilization by backward induction on the number of unqueried vertices, starting with terminal value zero. At any nonempty history, suppose ℓ labels have appeared. If no continuation using a fresh response is feasible, a fresh prediction is certainly wrong on every feasible child and is dominated by predicting any observed label. Otherwise $\ell \le b - 1$. Because $q \ge b + 1$, at least two unused names remain, and all fresh-response children are isomorphic under permutations fixing the old names. Against a fresh prediction, the adversary may therefore return a different fresh label, force a mistake, and reach the generic fresh child. Predicting an observed label has the same cost on that child and weakly improves its matching old-label child, so a fresh prediction is again dominated. Thus the Bellman recurrence depends only on the old-label children and one generic fresh child, not on q. The induction hypothesis makes their continuation values palette-independent. The legal connected-prefix query set depends only on Q, so the same induction applies there. At the empty history, for each first queried vertex all response children are isomorphic, and $q \ge 2$ lets the adversary make the first prediction wrong. Minimizing one plus the common child value over the first vertex therefore also gives a palette-independent value. Fixed tie-breaking after ordering observed blocks by first appearance yields a conservative equality-pattern, palette-blind optimum. Let Π_γ be the family of equality partitions whose crossing edges have total weight at most γ. Then $\mathcal{H}_\gamma^{(q)}(G) = K_q(\Pi_\gamma)$ for every q, and the first part of the proof shows that every member of Π_γ has

at most b blocks. Theorem 3.1 therefore gives the anchored threshold one palette earlier and all the adaptation assertions. □

For clarity, write $V^{(q)}_{\mathrm{conn}} = V_{\mathrm{conn}}(\mathcal{H}^{(q)}_{\gamma}(G); G)$ and define $V^{(q)}_{\mathrm{anch}}$ analogously. Define the first stabilization palettes

$$q_0^{\mathrm{free}}(G,\gamma) = \min\{q \geq 2 : m_s = m_q \text{ for every } s \geq q\},$$
$$q_0^{\mathrm{conn}}(G,\gamma) = \min\{q \geq 2 : V^{(s)}_{\mathrm{conn}} = V^{(q)}_{\mathrm{conn}} \text{ for every } s \geq q\},$$
$$q_0^{\mathrm{anch}}(G,\gamma) = \min\{q \geq 2 : V^{(s)}_{\mathrm{anch}} = V^{(q)}_{\mathrm{anch}} \text{ for every } s \geq q\}.$$

Theorem 3.2 gives

$$q_0^{\mathrm{free}}(G,\gamma), q_0^{\mathrm{conn}}(G,\gamma) \leq \beta_\gamma(G) + 1, \qquad q_0^{\mathrm{anch}}(G,\gamma) \leq \max\{2, \beta_\gamma(G)\}.$$

When there is no ambiguity, q_0 below denotes q_0^{free}.

The fragmentation bound is an eventual guarantee, not the exact first threshold. Before constructing every possible threshold, we record two operations that let weighted examples be turned into ordinary unweighted instances.

3.1 Two construction operations

The next two results let us transfer constructions to the stated graph model. The first removes integer edge weights; the second removes the temporary assumption that designated role vertices already have prescribed distinct labels.

Proposition 3.3 (Realization by a simple unweighted graph)**.** *Let J be a finite connected graph with positive integer edge weights and nonnegative integer budget γ. Aggregate parallel edges, let $W = \max_{e \in E(J)} w_e$ when $E(J) \neq \varnothing$ and let $W = 0$ otherwise, and choose an integer N with*

$$N \geq \gamma + 2, \qquad N^2 \geq W.$$

Replace each vertex of J by a disjoint K_N and realize a weight-w_{uv} edge by w_{uv} distinct edges between the two corresponding cliques. The condition $N^2 \geq W$ ensures that these edges can be chosen distinctly. Denote the resulting graph by $\widetilde{J}$. It is finite, connected, simple, and unweighted. Clique contraction identifies its feasible target class exactly with that of J for every palette. In particular, $m_q(J,\gamma) = m_q(\widetilde{J},\gamma)$, and the corresponding connected-prefix and neighbor-anchored values, β_γ, and every simultaneous policy-loss cap within each protocol are also preserved.

Proof. A nonconstant K_N cuts at least $N - 1 > \gamma$ internal edges, so every feasible target is constant on each replacement clique. A clique-constant target has exactly the weighted cost of its contraction.

A quotient policy lifts by using its action at the first queried vertex of a replacement clique and then filling the clique along a spanning tree while copying the revealed label, adding no mistake. After the initial clique, a connected-prefix lift enters each new clique at an endpoint of a representative edge from any already queried quotient neighbor. Under neighbor anchoring, it uses the representative edge from the particular neighbor supplying the predicted label. The same filling procedure then preserves the respective access condition. Conversely, simulate a policy on the expanded graph, issue a quotient query only on the first visit to a clique, and feed the forced response on later simulated visits. Suppressing those later visits cannot add mistakes. A first visit to a new clique contracts to a vertex adjacent to an already visited quotient vertex; if its prediction is anchored, the anchoring neighbor lies in such an adjacent clique and carries the same quotient

label. These transformations preserve all three protocol values and are compatible across palettes, so they also preserve simultaneous loss caps.

Finally, deleting at most $\gamma < N - 1$ internal edges cannot disconnect a replacement clique. Any deletion set of size at most γ therefore induces a quotient deletion: delete uv exactly when all w_{uv} edges between its two cliques were deleted. The quotient deletion has no greater weight and gives the same number of components. Conversely, deleting all representative edges of any quotient edge set has exactly its weighted cost and realizes the same components after contraction. Hence β_γ is preserved. □

Remark. Proposition 3.3 is an exact realizable reduction for positive integer weights. Positive rational weights, together with the budget, can first be multiplied by a common denominator. This argument does not give an exact finite realization of arbitrary real weights. Zero-weight edges may be discarded for the free-order game, but not automatically for a local-access game, because even a zero-weight edge can change the legal frontier or supply an anchor. The replacement may also be large and need not preserve structural parameters such as treewidth.

Most importantly, clique realization does not preserve agnostic regret. A realizable target is forced to be constant on each replacement clique, but an arbitrary agnostic response word may split such a clique. We therefore use the proposition only for the stated realizable values, access protocols, fragmentation numbers, and policy-loss comparisons; no unweighted agnostic hardness consequence is inferred from it.

The next lemma removes prescribed distinct label roles. It is stated for an arbitrary core so it can be reused in both the saturation and transfer constructions.

Lemma 3.4 (Releasing prescribed label roles)**.** *Let a core with positive integer edge weights and nonnegative integer budget Γ have live vertices X and role vertices $a_1, \dots, a_R$, where $3 \le R \le r < s$. In the conditioned free-order game, the roles are fixed boundary data carrying prescribed distinct labels and are not queried; only vertices of X are queried. Suppose the two conditioned target classes on X are nonempty and have free-order values b and c at palettes r and s.*

There is a positive-integer-weighted unconditioned graph with nonnegative integer budget whose two free-order values are $b + M$ and $c + M$ for one common integer M. More precisely, if an s-optimal fixed-role free-order policy has worst-case loss exactly ℓ on the conditioned r-label class, then its released free-order implementation has exact losses $\ell + M$ and $c + M$. Moreover, if no fixed-role free-order policy has losses at most (u, v) on the two classes, then no released free-order policy has losses at most $(u + M, v + M)$. The wrapper depends on the core and b, but not on s, so it preserves an entire stabilized upper-palette tail.

Proof. The wrapper has two kinds of feasible branch. When the roles are distinct, its midpoints form an independent Boolean cube and add a fixed loss. When two roles coincide, the same heavy edges make the branch strictly cheaper.

Put $N = |X|$, $d = 4$,

$$T = 4 + 2(N + R - b + 1), \qquad M = (d + T)\binom{R}{2},$$

and choose an integer $L > \Gamma$. Between each pair of roles add $d + T$ internally disjoint two-edge paths, giving both edges of every path weight L, and raise the budget to $\Gamma + ML$.

First condition on distinct role labels. Every midpoint must copy one of its endpoints: doing so costs L, whereas using a third label costs $2L$ and would exceed the budget because $L > \Gamma$. The conditioned class is therefore the product of the core with a Boolean M-cube. Lemma 2.2 gives values $b + M$ and $c + M$.

For the matching upper bound, use d paths per role pair to discover and verify the endpoint labels and reserve the other T paths as selectors. The scan in Appendix A.1 recovers and verifies all distinct roles, up to one palette permutation, within the allowance $d\binom{R}{2}$; the reserves then bring the total loss to at most M. After discovery, extend the role correspondence to a palette permutation and run the corresponding fixed-role core policy. On an r-label target, choose the extension to preserve $[r]$, so its lower-palette guarantee continues to apply. If verification instead finds $h \geq 1$ equal-label role pairs, let e be the number of reserve midpoints on those pairs that differ from the common endpoint label. The unequal pairs and these exceptions give

$$(M - (d+T)h + 2e)L \leq ML + \Gamma.$$

Since $\Gamma < L$, integrality gives $e \leq (d+T)h/2$. Even charging every other role, live vertex, and midpoint query as a mistake, the branch costs

$$M + N + R - Th + e \leq M + N + R - \frac{(T-d)h}{2} \leq M + b - 1.$$

Thus collision branches are cheaper than the distinct-role branch; note that $c \geq b$ by palette monotonicity. This proves both scalar values.

The same argument preserves policy losses. If an s-optimal fixed-role policy has lower-palette loss ℓ, run it after successful role discovery. Its two losses are at most $\ell + M$ and $c + M$, while a collision branch costs less than $M + \ell$ because $\ell \geq b$. On a canonical distinct-role assignment, the Boolean selector cube can force all M midpoint predictions to be wrong before a core target forces respectively ℓ or c further mistakes, so both bounds are exact. The same canonical assignment proves the cap statement: answer every midpoint query with an endpoint label different from the prediction, forcing M jointly feasible mistakes, and suppress role and midpoint queries. A released policy with caps $(u+M, v+M)$ would induce a fixed-role core policy with caps (u, v). The constants use the core and b but not s, so the wrapper preserves an entire stabilized upper tail. □

3.2 Every stabilization threshold occurs

Theorem 3.5. *For every integer $k \geq 2$, one can choose a connected simple unweighted graph G and a nonnegative integer budget γ such that*

$$q_0(G, \gamma) = k.$$

For every $k \geq 3$, the example can be chosen with an integer B such that

$$m_{k-1} = B, \qquad m_s = B + 1 \quad (s \geq k).$$

Proof. The cases $k = 2$ and $k = 3$ are immediate. A nontrivial tree at budget zero has value one for every palette. For $k = 3$, take $(G, \gamma) = (C_4, 3)$; Theorem 2.14 gives

$$m_2 = 3, \qquad m_s = 4 \quad (s \geq 3).$$

Now let $k \geq 4$ and set $r = k - 1$. We first build a weighted core with prescribed distinct role labels $1, \ldots, r$ at vertices $a_1, \ldots, a_r$. Distinguish an index $p \in [r]$, hence the role vertex a_p, add a center x, and add blocks $B_1, \ldots, B_r$ with

$$|B_p| = 1, \qquad |B_i| = 3 \ (i \neq p), \qquad n = 3r - 2.$$

For each $v \in B_i$, give xv weight $n+1$ and va_i weight n, and set $\Gamma = n(n+1)-1$. In this conditioned core, x and the block vertices are live queried vertices; the role vertices are fixed boundary conditions and are not queried until Lemma 3.4 removes the conditioning.

Here is the proof mechanism before the accounting. We compute every section obtained by revealing the center x: one distinguished old-label section and every fresh-label section have value $n-1$, whereas the other old-label sections have value $n-2$. Consequently, the palette with only the r role labels has value $n-1$, while the availability of any fresh label raises the value once, to n, and no later fresh label changes it. A two-section induction supplies the only difficult lower bound. Finally, role release and clique realization convert this conditioned weighted mechanism into the claimed ordinary unweighted instance without changing the first threshold.

We first compute the section obtained by fixing the label of x. Suppose $x = i$ is an old label and write $s_i = |B_i|$. Relative to the assignment that labels every block vertex i, let b_0 count vertices outside B_i that retain their role label, let c_0 count those using neither i nor their role label, and let d count non-i vertices in B_i. Feasibility is exactly

$$b_0 + (n+1)c_0 + (2n+1)d \le n(s_i+1) - 1.$$

Because $s_i \in \{1,3\}$, the inequality gives $d \le \lfloor s_i/2 \rfloor$; also $b_0 + c_0 \le n - s_i$. Predicting i on every remaining vertex therefore makes at most $n - \lceil s_i/2 \rceil$ mistakes. Equality follows from a Boolean cube: every vertex outside B_i independently chooses between i and its role label, and $\lfloor s_i/2 \rfloor$ chosen vertices of B_i independently choose between i and one fixed different old label. Hence the p-section has value $n-1$, and every other old-label section has value $n-2$.

Suppose instead that $x = f$ for a fresh label f. Let t be the number of block vertices labeled f, and let u be the number using neither f nor their role label. Their cut cost is

$$n(n+1) - t + nu.$$

Since $t \le n - u$, feasibility is equivalent to $u = 0$ and $t \ge 1$. Thus this section is a punctured Boolean n-cube and has value $n-1$. Querying x and predicting p now shows that the r-label core has value $n-1$.

With at least one fresh label, querying x gives the upper bound n. For the reverse inequality, fix a fresh label f and consider any first action. At x, one can return a wrong label among p and f and leave a section of value $n-1$. At a block vertex, any prediction other than f can be answered by f; fixing $x = f$ then leaves a Boolean cube on the other $n-1$ block vertices. If f is predicted at the unique vertex $w \in B_p$, answer p; fixing $x = p$ again leaves a Boolean $(n-1)$-cube.

It remains to consider a prediction f at $v \in B_i$ with $i \ne p$. Answer with the role label i and put

$$Z = \left(\bigcup_{j \ne p} B_j \right) \setminus \{v\}.$$

For $S \subseteq Z$, define two residual subclasses. In L_S, set $x = w = p$, fix $S \cup \{v\}$ to their role labels, and let each vertex of $Z \setminus S$ choose independently between p and its role label. In U_S, set $x = f$, fix $S \cup \{v\}$ to their role labels, and let w and the vertices of $Z \setminus S$ choose between f and their role labels, excluding the all-role assignment. The preceding two section calculations show that both subclasses are feasible. If $d = |Z \setminus S|$, then

$$v(L_S \cup U_S) \ge d + 1. \tag{8}$$

For $d = 0$, the two surviving assignments are $(x, w) = (p, p)$ and (f, f). For $d > 0$, every first action has a wrong response leaving either a Boolean or punctured Boolean class of value d, or the smaller

union $L_{S\cup\{z\}} \cup U_{S\cup\{z\}}$. This proves (8) by induction; Appendix A.2 gives the response-by-response verification.

Initially $d = n-2$, so the response at v leaves value $n-1$. Every possible first action therefore has a wrong-response child of value at least $n-1$, and the upper-palette core has value n for every palette at least $r+1$.

Lemma 3.4 now removes the prescribed role labels, adding one common constant M at every relevant palette. The released weighted graph has values $M+n-1$ at palette r and $M+n$ at every palette at least $r+1$. Proposition 3.3 converts it to a connected simple unweighted graph without changing this sequence. Its strict jump $m_r < m_{r+1}$ rules out every threshold at most r, while the constant tail from $r+1$ onward gives $q_0 = r+1 = k$. □

3.3 A weighted treewidth scale

The preceding unweighted examples can have large treewidth after unweighting. Retaining nonuniform integer weights reveals a secondary structural phenomenon: every treewidth $w \geq 2$ supports an example whose first stabilization palette is $w+2$.

A *tree decomposition* of G is a tree-indexed family of bags whose union is $V(G)$, such that every graph edge lies in some bag and the bags containing any fixed graph vertex form a connected subtree. Its width is one less than its largest bag size, and $\mathrm{tw}(G)$ is the least possible width.

Theorem 3.6. *For every integer $w \geq 2$, one can choose a connected simple graph G_w with positive integer edge weights and an integer budget γ_w such that*

$$\mathrm{tw}(G_w) = w, \qquad q_0(G_w, \gamma_w) = w+2.$$

The construction is weighted: its edge weights are not all equal.

Proof. First let $r = w \geq 3$. Before releasing the roles, prescribe distinct labels on $a_1, \ldots, a_r$ and distinguish an index $p \in [r]$, hence the role vertex a_p. Take $n = 8r-3$ leaves: five in the p-block and eight in every other role block. Join each leaf to a center x by an edge of weight two and to its role by an edge of weight one, and give this core budget $2n-1$. Here the center and leaves are the live queried vertices, while the role vertices are fixed boundary conditions until the release wrapper is applied.

The fixed-role core has values $n-2, n-1, n$ at palettes $r, r+1$, and $s \geq r+2$, respectively. The mechanism is short. If x receives an old label i, its role block has size s_i, and Q leaves disagree with x, then every such target has cost at least

$$n - s_i + Q + 2\big(Q - (n-s_i)\big)_+,$$

where $(z)_+ = \max\{z, 0\}$. The appendix uses this inequality to give the needed upper bounds for every old-center section. For the matching lower bound, give x the label prescribed at a_p, vary the $n-5$ outside leaves independently between p and their role labels, and let the five remaining leaves range over a Boolean Hamming ball of radius three. This subclass has value $(n-5)+3 = n-2$, so the palette-r core value is $n-2$.

One fresh center label exposes a punctured Boolean n-cube of value $n-1$. Together with the old-center bounds, querying x and predicting that fresh label proves that the palette-$(r+1)$ value is $n-1$.

With two fresh labels, a wrong fresh response at every possible first query leaves residual value $n-1$. Conversely, query x and then copy its revealed label on every leaf. More than n total mistakes

would require all n center edges to cross, at cost $2n > 2n - 1$. Thus the value is n. Appendix A.3 supplies the remaining section substitutions and verifies the stated wrong-response subclasses.

Release the roles with Lemma 3.4. The wrapper contributes

$$M = (2r + 16)\binom{r}{2}$$

selector coordinates and adds M to every value, so

$$m_r = M + n - 2, \qquad m_{r+1} = M + n - 1, \qquad m_s = M + n \quad (s \geq r + 2).$$

For $w = 2$, the analogous blocks of sizes five and eight give a 13-leaf core with values $11, 12, 13$. The two-role release gadget in Appendix A.3 adds sixteen selectors and yields

$$m_2 = m_3 = 28, \qquad m_s = 29 \quad (s \geq 4).$$

It remains to certify the width. For $r \geq 3$, one bag $\{x, a_1, \ldots, a_r\}$, the bags $\{x, a_i, v\}$ for block leaves v, and the bags $\{a_i, a_j, z\}$ for selector midpoints z form a tree decomposition of width r when every smaller bag is attached to the central one. Conversely, one leaf from each block gives internally disjoint x–a_i paths, and one selector path for each role pair gives a subdivision of K_{r+1}. Hence the treewidth is r. For $w = 2$, every auxiliary midpoint in the special wrapper is adjacent only to a_1 and a_2, so it fits a bag $\{a_1, a_2, z\}$. The same decomposition therefore has width two, while two internally disjoint a_1–a_2 selector paths form a cycle. Hence its treewidth is exactly two. The displayed final strict increase and ensuing plateau give $q_0 = w + 2$. □

Remark. Theorem 3.6 is an existence theorem for weighted graphs, not a general treewidth bound. Proposition 3.3 preserves the target class and minimax values, but replaces vertices by large cliques of size N and can therefore increase treewidth far beyond the original width. Thus the construction neither proves nor refutes the open unweighted inequality

$$q_0(G, \gamma) \leq \mathrm{tw}(G) + 2.$$

3.4 Hardness of the mistake bound

The preceding certificates and palette-blind policies are explicit, but the unrestricted minimax value itself is hard to evaluate in general. This is already true with two labels.

Theorem 3.7. *For connected simple unweighted graphs with a nonnegative integer budget, deciding whether*

$$m_2(G, \gamma) \leq |V(G)| - 1$$

is NP-complete. Its complementary decision $m_2(G, \gamma) = |V(G)|$ is coNP-complete, and exact evaluation is NP-hard. For connected simple graphs with positive integer edge weights and N vertices, under the promise $m_2 \in \{N - 1, N\}$, deciding the lower value is NP-complete and deciding the upper value is coNP-complete.

Proof. A nonempty class $\mathcal{C} \subseteq \{0, 1\}^n$ has minimax value n if and only if it is the Boolean cube. The Boolean cube permits the adversary to answer every prediction incorrectly. Conversely, if $z \notin \mathcal{C}$, predict $1 - z(x)$ whenever coordinate x is queried. The unique history on which every prediction is wrong is z, so every target in $\mathcal{C}$ gives at least one correct prediction.

Now let G have n vertices. Constant labelings are feasible, so its bounded-cut class is nonempty. This class is the Boolean cube exactly when every cut has size at most γ. Therefore

$$m_2(G,\gamma) \le n-1 \quad \Longleftrightarrow \quad \mathrm{MaxCut}(G) \ge \gamma+1.$$

A cut is an NP witness. Max-Cut remains NP-hard after imposing connectedness (Garey et al., 1976): connect the source components by a tree of bridges and increase the requested cut size by the number of added bridges. Independently flipping the two sides within each source component makes every bridge cross, so this operation increases the maximum cut by exactly that number. This proves NP-completeness; the complementary and exact-evaluation claims follow.

For the promise strengthening, start with a connected Max-Cut instance (G,K), where $n=|V(G)|$, $m=|E(G)|$, and $1 \le K \le m$. Choose $z \in V(G)$, add a leaf a, give az weight $H=m-K+1$, retain unit weights on G, and use budget m. Put $N=n+1$. The slice $h(a)=h(z)$ contains a Boolean n-cube, so the value is at least $N-1$. If $\mathrm{MaxCut}(G) \le K-1$, every slice with $h(a)\neq h(z)$ is also feasible because its cost is at most $(K-1)+H=m$; the whole class is the Boolean N-cube and has value N. If $\mathrm{MaxCut}(G) \ge K$, extend a cut of size at least K by assigning a the label opposite to that of z. This labeling has cost at least $K+H=m+1$. The class is proper, so the opening observation gives value at most $N-1$, and equality follows from the retained n-cube. Thus the lower value is equivalent to the Max-Cut yes-instance. Under the promise, an over-budget labeling certifies the lower value. The reduction always satisfies the promise, and taking complements gives the coNP-completeness of the upper-value case. □

Fragmentation therefore explains why palette stabilization must eventually occur, while the constructions show that its first threshold can vary arbitrarily. Exact evaluation is nevertheless NP-hard even in the binary case.

4 Policy transfer

Recall that m_q records the optimal scalar loss, whereas the transfer quantities compare the policies attaining that loss.

The equality case in (3) is benign: if $m_r=m_s$, every upper-palette optimum transfers perfectly. We now show that a strict rise, even by one, can change every optimum. Adjacent palettes can force every richer-palette optimum to pay an arbitrarily large full gap; equally optimal policies can realize every transfer loss across an interval; and exact zero-overhead transfer can fail for every nontrivial palette pair. We begin with the strongest adjacent-palette statement.

Theorem 4.1. *For every $r \ge 2$ and every integer $D \ge 1$, there is a connected simple unweighted graph and an integer cut budget such that*

$$m_{r+1}=m_r+D, \qquad \Delta^-_{r\leftarrow r+1}=\Delta^+_{r\leftarrow r+1}=D.$$

Equivalently, every $(r+1)$-optimal policy makes exactly D more worst-case mistakes on r-label targets than an r-optimal policy.

Proof. We first treat $r \ge 3$. Put $t=D$ and

$$W=2t, \qquad a=W+3, \qquad b=2W+4, \qquad n=a+(r-1)b, \qquad N=tn.$$

In the fixed-role core, distinct labels are prescribed on role vertices $A_1,\dots,A_r$, and one role p is distinguished. There are t stars. Star j has center x_j and leaf blocks $B_{j,1},\dots,B_{j,r}$, where $|B_{j,p}|=a$

and $|B_{j,i}| = b$ for $i \neq p$. Each leaf is joined to x_j with weight W and to its role vertex with weight one. The pooled budget is

$$\Gamma = WN - t(W-1).$$

The lower branch R_j fixes $x_j = p$. Leaves outside $B_{j,p}$ are independent Boolean coordinates between p and their role labels, while $B_{j,p}$ ranges over the radius-$(a-2)$ Hamming ball about the all-p labeling, using p and one other old label. The upper branch U_j fixes x_j to the fresh label $f = r+1$, lets every leaf choose between f and its role label, and deletes the all-role labeling. Hence

$$v(R_j) = n-2, \qquad v(U_j) = n-1,$$

and every target in either branch has cut cost at most $Wn-(W-1)$. Consequently

$$\mathcal{P}_r := \prod_{j=1}^{t} R_j, \qquad \mathcal{P}_{r+1} := \prod_{j=1}^{t} (R_j \cup U_j)$$

are feasible under the single pooled budget. The first product and the subproduct $\prod_j U_j \subseteq \mathcal{P}_{r+1}$ therefore give

$$B := N - 2t \leq m_r, \qquad C := N - t = B + t \leq m_{r+1}.$$

The pooled-budget calculation in Appendix B.1 gives matching policies for the entire bounded-cut classes, including targets that distribute cut cost unevenly among the stars. Its lower-palette estimate has an especially simple form: if the policy predicts p at a center and then copies the response, a factor with ℓ_j mistakes and cut cost E_j satisfies

$$E_j \geq (W+1)\ell_j - n + a. \tag{9}$$

For the upper-palette policy, write $d_j = n - \ell_j$ and $\sigma_j = Wn - E_j$. In the range relevant to a putative $(C+1)$st mistake, the calculation gives $\sigma_j \leq (W-1)d_j$ after a fresh center and $\sigma_j \leq (W+1)d_j - 2$ after an old center. Summing these estimates over the factors rules out the extra mistake and proves

$$m_r = B, \qquad m_{r+1} = C.$$

This calculation is essential: the products certify the lower bounds, but a shared cut budget can admit additional targets that a Cartesian-product argument alone does not control.

We now prove the optimizer statement. Fix an $(r+1)$-optimal policy. Along one common history, retain $\mathcal{P}_r$ as a lower-class version space and $\mathcal{P}_{r+1}$ as an upper-class version space. Consider an unresolved star. At its center or any leaf, a prediction different from f can be answered by f with a mistake while preserving the full upper residual value, so upper optimality forces prediction f. At a leaf outside $B_{j,p}$, answer with its role label. This is an *ordinary* mistake and lowers both retained values by one. A leaf of $B_{j,p}$ cannot be queried while the star remains unresolved: after the forced prediction f, response p does not lower the current value of the residual union $R_j \cup U_j$. Appendix B.1 verifies this two-section claim. Thus the mistake plus the residual value exceeds the upper allowance. Thus, apart from further outside-block queries, the only action consistent with upper optimality is to query the center; answer p there and select R_j. This mistake lowers the upper value by one but leaves the lower value unchanged; call it a *conflict*. Afterward the two residual factors coincide. Against each later query, take a tight Bellman response in their common residual class. Upper optimality rules out a branch cost above the current allowance, so a tight response either is correct and preserves the common value or is wrong and lowers it by exactly one; count the latter as ordinary. Because every vertex must eventually be queried, a factor can postpone its

conflict only until its finitely many outside-block leaves have been exhausted. Interleaving the stars does not affect the argument.

Thus, after e ordinary value-lowering mistakes and k conflicts, the retained values are

$$B - e, \qquad C - e - k,$$

while $e + k$ mistakes have occurred. When the tth star conflicts, the lower class forces total loss

$$(e + t) + (B - e) = B + t = C.$$

Every $(r + 1)$-optimal policy therefore has lower-class loss at least C; the scalar transfer sandwich gives equality. Lemma 3.4 adds the same selector cost to both scalar values and to this loss frontier, and Proposition 3.3 converts the construction to a connected simple unweighted graph.

For $r = 2$, use the preceding core and parameters with $t = \tau := D + 1$, and write $B := N - 2\tau$ and $C := N - \tau = B + \tau$ for its two values. The conflict coupling rules out a policy having loss at most $C - 1$ on the fixed-role binary class and at most C on the fixed-role ternary class. The generic role-release lemma is unavailable because there are only two old labels. Appendix B.2 therefore supplies a two-role wrapper with $K = 6\tau + 10$ equal-weight selectors. Distinct roles make the selectors independent Boolean coordinates; the appendix's discovery, orientation, and collision analysis shows that coincident roles are strictly cheaper and proves

$$m_2 = B + K + 1, \qquad m_3 = C + K.$$

It also shows that a ternary optimum with binary loss below $C + K$ would induce the forbidden fixed-role caps $(C - 1, C)$. Hence every ternary optimum has binary loss $C + K$, and

$$m_3 - m_2 = (C + K) - (B + K + 1) = \tau - 1 = D.$$

Proposition 3.3 preserves both scalar values and the two-palette loss frontier. □

The theorem says more than that some poorly chosen upper optimum transfers badly: every upper optimum does. It also explains why ordinary product amplification is insufficient. The construction verifies a global cut-cost inequality for all targets admitted by the pooled budget while retaining literal products only as lower-bound subclasses.

4.1 Transfer spectra among upper-palette optima

The full-gap theorem makes every upper-palette optimum transfer badly. A different family shows that tie-breaking among equally optimal policies can realize every possible transfer quality.

Theorem 4.2. *For all integers $r \geq 3$ and $g \geq 1$, there is a connected simple unweighted graph, together with an integer cut budget and an integer B, such that*

$$m_r = B, \qquad m_{r+1} = B + g, \qquad m_s = B + g + 1 \quad \text{for every } s \geq r + 2.$$

For the adjacent pair,

$$\{L_r(A) : A \in \mathrm{Opt}_{r+1}\} = \{B + 1, B + 2, \ldots, B + g\}. \tag{10}$$

Consequently

$$\Delta^{-}_{r \leftarrow r+1} = 1, \qquad \Delta^{+}_{r \leftarrow r+1} = g.$$

For every $s \geq r+2$,

$$\{L_r(A) : A \in \mathrm{Opt}_s\} = \{B, B+1, \ldots, B+g+1\}, \tag{11}$$

and hence

$$\Delta^{-}_{r\leftarrow s} = 0, \qquad \Delta^{+}_{r\leftarrow s} = g+1.$$

Proof. The fixed-role seed and its exact loss frontier are proved in Proposition C.1. Write $n = r(2g+3)-2$ for its number of leaves, so including the center there are $n+1$ live vertices, and put $b = n-g-1$. Its values at palettes r, $r+1$, and $s \geq r+2$ are respectively

$$b, \qquad n-1 = b+g, \qquad n = b+g+1.$$

At palette $r+1$, predicting the unique fresh label at the center leaves $g-1$ units of slack in the distinguished old-label section, after charging the center mistake. Deliberately spending $j = 0, \ldots, g-1$ of those units before continuing optimally gives exact lower-palette losses $b+1+j$. A moving pair of retained Boolean subcubes rules out loss b, producing the entire interval in (10).

With at least two fresh labels, predicting an old center label gives a simultaneous optimum of lower-palette loss b. Predicting one fresh label instead leaves another fresh response available to certify the upper value n; spending $j = 0, \ldots, g$ units in the distinguished old section gives lower-palette losses $b+1+j$. This fills the interval in (11).

Lemma 3.4 removes the prescribed roles and realizes every listed loss after adding one common selector cost M. In the adjacent case, its cap-preservation clause also excludes lower loss $b+M$; the transfer sandwich excludes values outside the displayed interval. In the separated case, the realized interval already fills the entire sandwich. Taking $B = b+M$ proves both spectra. The same wrapper works throughout the upper-palette tail. Apply Proposition 3.3 once: it preserves the values and simultaneous loss caps, and lifts each constructed policy with its exact losses. Thus this single connected simple unweighted graph is fixed independently of s and supports the entire asserted tail. □

Thus scalar values conceal substantial optimizer-level geometry. For adjacent palettes, every richer-class optimum is incompatible with the lower optimum, yet its added loss can be any integer from one through the full scalar gap. Once two fresh labels are available, this family also contains a perfectly transferring optimum, while other optima still lose the entire gap.

4.2 A construction for nonadjacent palettes

For nonadjacent palettes, two fresh labels destroy the asymmetric mechanism used by the adjacent construction. The following construction instead forces the two palettes to prefer different global role phases.

Lemma 4.3. *For every $r \geq 3$ and $s \geq r+2$, there is a graph with positive integer edge weights and an integer budget, together with r anchor vertices conditioned to have prescribed distinct labels, such that the resulting r- and s-label classes have values $n-1$ and n, respectively, where $n = 7r-2$, and no policy is minimax for both palettes.*

Proof. Call the old labels $1, \ldots, r$, put $u = 1$ and $p = 2$, and prescribe label i at anchor A_i. For every $i \neq p$, the construction has a role vertex U_i. Its role-edge system isolates exactly two feasible phases:

$$I : \ U_i = i \quad (i \neq p), \qquad C : \ U_i = p \quad (i \neq p).$$

The two pure phases are balanced at the common budget, while a coherent mix with j private roles incurs an additional positive term proportional to $j(r-1-j)$, chosen larger than the available payload slack; assigning a role any third label costs still more. Thus no intermediate role phase is feasible. The remaining live vertices form the payload: a center x, leaf vertices partitioned into blocks, and tag vertices. Together with the U_i, these are all unconditioned vertices. A complete specification of the graph and its integer weights, together with the budget and the verification that every other role assignment is infeasible, appears in Appendix B.3.

Set $a = n-2$, $b = n-1$, and $c = n$. The payload attached to the roles has the following residual values:

	I	C
r-label payload	a	b
s-label payload	c	b

. (12)

In phase I, the payload exposes full and punctured Boolean cubes through the private roles; in phase C, the roles collapse to p and the payload becomes the punctured Boolean cube $\{p,u\}^n \setminus \{u^n\}$. Appendix B.3 derives the four entries in (12), uniformly for every $s \geq r+2$, directly from the construction.

The lower value is b. Indeed, query any U_i and predict p. Response p enters phase C with residual value b, whereas response i costs one mistake and enters phase I with residual value a; both branches cost at most b. The phase-C payload gives the matching lower bound. Similarly, the upper value is c: query U_i and predict i. Its phase-I branch has value c, while response p costs one mistake and leaves value $b = c-1$. Phase I supplies the matching lower bound.

It remains to show that these two optimal allowances are incompatible. The conditioned anchors have already been revealed, and every live vertex is of one of the three types in the following table. The tight response sections are

queried vertex	response leaving lower residual b	response leaving upper residual c
U_i	p	i
x or a block vertex	p	a fresh label f
a tag vertex	p	u

. (13)

Appendix B.3 verifies these response-section values directly from the construction. For completeness, conditioning on a response only restricts the target class and removes the queried coordinate, so each section is bounded above by the already established whole-game value b or c. In every entry of the table, the indicated pure phase remains available and contains a subclass of that same value. The displayed section values are therefore equalities, not merely lower bounds from the pure phases.

At a role vertex, the lower allowance forces prediction p, after which response i already costs one mistake and leaves the full upper value c. At x or a block vertex, the same lower requirement forces prediction p, while a fresh response leaves upper value c. At a tag, the upper allowance forces prediction u, while response p costs one mistake and leaves lower value b. Thus every possible first action violates one of the two minimax allowances. □

Corollary 4.4. *For every $r \geq 3$ and $s \geq r+2$, a connected simple unweighted graph and an integer budget satisfy $m_s = m_r + 1$ and have no policy minimax for both palettes.*

Proof. Release the prescribed anchors using Lemma 3.4, which adds the same selector loss to both values and preserves incompatibility. Then apply Proposition 3.3. □

4.3 The binary lower palette

The adjacent theorem handles $2 \to 3$. A fixed finite construction supplies every pair $2 \to s$ with $s \geq 4$.

Lemma 4.5. *There is a connected graph $J_\star$ with positive integer edge weights and budget* 200000 *such that*

$$m_2(J_\star, 200000) = 3, \qquad m_s(J_\star, 200000) = 4 \quad (s \geq 3),$$

and, for any $s \geq 3$, no s-label policy is minimax for both the binary and s-label classes.

Proof. The graph consists of two identical petals meeting only at a vertex o. Appendix B.4 gives the complete weighted edge list and budget. After fixing $o = 1$, it also lists all twenty-three feasible equality-partition representatives of the one-petal residual class and the exact Bellman recurrence. Let $\mathcal{P}_q$ denote that residual class at palette size q. Direct exact evaluation gives

q	2	3	4
$v(\mathcal{P}_q)$	2	3	3
$m_q(J_\star, 200000)$	3	4	4

.

The action-level output is also short. At the distinguished vertex t, the response-section values, in label order, are

$$(1,2), \qquad (1,3,3), \qquad (1,3,3,3) \quad \text{for } q = 2,3,4,$$

respectively; every other petal query has binary sections $(2,2)$. Thus on $\mathcal{P}_2$, loss two uniquely forces the action "query t and predict label 2," whereas that action costs four on $\mathcal{P}_3$ and $\mathcal{P}_4$. No conditioned-petal policy therefore has respective loss caps $(2,3)$. On the two-petal binary class, the response sections are $(2,2)$ at b_i, c_i, d_i, e_i and $(3,3)$ at every other live vertex. Consequently every first action of cost at most three queries one of the four listed vertices in some petal.

The exhaustive computation also shows that a nonconstant petal costs at least 102856, more than half the budget. Hence every nonconstant feasible target activates exactly one petal. Suppose a three- or four-label policy were minimax for both palettes. By the two-petal section values, its binary first query lies in one petal at one of the listed vertices. Return one of the two lower-palette labels different from its prediction, complete that petal and o constantly, and leave the other petal unresolved. A constant petal uses no budget, so every target in the other conditioned petal remains feasible. By label symmetry, for either binary response the latter residual class is a label-renamed copy of $\mathcal{P}_q$. Following the forced mistake, suppressing queries in the fixed petal would therefore produce a conditioned-petal policy with caps $(2,3)$, which is impossible by the preceding action calculation.

The enumeration shows that no feasible petal partition has four blocks. Nor can it have more: merging blocks down to four could only decrease the cut cost. Since at most one petal is nonconstant, no feasible partition of $J_\star$ has more than three blocks. Indeed, if $\beta_{200000}(J_\star) \geq 4$, assigning four distinct labels to four components produced by an admissible deletion would give a feasible four-block equality partition, a contradiction. Thus $\beta_{200000}(J_\star) \leq 3$, and Theorem 3.2 gives $m_s(J_\star, 200000) = 4$ for every $s \geq 4$. If a simultaneous policy existed for some $s > 4$, choose four labels containing the two binary labels and simulate that policy internally on their targets. Whenever it predicts outside the four labels, output a fixed retained label instead, but feed the revealed label to the original simulation. This modification cannot increase the number of mistakes, so it would contradict the four-label case.

All entries are obtained from the printed integer weights and the exact Bellman recurrence. The archived source `binary_nonadjacent_certificate.py` (Zhou, 2026) reruns the exhaustive calculation; it is a reproducibility aid rather than an additional premise. □

These ingredients yield the cleanest form of policy instability: it occurs for every strict enlargement of the palette, even across a one-unit rise in the scalar minimax value.

Theorem 4.6. *For every $2 \le r < s$, there is a connected simple unweighted graph and an integer budget such that*

$$m_s = m_r + 1, \qquad \Delta^{-}_{r \leftarrow s} = \Delta^{+}_{r \leftarrow s} = 1.$$

Thus no upper-palette optimum is minimax on the lower-palette subclass: zero-overhead transfer can fail for every strict palette enlargement.

Proof. There are four cases. For $(r, s) = (2, 3)$, use Theorem 4.1 with $D = 1$. For $r = 2$ and $s \ge 4$, use Lemma 4.5 followed by Proposition 3.3. For $r \ge 3$ and $s = r + 1$, again use Theorem 4.1 with $D = 1$. Finally, for $r \ge 3$ and $s \ge r + 2$, use Corollary 4.4.

Each construction has values $m_r = B$ and $m_s = B + 1$ and excludes a simultaneous optimum. The sandwich (3) permits the lower-class loss of an s-optimal policy to be only B or $B + 1$. The first possibility would give a simultaneous optimum, so every such policy has loss $B + 1$. □

Open problem. For fixed $2 \le r$ and $s \ge r + 2$, can the best-choice overhead $\Delta^{-}_{r \leftarrow s}$ be arbitrarily large? The separated-palette part of Theorem 4.2 shows why a large scalar gap alone cannot settle this question: for every $r \ge 3$ and $s \ge r + 2$, that family has arbitrarily large $m_s - m_r$ while $\Delta^{-}_{r \leftarrow s} = 0$.

5 The price of local access

The connected-prefix rule limits where the next piece of information can be acquired. Neighbor anchoring additionally limits which label may be proposed there. We track the two restrictions separately to distinguish constraints on queries from constraints on predictions.

A close predecessor is the adaptive exploration protocol of Cesa-Bianchi et al. (2011). There the graph is initially unknown, the learner chooses an exposed anchor, and the environment supplies a new neighbor; their principal prediction copies the anchor label. Here the entire graph is known and the learner chooses the exact frontier vertex. Our comparison therefore isolates access order from neighbor-copy prediction and asks for the exact minimax cost of each restriction on one fixed bounded-cut class.

The central phenomenon is an information-order gap: free play may inspect distant endpoints before entering a path, whereas connected play must commit to a frontier without that information. The following elementary module isolates the obstruction needed for the near-factor-two bouquet construction.

Lemma 5.1 (Theta module)**.** *Let $t \ge 3$, and let J_t consist of the t internally disjoint paths*

$$p - a_i - b_i - z \qquad (1 \le i \le t)$$

whose edge weights are $t - 1, 1, t - 1$. Restrict to targets of cut cost at most t, over any palette $q \ge 2$. The module has two properties:

(i) If the history consists only of the revealed response $h(p) = \alpha$, then the residual connected-prefix and neighbor-anchored values are both two.

(ii) Whatever first vertex and prediction the learner chooses, the realizable adversary has a wrong response that leaves connected-prefix residual value at least two.

Proof. Suppose first that p has revealed label α. Query all a_i from p, then b_1 from a_1, then z from b_1, and finally every remaining b_i from z, always copying the queried neighbor's label. This policy is neighbor-anchored connected-prefix. Let X be the number of bichromatic weight-$(t-1)$ edges and let $\ell = \mathbf{1}\{h(a_1) \neq h(b_1)\}$. Its residual loss is at most $X + \ell$. Feasibility gives

$$(t-1)X + \ell \leq C_{J_t}(h) \leq t.$$

Since $t \geq 3$, at most one heavy edge is bichromatic, and the displayed inequality yields $X + \ell \leq 2$. Thus both residual local values are at most two.

For the reverse bound, the realizable adversary maintains a simple invariant: as long as the learner remains on the a-side, answer α; the first attempt to cross an arm through some b_i can then be answered incorrectly while preserving at least one further unavoidable mistake. To formalize this strategy, fix $\beta \neq \alpha$ and retain the following targets: the constant-α target; for each i, the target equal to β only at a_i or only at b_i; and the central target equal to α on $p, a_1, \ldots, a_t$ and to β on $b_1, \ldots, b_t, z$. Each nonconstant target has cut cost exactly t.

Answer α whenever the learner queries an a_i. Consider the first subsequent query at some b_i. Its α-response section contains the constant target and a singleton target at b_j for some $j \neq i$, while its β-response section contains the singleton target at b_i and the central target. Each section therefore contains two targets that still differ on an unqueried vertex. The realizable adversary chooses whichever of α, β differs from the prediction, causing a mistake and leaving residual value at least one. If an earlier prediction at an a_i was not α, answering α already caused a mistake; otherwise connected play must eventually reach such a b_i. In either case the residual connected-prefix value is at least two. This lower argument remains valid if any subset of the a_i has already been revealed as α.

It remains to justify the arbitrary first-query clause. By label symmetry, the realizable adversary chooses a first response α different from the prediction. A first query at p or z leaves the rooted state just analyzed, up to reflection. If the first query is at a_i, the frontier is $\{p, b_i\}$. At b_i the two response sections from the preceding paragraph again let the adversary force a mistake while retaining value one. At p, return α: a non-α prediction is a mistake with two feasible completions left, whereas an α prediction enters the rooted obstruction with a_i already revealed. Thus either action leaves residual value at least two. A first query at b_i is symmetric. □

The lemma isolates both contributions needed below: after the shared vertex is known, each untouched module costs two local mistakes; a module containing the initial query can be forced to contribute three including that first mistake.

Theorem 5.2. *For every integer $N \geq 1$, there is a connected simple unweighted graph G_N and an integer budget γ_N such that, simultaneously for every palette size $q \geq 2$,*

$$M_{sd}(\mathcal{H}^{(q)}_{\gamma_N}, G_N) = N + 1, \qquad V_{\mathrm{conn}} = V_{\mathrm{anch}} = 2N + 1.$$

Equivalently, for every integer $M \geq 2$ there is an instance satisfying

$$V_{\mathrm{free}} = M, \qquad V_{\mathrm{conn}} = V_{\mathrm{anch}} = 2M - 1.$$

Hence neither local protocol admits a universal multiplicative comparison with the free value by a factor smaller than two.

Proof. Choose an integer $t > 2N + 2$. Take N copies of the module in Lemma 5.1 and identify their vertices p. Module j has right endpoint z_j and paths

$$p - a_{j,i} - b_{j,i} - z_j \qquad (1 \leq i \leq t)$$

with edge weights $t-1, 1, t-1$. Give the whole weighted bouquet budget $\Gamma = Nt$.

Free upper bound. Query p, then every z_j while predicting the label of p, and finally query $a_{j,i}$ predicting the label of p and $b_{j,i}$ predicting the label of z_j. Let Z be the number of modules whose two endpoints differ, X the number of bichromatic weight-$(t-1)$ edges, and Y the number of bichromatic weight-one edges. Apart from the possible first mistake, this policy makes at most $Z + X$ mistakes. Each module counted by Z has, on every arm, either a bichromatic heavy edge or a bichromatic middle edge, so $Y \geq tZ - X$. Feasibility gives

$$Nt \geq (t-1)X + Y \geq tZ + (t-2)X.$$

Thus $Z \leq N$ and, because $t - 2 > 2N$,

$$X \leq \left\lfloor \frac{t(N-Z)}{t-2} \right\rfloor = N - Z.$$

The free loss is at most $N + 1$.

Free lower bound. Select one vertex $b_{j,1}$ in each module. Its singleton boundary has weight t, and the complement of the N selected vertices is connected because every module has another intact arm. These singletons and the complement form an $(N+1)$-part partition with total boundary Nt. Assigning two labels independently to the parts gives a feasible Boolean $(N+1)$-cube. Hence the unrestricted minimax value is $N + 1$ for every palette.

Local upper bound. There is a neighbor-anchored policy with at most $2N + 1$ mistakes. Root at p. In each module query all $a_{j,i}$ from p, then $b_{j,1}$ from $a_{j,1}$, then z_j from $b_{j,1}$, and finally the remaining $b_{j,i}$ from z_j, always copying the predecessor's label. If ℓ_j records whether $a_{j,1}b_{j,1}$ is bichromatic, its loss is at most

$$1 + X + \sum_j \ell_j.$$

The cut cost bound gives $X \leq \lfloor Nt/(t-1) \rfloor = N$, and $\sum_j \ell_j \leq N$, proving the upper bound.

Local lower bound. Retain the global targets whose restriction to every module has cut cost at most t; their total cost is at most Nt. It is enough to prove the connected-prefix lower bound, since neighbor-anchored play is more restrictive. Once the response at p is fixed, the retained class is the Cartesian product of the rooted module classes. For a current history, let K_j be the residual class in module j, let u_j be its connected-prefix Bellman value with p already queried, and let e count mistakes made since the response at p was fixed. Define

$$\Psi = e + \sum_{j=1}^{N} u_j.$$

Every legal query belongs to one module, say j. The Bellman recurrence for K_j supplies a feasible response for which the immediate mistake indicator δ and new value u'_j satisfy $\delta + u'_j \geq u_j$. All other factors are unchanged, and the product structure lets this response be combined with arbitrary completions in them. Hence Ψ never decreases. Initially every untouched rooted module has value two by Lemma 5.1(i), and at termination all u_j vanish. Thus the rooted factors force two mistakes each.

If the learner starts at p, answer the first prediction incorrectly and then apply this potential, forcing $1 + 2N$ mistakes. Otherwise its first query lies in one active module, and before p is queried

no other module is reachable. Lemma 5.1(ii) answers that first prediction incorrectly and leaves active-module value at least two. Let e now count subsequent mistakes and u the active-module residual value. At each query before and including the query of p, the same one-factor Bellman inequality preserves $e + u \geq 2$. When p is revealed, the other $N - 1$ modules become untouched rooted factors of value two, and the product potential applies. The total loss is therefore at least

$$1 + e + u + 2(N - 1) \geq 2N + 1.$$

Finally apply Proposition 3.3. The replacement cliques cannot be split within budget, so contraction preserves all three values. This gives the asserted simple unweighted graph. □

The lower bound $2M - 1$ is asymptotically decisive: no factor below two can hold. The complementary statement

$$V_{\mathrm{conn}}, V_{\mathrm{anch}} \leq 2\, M_{sd}(\mathcal{H}_\gamma^{(q)}, G)$$

for every bounded-cut instance remains open. The bouquet theorem gives a necessary lower bound of two on any universal comparison constant; it does not establish a universal upper bound.

6 Deterministic agnostic regret

The realizable game assumes that the revealed labeling meets the cut budget. We now allow arbitrary responses and compare the learner's mistakes with those of the best bounded-cut labeling in hindsight. This changes the governing geometry: the graph still selects the structured comparator class, but Hamming distance determines the comparator's loss.

Let V be a known set of n coordinates, let $q \geq 2$, and let $\varnothing \neq \mathcal{H} \subseteq [q]^V$. In this section a *word* and a labeling are the same object, with vertices serving as coordinates. Put

$$d(f, g) = |\{v \in V : f(v) \neq g(v)\}|, \qquad d(y, \mathcal{H}) = \min_{h \in \mathcal{H}} d(y, h).$$

The learner knows $\mathcal{H}$ and knows that it is playing the agnostic game, but the response word $y \in [q]^V$ is arbitrary. The comparator is chosen only in hindsight. The deterministic agnostic value is

$$R_{\mathrm{sd}}^{\mathrm{det}}(\mathcal{H}) := \min_{A} \max_{y \in [q]^V} \big(M(A, y) - d(y, \mathcal{H})\big).$$

An adaptive adversary is legitimate here: once a deterministic learner is fixed, its response history defines one completed word y. This is the deterministic finite-domain agnostic analogue of classical self-directed learning (Goldman and Sloan, 1994). Devulapalli and Hanneke (2024) study a broader binary formulation with randomized expected regret; retaining the present deterministic protocol allows exact graph-dependent worst-case values.

For later use, the exact backward recursion may be written directly on a partial word $z \in ([q] \cup \{*\})^V$. If $z^{x \leftarrow a}$ fills coordinate x with a, define

$$F_{\mathcal{H}}(z) = \begin{cases} -d(z, \mathcal{H}), & z \in [q]^V, \\ \min\limits_{x : z_x = *} \min\limits_{p \in [q]} \max\limits_{a \in [q]} \big(\mathbf{1}\{p \neq a\} + F_{\mathcal{H}}(z^{x \leftarrow a})\big), & \text{otherwise.} \end{cases} \tag{14}$$

Backward induction gives $R_{\mathrm{sd}}^{\mathrm{det}}(\mathcal{H}) = F_{\mathcal{H}}(*^V)$.

For a nonempty class in Hamming space, write

$$\begin{aligned}\operatorname{covrad}(\mathcal{H}) &= \max_{y\in[q]^V} d(y,\mathcal{H}),\\ \operatorname{rad}(\mathcal{H}) &= \min_{p\in[q]^V}\max_{h\in\mathcal{H}} d(p,h),\\ \operatorname{diam}(\mathcal{H}) &= \max_{f,g\in\mathcal{H}} d(f,g).\end{aligned}$$

The radius, also called the *Chebyshev radius*, is the performance of the best fixed prediction word; the covering radius measures the farthest possible response word from the comparator set; and the diameter measures the comparator set's spread.

Proposition 6.1. *For every nonempty $\mathcal{H}\subseteq[q]^V$,*

$$\max\left\{n-\operatorname{covrad}(\mathcal{H}),\left\lceil\frac{\operatorname{diam}(\mathcal{H})}{2}\right\rceil\right\}\le R_{\mathrm{sd}}^{\mathrm{det}}(\mathcal{H})\le\operatorname{rad}(\mathcal{H})\le\operatorname{diam}(\mathcal{H}). \tag{15}$$

Consequently, a fixed Chebyshev-center word is always a factor-two approximation to optimal deterministic self-directed regret.

Proof. If the learner predicts a fixed word p and h_y is a nearest comparator to y, the triangle inequality gives

$$d(p,y)-d(y,\mathcal{H})\le d(p,h_y).$$

For the reverse inequality for a fixed word p, take a farthest $h\in\mathcal{H}$ as the response word: its comparator loss is zero. Thus the worst-case regret of p is exactly $\max_{h\in\mathcal{H}} d(p,h)$. Minimizing over p proves the middle upper bound. Choosing $p\in\mathcal{H}$ gives $\operatorname{rad}(\mathcal{H})\le\operatorname{diam}(\mathcal{H})$.

For the covering-radius lower bound, answer every prediction incorrectly. The resulting word gives regret at least $n-d(y,\mathcal{H})\ge n-\operatorname{covrad}(\mathcal{H})$. For the diameter bound, choose $f,g\in\mathcal{H}$ at distance D. At each coordinate where they differ, return one of $f(v),g(v)$ that is not the prediction; at their common coordinates return the common label. The learner makes at least D mistakes, while the two comparator losses sum to D. One comparator loses at most $\lfloor D/2\rfloor$, giving regret at least $\lceil D/2\rceil$. Since $\operatorname{diam}(\mathcal{H})\le 2R_{\mathrm{sd}}^{\mathrm{det}}(\mathcal{H})$, the factor-two claim follows. □

In binary Hamming space, complementation gives the standard antipodal radius–covering-radius identity (Mazumdar et al., 2013), making the two geometric bounds coincide.

Theorem 6.2. *For every nonempty $\mathcal{H}\subseteq\{0,1\}^V$,*

$$R_{\mathrm{sd}}^{\mathrm{det}}(\mathcal{H})=\operatorname{rad}(\mathcal{H})=n-\operatorname{covrad}(\mathcal{H}).$$

An optimal learner is nonadaptive: choose a Chebyshev center, query in any order, and predict that center word. Thus deterministic adaptivity and query order have exactly zero value relative to the best fixed word for every binary comparator class.

Proof. For the bitwise complement $\bar{c}$, $d(c,h)=n-d(\bar{c},h)$. Since complementation is a bijection,

$$\min_c\max_{h\in\mathcal{H}} d(c,h)=n-\max_z\min_{h\in\mathcal{H}} d(z,h)=n-\operatorname{covrad}(\mathcal{H}).$$

The two matching sides of (15) force equality. This equality also has a direct worst-case witness for the stated fixed strategy. If c is a Chebyshev center of radius R and the response is $y=\bar{c}$, then the learner predicting c makes n mistakes, while

$$d(y,\mathcal{H})=\min_{h\in\mathcal{H}}\bigl(n-d(c,h)\bigr)=n-R.$$

Its regret on this word is therefore exactly R. □

The conclusion changes immediately when a third response symbol is available. Let $\mathcal{K}_q(V) = \{a^V : a \in [q]\}$ be the class of constant words.

Theorem 6.3. *For every $n \geq 1$ and $q \geq 2$,*

$$R_{\mathrm{sd}}^{\det}(\mathcal{K}_q(V)) = \left\lceil \frac{n}{2} \right\rceil, \qquad \mathrm{rad}(\mathcal{K}_q(V)) = n - \left\lfloor \frac{n}{q} \right\rfloor.$$

For $q = 3$ and $n = 2$, adaptive regret is 1, while every fixed word has worst-case regret 2. Hence the factor two in Proposition 6.1 is sharp, and a third symbol already makes feedback strictly more powerful than every fixed prediction word.

Proof. Two distinct constant words are distance n apart, so the diameter lower bound gives $\lceil n/2 \rceil$. For the upper bound, pair the coordinates, with at most one singleton. Predict arbitrarily on the first coordinate of each pair and copy its response on the second. If the two responses agree, the pair costs at most one mistake. If they differ, it costs at most two, but every constant comparator disagrees on at least one coordinate of the pair. Thus, against every constant comparator, the regret is at most one per pair and at most one on the singleton. The same bound holds against the best constant chosen in hindsight.

If a fixed word p uses label a exactly n_a times, its farthest constant word is at distance $n - \min_a n_a$. Balancing the counts n_a proves the radius formula and the stated two-coordinate example. □

6.1 Equality patterns and palette stabilization

For $h \in [q]^V$, let $\Pi(h)$ be its *equality partition*, defined by

$$u \sim v \quad \Longleftrightarrow \quad h(u) = h(v).$$

It records which coordinates share a label while forgetting the label names. For example, $(1, 1, 2, 3, 2)$ and $(3, 3, 1, 2, 1)$ induce the same partition. If $\mathfrak{P}$ is a nonempty family of set partitions of V, viewed as the collection of allowed equality patterns, define

$$\mathcal{H}_q(\mathfrak{P}) := \{h \in [q]^V : \Pi(h) \in \mathfrak{P}\}.$$

Lemma 6.4. *Let P be a partition of V with at most q blocks, and let $\mathcal{R}_q(P) = \{h \in [q]^V : \Pi(h) = P\}$ be its set of label realizations. For a response word y, put $Y_a = y^{-1}(a)$ for $a \in [q]$, allowing empty classes. Then*

$$d(y, \mathcal{R}_q(P)) = n - \max_{\sigma : P \hookrightarrow [q]} \sum_{B \in P} |B \cap Y_{\sigma(B)}|,$$

where the maximum is over injections, equivalently over maximum-weight matchings that cover the blocks of P. In particular, when $q \geq n$, the right side depends only on the two set partitions P and $\Pi(y)$, not on the ambient palette.

Proof. A realization of P is exactly an injection σ assigning distinct label names to its blocks. If block B receives name $\sigma(B)$, it agrees with y on exactly $|B \cap Y_{\sigma(B)}|$ coordinates. Maximizing these agreements and subtracting from n proves the formula. The positive matching weights depend only on intersections between blocks of P and $\Pi(y)$. A maximum-weight partial matching between these two partitions can be extended to cover P using unused label names with zero additional weight. Indeed, an unmatched block and an unmatched observed label cannot have positive intersection, since adding that pair would improve the matching, and $q \geq n \geq |P|$ supplies enough names. Extra unused names therefore do not change the optimum. □

Theorem 6.5. *For each fixed q for which $\mathcal{H}_q(\mathfrak{P})$ is nonempty, Bellman values of partial histories depend only on the queried set and the equality partition of the observed labels. If $n = |V|$, then*

$$R_{\mathrm{sd}}^{\det}(\mathcal{H}_q(\mathfrak{P})) = R_{\mathrm{sd}}^{\det}(\mathcal{H}_{n+1}(\mathfrak{P})) \qquad (q \geq n+1).$$

Proof. Permuting label names preserves the comparator class, Hamming distance, and every action in the game. Relative to a partial history, a prediction is therefore either one specified old label or a fresh label. A response is the prediction, another specified old label, or a different fresh label. Before the terminal round at most $n-1$ names have appeared, so every one of these orbit types is available once $q \geq n+1$. At a terminal history, Lemma 6.4 gives

$$d(y, \mathcal{H}_q(\mathfrak{P})) = \min_{\substack{P \in \mathfrak{P} \\ |P| \leq q}} d(y, \mathcal{R}_q(P)),$$

and for $q \geq n$ the restriction $|P| \leq q$ is vacuous. The quantity then depends only on the equality partition of y. These palette-independent terminal values start a backward induction in (14); the common nonterminal orbit types then prove both assertions. □

Proposition 6.6. *Suppose $\mathfrak{P}$ is closed under merging blocks. If $q \geq \ell \geq 2$, then*

$$R_{\mathrm{sd}}^{\det}(\mathcal{H}_q(\mathfrak{P})) \geq R_{\mathrm{sd}}^{\det}(\mathcal{H}_\ell(\mathfrak{P})).$$

In particular, for bounded-cut graph comparators the agnostic value is nondecreasing in the palette and is constant from palette $n+1$ onward. The number $n+1$ is a universal sufficient threshold, not necessarily the first one.

Proof. Choose a map $\phi : [q] \to [\ell]$ that fixes $[\ell]$. Against a q-label learner, simulate an optimal ℓ-label adversary using prediction $\phi(p)$. Every simulated mistake is a real mistake. For a response word $y \in [\ell]^V$, the comparator distances in the two classes agree: one inequality is inclusion, while applying ϕ to a q-label comparator merges blocks, preserves feasibility, and cannot increase its distance from y. For graph classes, merging labels only removes bichromatic edges. The monotonicity and Theorem 6.5 give the final claim. □

6.2 Exact agnostic graph benchmarks

For the rest of this subsection, graphs are unweighted, the budget t is a nonnegative integer, and the comparator is $\mathcal{H}_t^{(q)}(G)$ from (1); the response word remains arbitrary. The one technical ingredient needed for the exceptional cycle case is recorded next.

Lemma 6.7. *Let $v_0, v_1, \ldots, v_\ell, v_{\ell+1}$ be a path whose endpoint labels are fixed and whose ℓ internal vertices remain unqueried. If $q \geq 3$, $a \in \mathbb{Z}_{\geq 0}$, and $\ell \geq 2a+2$, the adversary can answer every remaining prediction incorrectly and leave a response word y and a comparator h with the prescribed endpoint labels such that*

$$d(y,h) \leq a, \qquad |\{i \in \{0,\ldots,\ell\} : h(v_i) \neq h(v_{i+1})\}| \leq \ell - 2a.$$

Proof. The parity-sensitive induction is given in Appendix D. □

For K_n, define

$$u_t = \max\left\{u \in \mathbb{Z} : 0 \leq u \leq \left\lfloor \frac{n}{2} \right\rfloor,\ u(n-u) \leq t\right\}.$$

For P_n, put $b = \min\{t, n-1\}$. For C_n, where $n \geq 3$, put

$$\tau = \min\{t, n\}, \qquad s_t = 2\min\left\{\left\lfloor \frac{\tau}{2} \right\rfloor, \left\lfloor \frac{n}{2} \right\rfloor\right\}.$$

Theorem 6.8. *For every $q \geq 2$,*

$$R_{\mathrm{sd}}^{\mathrm{det}}(\mathcal{H}_t^{(q)}(K_n)) = \left\lceil \frac{n}{2} \right\rceil + u_t, \tag{16}$$
$$R_{\mathrm{sd}}^{\mathrm{det}}(\mathcal{H}_t^{(q)}(P_n)) = \left\lceil \frac{n+b}{2} \right\rceil.$$

For $q = 2$, the corresponding covering radii are $\lfloor n/2 \rfloor - u_t$ and $\lfloor (n-b)/2 \rfloor$.

On cycles,

$$R_{\mathrm{sd}}^{\mathrm{det}}(\mathcal{H}_t^{(2)}(C_n)) = \left\lceil \frac{n}{2} \right\rceil + \frac{s_t}{2}, \tag{17}$$

$$R_{\mathrm{sd}}^{\mathrm{det}}(\mathcal{H}_t^{(q)}(C_n)) = \begin{cases} \lceil n/2 \rceil, & \tau \leq 1, \\ \left\lceil \dfrac{n + \min\{\tau, n-1\}}{2} \right\rceil, & \tau \geq 2, \end{cases} \qquad (q \geq 3). \tag{18}$$

Proof. On K_n, a binary word whose minority label occurs u times cuts exactly $u(n-u)$ edges. The feasible binary class is therefore the union of the radius-u_t Hamming balls centered at 0^n and 1^n, and its covering radius is $\lfloor n/2 \rfloor - u_t$. The binary collapse and palette monotonicity give the claimed lower bound for every palette.

For the reverse bound, suppose first that $\lceil n/2 \rceil + u_t < n$. Then $u_t < \lfloor n/2 \rfloor$, so maximality in the definition of u_t gives

$$t < (u_t + 1)(n - u_t - 1) \leq \left\lfloor \frac{n}{2} \right\rfloor \left\lceil \frac{n}{2} \right\rceil \leq \frac{n^2}{4}.$$

Every feasible multiclass comparator therefore has a majority label. Indeed, if its label-class sizes a_i were all at most $n/2$, then

$$C_{K_n}(h) = \frac{1}{2}\left(n^2 - \sum_i a_i^2\right) \geq \frac{n^2}{4} > t.$$

If the majority has size $n - u$, merging all minority labels gives a binary cut of size $u(n-u)$, so $u \leq u_t$. Thus every feasible comparator is within u_t of a constant word. Equivalently,

$$d(y, \mathcal{H}_t^{(q)}(K_n)) \geq d(y, \mathcal{K}_q(V)) - u_t.$$

The optimal constant-comparator policy from Theorem 6.3 now has regret at most $\lceil n/2 \rceil + u_t$. If that expression equals n, the trivial upper bound suffices. This proves (16).

For P_n, take any binary response word, preserve its first b vertices, and make the remaining interval constant in its majority label. The resulting comparator has at most b transitions and changes at most $\lfloor (n-b)/2 \rfloor$ coordinates. Binary collapse and monotonicity again give the stated lower bound. For the upper bound, query along the path and copy the preceding response after the first query. If the response word has c transitions, this policy makes at most $1 + c$ mistakes. When $c > b$, changing one coordinate can remove at most two transitions, so

$$d(y, \mathcal{H}_t^{(q)}(P_n)) \geq \left\lceil \frac{c-b}{2} \right\rceil.$$

It follows that the regret is at most

$$1 + c - \left\lceil \frac{c-b}{2} \right\rceil \leq \left\lceil \frac{n+b}{2} \right\rceil.$$

If $c \le b$, the bound $1 + c \le 1 + b \le \lceil (n+b)/2 \rceil$ applies.

For a binary cycle, every cut count is even, so budget t is equivalent to budget s_t. In an arbitrary binary response word, choose a minority label. If it has at most $s_t/2$ runs, change nothing; otherwise retain its $s_t/2$ largest runs and flip the others. In the latter case this changes at most $\lfloor n/2 \rfloor - s_t/2$ vertices, and in both cases it leaves at most s_t transitions. Conversely, start with a binary word having $2\lfloor n/2 \rfloor$ transitions. A single coordinate change destroys at most two transitions, so every feasible comparator is at least $\lfloor n/2 \rfloor - s_t/2$ away. Thus,

$$\operatorname{covrad}(\mathcal{H}_t^{(2)}(C_n)) = \left\lfloor \frac{n}{2} \right\rfloor - \frac{s_t}{2},$$

and binary collapse proves (17).

Now let $q \ge 3$. If $\tau \le 1$, the comparator class consists only of constants. If $\tau \ge 2$, delete one cycle edge and apply the path upper bound with budget $\min\{\tau, n-1\}$. The binary lower bound matches this quantity except when n is even and τ is odd with $3 \le \tau \le n-1$. In that exceptional case put

$$a = \frac{n - \tau - 1}{2}.$$

Answer the first cycle query incorrectly and view the remaining cycle as a path whose two endpoints are copies of that revealed vertex. Apply Lemma 6.7 to the $n-1$ unqueried internal vertices with allowance a. Every prediction is wrong, while a comparator at distance at most a has at most $(n-1) - 2a = \tau$ bichromatic edges. The regret is at least $n - a = (n + \tau + 1)/2$, matching the path upper bound. This proves (18). □

6.3 Computational complexity

The binary theorem turns the agnostic game into a static center problem, but it does not make exact evaluation easy. In fact, the same Max Cut endpoint that gives realizable hardness also settles binary unweighted agnostic hardness.

Corollary 6.9. *On connected simple unweighted graphs with n vertices and a nonnegative integer budget γ, deciding whether $R_{\mathrm{sd}}^{\det}(\mathcal{H}_\gamma^{(2)}(G)) \le n-1$ is NP-complete, and deciding whether it equals n is coNP-complete. Exact evaluation is NP-hard.*

Proof. By Theorem 6.2,

$$\begin{aligned} R_{\mathrm{sd}}^{\det}(\mathcal{H}_\gamma^{(2)}(G)) = n \quad &\iff \quad \operatorname{covrad}(\mathcal{H}_\gamma^{(2)}(G)) = 0 \\ &\iff \quad \mathcal{H}_\gamma^{(2)}(G) = [2]^V \\ &\iff \quad \operatorname{MaxCut}(G) \le \gamma. \end{aligned}$$

Thus the value is at most $n-1$ exactly when a cut of size at least $\gamma + 1$ exists. A cut certifies this condition in polynomial time, and the connected unweighted Max Cut reduction in the proof of Theorem 3.7 proves NP-hardness. Taking complements gives the second claim, and exact evaluation decides either endpoint condition. □

For each fixed palette $q \ge 3$, the binary radius identity is unavailable. The next reduction establishes hardness with positive integer edge weights directly at the root of the adaptive game, without reducing to a static comparator-distance problem.

Theorem 6.10. *For every fixed $q \ge 2$, exact evaluation of $R_{\mathrm{sd}}^{\det}(\mathcal{H}_\gamma^{(q)}(G))$ is NP-hard on connected simple graphs with positive integer edge weights. More precisely, deciding whether the value is at most $|V(G)| - 1$ is NP-hard, while deciding whether it equals $|V(G)|$ is coNP-hard.*

Proof. For a weighted graph J on N vertices, define

$$\Phi_q(J) = \max_A \min\{C_J(y) : y \text{ is an all-mistake response word against } A\}.$$

Then

$$R_{\mathrm{sd}}^{\det}(\mathcal{H}_\gamma^{(q)}(J)) = N \quad \Longleftrightarrow \quad \Phi_q(J) \le \gamma.$$

If $\Phi_q(J) \le \gamma$, every learner admits an all-wrong word that is itself a feasible comparator. If $\Phi_q(J) > \gamma$, one learner ensures that every response word either contains a correct prediction or, if all predictions are wrong, lies outside the comparator class and has comparator distance at least one. Its regret is therefore at most $N - 1$.

Reduce a Max Cut instance (G, B) with $1 \le B \le m$, where G has n vertices, $m > 0$ edges, and maximum cut size M. Add vertices a, b. Retain the original edges with weight one, give each edge av and bv weight $D = m$, and give ab weight

$$W = Dn + m.$$

Call the resulting connected simple weighted graph J. We claim

$$\Phi_q(J) = W + Dn + M. \tag{19}$$

For the lower bound, a learner queries a, receives a wrong label α, then queries b while predicting α and receives $\beta \neq \alpha$. Fix a maximum cut of G. At each original vertex, predict the anchor label opposite the label prescribed by its shore. An all-wrong response in $\{\alpha, \beta\}$ is forced to be the prescribed label and contributes D on the two anchor edges. If no original vertex receives a third label, the old edges contribute exactly M. If at least one does, the vertex–anchor edges contribute at least $Dn + D = Dn + m$, while the old-edge contribution is nonnegative; since $M \le m$, the total is again at least $W + Dn + M$.

Conversely, fix two labels and, against any learner, always answer with one of them different from the prediction. The resulting binary cut has weight at most $W + Dn + M$. If the anchors are separated, the anchor edge and the vertex–anchor edges contribute exactly $W + Dn$, and the old graph contributes at most M. If the anchors agree, the total is at most $2Dn + m = W + Dn$. This proves (19).

Set $\gamma = W + Dn + B - 1$. If $M \ge B$, then $\Phi_q(J) > \gamma$ and the value is at most $N - 1$; if $M < B$, then $\Phi_q(J) \le \gamma$ and the value is N. Max Cut is NP-complete (Garey et al., 1976), proving both hardness orientations. □

Proposition 3.3 does not remove the weights from this reduction: its replacement cliques force feasible comparators to be constant, but an agnostic response word is unrestricted and need not be constant on a replacement clique. Binary unweighted hardness is settled separately by Corollary 6.9. For each fixed $q \ge 3$, the complexity of exact evaluation on connected simple unweighted graphs remains open.

7 Discussion

The theory has two complementary regimes. Under realizability, four layers remain distinct. Exchange blocks describe pair-specific ambiguity inside the target class. Palette saturation asks when extra category names stop changing the best attainable score. Policy transfer asks the finer question of whether a procedure attaining that score survives a change of alphabet. Local

exploration changes neither the graph nor the target class; it changes only when information can become available. In the agnostic regime, the graph class becomes a hindsight comparator and Hamming geometry replaces version-space containment as the organizing principle.

Several contrasts are exact. Fragmentation gives a universal lower bound and an eventual palette threshold, but Theorem 3.5 shows that the first threshold can be any prescribed palette. The stabilized tail has one common palette-blind equality-pattern policy schema, yet Theorem 4.6 shows that every strict palette pair can force optimizer incompatibility, and Theorem 4.2 shows that equally optimal richer-class policies may realize a complete interval of transfer losses. The exchange certificate, in turn, approximates the exact tree value within a sharp factor two but can fall short by a factor $|V|$ on cliques. Thus neither a scalar mistake bound nor a single root certificate captures the full decision-tree structure. The theta bouquet adds a different obstruction: retaining the same target class but restricting information order can nearly double its value.

The agnostic results expose a second sharp contrast. Binary deterministic regret is completely static: it is exactly the Chebyshev radius for every comparator class. Multiclass feedback can instead improve on the best fixed word by the largest possible factor, and cycles realize a strict palette jump within the same bounded-cut family. Thus palette effects arise both from the feasible version space under realizability and from the geometry of the hindsight benchmark without realizability, but by different mechanisms.

Interpretation and scope

The cleanest operational interpretation is exhaustive verification of a piecewise-constant categorical graph signal. A vertex is an item requiring review, an edge records spatial adjacency or a trusted similarity relation, and a mistake is a proposed label corrected upon inspection. On a grid, the cut budget is a discrete boundary-length bound; on a weighted similarity graph, it bounds the affinity crossing between categories. Connected-prefix play grows one verified region, while neighbor anchoring gives every noninitial proposal one-hop provenance in an already verified neighbor.

This idealization is narrower than a deployed annotation or sensing system. The realizable part assumes a known static graph, exact feedback, a hard cut budget, equal correction costs, and full coverage; the agnostic part allows arbitrary responses and measures the learner's excess mistakes over the best bounded-cut comparator. It contains no unary evidence, reviewer noise, semantic distance between labels, model retraining, route length, or battery cost. The locality gap is therefore an information-order lower bound, not a travel-time lower bound. Likewise, palette growth adds symmetric atomic names; it does not model a hierarchical ontology or semantic distance. The exchange theorem itself extends to unary Potts terms as noted after Theorem 2.5, but the later palette and locality constructions should not be read as quantitative claims for arbitrary unary-augmented systems.

8 Conclusion

Bounded-cut self-directed learning turns a simple graph-cut constraint into two distinct prediction theories. Under realizability, our results isolate distinct roles for exchange geometry, palette size, optimizer transfer, and information order; local access can cost almost a factor two. Without realizability, the same graph family becomes a comparator code: binary regret collapses to a static Hamming radius, while multiclass feedback can achieve a sharp factor-two advantage. Together, the results separate the roles of graph structure, palette size, adaptive order, and hindsight comparison in full-coverage categorical prediction.

Open problems

Four questions now mark natural boundaries of the theory.

1. **A universal locality upper bound.** Does every bounded-cut class satisfy $V_{\mathrm{conn}}, V_{\mathrm{anch}} \leq 2m_q$? Theorem 5.2 shows that the constant two would be best possible.

2. **Best-choice transfer between nonadjacent palettes.** For fixed $s \geq r+2$, is $\Delta^{-}_{r\leftarrow s}$ unbounded over connected simple unweighted graphs, or does the presence of two fresh labels impose a finite bound? Theorem 4.6 proves strict failure, but the adjacent amplification uses the unique fresh label in an essential way. The complete-spectrum family shows why a large scalar gap alone cannot answer the question: with two fresh labels it may also contain a perfectly transferring optimum.

3. **Treewidth in unweighted graphs.** Is $q_0(G,\gamma) \leq \mathrm{tw}(G)+2$ for every connected simple unweighted graph? Theorem 3.6 shows that this bound would be sharp for weighted graphs, but exact unweighted realization does not preserve treewidth.

4. **Uniform multiplicative palette growth.** For fixed r, how large can the ratio

$$\frac{m_{r+1}(G,\gamma)}{m_r(G,\gamma)}$$

be over unweighted instances? The additive palette and transfer gaps can be arbitrarily large, but their current witnesses also make the denominator grow. Fragmentation and spanning-tree bounds do not determine this ratio.

9 Acknowledgements

I am grateful to my mentor, Dr. Jesse Geneson of San José State University, for proposing the model studied in this paper: self-directed prediction of a bounded-cut clustered labeling on a known graph. He also provided high-level guidance on the scope and direction of the project, including advising against directions that appeared unlikely to be productive. The specific division of mathematical, computational, and AI-assisted contributions is described below. This research was conducted through the MIT PRIMES-USA program, and I am grateful to its organizers for providing me with this research opportunity.

During June and early July 2026, I used Google Gemini 3.1 Pro and Anthropic Claude Sonnet 4.6 and Sonnet 5 intermittently for preliminary research contextualization, including exploring possible directions and identifying potentially related areas of literature. These exploratory interactions did not contribute results, proofs, constructions, counterexamples, or other mathematical arguments used in the final paper. Any literature ultimately cited in the manuscript was checked against the original sources.

I used OpenAI Codex GPT-5.6 Sol during August 2026 for targeted counterexample search and computational exploration. I conjectured that, across nested palettes, some optimal upper-palette policy would always remain optimal on the lower-palette target class. After encountering structural obstructions while attempting to prove it, I supplied those obstructions to Codex and asked it to search for counterexamples. Codex constructed and verified a finite weighted binary-to-ternary counterexample based on a weighted petal and a one-active-petal wedge, and generated the exact executable verification program used to certify the construction. I manually reviewed the construction, checked the Bellman computation and certificate, and verified the reported results.

This example motivated my subsequent development of the general adjacent-palette construction and arbitrary-gap result in Theorem 4.1. The original finite construction, which also stabilizes above the ternary palette, is retained as the AI/computer-assisted Lemma 4.5 and supplies the binary nonadjacent case used in Theorem 4.6.

For nonadjacent palettes with $r \geq 3$, I developed the conceptual blueprint for the two-phase construction of Lemma 4.3, including the private and collapsed role phases, the center–leaf–tag payload, the intended phase-value profile, the mechanism isolating the two role phases, and the response-section strategy for proving policy incompatibility. I supplied this construction sketch to Codex and asked it to complete the detailed parameterization and verification. Codex derived explicit parameter choices and filled in technical accounting for the phase-isolation, payload, and response-section claims. I manually checked these calculations and arguments and rewrote the verified details in my own presentation in Appendix B.3. I independently developed and proved the role-release and weighted-to-unweighted realization lemmas used to obtain the final unconditioned simple-unweighted results; for some constructions with large integer parameters, I used Codex to carry out or check numerical applications of the latter lemma.

I also used Codex in early August 2026 to search for counterexamples to my conjectures that connected-prefix and neighbor-anchored connected-prefix policies could attain the unrestricted minimax value. I identified theta graphs as a promising family in which to search for an obstruction and asked Codex to investigate this family. Codex constructed and verified a specific binary weighted theta-graph counterexample with three paths having edge weights $(4, 1, 4)$ and cut budget 5. I subsequently abstracted this obstruction into the parameterized theta module of Lemma 5.1 and independently developed the bouquet amplification, global cut accounting, and potential argument yielding the general M versus $2M - 1$ separation of Theorem 5.2.

During August and September 2026, I also used OpenAI ChatGPT-5.6 Sol and Codex GPT-5.6 Sol repeatedly as secondary tools for proof checking and attempted falsification, literature and reference checking, and suggestions concerning organization and expository clarity. I critically reviewed suggested changes, manually verified mathematical content incorporated into the manuscript, and checked the original sources underlying the paper's citations and prior-work claims. I take responsibility for the final mathematical statements, proofs, citations, and exposition.

A Technical verification for the palette constructions

A.1 Role discovery and selector accounting

We give the query-level details for Lemma 3.4. Whenever a role has already been queried, reuse its known label instead of issuing another query. Each role-pair bundle has $d = 4$ *discovery* midpoints and T reserve midpoints. For one pair, query a discovery midpoint and call its response λ. Predict λ on successive discovery midpoints until either a different response μ appears or all four have been queried. When the endpoint labels are distinct, every midpoint carries one of those two labels. If μ first appears in position $j \geq 2$, the first query, the query that reveals μ, and the remaining $4 - j$ discovery midpoints cost at most

$$1 + 1 + (4 - j) \leq 4$$

mistakes; the responses λ and μ identify the unordered pair of endpoint labels. If all four responses are λ, query both endpoints while predicting λ. One endpoint has label λ, so this entire scan, including the possibly mistaken first midpoint, costs at most two mistakes.

Run this routine on the three bundles joining pairs among a_1, a_2, a_3. If none of the three scans is exhausted, pairwise intersection of the three unordered label pairs identifies the label at each role. If a scan is exhausted, querying its endpoints directly identifies and orients that pair, after which the other pairs can be oriented as well. Now the label of a_1 is known. For each $j \geq 4$, predict that label on the four discovery midpoints of the a_1a_j bundle. A different response identifies the label of a_j; if all four predictions are correct, query a_j while predicting the label of a_1. Once a bundle has identified both endpoint labels, query all of its still-unqueried discovery midpoints; also query the discovery midpoints in every remaining bundle after its endpoint labels are known. If the observed pairs cannot be consistently oriented, abandon inference and query every remaining role directly. Otherwise verify every inferred role. Compare all verified labels before querying any reserve midpoint. Inconsistent inference cannot occur with distinct roles, so it leads to the collision bound below. On a distinct-role branch this procedure costs at most $d\binom{R}{2}$ mistakes, and the reserves cost at most $T\binom{R}{2}$. The total is therefore $M = (d + T)\binom{R}{2}$.

The verification step detects every noninjective assignment before any reserve is used. Suppose $h \geq 1$ role pairs have equal endpoint labels. Every midpoint in an unequal pair contributes at least L to the cut, for a total of $(M - (d + T)h)L$. If e reserve midpoints in equal pairs differ from their common endpoint label, each contributes $2L$. Since the budget is $\Gamma + ML$ and $\Gamma < L$,

$$e \leq \frac{(d + T)h}{2}.$$

After the collisions are known, predict their common label on the reserve midpoints of every equal pair; precisely the e exceptional reserves can be mistakes. Charging every other midpoint, role, and core query as a mistake gives

$$M + N + R - Th + e \leq M + N + R - \frac{(T - d)h}{2} \leq M + b - 1,$$

which is the strict collision bound used in the main proof.

For the suppression argument, fix the canonical distinct role labels and any feasible core target. Whenever the released policy queries a midpoint, answer with an endpoint label different from its prediction; if it predicts neither endpoint label, choose either one. These M choices are independent Boolean coordinates, so they remain jointly feasible with the fixed roles and with every subsequent core history, and every midpoint query is a mistake. Feed the fixed role responses to the simulation.

Deleting its role and midpoint actions leaves a causal fixed-role core policy. Consequently, caps $(u + M, v + M)$ for the released policy would give caps (u, v) for this core policy, because every complete simulation has already spent M mistakes on midpoints.

A.2 The difficult upper-palette child

We complete the first-action check in Theorem 3.5. Fix the fresh label f. At x, return f if the prediction was p and return p otherwise; the wrong response leaves a section of value $n - 1$. At a block vertex v, answer a prediction other than f by f, set $x = f$, and vary the other $n - 1$ vertices independently between f and their role labels. Because $v = f$, this is a feasible full $(n - 1)$-cube. If the prediction is f and v is the unique vertex of B_p, answer p, set $x = p$, and vary all $n - 1$ outside-block vertices between p and their role labels.

It remains to verify (8). When $d = 0$, the two surviving assignments on (x, w) are (p, p) and (f, f), so their union has value one. Suppose $d > 0$. A non-f prediction at x is answered by f and leaves a punctured Boolean cube on $d + 1$ coordinates, of value d; the same response at w or at $z \in Z \setminus S$ leaves a full d-cube. An f prediction at x or w is answered by p, again leaving a full d-cube. Finally, at $z \in Z \setminus S$, return the role label of z; this is a mistake and leaves $L_{S\cup\{z\}} \cup U_{S\cup\{z\}}$, whose value is at least d by induction. Every next action has a wrong child of value at least d, proving the invariant and the required residual value $n - 1$.

A.3 The weighted treewidth construction

We supply the arithmetic deferred from Theorem 3.6. First let $r \geq 3$. If the center has old label i, its block has size s_i, and Q leaves disagree with it, direct minimization gives cut cost at least

$$n - s_i + Q + 2\big(Q - (n - s_i)\big)_+.$$

If $Q \leq n - s_i$, the desired upper bounds are immediate. Otherwise the displayed inequality and the budget $2n - 1$ give

$$Q \leq \left\lfloor \frac{3n - s_i - 1}{3} \right\rfloor = \begin{cases} n - 2, & s_i = 5, \\ n - 3, & s_i = 8. \end{cases}$$

Predicting p at the center and then copying its response therefore costs at most $n - 2$: there is no center mistake when $i = p$, and at most one otherwise. For equality, fix $x = p$, vary the $n - 5$ outside leaves independently between p and their role labels, and allow at most three of the five remaining leaves to use a second old label. This product class has value $(n - 5) + 3 = n - 2$ and maximum cut cost $2(n - 5) + 3 \cdot 3 = 2n - 1$.

With one fresh label f, if $x = f$ and Q leaves disagree with it, the cut cost is at least $n + Q$, so the center-copy policy loses at most $n - 1$. Conversely, the labelings with $x = f$ in which each leaf chooses between f and its role label, excluding the all-role word of cut cost $2n$, form the punctured Boolean n-cube of value $n - 1$. With two fresh labels, a first query at x can be answered by a wrong fresh label and retain this cube. At a first leaf query, give that leaf and x a wrong fresh label and vary the other leaves between it and their roles; the maximum cut cost is $1 + 2(n - 1) = 2n - 1$. Every first action thus has a wrong child of value $n - 1$. The center-copy policy makes at most n mistakes, because a center mistake and n leaf mistakes would cut center edges of total weight $2n$. This proves the fixed-role values $n - 2, n - 1, n$.

For width two, the role blocks have sizes five and eight, the core budget is 25, and the fixed-distinct-role values are $11, 12, 13$. Add sixteen selectors, each joined to both roles by weight-L edges for $L > 25$, and add $16L$ to the budget. Distinct roles give an independent binary 16-cube; if the

roles coincide, at most eight selectors differ from their common label. For two labels, give the two roles opposite labels. Let the center and two leaves of the five-leaf block copy that block's role. The other eleven leaves and the sixteen selectors vary independently between the two labels, while global interchange of the role labels supplies one additional binary coordinate. This gives a full Boolean 28-cube. Every such assignment has core cut cost at most

$$8 \cdot 2 + 3 \cdot 3 = 25,$$

while the selectors contribute $16L$, so the entire cube is feasible. Hence $m_2 \geq 28$. Querying the roles orients them with at most one mistake, so the matching upper bound is $1 + 16 + 11 = 28$; collision branches are cheaper.

For three or more labels, use four selectors for discovery and twelve as reserves. Two observed labels give a candidate unordered role pair. If all four responses coincide, query both roles while first predicting that label. In a distinct-role branch the combined discovery and identification cost is at most two; any failure enters the collision bound below. Three observed labels also certify a collision before any reserve is touched.

After obtaining a two-label candidate pair, postpone verification and run an orientation-free core policy: choose a label outside the pair, query the center with that prediction, and then copy its response on every leaf. To account for the later verification, let E, E' be the core cut costs under the two candidate orientations, let Q count leaf–center disagreements, and let F count leaves whose labels lie outside the candidate pair. Direct summation gives

$$E + E' = 4Q + 13 + F.$$

With three labels, twelve core mistakes imply $Q \geq 11$, so two feasible orientations would give $E + E' \geq 57 > 2 \cdot 25$. With four or more labels, thirteen core mistakes are also impossible under two feasible orientations: if the center prediction is correct, thirteen leaf disagreements already cost $26 > 25$, while otherwise $Q \geq 12$ and $E + E' \geq 61 > 2 \cdot 25$. Thus, if both orientations remain feasible after the core is exposed, its loss is at most 11 or 12, respectively, leaving one mistake to query one role, orient the pair, and then verify the other. In a distinct-role branch this verification costs at most one mistake. If exactly one orientation is feasible, predict it and verify both roles without a mistake; if neither is feasible, or if any verification fails, use the collision bound below. No reserve is queried before this verification.

Consequently discovery, verified core play, and the reserves cost at most $4 + 12 + 12 = 28$ with three labels and $4 + 13 + 12 = 29$ with four or more. The fixed-role core classes times the selector cube give the matching lower bounds. In a collision branch, at most eight selectors can differ from the common role label; charging four discovery queries, both roles, all fourteen core vertices, and those eight exceptions gives at most $4 + 2 + 14 + 8 = 28$. Hence $m_2 = m_3 = 28$ and $m_s = 29$ for $s \geq 4$.

B Detailed transfer-gadget verifications

B.1 Pooled-star scalar accounting

We verify the two scalar values used in Theorem 4.1, retaining its notation. For the lower palette, query each center while predicting p and then predict its observed label on every leaf in that star. Let ℓ_j and E_j denote the number of mistakes and the cut cost in star j. If the center is p, the $n - a$ outside-block mistakes are cheapest; after they are exhausted, each further mistake costs two additional units. If the center is another old label, there is one center mistake and the corresponding

cheap block has size $n-b$. In the two cases, put respectively $(s,\ell)=(a,\ell_j)$ and $(s,\ell)=(b,\ell_j-1)$. Direct minimization gives

$$E_j \geq (n-s)+(W-1)\ell+2\big(\ell-(n-s)\big)_+.$$

Since $b=a+W+1$, both cases imply (9). Therefore $B+1=N-2t+1$ mistakes would force cut cost at least

$$(W+1)(N-2t+1)-N+at=\Gamma+W+1,$$

contradicting feasibility. The product $\prod_j R_j$ supplies the matching lower bound, so $m_r=B$.

For the upper palette, first fix every center to the fresh label f and let every leaf choose between f and its role label. If F leaves use f, the cut cost is $WN-(W-1)F$. Hence the budget contains the tail $F\geq t$. Predicting f makes at most $N-t$ leaf mistakes, whereas fixing any t leaves to f leaves a full $(N-t)$-cube. This proves the lower bound $C=N-t$.

For the reverse bound, query each center predicting f and copy its response on the leaves. Put $d_j=n-\ell_j$ and $\sigma_j=Wn-E_j$. A fresh center gives

$$\sigma_j\leq(W-1)d_j.$$

For an old center whose matching block has size $s\in\{a,b\}$, whenever $d_j<s-1$ direct minimization gives

$$E_j\geq W(n-s)+(W+1)(s-d_j-1),\qquad \sigma_j\leq(W+1)d_j+W+1-s\leq(W+1)d_j-2.$$

Suppose at least $C+1$ mistakes occur and put $D_0=\sum_j d_j\leq t-1$. Since every $d_j\geq-1$,

$$d_j\leq D_0+t-1\leq 2t-2=W-2<a-1,$$

so the old-center estimate applies in every factor. Let P be the set of old-center stars and put $D_P=\sum_{j\in P}d_j$. Fresh-center defects are nonnegative, hence $D_P\leq D_0$. If $P=\varnothing$, then

$$\sum_j\sigma_j\leq(W-1)D_0<t(W-1).$$

If $P\neq\varnothing$, then $D_P-|P|\leq D_0-1\leq t-2$, and

$$\begin{aligned}\sum_j\sigma_j&\leq(W-1)D_0+2(D_P-|P|)\\&\leq(W-1)(t-1)+2(t-2)\\&=t(W-1)-3.\end{aligned}$$

In either case the total cut cost is greater than $WN-t(W-1)=\Gamma$, a contradiction. Thus $m_{r+1}=C$.

We also record the residual fact used in the optimizer argument. Fix $v\in B_{j,p}$ and the response $v=p$. For $0\leq k\leq n-a$, let S_k be the resulting $(R_j\cup U_j)$-section after k outside-block leaves have additionally been fixed to their role labels. Then

$$v(S_k)=n-1-k.$$

For the upper bound, query x_j: its p- and f-sections are the corresponding R_j- and U_j-sections, each of value $n-2-k$. For the lower bound, induct downward on the number of unresolved outside-block

leaves. At such a leaf, answer a non-f prediction by f, leaving a full Boolean cube of dimension $n-2-k$, or answer prediction f by its role label, leaving S_{k+1}. At x_j, choose p or f different from the prediction, again leaving value $n-2-k$. At another leaf of $B_{j,p}$, answer prediction f by p; the R_j-section contains a full $(n-2-k)$-cube because two fixed p coordinates make its radius constraint vacuous. For any other prediction answer f; the U_j-section is then an unpunctured cube of the same dimension. These responses also prove the base case $k = n-a$. Every action therefore has a wrong child of value $n-2-k$, completing the induction.

B.2 The two-role release used for the binary palette

Put $\tau = D+1$ and use the fixed-role core from Theorem 4.1 with two prescribed distinct roles and $t = \tau$. Thus

$$W = 2\tau, \quad a = W+3, \quad b = 2W+4, \quad n = a+b = 3W+7, \quad N = \tau n,$$

the core budget is $\Gamma = WN - \tau(W-1)$, and its two values are

$$B = N - 2\tau, \qquad C = N - \tau.$$

The preceding conflict invariant excludes a common policy with loss caps $(C-1, C)$.

Put $d = 4$, $K = 6\tau + 10$, and $T = K - d = 6\tau + 6$. Choose an integer $L > \Gamma$, add K selector vertices, join each selector to both roles by edges of weight L, and raise the budget to $\Gamma + KL$. Distinct roles force every selector to use one of their labels, so these selectors are independent binary coordinates. If instead the roles coincide and e selectors differ from their common label, then, writing E for the core cut cost, feasibility gives

$$E + 2eL \le \Gamma + KL.$$

Since $E \ge 0$, $L > \Gamma$, and $2e - K$ is integral, this implies $2e \le K$. Thus at most $K/2$ selectors differ from the common label.

In the binary game, query the first role and query the second while predicting the other binary label. Distinct roles are oriented with at most one mistake; then use the B-mistake core policy and arbitrary selector predictions. If the roles coincide, even charging every role and core query as a mistake gives

$$2 + (N+\tau) + K/2 = N + 4\tau + 7 < B + K + 1.$$

Conversely, there is a literal disjoint-block Boolean cube of dimension $B+K+1$. In every star, use all leaves outside $B_{j,p}$ and any $a-2$ leaves of $B_{j,p}$ as singleton binary coordinates; these give the $B = N - 2\tau$ core coordinates. Put the two roles, all centers, and the two remaining $B_{j,p}$ leaves in one block whose two patterns globally swap the role labels. Finally, use each selector as a singleton coordinate. For either orientation, every assignment to the singleton coordinates lies in $\prod_j R_j$, and every selector uses one of its two endpoint labels, so the whole cube is feasible. Hence

$$m_2 = B + K + 1. \tag{20}$$

For the ternary upper bound, designate four selectors for discovery. Query the first arbitrarily, call its response λ, and predict λ on the other three. If a second label μ appears, then in a distinct-role section $\{\lambda, \mu\}$ is precisely the unordered pair of role labels. If only λ appears, the scan costs at most one mistake; query the first role predicting λ. If its response differs, the pair is known; otherwise query the second role with the same prediction. On a distinct-role branch these queries cost at most one further mistake. Three discovery labels signal a collision. If a pair has

been inferred but not oriented, postpone role verification until after the orientation-free core play below. In every case verify both roles before querying any reserve selector.

It remains to show that orienting a distinct pair fits inside the core allowance. Run the fresh-center policy, let Q be the number of leaf–center disagreements, let P be the number of old-labeled centers, and let F be the number of fresh-labeled leaves. If E, E' are the core cut costs under the two possible role orientations, then

$$E + E' = 2WQ + N + F.$$

If the core makes at least $C = N - \tau$ mistakes, then $P \le \tau$ and $Q \ge N - 2\tau$. The two orientations cannot both be feasible, since

$$2W(N - 2\tau) + N - 2\Gamma = N - 2\tau(W + 1) = \tau(W + 5) > 0.$$

Thus orientation ambiguity leaves the one mistake needed to query a role. If verification instead finds equal roles, discovery, the roles, the entire core, and all exceptional selectors cost at most

$$d + 2 + (N + \tau) + K/2 = N + 4\tau + 11 = B + K + 1 \le C + K.$$

The resulting ternary policy has loss at most $C + K$. Fixing distinct roles, using the value-C core subclass, and varying all selectors independently gives the reverse inequality, so

$$m_3 = C + K. \tag{21}$$

Finally suppose a ternary optimum had binary loss at most $C + K - 1$. Restrict to a canonical distinct-role section and, whenever a selector is queried, return the endpoint label different from the prediction. The selector cube makes all K responses jointly feasible and forces K mistakes. Suppress role and selector queries. The induced fixed-role core policy has caps $(C - 1, C)$, a contradiction. Combining (20) and (21) gives the asserted scalar gap and transfer loss.

B.3 Separated-palette construction accounting

Retain the notation of Lemma 4.3: the old labels are $1, \dots, r$, with $u = 1$ and $p = 2$; anchor A_i has label i; and the role vertices are U_i for $i \ne p$. In phase I, every U_i has its private label i; in phase C, every U_i has label p. Put

$$L = 10n, \qquad W = 2n + 2, \qquad H = 100n.$$

Create blocks $B_1, \dots, B_r$, with $|B_1| = 5$ and every other block of size seven. Let $R_2 = A_2$ and $R_i = U_i$ for $i \ne 2$. Add a center x, a shared tag y, and ordinary tags $z_1, \dots, z_{n-1}$, with edges

$$\begin{aligned} w(xv) &= L + 1, & w(vR_i) &= L & &(v \in B_i),\\ w(xy) &= W, & w(U_1 y) &= 1, & w(yA_1) &= W - 1,\\ w(U_1 z_j) &= H + 2, & w(z_j A_1) &= H & &(1 \le j < n). \end{aligned}$$

The blocks contain $5 + 7(r - 1) = n$ leaves. In phase I, every B_i is therefore attached to label i, and $U_1 = A_1 = u$. In phase C, all leaves are attached to p, while $U_1 = p$ and $A_1 = u$. The shared tag y and the $n - 1$ ordinary tags are the n tag coordinates used below.

The role-edge system constructed at the end of this subsection will leave the payload allowance

$$T_I = n(L + 1) - 1 + W = 10n^2 + 3n + 1$$

in phase I, and the allowance

$$T_C = (W-1) + (n-1)H + 2(n-1) = 100n^2 - 96n - 1$$

in phase C.

We first verify the phase-I profile. Because $U_1 = A_1 = u$, changing any ordinary tag away from u adds exactly $2H+2$. To see that the budget never permits this change, minimize the cost of the star together with the shared tag for each possible type of center label:

center label	minimum cost	remaining slack
$x = u$	$L(n-5)$	$5L + n - 1 + W$
$x = i \neq u$ old	$L(n-7) + W$	$7L + n - 1$
x fresh	$nL + W$	$n - 1$.

The slack is always at most $7L + n - 1 < 2H + 2$. Hence every ordinary tag is forced to u.

Now fix an old center label $x = i$, fix the shared-tag label y, and write $t_i = |B_i|$. Let d count the leaves of B_i not labeled i. Among leaves outside B_i, let e use their attachment label and let c use neither their attachment label nor i; all remaining outside leaves use i. After subtracting the base star cost $L(n - t_i)$ and the cost of the shared tag, feasibility is

$$e + (L+1)c + (2L+1)d \leq t_i L + n - 1 + \delta, \tag{22}$$

where the four cases are

$$\delta = \begin{cases} W, & i = u \text{ and } y = u, \\ -W, & i = u \text{ and } y \neq u, \\ 0, & i \neq u \text{ and } y \in \{u, i\}, \\ -W, & i \neq u \text{ and } y \notin \{u, i\}. \end{cases}$$

Predict i on every leaf. Replacing a c-type leaf by its attachment label saves L without changing whether that prediction is wrong, so a loss-maximizing feasible assignment may be taken with $c = 0$. All $n - t_i$ outside leaves can then use their attachment labels. The right side of (22) is large enough to take $e = n - t_i$. Moreover, $n - t_i < 2L + 1$, so replacing one costly d-deviation, when necessary, by all missing cheap outside deviations never decreases the number of mistakes. There is therefore a maximum with $e = n - t_i$. Put

$$k_i = \left\lfloor \frac{t_i(L+1) - 1 + \delta}{2L+1} \right\rfloor.$$

The fixed-(x, y) value is at most $n - t_i + k_i$. This bound is exact: let all outside leaves vary independently between i and their attachment labels, and let any fixed set of k_i leaves in B_i vary independently between i and one fixed alternative label. For $t_i = 5$, we have $k_i = 2$ for both possible values of δ; for $t_i = 7$, we have $k_i = 3$ for $\delta \in \{0, -W\}$. Thus the fixed-(x, y) value is $n - 3$ when $x = u$ and $n - 4$ for every other old center.

For $x = u$, the same $(n-3)$-cube survives for $y = u$ and for one alternative tag label. For $x = i \neq u$, the same $(n-4)$-cube survives for $y = u$ and $y = i$. These two choices of y add one independent Boolean coordinate; querying y gives the matching upper bound. Hence, when y is not fixed,

observed center label	u	$i \neq u$
residual value	$n - 2$	$n - 3$.

Querying x and predicting u now gives the phase-I lower-palette upper bound $n - 2$, and the $x = u$ section gives equality. Thus the phase-I lower-palette value is $a = n - 2$.

If $x = f$ is fresh, let t leaves equal f and let h use neither f nor their attachment label. Their star cut cost is $n(L+1) - t + Lh$. After reserving the minimum shared-tag cost W, feasibility forces $h = 0$ and $t \geq 1$. These conditions are sufficient when $y \in \{u, f\}$, so the leaves form a punctured Boolean n-cube of value $n-1$. The shared tag can have only label u or f: a third label raises its cost from W to $2W$, but $W > n-1$. For either allowed tag label the leaf class is the same, so the shared tag is an independent Boolean coordinate. The fresh-center section consequently has value n.

For the response sections used in the conflict table, put $\Lambda = T_I - W = n(L+1) - 1$. If $y = f$ is fixed fresh, then $x = f$ leaves the punctured Boolean n-cube. Any other fresh center has tag cost $2W$ and therefore leaves star allowance

$$\Lambda - W = nL - n - 3 < nL,$$

below the least possible fresh-center star cost. The old-center formula bounds all remaining children. Querying x while predicting f gives the upper bound $n-1$, and the $x = f$ child gives equality. Hence the whole $y = f$ section has value $n-1$.

Now fix $y = u$ and suppose $s \geq r+2$, so at least two fresh labels are available. Forced ordinary tags carry no information and may be ignored. At a center query, return a fresh label different from the prediction and retain a punctured Boolean n-cube on the leaves. At a leaf query, return such a fresh label, give x that label, and retain a full cube on the other $n-1$ leaves. Either response causes a mistake and leaves residual value $n-1$. This proves a lower bound of n; querying x and using the fixed-center bounds gives the reverse inequality. Thus the $y = u$ section has value n. For any other old value of y, fresh centers are infeasible and the preceding old-center calculation bounds the section by $n-3$. Querying y while predicting u therefore proves that the phase-I upper-palette value is $c = n$.

Finally, fix a fresh response at a leaf v. Setting $x = f$ leaves a full cube on the other $n-1$ leaves, while y remains the independent choice between u and f. This section has value n, as required for the block-vertex row of (13); other center labels are bounded as above.

We next verify phase C. With x, every leaf, and every tag labeled p, the tag cost is

$$B_{\text{tag}} = (W-1) + (n-1)H, \qquad T_C = B_{\text{tag}} + 2(n-1).$$

If $x = p$, changing one leaf away from p adds $2L+1 > 2(n-1)$, so no leaf can change. If $x \neq p$, the star costs at least nL; changing the shared tag can save at most $W-2$ from B_{tag}, and

$$nL - (W-2) > 2(n-1).$$

Thus x and every leaf are forced to p. Relative to B_{tag}, the shared tag has increments $0, 2, W+1$ at p, u, and any third label, respectively; an ordinary tag has increments $0, 2, H+2$. Consequently the feasible tag words are exactly $\{p, u\}^n \setminus \{u^n\}$. This punctured Boolean cube has value $b = n-1$. Fixing one tag to p leaves a full $(n-1)$-cube, still of value b, whereas fixing it to u leaves a punctured $(n-1)$-cube of value $n-2$. This proves the phase profile and every phase-C entry in (13).

It remains to isolate the phases. Put $m = r-1$ and let

$$E_* = n(2L+1) + 2W + (n-1)(2H+2)$$

be the total payload edge mass. Define

$$\begin{aligned} K &= 2(T_C + 1), \\ B_0 &= K(m-1) + E_* + T_C + (T_C - T_I) + 1, \\ A_0 &= B_0 + \frac{K(m-1)}{2} + \frac{T_I - T_C}{m}. \end{aligned}$$

Here $n = 7r - 2 \geq 19$, so

$$T_C - T_I = 90n^2 - 99n - 2 > 0.$$

For every $i \neq 2$, add U_iA_i with weight A_0 and U_iA_2 with weight B_0, and join every pair of role vertices by an edge of weight K. Set

$$S_I = mB_0 + K\binom{m}{2}, \qquad S_C = mA_0, \qquad \Theta = S_I + T_I = S_C + T_C.$$

These are positive rational weights. In particular, $B_0 > 0$, and, because $K/2 = T_C + 1$,

$$m(A_0 - B_0) = T_C\big(m(m-1) - 1\big) + T_I + m(m-1) > 0.$$

Only A_0 can be fractional. All calculations below use this rational normalization. After the verification, multiplying every edge weight, Θ, and hence the two payload allowances by $2m$ clears all possible denominators and gives positive integers without changing the feasible class. A coherent assignment with j private roles has role cut cost

$$\begin{aligned} S_j &= jB_0 + (m-j)A_0 + K\left[\binom{j}{2} + j(m-j)\right] \\ &= \frac{j}{m}S_I + \left(1 - \frac{j}{m}\right)S_C + \frac{K}{2}j(m-j). \end{aligned}$$

Consequently

$$\Theta - S_j = \frac{j}{m}T_I + \left(1 - \frac{j}{m}\right)T_C - \frac{K}{2}j(m-j) < 0 \qquad (0 < j < m).$$

Indeed, the convex combination of T_I and T_C is at most T_C, whereas

$$\frac{K}{2}j(m-j) = (T_C + 1)j(m-j) \geq T_C + 1.$$

Thus $\Theta - S_j < 0$ whenever $0 < j < m$, and every mixed coherent phase is infeasible.

It remains to exclude assignments in which some role uses neither its private label nor p. Let J be the set of such roles and put $k = |J| \geq 1$. Relabel every U_i with $i \in J$ as p, leaving all payload labels fixed, and call the resulting labeling $\widehat{h}$. Each replacement removes the B_0 contribution of U_iA_2, leaves U_iA_i cut, and can add at most $K(m-1)$ on role–role edges. Writing C_{role} for the contribution of the anchor and role–role edges, we obtain

$$C_{\text{role}}(h) - C_{\text{role}}(\widehat{h}) \geq k\big(B_0 - K(m-1)\big).$$

Changing the role labels can alter the payload cost by at most its total edge mass E_*. The roles of $\widehat{h}$ are coherent and include at least one role labeled p. Their role cost is therefore at least $S_C = \Theta - T_C$: phase C attains this value, while every mixed coherent phase already exceeds Θ. It follows that

$$\begin{aligned} C_G(h) &\geq S_C + k\big(B_0 - K(m-1)\big) - E_* \\ &\geq \Theta - T_C + \big(E_* + T_C + (T_C - T_I) + 1\big) - E_* \\ &= \Theta + (T_C - T_I) + 1 > \Theta. \end{aligned}$$

Thus every noncoherent role assignment is infeasible. The feasible class is exactly the union of phase I with allowance T_I and phase C with allowance T_C, completing the technical verification.

B.4 Exact exhaustive computation for the universal binary construction

The graph $J_\star$ in Lemma 4.5 consists of two copies of one petal sharing only the vertex o, with total budget 200000. A petal has vertices o, t, a, b, c, d, e and the following weighted edges; omitted pairs are nonedges.

edge	weight	edge	weight	edge	weight
oa	39998	oc	28572	od	28572
oe	5714	ta	34285	tb	48571
tc	2856	te	17144	ab	20001
ae	8572	bc	48572	bd	82858
cd	14284	ce	8572	de	74288

Fix the label of o and denote it abstractly by digit 0. In vertex order (t, a, b, c, d, e), the feasible equality partitions of one petal have representatives

$$\begin{aligned}\mathcal{B} = \{&000000, 000001, 000011, 000100, 001111, 010000, 100000,\\ &100001, 100100, 101011, 101111, 110000, 110001,\\ &111000, 111011, 111100, 111111\},\\ \mathcal{T} = \{&101211, 102222, 111211, 112222, 120000, 121111\}.\end{aligned}$$

Here $\mathcal{B}$ is the complete list of feasible binary representatives. For $q \in \{2, 3, 4\}$, replace the nonzero digit in each nonconstant pattern of $\mathcal{B}$ by any of the $q-1$ nonzero labels, and replace digits $1, 2$ in each pattern of $\mathcal{T}$ by any ordered pair of distinct nonzero labels. Together with the constant pattern, these are all feasible labelings.

This list is checked without numerical approximation: generate all set partitions of the seven named vertices, and for each partition sum exactly the printed integer edge weights crossing its blocks. Precisely these twenty-three partitions have cost at most 200000; the smallest omitted cost is 200002, and the smallest retained nonconstant cost is 102856. If $\mathcal{P}_q$ denotes the one-petal class conditioned on the label of o, its sizes are consequently $17, 45, 85$ at palettes two, three, and four.

For completeness, the Bellman computation can be reproduced directly. A state is a pair (S, R), where S is the subset of listed labelings consistent with the history and R is the set of unqueried coordinates. With $V(S, \varnothing) = 0$, evaluate for increasing $|R|$

$$V(S, R) = \min_{x \in R} \min_{p \in [q]} \max_{z:\, S_{x=z} \neq \varnothing} \big(\mathbf{1}\{p \neq z\} + V(S_{x=z}, R \setminus \{x\})\big).$$

This exact integer recurrence gives

q	$\lvert\mathcal{P}_q\rvert$	$v(\mathcal{P}_q)$	t-section values	a, b, c, d, e section values
2	17	2	$(1, 2)$	$(2, 2)$
3	45	3	$(1, 3, 3)$	$(2, 2, 2)$
4	85	3	$(1, 3, 3, 3)$	$(2, 2, 2, 2)$

where each tuple is ordered by the possible response label and the last column applies to each of the five listed query vertices.

Since two nonconstant petals cost more than the global budget, a feasible target is constant or has exactly one active petal. The full class therefore has size $q(1 + 2(|\mathcal{P}_q| - 1))$. Applying the same recurrence to these thirteen-coordinate classes gives

q	$\lvert\mathcal{H}^{(q)}_{200000}(J_\star)\rvert$	$m_q(J_\star, 200000)$	b_i, c_i, d_i, e_i sections	all other live sections
2	66	3	$(2, 2)$	$(3, 3)$
3	267	4	$(3, 3, 3)$	$(3, 3, 3)$
4	676	4	$(3, 3, 3, 3)$	$(3, 3, 3, 3)$

for either petal $i \in \{1, 2\}$. Thus the graph, feasible partition list, recurrence, and both exact output tables are printed here.

The certificate is archived on Zenodo as version v1.0.3 (Zhou, 2026); the reference also links its GitHub repository. The script requires Python 3.10 or later, uses only the standard library, and performs exact integer arithmetic.

The script checks the displayed binary and ternary representative lists, verifies the three cut cost extrema and every entry of both profile tables by the displayed Bellman recurrence, and independently checks all four common-policy incompatibilities. In those coupled checks, each response subtracts the immediate mistake from both remaining loss caps; the caps are carried into the recursive call, not reset to the scalar values of the residual classes.

After downloading and unpacking the archived release, reproduce these checks by running the following command from the directory containing the script:

```
python3 binary_nonadjacent_certificate.py
```

This exhaustive computation is used only in Lemma 4.5. Among the later headline results, only the case $r = 2$ and $s \geq 4$ of Theorem 4.6 depends on that lemma.

C The transfer-spectrum seed

This appendix supplies the fixed-role calculation used in Theorem 4.2. The role-release and unweighting steps are already proved in Lemma 3.4 and Proposition 3.3.

Proposition C.1 (Fixed-role transfer-spectrum seed)**.** *Fix integers $r \geq 3$ and $g \geq 1$. Prescribe distinct labels $1, \dots, r$ on role vertices $a_1, \dots, a_r$, and distinguish $p \in [r]$. Add a live center x and disjoint leaf blocks $B_1, \dots, B_r$ with*

$$|B_p| = 2g + 1, \qquad |B_i| = 2g + 3 \quad (i \neq p), \qquad n = r(2g + 3) - 2.$$

For each $v \in B_i$, give xv weight $n + 1$ and va_i weight n, and use budget $\Gamma = n(n + 1) - 1$. Let $X = \{x\} \dot\cup \bigcup_i B_i$, and let $\mathcal{C}_k$ be the trace on X of the k-label budget class under the prescribed roles. Thus $|X| = n + 1$. Put $b = n - g - 1$. Then

$$v(\mathcal{C}_r) = b, \qquad v(\mathcal{C}_{r+1}) = n - 1, \qquad v(\mathcal{C}_s) = n \quad (s \geq r + 2).$$

If $L_k^{\mathcal{C}}(A) = \max_{h \in \mathcal{C}_k} M(A, h)$, then

$$\{L_r^{\mathcal{C}}(A) : L_{r+1}^{\mathcal{C}}(A) = n - 1\} = \{b + 1, b + 2, \dots, n - 1\},$$
$$\{L_r^{\mathcal{C}}(A) : L_s^{\mathcal{C}}(A) = n\} = \{b, b + 1, \dots, n\} \qquad (s \geq r + 2).$$

Proof. We first compute every center section. Fix an old center response $x = i$ and write $t_i = |B_i|$. Relative to the all-i assignment on the leaves, let b_0 count vertices outside B_i that retain their role label, let c_0 count those using neither i nor their role label, and let d count non-i vertices in B_i. Direct summation of the two incident edge costs shows that feasibility is equivalent to

$$b_0 + (n + 1)c_0 + (2n + 1)d \leq n(t_i + 1) - 1. \tag{23}$$

Predicting i on every leaf makes $b_0 + c_0 + d$ mistakes. The inequality implies $b_0 + c_0 \leq n - t_i$ and $d \leq \lfloor t_i/2 \rfloor$, so the section value is at most

$$n - t_i + \left\lfloor \frac{t_i}{2} \right\rfloor = n - \left\lceil \frac{t_i}{2} \right\rceil.$$

For equality, let every outside-block vertex independently choose between i and its role label, and let any fixed $\lfloor t_i/2 \rfloor$ vertices in B_i independently choose between i and a second old label. Equation (23) makes this a feasible full cube of the displayed dimension. Hence the distinguished old section has value b, and every other old section has value $b-1$.

Let f be fresh and fix $x = f$. Starting from the all-role assignment on the leaves, suppose t leaves change to f and u leaves use neither f nor their role label. Their cut cost is

$$n(n+1) - t + nu.$$

Thus feasibility is exactly $u = 0$ and $t \geq 1$. The section is the punctured Boolean n-cube and has value $n-1$. Querying x and predicting p proves $v(\mathcal{C}_r) = b$; predicting f proves $v(\mathcal{C}_{r+1}) = n-1$, and the corresponding center sections give the matching lower bounds.

If $s \geq r+2$, every first action has a wrong-response child of value $n-1$. At x, choose a fresh response different from the prediction and retain the punctured Boolean n-cube. At a leaf, return such a fresh label, give x the same label, and let all other $n-1$ leaves vary independently between that label and their role labels. Conversely, query x while predicting a fresh label. Every fresh-response section has value $n-1$, and every old section has value at most b. Therefore $v(\mathcal{C}_s) = n$.

We next exclude restricted loss b from an adjacent-palette optimum. Put $O = \bigcup_{i \neq p} B_i$. After t queried vertices of O have returned their role labels, each response having caused a mistake, the remaining lower and upper loss allowances are $b-t$ and $n-1-t$. Retain two classes. For each choice of g vertices of B_p, the lower class contains the full cube obtained by fixing $x = p$, varying every remaining vertex of O between p and its role label, and varying the chosen vertices of B_p between p and a second old label, with all other vertices of B_p fixed to p. Each such cube L_t has value $b-t$; the actual lower version space contains all these choices. In the upper class, fix $x = f$, let every remaining block vertex choose between f and its role label, and remove the all-role assignment. This is a punctured Boolean cube U_t of value $n-1-t$.

Suppose one policy were to meet the two caps $(b, n-1)$. An old-label prediction can be answered by f without lowering the upper residual value. Predicting f at x can be answered by p without lowering the lower value, and the same is true at a vertex of B_p: choose a retained cube whose g varying coordinates omit that vertex, and return p. No vertex of B_p has yet been queried, so this choice is available. Each response is a mistake and therefore violates the corresponding tight cap. The only temporarily viable action is to query an unqueried $v \in B_i$, $i \neq p$, and predict f. Return i; this mistake lowers both cube values by one and recreates the same state with t replaced by $t+1$. Once O is exhausted, no viable action remains. Thus no adjacent upper optimum has restricted loss b.

It remains to realize every other loss. For $0 \leq j \leq g-1$, query x and predict f. In the branch $x = p$, query j prescribed vertices of B_p, predicting f each time, and then continue optimally; use optimal section policies in every other branch. The p-branch costs at most $1+j+b \leq n-1$, and the fresh branch has value $n-1$, so the policy is $(r+1)$-optimal. On an r-label target, its loss is exactly $b+1+j$: answer p on the j sacrificial queries and retain the b-cube from the distinguished old section. There are enough untouched vertices because $|B_p| - j \geq g+2$. The general scalar bounds and the preceding incompatibility exclude every other restricted loss, proving the first spectrum.

Finally let $s \geq r+2$. Querying x and predicting p, followed by an optimal section policy, is s-optimal and remains r-optimal, giving loss b. Alternatively, predict a fresh label at x and make $j = 0, \ldots, g$ sacrifices in the p-branch. A different fresh response at x costs one mistake and leaves a section of value $n-1$, so every such policy remains s-optimal. The same retained cube makes its exact restricted loss $b+1+j$; now $|B_p| - j \geq g+1$. These policies fill every integer through n, and scalar optimality excludes all others. □

For reference, here $|X| = n+1$, $R = r$, and $b = n-g-1$, so Lemma 3.4 uses

$$T = 4 + 2(|X| + R - b + 1) = 2r + 2g + 10, \qquad M = (4+T)\binom{r}{2} = (2r+2g+14)\binom{r}{2}$$

midpoint vertices on this core. It translates every scalar and restricted loss in the proposition by M before the simple-unweighted realization.

D Proof of the agnostic path-completion lemma

Proof of Lemma 6.7. We begin with a simple strategy used repeatedly. If a path has distinct endpoint labels $c \neq d$ and m internal vertices, answer every query with a member of $\{c, d\}$ different from the prediction. Choose $a_0 \in \{c, d\}$ that occurs most often among the internal responses (arbitrarily if $m = 0$), and let h assign a_0 to every internal vertex. Then $d(y, h) \leq \lfloor m/2 \rfloor$, and h has exactly one bichromatic edge.

We next prove the two critical lengths. Let $\ell = 2a + 2$. If both endpoints have label c, fix $d \neq c$ and answer from $\{c, d\}$, always incorrectly. If at most a responses are d, use c throughout the comparator; if at most a are c, use d throughout the interior. Otherwise each response count is $a + 1$; use c internally except at one position whose response is d. In every case the comparator differs from the response word in at most a positions and has at most two bichromatic edges.

Now suppose the endpoint labels are $c \neq d$, and choose $e \notin \{c, d\}$. At the first query, let L, R be the numbers of remaining internal vertices on its two sides, so $L + R = 2a + 1$. If the prediction is not e, answer e and use the distinct-endpoint strategy on both sides. Since L and R have opposite parity, the number of Hamming disagreements is

$$\left\lfloor \frac{L}{2} \right\rfloor + \left\lfloor \frac{R}{2} \right\rfloor = a,$$

and there are two bichromatic edges. If the prediction is e, suppose $L = 2u$ and $R = 2v + 1$, and answer c. Answer later on the left from $\{c, e\}$ and on the right from $\{e, d\}$. Let $z \in \{e, d\}$ be a majority response on the right. If the left has at most u responses e, use comparator label c on the left and at the first queried vertex, and z on the right. Otherwise the left has at most $u - 1$ responses c; use e on the left and at the first queried vertex, paying one Hamming disagreement there, and use z on the right. Both choices use at most $u + v = a$ Hamming disagreements and two bichromatic edges. The case in which L is odd is symmetric.

Let next $\ell = 2a + 3$. With distinct endpoint labels, answer from those two labels. If the minority response count is at most a, use the majority label throughout the interior. Otherwise the counts are $a + 1$ and $a + 2$; use the majority label except at one minority occurrence. This uses at most a Hamming disagreements and creates at most three bichromatic edges. With equal endpoints c, c, answer the first query with a label z different from both c and the prediction. The two side lengths have the same parity. If both are odd, apply the distinct-endpoint strategy on both. If they are $2u, 2v$, where $u + v = a + 1$, choose a side with positive half-length, say the $2v$ side. Use the elementary distinct-endpoint strategy on the $2u$ side, at distance at most u and with one bichromatic edge, and use the $\ell = 2(v - 1) + 2$ critical construction on the $2v$ side, at distance at most $v - 1$ and with at most two bichromatic edges. If $u = 0$, the first side contributes only its boundary edge. The total distance is at most a and there are at most three bichromatic edges.

We finish by strong induction in the following strengthened form. For a positive subpath with m internal vertices and endpoint labels c, d, and for every integer $0 \leq a' \leq K(m; c, d)$, there is an

all-wrong strategy whose comparator is within distance a' and has at most $m - 2a'$ bichromatic edges, where

$$K(m; c, d) = \begin{cases} \lfloor (m-1)/2 \rfloor, & c \neq d, \\ \lfloor (m-2)/2 \rfloor, & c = d \end{cases}.$$

The elementary distinct-endpoint strategy covers $m = 2a' + 1$ when the endpoints differ. The two critical constructions cover $m = 2a' + 2$ and $m = 2a' + 3$ for either relation between the endpoint labels. These include every admissible case with $m \leq 2a' + 3$; the remaining cases have $m \geq 2a' + 4$ and follow by the induction step below. Assume $\ell \geq 2a + 4$, and first suppose that the learner's first query leaves positive side lengths L, R. Write c, d for the outer endpoint labels. For a proposed wrong response z, used also as the comparator label at the queried vertex, the total available allowance on the two sides is

$$B - \mathbf{1}\{L \text{ is odd},\ z = c\} - \mathbf{1}\{R \text{ is odd},\ z = d\},$$

where

$$B = \left\lfloor \frac{L-1}{2} \right\rfloor + \left\lfloor \frac{R-1}{2} \right\rfloor.$$

If L, R have opposite parity, then $B = \ell/2 - 2 \geq a$; choose a response different from the prediction and from the outer endpoint of the odd side. If both are even, then $B = (\ell - 5)/2 \geq a$ and there is no penalty. If both are odd, then $B = (\ell - 3)/2 \geq a + 1$. For equal outer endpoint labels, avoid that label and the prediction. For distinct outer labels, any wrong response incurs at most one penalty; if a side has length one, also avoid its outer endpoint. At most two labels are excluded in every case, so $q \geq 3$ makes the choice possible.

Choose nonnegative allowances a_L, a_R with $a_L + a_R = a$ within the two available bounds, and run the two inductive strategies, interleaving them as the learner requires. Their distances add to at most a, and their numbers of bichromatic edges add to at most

$$(L - 2a_L) + (R - 2a_R) = \ell - 1 - 2a \leq \ell - 2a.$$

If the first query leaves one side empty, give any wrong response, count its boundary edge, and apply induction with allowance a to the other side, whose length is $\ell - 1 \geq 2a + 3$. The total is at most $1 + (\ell - 1 - 2a) = \ell - 2a$. This completes the induction. Since $\ell \geq 2a + 2$ implies $a \leq K(\ell; c, d)$ for either relation between the endpoint labels, the strengthened statement proves Lemma 6.7. □

References

Peyman Afshani, Ehsan Chiniforooshan, Reza Dorrigiv, Arash Farzan, Mehdi Mirzazadeh, Narges Simjour, and Hamid Zarrabi-Zadeh. On the complexity of finding an unknown cut via vertex queries. In *Computing and Combinatorics*, volume 4598 of *Lecture Notes in Computer Science*, pages 459–469. Springer, 2007. doi: 10.1007/978-3-540-73545-8_45. URL https://doi.org/10.1007/978-3-540-73545-8_45.

Mykhaylo Andriluka, Jasper R. R. Uijlings, and Vittorio Ferrari. Fluid annotation: A human–machine collaboration interface for full image annotation. In *Proceedings of the 26th ACM International Conference on Multimedia*, pages 1957–1966. ACM, 2018. doi: 10.1145/3240508.3241916. URL https://doi.org/10.1145/3240508.3241916.

Shai Ben-David and Nadav Eiron. Self-directed learning and its relation to the VC-dimension and to teacher-directed learning. *Machine Learning*, 33(1):87–104, 1998. doi: 10.1023/A:1007510732151. URL https://doi.org/10.1023/A:1007510732151.

Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *Proceedings of the 22nd Conference on Learning Theory*, 2009. URL https://www.learningtheory.org/colt2009/papers/032.pdf.

Avrim Blum and Shuchi Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 19–26. Morgan Kaufmann, 2001. URL https://www.cs.cmu.edu/~avrim/Papers/mincut.pdf.

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. doi: 10.1109/34.969114. URL https://doi.org/10.1109/34.969114.

Yuri Y. Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 105–112, 2001. doi: 10.1109/ICCV.2001.937505. URL https://doi.org/10.1109/ICCV.2001.937505.

Mickey Brautbar. Online learning a binary labeling of a graph. In *Proceedings of the Joint 7th Workshop on Mining and Learning with Graphs and the 6th International Workshop on Statistical Relational Learning, MLG/SRL@ILP 2009*, Leuven, Belgium, 2009. URL https://dtai.cs.kuleuven.be/events/ilp-mlg-srl/USBStick/papers/MLG09-6.pdf.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. ISBN 978-0-521-84108-5. doi: 10.1017/CBO9780511546921. URL https://doi.org/10.1017/CBO9780511546921.

Nicolò Cesa-Bianchi, Claudio Gentile, and Fabio Vitale. Fast and optimal prediction on a labeled tree. In *Proceedings of the 22nd Conference on Learning Theory*, pages 145–156, 2009. URL https://www.learningtheory.org/colt2009/papers/015.pdf.

Nicolò Cesa-Bianchi, Claudio Gentile, Fabio Vitale, and Giovanni Zappella. Active learning on trees and graphs. In *Proceedings of the 23rd Conference on Learning Theory*, pages 320–332, 2010. URL https://www.learningtheory.org/colt2010/papers/67Vitale.pdf.

Nicolò Cesa-Bianchi, Claudio Gentile, and Fabio Vitale. Predicting the labels of an unknown graph via adaptive exploration. *Theoretical Computer Science*, 412(19):1791–1804, 2011. doi: 10.1016/j.tcs.2010.12.056. URL https://doi.org/10.1016/j.tcs.2010.12.056.

Nicolò Cesa-Bianchi, Claudio Gentile, Fabio Vitale, and Giovanni Zappella. Random spanning trees and the prediction of weighted graphs. *Journal of Machine Learning Research*, 14(38):1251–1284, 2013. URL https://www.jmlr.org/papers/v14/cesa-bianchi13a.html.

David A. Cohn, Les E. Atlas, and Richard E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994. doi: 10.1007/BF00993277. URL https://doi.org/10.1007/BF00993277.

Gautam Dasarathy, Robert Nowak, and Xiaojin Zhu. S2: An efficient graph based active learning algorithm with application to nonparametric classification. In *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 503–522. PMLR, 2015. URL https://proceedings.mlr.press/v40/Dasarathy15.html.

Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems 18*, pages 235–242, 2005. URL https://papers.nips.cc/paper_files/paper/2005/hash/6e82873a32b95af115de1c414a1849cb-Abstract.html.

Pramith Devulapalli and Steve Hanneke. The dimension of self-directed learning. In *Proceedings of The 35th International Conference on Algorithmic Learning Theory*, volume 237 of *Proceedings of Machine Learning Research*, pages 544–573. PMLR, 2024. URL https://proceedings.mlr.press/v237/devulapalli24a.html.

Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Self-directed linear classification. In *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 2919–2947. PMLR, 2023. URL https://proceedings.mlr.press/v195/diakonikolas23c.html.

Jittat Fakcharoenphol and Boonserm Kijsirikul. Low congestion online routing and an improved mistake bound for online prediction of graph labeling. *CoRR*, abs/0809.2075, 2008. doi: 10.48550/arXiv.0809.2075. URL https://arxiv.org/abs/0809.2075.

Dylan J. Foster, Satyen Kale, Mehryar Mohri, and Karthik Sridharan. Parameter-free online learning via model selection. In *Advances in Neural Information Processing Systems 30*, pages 6020–6030, 2017. URL https://papers.nips.cc/paper/7183-parameter-free-online-learning-via-model-selection.

Michael R. Garey, David S. Johnson, and Larry Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976. doi: 10.1016/0304-3975(76)90059-1. URL https://doi.org/10.1016/0304-3975(76)90059-1.

Sally A. Goldman and Robert H. Sloan. The power of self-directed learning. *Machine Learning*, 14(3):271–294, 1994. doi: 10.1007/BF00993977. URL https://doi.org/10.1007/BF00993977.

Sally A. Goldman, Ronald L. Rivest, and Robert E. Schapire. Learning binary relations and total orders. *SIAM Journal on Computing*, 22(5):1006–1034, 1993. doi: 10.1137/0222062. URL https://doi.org/10.1137/0222062.

Igor Gridchyn and Vladimir Kolmogorov. Potts model, parametric maxflow and k-submodular functions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2320–2327. IEEE, 2013. doi: 10.1109/ICCV.2013.288. URL https://doi.org/10.1109/ICCV.2013.288.

Andrew Guillory and Jeff A. Bilmes. Label selection on graphs. In *Advances in Neural Information Processing Systems 22*, pages 691–699, 2009. URL https://papers.nips.cc/paper_files/paper/2009/hash/90794e3b050f815354e3e29e977a88ab-Abstract.html.

Steve Hanneke. An analysis of graph cut size for transductive learning. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 393–399. ACM, 2006. doi: 10.1145/1143844.1143894. URL https://doi.org/10.1145/1143844.1143894.

Mark Herbster and Massimiliano Pontil. Prediction on a graph with a perceptron. In *Advances in Neural Information Processing Systems 19*, pages 577–584. MIT Press, 2006. URL https://papers.nips.cc/paper_files/paper/2006/hash/cc3d69ed781b16bce06687822ae56e6d-Abstract.html.

Mark Herbster, Massimiliano Pontil, and Lisa Wainer. Online learning over graphs. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 305–312. ACM, 2005. doi: 10.1145/1102351.1102390. URL https://doi.org/10.1145/1102351.1102390.

Mark Herbster, Stephen Pasteris, and Shaona Ghosh. Online prediction at the limit of zero temperature. In *Advances in Neural Information Processing Systems 28*, pages 2935–2943, 2015. URL https://papers.nips.cc/paper/5690-online-prediction-at-the-limit-of-zero-temperature.

Alexander Jung, Alfred O. Hero, III, Alexandru Cristian Mara, Saeed Jahromi, Ayelet Heimowitz, and Yonina C. Eldar. Semi-supervised learning in network-structured data via total variation minimization. *IEEE Transactions on Signal Processing*, 67(24):6256–6269, 2019. doi: 10.1109/TSP.2019.2953593. URL https://doi.org/10.1109/TSP.2019.2953593.

Jon Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *Journal of the ACM*, 49(5):616–639, 2002. doi: 10.1145/585265.585268. URL https://doi.org/10.1145/585265.585268.

Victor Lempitsky, Carsten Rother, Stefan Roth, and Andrew Blake. Fusion moves for Markov random field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1392–1405, 2010. doi: 10.1109/TPAMI.2009.143. URL https://doi.org/10.1109/TPAMI.2009.143.

Guy Lever. Relating function class complexity and cluster structure in the function domain with applications to transduction. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 437–444. PMLR, 2010. URL https://proceedings.mlr.press/v9/lever10a.html.

Oscar Hernan Madrid Padilla, James Sharpnack, James G. Scott, and Ryan J. Tibshirani. The DFS fused lasso: Linear-time denoising over general graphs. *Journal of Machine Learning Research*, 18 (176):1–36, 2018. URL https://www.jmlr.org/papers/v18/16-532.html.

Arya Mazumdar, Yury Polyanskiy, and Barna Saha. On Chebyshev radius of a set in Hamming space and the closest string problem. In *Proceedings of the IEEE International Symposium on Information Theory*, pages 1401–1405, 2013. doi: 10.1109/ISIT.2013.6620457. URL https://doi.org/10.1109/ISIT.2013.6620457.

Jean-Claude Picard and Maurice Queyranne. On the structure of all minimum cuts in a network and applications. In *Combinatorial Optimization II*, volume 13 of *Mathematical Programming Studies*, pages 8–16. Springer Berlin Heidelberg, 1980. doi: 10.1007/BFb0120902. URL https://doi.org/10.1007/BFb0120902.

Alexander Rakhlin and Karthik Sridharan. Hierarchies of relaxations for online prediction problems with evolving constraints. In *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 1457–1479. PMLR, 2015. URL https://proceedings.mlr.press/v40/Rakhlin15.html.

Alexander Rakhlin and Karthik Sridharan. Efficient online multiclass prediction on graphs via surrogate losses. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1403–1411. PMLR, 2017. URL https://proceedings.mlr.press/v54/rakhlin17a.html.

Eric Ringger, Marc Carmen, Robbie Haertel, Kevin Seppi, Deryle Lonsdale, Peter McClanahan, James Carroll, and Noel Ellison. Assessing the costs of machine-assisted corpus annotation through a user study. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, pages 3318–3324. European Language Resources Association, 2008. URL https://aclanthology.org/L08-1367/.

Georgy Sokolov, Maximilian Thiessen, Margarita Akhmejanova, Fabio Vitale, and Francesco Orabona. Self-directed node classification on graphs. In *Proceedings of The 36th International Conference on Algorithmic Learning Theory*, volume 272 of *Proceedings of Machine Learning Research*, pages 1138–1168. PMLR, 2025. URL https://proceedings.mlr.press/v272/sokolov25a.html.

Peter Tittmann, Ilia Averbouch, and Johann A. Makowsky. The enumeration of vertex-induced subgraphs with respect to the number of components. *European Journal of Combinatorics*, 32(7): 954–974, 2011. doi: 10.1016/j.ejc.2011.03.017. URL https://doi.org/10.1016/j.ejc.2011.03.017.

Fabio Vitale, Nicolò Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. See the tree through the lines: The Shazoo algorithm. In *Advances in Neural Information Processing Systems 24*, pages 1584–1592. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/hash/8b6dd7db9af49e67306feb59a8bdc52c-Abstract.html.

F. Y. Wu. The Potts model. *Reviews of Modern Physics*, 54(1):235–268, 1982. doi: 10.1103/RevModPhys.54.235. URL https://doi.org/10.1103/RevModPhys.54.235.

Anna Zhou. brainSTEM-Inc/learning: Exact computational certificate for binary nonadjacent transfer. Zenodo, September 2026. URL https://doi.org/10.5281/zenodo.22729770. Version v1.0.3.

Baojian Zhou, Yifan Sun, and Reza Babanezhad Harikandeh. Fast online node labeling for very large graphs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 42658–42697. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhou23k.html.