RECONFIGURATION OF HYPOTHESES: A CONNECTIVITY PERSPECTIVE ON LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconfiguration asks whether one feasible solution can be transformed into another by elementary, feasibility-preserving steps. We ask this for learning: configurations are the hypotheses of a concept class, feasibility is consistency with a sample, and the version space becomes a graph under local edits. For every efficiently evaluable class and polynomial-time-decidable local move relation, reachability lies in polynomial space. For classes admitting bidirectional polynomial-time translations between samples and Boolean constraint formulas, the satisfiability connectivity dichotomy transfers: hypotheses evaluating clauses give polynomial-space completeness, while linear concepts over the binary field admit polynomial-time reachability. The central object is the agnostic relaxation, where feasibility bounds the empirical loss: connectivity is governed by a bottleneck, the least budget admitting a path. For a clause class the bottleneck is polynomial-space-hard to approximate within any factor, yet collapses to the larger endpoint loss on finite landscapes without bad local minima whose minimizers are connected, a discrete form of mode connectivity. For halfspaces on a grid, sufficiently large functional margin guarantees zero-error paths, with a threshold sharp in the plane; positive error budgets can instead produce barriers at every resolution. In the chamber model, reachability is polynomial-time in every fixed dimension and for every fixed error budget. For two-layer networks with convex loss and width at least twice the sample size, lattice paths connect low-loss endpoints with an explicit loss allowance that vanishes under grid refinement.

1 INTRODUCTION

A learning algorithm searches a space of hypotheses for one that fits the data. The space of hypotheses consistent with a sample, the *version space*, has long been a central object in computational learning theory. Yet it is almost always studied as a static set: one asks for its size, its boundaries, or a single point inside it. We ask a different question, borrowed from a now mature area of theoretical computer science. We turn the version space into a graph. Each hypothesis is a node, and two nodes are adjacent when a single elementary edit transforms one hypothesis into the other. We study the graph's connectivity and the complexity of navigating it.

This is precisely the kind of question studied under the heading of *reconfiguration*. Given a source and a target solution of some problem, a reconfiguration problem asks whether one can be transformed into the other through a sequence of intermediate solutions. Each intermediate solution must be feasible, and each is obtained from the previous one by an atomic move. Reconfiguration has been studied for satisfiability, independent sets, colorings, vertex covers, and many other problems. A recurring theme is that the reachability question is typically either solvable in polynomial time or complete for polynomial space. We bring reconfiguration into the learning setting, identify what existing theory implies, and study the resulting boundaries and new questions.

The motivation is twofold. On the one hand, several questions in learning are about movement through a hypothesis space rather than about a single output. Whether a learner can migrate from a deployed model to a retrained one without ever passing through a model that misclassifies the current data is one such question. Whether two empirically optimal models are connected by a path of empirically optimal models is another. Whether local search over hypotheses can reach a given target at all is a third.

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

On the other hand, the geometry of solution spaces is exactly where computational learning theory, the complexity of satisfiability, and statistical physics already meet. They meet through the clustering of solutions in random instances and through algorithmic barriers such as the overlap gap property. Reconfiguration is the decision-theoretic face of that geometry.

For one concept class the graph itself is already a familiar object. The single-flip solution graph of the version space of a halfspace with binary weights is what the binary perceptron literature studies on random instances. There typical solutions are predicted to be isolated, with a conditional proof in the symmetric variant of the model, while at sufficiently low constraint density an efficient algorithm can find a rare well-connected cluster (Krauth & Mézard, 1989; Huang & Kabashima, 2014; Baldassi et al., 2015; Perkins & Xu, 2024; Abbe et al., 2022; Gamarnik et al., 2022). The worst-case decision problem for general concept classes remains unexplored, to our knowledge; that is the question we study here.

1.1 Our Results

For the Boolean models, a concept class is presented by a parameter space $\{0,1\}^m$, a polynomial-time evaluation procedure that maps a parameter string and an input to a label, and a local move relation on parameter strings. A sample is a finite sequence of labeled examples, and a parameter string is feasible when the hypothesis it encodes is consistent with the sample. Precise definitions are given in Section 2.

The realizable case. Our first observation places the basic reachability problem within polynomial space, as is typical of reconfiguration problems.

Theorem 1.1. *For every concept class with polynomial-time evaluation and every polynomial-time-decidable move relation,* Hypothesis-Reconfiguration *is in* PSPACE.

Our second result transfers a known classification into the learning setting. Call a class Γ-*representable* under single-bit moves if samples and formulas over a fixed Boolean constraint language Γ translate in both directions in polynomial time, preserving exactly the feasible parameter strings (Definition B.1). Under this correspondence, the sample itself plays the role of the formula. For such classes the complexity and the geometry of reconfiguration are inherited verbatim from the connectivity dichotomy of Gopalan et al. (2009), in the corrected form of Schwerdtfeger (2014). In that corrected form the boundary of the dichotomy is the safe tightness of Γ (Definition 2.8).

Theorem 1.2. *Let a concept class be Γ-representable under single-bit moves. If Γ is safely tight, then* Hypothesis-Reconfiguration *is in* P*,* Hypothesis-Connectivity *is in* coNP*, and every connected component of every solution graph has diameter linear in the number of parameters. If Γ is not safely tight, then both* Hypothesis-Reconfiguration *and* Hypothesis-Connectivity *are* PSPACE*-complete, and the diameter can be superpolynomial in the number of parameters.*

Both sides of the dichotomy are populated by natural concept classes. The hard side is reached already by a class as simple as evaluating short clauses.

Corollary 1.3. *There is a fixed, polynomial-time evaluable concept class of Vapnik–Chervonenkis dimension m for which* Hypothesis-Reconfiguration *is* PSPACE*-complete.*

On the easy side, the hypothesis is a linear functional over the binary field.

Corollary 1.4. *For the class of $\mathrm{GF}(2)$-linear concepts on m bits under single-bit moves,* Hypothesis-Reconfiguration *is in* P *and every connected component of every solution graph has linear diameter.*

Theorem 1.1 is proved in Appendix A, and Theorem 1.2 and Corollaries 1.3 and 1.4 in Appendix B.

Unless P = PSPACE, width three is the least that is hard: for clauses of width at most two the language is safely tight, so reachability is polynomial-time (Corollary B.5).

The agnostic relaxation. Exact consistency is the realizable case of a more general object. In practice a learner fits a sample only up to a loss tolerance. Appendix C develops this agnostic relaxation as the central problem of the framework. There the feasible set F_k is the set of hypotheses

that misclassify at most k examples. Connectivity of two hypotheses in F_k is monotone in k, and it is controlled by a single quantity, the *reconfiguration bottleneck* $\beta_S(w_0, w_1)$. This bottleneck is the least k for which w_0 and w_1 are connected in F_k (Definition C.2 and Lemma C.4). The problem stays PSPACE-complete in the worst case (Proposition C.3). For the clause class the bottleneck is hard to approximate within any multiplicative factor, and, in loss terms, within a fixed additive constant (Corollary C.7). This inapproximability is inherited through the framework from the inapproximability of maxmin satisfiability reconfiguration (Hirahara & Ohsaka, 2024).

Theorem 1.5 (restated as Theorem C.6)**.** *There is a rational constant $\epsilon_0 > 0$ such that, for the clause class $\mathcal{C}_3$ of Corollary 1.3, the following promise problem is* PSPACE*-hard: given a sample S of size t and two consistent hypotheses w_0 and w_1, decide whether $\beta_S(w_0, w_1) = 0$ or $\beta_S(w_0, w_1) > \epsilon_0 t$. Consequently, unless* P $=$ PSPACE*, no polynomial-time algorithm approximates $\beta_S(w_0, w_1)$ within any multiplicative factor.*

The relaxation collapses to a triviality, a discrete counterpart of mode connectivity, on any finite landscape that has no bad local minimum and whose set of minimizers is connected.

Theorem 1.6 (restated as Theorem C.11)**.** *Let $G = (V, E)$ be a finite nonempty graph and $e\colon V \to \mathbb{R}_{\geq 0}$ a loss with no bad local minimum, that is, every vertex of loss above the minimum $m_\star$ has a neighbor of strictly smaller loss. Suppose the minimizers of e induce a connected subgraph. Then for every $k \geq m_\star$ the sublevel set $\{v : e(v) \leq k\}$ induces a connected subgraph of G. In particular, taking G to be the hypercube and $e = e_S$, the sample error count, if e_S has no bad local minimum and its minimizers induce a connected subgraph, then any two feasible hypotheses are connected in F_k, so* AGNOSTIC-RECONFIGURATION *is a yes-instance and $\beta_S(w_0, w_1) = \max(e_S(w_0), e_S(w_1))$.*

Halfspaces. Appendix D carries the collapse out concretely for halfspaces whose weights are discretized to a grid. The grid is $\Lambda_s = (\frac{1}{s}\mathbb{Z})^d$, with single-coordinate moves of size $\frac{1}{s}$ (Definition D.1). The data are assumed to have Euclidean norm at most R. The collapse holds, by a routing argument, once the grid resolves the functional margin. A two-point instance shows that the constant in this margin threshold is sharp in the plane (Proposition D.4).

Theorem 1.7 (restated as Theorem D.3)**.** *In the model of Definition D.1, any two weights $p, q \in \Lambda_s$ whose functional margins are both at least μ, with $\mu > \frac{R\sqrt{d}}{2s}$, are joined by a coordinate-move path lying entirely in the zero-error set, every vertex of which has functional margin at least $\mu - \frac{R\sqrt{d}}{2s}$. Consequently, for every $\gamma > 0$, every $\rho > 0$ and every resolution $s > \frac{R\sqrt{d}}{2\gamma\rho}$, any two weights $p, q \in \Lambda_s$ with $\|p\|_2, \|q\|_2 \geq \rho$ and geometric margin $\min_i y_i\langle \cdot, x_i\rangle / \|\cdot\|_2$ at least γ are connected in the zero-error set, and their bottleneck is $\beta(p, q) = 0 = \max(e(p), e(q))$. All such weights lie in a single connected component of the zero-error set. The norm condition cannot be dropped (Proposition D.4).*

The agnostic regime is different. With a positive error budget the feasible region can split into clusters, separated by a barrier whose height is a constant fraction of the sample size, independently of the resolution. Thus a finer grid alone need not remove an agnostic barrier.

Proposition 1.8 (restated as Proposition D.6)**.** *For every integer $r \geq 1$ there is a sample of $t = 3r$ points in $\mathbb{R}^2$ with a common label with the following property, for every $s \geq 1$. The grid weights $u = \left(-\frac{2}{s}, 0\right)$ and $v = \left(\frac{2}{s}, \frac{2}{s}\right)$ have $e(u) = e(v) = r$, and every coordinate-move path in Λ_s from u to v, as well as every continuous path in $\mathbb{R}^2$ from u to v, meets a weight of error at least $2r$. Hence the bottleneck satisfies $\beta(u, v) \geq 2r = 2\max(e(u), e(v))$, a gap of $r = t/3$ that no refinement of the grid removes; the bound is attained.*

In the chamber model, reachability is polynomial-time whenever the dimension or the error budget is fixed. On rational data, the natural scale-free configuration space here is the *chamber model* (Appendix D.6). Its vertices are the chambers of the arrangement of the hyperplanes $\{w : \langle w, x_i\rangle = 0\}$. Two chambers are adjacent when they share a facet, and a chamber is feasible when $e \leq k$.

Corollary 1.9 (restated as Corollary D.12)**.** *For every fixed d, agnostic halfspace reconfiguration in the chamber model is solvable in polynomial time, and the bottleneck β is computable in polynomial time.*

Fixing the error budget also makes reachability tractable, and computes the bottleneck when it does not exceed that budget.

Proposition 1.10 (restated as Proposition D.13)**.** *For every fixed k, agnostic halfspace reconfiguration in the chamber model with error budget k is solvable in polynomial time, uniformly in the dimension d; the bottleneck β is computed in the same time whenever $\beta \leq k$.*

Networks. For two-layer networks spare neurons allow paths built using convexity in the output weights. The resulting near-collapse is a discrete counterpart of mode connectivity. When the width is at least twice the sample size, over-parameterization lets the sample predictions of any low-loss network be re-expressed on a reserve of neurons, up to rounding error, and any two low-loss networks are then joined by a low-loss path. The model (Appendix D.7) has n samples and m rectified-linear neurons, with parameters confined to the grid of resolution s. The hidden weights are confined to a box on which every activation is at most G. The loss L is the sample average of a per-example loss that is convex and λ-Lipschitz.

Theorem 1.11 (restated as Theorem D.17)**.** *Let $m \geq 2n$ and let θ_A, θ_B be two-layer networks with $L(\theta_A), L(\theta_B) \leq \eta$ and output weights of ℓ_∞-norm at most K. Then they are joined by a single-coordinate-move lattice path along which the loss never exceeds $\eta + \frac{\lambda G(m+n)}{2s}$. On the refined grid of resolution $s' = ks$ with $k \in \mathbb{Z}_{\geq 1}$, the same holds with s' in place of s, so the bottleneck is at most $\eta + \epsilon$, for every $\epsilon > 0$, once $s' \geq \frac{\lambda G(m+n)}{2\epsilon}$.*

Deeper networks lie outside the two-layer argument. There the output layer is still linear in its weights, but the prediction is not linear in the layers below it. What replaces convexity in that setting is one of the questions collected in Section 1.5.

The technical content of Theorems 1.1 and 1.2 and Corollaries 1.3 and 1.4 is modest, a matter of recognizing consistency as a constraint language. The contribution is the framework of Appendices A to C, the halfspace analysis of Appendix D, and the program of Section 1.5.

1.2 Our Techniques

Generic reachability in polynomial space. The upper bound of Theorem 1.1 is the standard argument for reconfiguration. The solution graph has at most 2^m vertices, and feasibility and adjacency are decidable in polynomial time. Reachability in an exponentially large but succinctly described graph can then be decided by a nondeterministic procedure that walks from the source while maintaining only the current vertex and a step counter bounded by 2^m. Since nondeterministic polynomial space equals polynomial space, the bound follows. Nothing here is specific to learning; what matters is only that hypotheses are evaluated efficiently and that moves are recognized efficiently.

Consistency as a constraint language. The heart of Theorem 1.2 is a dictionary rather than a new hardness proof. Once the parameters of a hypothesis are fixed as Boolean variables, each labeled example imposes one constraint on those variables: that the hypothesis assigns the example its given label. A Γ-representation translates each sample to a conjunction of relations from Γ and each such formula back to a sample, in polynomial time and without changing the parameter bits. The version space and the formula's solution set then coincide. Under single-bit moves the version space, with its induced hypercube adjacency, is then literally the solution graph studied in the connectivity dichotomy for Boolean satisfiability. The complexity and diameter classifications transfer because the two solution graphs coincide and the reductions in both directions are the identity on solutions.

Two natural instantiations. Corollary 1.3 is obtained by taking the inputs to be short clauses and the hypothesis to be an assignment evaluated on a clause. Consistency with a sample of clauses is then satisfaction of the corresponding bounded-width formula, where a clause labeled false contributes the unit clauses of its negated literals. The language of clauses is not tight, so the hard side of the dichotomy applies. Corollary 1.4 is obtained by taking the hypothesis to be a linear functional over the binary field. Consistency with a sample is then a system of linear equations over the parameters, and the components of the version space are the subcubes on the coordinates no example uses. This is proved directly. For examples of Hamming weight at most three, the same problem is also the easy side of the dichotomy, since the affine language is safely tight. In both cases the moves are single-bit flips of the parameters, matching the adjacency of the dichotomy.

Bottlenecks and geometric paths. Gap hardness for satisfiability reconfiguration transfers by counting violated clauses as errors. For halfspaces, a lattice path stays within half a grid step in each coordinate of the segment between its endpoints; a margin larger than the resulting perturbation preserves every label. For chambers, rational linear programming tests sign patterns, while fixed dimension or fixed error budget bounds how many must be tested. For networks, a maximum-volume basis compresses the sample predictions into at most n active neurons. Spare neurons then permit output-layer interpolation; rounding and coordinate routing give the explicit loss allowance.

1.3 Other Related Work

Reconfiguration. The reconfiguration framework and its prevailing PSPACE-hardness were crystallized in the work of Ito et al. (2011), and the area is surveyed by van den Heuvel (2013). More recently the inapproximability of optimization variants has been established through a reconfiguration analogue of the probabilistically checkable proof theorem due to Hirahara & Ohsaka (2024). That theorem proves the reconfiguration inapproximability hypothesis, which Ohsaka (2026) put forth together with gap-preserving reductions between reconfiguration problems. An independent proof of the hypothesis is due to Guruswami et al. (2025). For satisfiability the approximation threshold is now known up to the constant: Hirahara & Ohsaka (2025) show that the optimal factor for Maxmin Ek-SAT Reconfiguration is $1 - \Theta(1/k)$, with PSPACE-hardness within $1 - \frac{1}{10k}$ for every sufficiently large k.

Connectivity of solution spaces. The classification we transfer is that of Gopalan et al. (2009), who refined Schaefer's satisfiability dichotomy (Schaefer, 1978) into a connectivity dichotomy. The source-to-target reachability problem is in polynomial time for tight constraint languages and PSPACE-complete otherwise, and the diameter of components passes from linear to possibly superpolynomial along the same boundary. The boundary was later corrected to the safe strengthening of tightness (Schwerdtfeger, 2014), which is the form we use. The study of Gopalan, Kolaitis, Maneva, and Papadimitriou was itself motivated by the clustering picture of solution spaces from statistical physics, as in the analysis of random satisfiability by Mézard et al. (2002). The same geometry underlies the overlap gap property of Gamarnik (2021), a topological obstruction to stable algorithms for optimization over random structures. This obstruction bears directly on learning over random instances.

Learning theory. The version space, the set of hypotheses consistent with the observed data, was introduced by Mitchell (1982), whose candidate-elimination algorithm maintains its most specific and most general boundaries. Consistency with a sample is the basic object of the learning model of Valiant (1984); see (Kearns & Vazirani, 1994). Our feasible set is exactly the version space. We ask a new question of it: whether it is reachable under local edits, posed as a decision problem over a concept class. This question appears not to have been studied in this generality. The chamber graph of Appendix D.6 is the tope graph of the oriented matroid of the data (Björner et al., 1999). When no two sample points define the same hyperplane, the chamber graph is the one-inclusion graph restricted to strict sign patterns of homogeneous halfspaces on the sample: two such patterns are adjacent when they differ on one example (Haussler et al., 1994). A parallel class of several points is crossed in one move of the chamber graph, whereas the two resulting strict sign patterns are not adjacent in the one-inclusion graph.

The solution space of the binary perceptron. For halfspaces with binary weights the single-flip solution graph of the version space is the object of a long line of work on the storage problem of the perceptron, beginning with Krauth & Mézard (1989). On random instances the typical solutions are predicted to be isolated, each a singleton component at linear Hamming distance from every other (Huang & Kabashima, 2014), and Perkins and Xu prove this picture for the symmetric variant under an assumption on the critical points of a real-valued function (Perkins & Xu, 2024). At sufficiently low constraint density a rare, dense, well-connected cluster exists in the symmetric and the original model alike, and an efficient multiscale majority algorithm can reach it (Baldassi et al., 2015; Abbe et al., 2022). In the symmetric model the overlap gap property of that landscape obstructs stable algorithms (Gamarnik et al., 2022), and in the original model no stable algorithm reliably locates a strongly isolated solution (Gong et al., 2026). That literature is typical-case and structural; the present paper is worst-case and decision-theoretic. Our Remark B.4 is the determin-

istic shadow of the isolation phenomenon. The bottleneck of Appendix C is the saddle height of discrete landscape theory (Flamm et al., 2002), taken as a computational object.

Loss landscapes and mode connectivity. The agnostic side of our framework is the decision form of mode connectivity, the empirical observation that independently trained networks are joined by paths of near-constant low loss (Garipov et al., 2018; Draxler et al., 2018). On the theoretical side, Freeman & Bruna (2017) prove that the level sets of a half-rectified one-layer network are asymptotically connected, with explicit bounds reflecting the interplay of data and architecture. For distinct inputs, convex loss, and suitable strictly monotone onto activations such as leaky ReLU, Nguyen (2019) proves connected sublevel sets when the first hidden layer has width at least $2N$ and the subsequent widths strictly decrease. A subsequent note (Nguyen, 2021) lowers that first-layer bound to $N + 1$ for N samples under the same restrictions. Its two-layer examples show that width N does not suffice in general.

Venturi et al. (2019) isolate the property behind these results, a nonincreasing path to a global minimizer from every point, and show that it excludes spurious valleys. Kuditipudi et al. (2019) connect two solutions that each use at most half the neurons through the reserve. Nguyen et al. (2021) connect pairs of solutions with order $\sqrt{N}$ neurons in the last two hidden layers, under assumptions on the intermediate features; this is a guarantee for those pairs rather than for whole sublevel sets. Şimşek et al. (2021) show how adding one neuron per layer connects permutation copies through networks representing the same function. The role of the same permutation symmetries in linear mode connectivity is the subject of Entezari et al. (2022); Ainsworth et al. (2023).

Our network result is the discrete, reconfiguration-theoretic counterpart of this picture. Theorem C.11 gives a sufficient condition for exact sublevel connectivity on finite configuration graphs; Theorem D.17 separately constructs lattice paths for two-layer networks, with an explicit loss allowance. We claim neither a new threshold nor a new mechanism. Theorem C.11 uses the descent principle of Venturi et al. (2019); the network proof adapts the reserve-of-neurons mechanism of Nguyen (2019, Theorem 5.1), with different activation assumptions and a lattice error bound. What the discrete view contributes is the reconfiguration-theoretic formulation and the discretization by lattice routing with explicit slack.

1.4 Discussion

Theorem 1.2 applies to a broad family of classes: for them, hypothesis reconfiguration translates to connectivity in a constraint satisfaction problem. This correspondence has two consequences. Statistical physics and the connectivity dichotomy each attach a geometry to solution spaces in general. For learning, that geometry is the geometry of the version space. The relevant expressiveness here is exact representation by a constraint language; the theorem does not classify navigation by Vapnik–Chervonenkis dimension. The formal claims of Appendices A and B are kept to what this dictionary between reconfiguration and constraint satisfaction yields cleanly.

The agnostic relaxation of Appendix C adds a global error budget, so the consistency dictionary alone does not classify its complexity. There we show that the relaxation is no easier than the realizable case in the worst case, yet it collapses to a triviality on any finite landscape that has no bad local minimum and whose set of minimizers is connected. For halfspaces on a grid, Appendix D pins the realizable collapse to a threshold relating the margin to the grid resolution. A three-fold barrier, however, shows that the agnostic regime can fragment independently of resolution. The convex realizable region supports the margin argument, while the agnostic barrier survives refinement.

For two-layer networks this same convexity, supplied by the spare neurons of an over-parameterized model rather than by a margin, yields a discrete counterpart of mode connectivity once the width is at least twice the sample size. On the network side we claim neither the threshold nor the mechanism (Section 1.3). The contribution there is the reconfiguration-theoretic formulation and the lattice discretization with explicit slack. We leave deeper networks and the removal of the grid slack open (Section 1.5).

1.5 Open Problems

Complexity of the agnostic bottleneck for halfspaces. Theorem C.6 settles the question for the clause class $\mathcal{C}_3$: there the bottleneck is PSPACE-hard to approximate within any multiplicative factor. This inapproximability is inherited from the inapproximability of maxmin satisfiability reconfiguration (Hirahara & Ohsaka, 2024). As $\mathcal{C}_3$ is the learning-theoretic image of 3-SAT, this result is the expected counterpart of the satisfiability result. What it leaves open is the geometric classes of Appendix D. In the chamber model, Corollary D.12 gives polynomial-time reachability and bottleneck computation in fixed dimension, and Proposition D.13 gives polynomial-time reachability for fixed error budget. The open complexity of chamber reachability therefore concerns growing dimension and budget. Whether agnostic halfspace reconfiguration is PSPACE-hard there, or even NP-hard, and whether the bottleneck is hard to approximate, remains open. We conjecture PSPACE-completeness (Remark D.14); such a result would be the first hardness for the agnostic problem on a natural geometric class. But the reductions stall for the reasons given in Remark D.14, and the tractability on each axis separately leaves the question open in both directions.

The margin threshold beyond the plane. The threshold $R\sqrt{d}/(2s)$ of Theorem D.3 is sharp in the plane (Proposition D.4), and one of order R/s is necessary in every dimension $d \geq 2$ (Appendix D.4). Whether the factor $\sqrt{d}$ is needed for $d \geq 3$ is open; the diagonal weights that witness sharpness in the plane do not witness it there.

Over-parameterization and the collapse for networks. Theorem D.17 resolves the question for two-layer networks in the over-parameterized regime $m \geq 2n$: any two low-loss networks are connected through low loss on a fine grid, a discrete counterpart of the continuous connectivity results (Nguyen, 2019; Freeman & Bruna, 2017). What remains is to go deeper. The two-layer argument uses output-layer convexity and spare neurons whose hidden weights can move while their output weights are zero. A network with further layers below the last hidden one has a *trunk*: the layers below the output layer. In such a network the prediction is not linear in the trunk, and moving one trunk to another while holding the loss low is the step that does not discretize by convexity. Continuous sublevel connectivity is known with first-layer width $N + 1$ for N distinct samples, suitable activations, and strictly decreasing subsequent widths (Nguyen, 2019; 2021). Whether those arguments survive discretization to single-coordinate moves with explicit slack, and what the exact width threshold below $m = 2n$ is, are both open. The width-one example of Remark D.19 shows that a network barrier can persist under refinement.

A landscape dichotomy. One might hope for a single structural property that decides tractability, in the spirit of the safely tight versus not safely tight boundary of Theorem 1.2 but for the agnostic case. Our sufficient conditions concern two different questions: whether feasible endpoints are always connected, and whether their connectivity can be decided efficiently. The three-fold barrier gives disconnected feasible sets in a tractable chamber model, so disconnection alone does not imply hardness. A structural complexity classification remains open beyond the fixed-dimension and fixed-budget regimes.

1.6 Paper Organization

The main text states the results and fixes the preliminaries. The technical sections, with every proof, are placed in the appendices, rather than in the body, because of the page limit. Section 2 fixes the learning-theoretic and reconfiguration definitions and recalls the connectivity dichotomy we rely on. Appendix A proves the polynomial-space upper bound of Theorem 1.1 for the reconfiguration problems of Definition 2.5. Appendix B introduces Γ-representability, proves the transfer theorem of Theorem 1.2, and derives the two corollaries and the easiness of width two. Appendix C develops the agnostic relaxation, where feasibility is a loss threshold, and proves the membership, hardness, and landscape-collapse results, together with the tractable case for explicitly represented hypothesis spaces. Appendix D specializes to halfspaces on a grid. There margin forces the collapse above a resolution threshold that is sharp in the plane, while a three-fold barrier shows that the agnostic regime can fragment independently of resolution. The chamber graph then makes the agnostic problem tractable in every fixed dimension and for every fixed error budget. The section closes

with an over-parameterization argument that gives a discrete counterpart of mode connectivity for two-layer networks of width at least twice the sample size.

2 PRELIMINARIES

We write $\{0,1\}^m$ for the set of binary strings of length m. For $w, w' \in \{0,1\}^m$, we write $d_H(w, w')$ for their Hamming distance. All classes, samples, and move relations below are presented succinctly, so that the relevant objects are decidable in time polynomial in the description length.

2.1 CONCEPT CLASSES, SAMPLES, AND CONSISTENCY

A hypothesis is a string of parameters, and a class says how each string labels each instance.

Definition 2.1 (Parameterized concept class)**.** A *parameterized concept class* is a triple $\mathcal{C} = (m, \mathsf{E}, X)$, where X is an instance set, $m \geq 1$ is a parameter length, and $\mathsf{E}\colon \{0,1\}^m \times X \to \{0,1\}$ is an evaluation procedure computable in time polynomial in m and the description length of its input. For $w \in \{0,1\}^m$ the *hypothesis* h_w is the map $x \mapsto \mathsf{E}(w, x)$. The complexity statements below concern a class given for every $m \geq 1$ by one procedure E taking m as an argument, and measure time and space in the length of the input, which includes m.

In words, a parameterized concept class is a family of $\{0,1\}$-valued functions on X indexed by m-bit strings. Under the single-bit move relation, reconfiguration edits this string one bit at a time. Appendix B reads this string as a truth assignment and as a linear functional over the binary field, and Appendix D replaces it by a grid point.

Definition 2.2 (Sample and consistency)**.** A *sample* is a finite sequence $S = ((x_1, y_1), \ldots, (x_t, y_t))$ of labeled examples $(x_i, y_i) \in X \times \{0,1\}$, repetitions allowed; t is its size. A parameter string $w \in \{0,1\}^m$ is *consistent* with S if $\mathsf{E}(w, x_i) = y_i$ for every $i \in \{1, \ldots, t\}$, and we write $\mathsf{cons}_S(w)$ for this predicate. The *version space* of S is $V(S) = \{w \in \{0,1\}^m : \mathsf{cons}_S(w)\}$.

The predicate cons_S is decidable in polynomial time, since it is a conjunction of t evaluations of E. The version space is the static object that our reconfiguration questions endow with a graph structure.

A sample is a sequence, so repetitions are counted rather than merged. This lets the hardness reductions map a formula with C clauses, repeated clauses included, to a sample of size exactly C.

Definition 2.3 (Shattering and Vapnik–Chervonenkis dimension)**.** A set of instances $\{x_1, \ldots, x_r\} \subseteq X$ is *shattered* by $\mathcal{C}$ if every labeling is realized by some hypothesis, that is, for each $(b_1, \ldots, b_r) \in \{0,1\}^r$ there is a $w \in \{0,1\}^m$ with $\mathsf{E}(w, x_i) = b_i$ for all i. The *Vapnik–Chervonenkis dimension* of $\mathcal{C}$ is the largest r for which some set of r instances is shattered.

The Vapnik–Chervonenkis dimension measures the expressiveness of the class, and the Sauer–Shelah lemma (Lemma D.11) turns it into a bound on the labelings a sample admits.

2.2 RECONFIGURATION AND SOLUTION GRAPHS

We first fix which small changes between consistent hypotheses count as one move.

Definition 2.4 (Move relation and solution graph)**.** A *move relation* for $\mathcal{C}$ is a symmetric relation $R \subseteq \{0,1\}^m \times \{0,1\}^m$ decidable in time polynomial in m. The *single-bit relation* is $R = \{(w, w') : d_H(w, w') = 1\}$. Given a sample S, the *solution graph* $G(S)$ has vertex set $V(S)$ and an edge between w and w' whenever $(w, w') \in R$.

The two problems below ask whether two given consistent hypotheses lie in one component of this graph, and whether the whole graph is connected.

Definition 2.5 (Reconfiguration problems)**.** Fix a parameterized concept class $\mathcal{C}$ and a move relation R for it. The problem HYPOTHESIS-RECONFIGURATION takes as input a sample S and two consistent strings $w_0, w_1 \in V(S)$, and asks whether w_0 and w_1 lie in the same connected component of $G(S)$. The problem HYPOTHESIS-CONNECTIVITY takes as input S, together with the parameter length m written in unary, and asks whether $G(S)$ is connected.

In both problems, the parameter length is part of the input and is at most the input's length. For HYPOTHESIS-RECONFIGURATION, this holds because $w_0 \in \{0,1\}^m$ is written out; for HYPOTHESIS-CONNECTIVITY, because m is given in unary.

The *diameter* of a connected component of $G(S)$ is the largest shortest-path distance between two of its vertices. When R is the single-bit relation, $G(S)$ is the subgraph of the m-dimensional hypercube $\{0,1\}^m$ induced by the version space.

Two standard results are used in more than one appendix. The first bounds the space of deciding reachability: a nondeterministic walk that guesses one step at a time uses little space, and Savitch's theorem removes the nondeterminism at the cost of squaring that space.

Proposition 2.6 (Savitch's theorem Savitch, 1970)**.** *A decision problem accepted by a nondeterministic Turing machine in space $S(n) \geq \log n$ is decided by a deterministic Turing machine in space $O\big(S(n)^2\big)$. In particular, any problem decided by a nondeterministic machine in polynomial space lies in* PSPACE.

The second applies to a graph small enough to be written down explicitly. It computes the least threshold at which two vertices become connected using only vertices of weight at most that threshold, the same bottleneck quantity that Appendix C needs on an explicit graph.

Lemma 2.7 (Minimum-bottleneck path Pollack, 1960; Hu, 1961; see also Cormen et al., 2009, Problem 23-3)**.** *Let H be an explicitly given finite graph with binary rational vertex weights, and let u, v be two of its vertices. The minimum over u–v paths of the maximum vertex weight on the path, taken to be ∞ when no such path exists, is computable in time polynomial in the full weighted input length. So is comparison of this minimum with a binary rational threshold k, including its encoding length.*

2.3 BOOLEAN CONSTRAINT LANGUAGES AND THE CONNECTIVITY DICHOTOMY

We recall the framework and the result we transfer. A *constraint language* Γ is a finite set of Boolean relations. A Γ-*formula* over variables $z_1, \ldots, z_m$ is a conjunction of constraints. Each constraint applies a relation of Γ to a tuple whose entries are variables or the constants $0, 1$. Repeated entries are allowed. This setup is the model of Gopalan et al. (2009); Schwerdtfeger (2014), written $\mathrm{CNF}_C(\Gamma)$ in (Schwerdtfeger, 2014). Every result we import is proved in this model. A Γ-formula's solution set is a subset of $\{0,1\}^m$. Its *solution graph* has those solutions as vertices, with hypercube adjacency. The dichotomy we transfer is controlled by the following structural property of Γ.

Definition 2.8 (Tight constraint language Gopalan et al., 2009)**.** For a relation $R \subseteq \{0,1\}^k$, let $G(R)$ be the graph on R that joins two tuples at Hamming distance one. The relation R is *bijunctive* if it is closed under coordinatewise majority, that is, $\mathrm{maj}(a,b,c) \in R$ whenever $a, b, c \in R$, where each coordinate of $\mathrm{maj}(a,b,c)$ is the bit occurring at least twice there. It is *componentwise bijunctive* if every connected component of $G(R)$ is bijunctive. It is *OR-free* if the binary relation $\{01, 10, 11\}$ cannot be obtained from R by fixing all but two of its coordinates to constants, and *NAND-free* if $\{00, 01, 10\}$ cannot be so obtained. A constraint language Γ is *tight* if every relation in Γ is componentwise bijunctive, or every relation in Γ is OR-free, or every relation in Γ is NAND-free. For $R \subseteq \{0,1\}^k$ and indices $i < j$, the *identification* of the coordinates i and j is the relation $\{(a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_k) : a \in R, a_i = a_j\} \subseteq \{0,1\}^{k-1}$. The language Γ is *safely tight* (Schwerdtfeger, 2014) if the set of all relations obtainable from relations of Γ by repeated identifications, Γ itself included, is tight. We call the relations obtained from R by identifications its *identification minors*. Agreement with the source definition is justified in Remark B.2.

Safe tightness determines connectivity complexity as follows.

Proposition 2.9 (Gopalan et al., 2009; Schwerdtfeger, 2014)**.** *Let Γ be a constraint language. If Γ is safely tight, then source-to-target connectivity of Γ-formulas is in* P*, connectivity is in* coNP*, and the diameter of every connected component is linear in the number of variables. If Γ is not safely tight, then both connectivity problems are* PSPACE*-complete, and the diameter of a component can be superpolynomial in the number of variables, as large as $2^{\Omega(\sqrt{m})}$ for formulas on m variables (Schwerdtfeger, 2014, Table 2).*

AI USE STATEMENT

The authors posed the research question. Claude (Anthropic) helped develop the proofs, find and format references, and write the exposition. Codex (OpenAI) assisted with reviewing the mathematical arguments, citations, and prose of the ICLR submission. The authors reviewed the AI-assisted work and take responsibility for the paper and any errors.

Under the ICLR 2027 AI Policy for Authors, model development, claim formulation, and proof contributions and writing require disclosure; drafting, identifying literature, and formatting references are recommended for disclosure. Experiments, data, research code, and translation do not arise in this paper.

REPRODUCIBILITY STATEMENT

Every result of this paper is a theorem. The model of each is fixed where the result is stated, every assumption is part of the statement, and every proof is given in full in the appendices, which are self-contained given Section 2; no experiment, dataset, or code is involved.

REFERENCES

Emmanuel Abbe, Shuangping Li, and Allan Sly. Binary perceptron: Efficient algorithms can find solutions in a rare well-connected cluster. In *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 860–873, 2022.

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations (ICLR)*, 2023.

Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Subdominant dense clusters allow for simple learning and high computational performance in neural networks with discrete synapses. *Physical Review Letters*, 115:128101, 2015.

Anders Björner, Michel Las Vergnas, Bernd Sturmfels, Neil White, and Günter M. Ziegler. *Oriented Matroids*, volume 46 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2nd edition, 1999.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred A. Hamprecht. Essentially no barriers in neural network energy landscape. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1309–1318, 2018.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Christoph Flamm, Ivo L. Hofacker, Peter F. Stadler, and Michael T. Wolfinger. Barrier trees of degenerate landscapes. *Zeitschrift für Physikalische Chemie*, 216(2):155–173, 2002.

C. Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. In *International Conference on Learning Representations (ICLR)*, 2017.

David Gamarnik. The overlap gap property: A topological barrier to optimizing over random structures. *Proceedings of the National Academy of Sciences*, 118(41):e2108492118, 2021.

David Gamarnik, Eren C. Kızıldağ, Will Perkins, and Changji Xu. Algorithms and barriers in the symmetric binary perceptron model. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 576–587, 2022.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P. Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018.

Shuyang Gong, Brice Huang, Shuangping Li, and Mark Sellke. Stable algorithms cannot reliably find isolated perceptron solutions, 2026. arXiv:2604.00328.

Parikshit Gopalan, Phokion G. Kolaitis, Elitza Maneva, and Christos H. Papadimitriou. The connectivity of Boolean satisfiability: Computational and structural dichotomies. *SIAM Journal on Computing*, 38(6):2330–2355, 2009.

Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39(2):742–765, 2009.

Venkatesan Guruswami, Karthik C. S., Pasin Manurangsi, Xuandi Ren, and Kewen Wu. On inapproximability of reconfiguration problems: PSPACE-hardness and some tight NP-hardness results, 2025. arXiv:2312.17140v3, merging the December 2023 preprint of Karthik C. S. and Manurangsi with work of Guruswami, Ren, and Wu.

David Haussler, Nick Littlestone, and Manfred K. Warmuth. Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.

Shuichi Hirahara and Naoto Ohsaka. Probabilistically checkable reconfiguration proofs and inapproximability of reconfiguration problems. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 1435–1445, 2024.

Shuichi Hirahara and Naoto Ohsaka. Asymptotically optimal inapproximability of Ek-SAT reconfiguration. In *Proceedings of the 66th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 267–276, 2025. doi: 10.1109/FOCS63196.2025.00018.

T. C. Hu. The maximum capacity route problem. *Operations Research*, 9(6):898–900, 1961.

Haiping Huang and Yoshiyuki Kabashima. Origin of the computational hardness for learning with binary synapses. *Physical Review E*, 90:052813, 2014.

Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12–14):1054–1065, 2011.

David S. Johnson and Franco P. Preparata. The densest hemisphere problem. *Theoretical Computer Science*, 6(1):93–107, 1978.

Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

Leonid G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979. Translation of Doklady Akademii Nauk SSSR 244 (1979), 1093–1096.

Werner Krauth and Marc Mézard. Storage capacity of memory networks with binary couplings. *Journal de Physique*, 50(20):3057–3066, 1989.

Rohith Kuditipudi, Xiang Wang, Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Rong Ge, and Sanjeev Arora. Explaining landscape connectivity of low-cost solutions for multilayer nets. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.

Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

Tom M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, 1982.

Quynh Nguyen. On connected sublevel sets in deep learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 4790–4799, 2019.

Quynh Nguyen. A note on connectivity of sublevel sets in deep learning, 2021. arXiv:2101.08576.

Quynh N. Nguyen, Pierre Bréchet, and Marco Mondelli. When are solutions connected in deep networks? In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pp. 20956–20969, 2021.

Naoto Ohsaka. Gap preserving reductions between reconfiguration problems. *Journal of Computer and System Sciences*, 157:103733, 2026. doi: 10.1016/j.jcss.2025.103733.

Will Perkins and Changji Xu. Frozen 1-RSB structure of the symmetric Ising perceptron. *Random Structures & Algorithms*, 64(4):856–877, 2024.

Maurice Pollack. The maximum capacity through a network. *Operations Research*, 8(5):733–736, 1960.

Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1): 145–147, 1972.

Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 216–226, 1978.

Konrad W. Schwerdtfeger. A computational trichotomy for connectivity of Boolean satisfiability. *Journal on Satisfiability, Boolean Modeling and Computation*, 8(3–4):173–195, 2014.

Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

Berfin Şimşek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9722–9732, 2021.

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Jan van den Heuvel. The complexity of change. In *Surveys in Combinatorics 2013*, volume 409 of *London Mathematical Society Lecture Note Series*, pp. 127–160. Cambridge University Press, 2013.

Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16(2):264–280, 1971.

Luca Venturi, Afonso S. Bandeira, and Joan Bruna. Spurious valleys in one-hidden-layer neural network optimization landscapes. *Journal of Machine Learning Research*, 20(133):1–34, 2019.

A Hypothesis Reconfiguration and Its Complexity

We first record the polynomial-space upper bound. It holds in full generality, requiring nothing of the class beyond efficient evaluation and a move relation decidable in polynomial time. The one fact it needs from complexity theory is Savitch's theorem (Proposition 2.6), that nondeterminism costs at most a square in space.

Proof of Theorem 1.1. Fix the class $\mathcal{C} = (m, \mathsf{E}, X)$ and the move relation R, and let the input consist of a sample S and consistent strings w_0, w_1. We describe a nondeterministic procedure using space polynomial in the input length; membership in PSPACE then follows by Proposition 2.6.

The procedure maintains a current string w, initially w_0, and an integer counter c, initially zero. The counter is stored in binary using $m + 1$ bits, since it ranges over $0, \ldots, 2^m$. At each step, the procedure verifies $\mathsf{cons}_S(w)$, which takes polynomial time since it is a conjunction of t evaluations of E, each polynomial-time by Definition 2.1, and rejects on this branch if the check fails. If $w = w_1$,

it accepts. Otherwise, if c has reached 2^m, it rejects on this branch. If c has not reached 2^m, it guesses a string $w' \in \{0,1\}^m$ and verifies $(w, w') \in R$ and $\mathsf{cons}_S(w')$, both in polynomial time. It rejects on this branch if either check fails, and otherwise replaces w by w' and increments c. Every branch halts. The counter c grows by one with each guess, and the branch halts once c reaches 2^m, so a branch makes at most 2^m guesses.

Every accepting computation traces a walk in $G(S)$ from w_0 to w_1. Each string the procedure holds has been verified consistent, so it is a vertex of $G(S)$. Each replacement of w by w' has been verified to be a move of R, so consecutive strings are adjacent. The first string is w_0, and the string held at acceptance is w_1. So acceptance implies that w_0 and w_1 are in the same component.

Conversely, suppose w_0 and w_1 are in the same component. Then some simple path connects them. A simple path visits each vertex at most once, so it has at most $|V(S)| - 1 \le 2^m - 1$ edges. So the branch that guesses the vertices of this path in order reaches w_1 after at most $2^m - 1$ increments of c. At each vertex of the path before w_1 the counter is at most $2^m - 2$, so it has not reached 2^m and the branch guesses the next vertex of the path. That vertex is consistent and is a move of R from the current string, so both checks pass. When the branch holds w_1, the test $w = w_1$ succeeds, and the branch accepts.

It remains to bound the space the procedure uses. The procedure stores three items in memory: the current string w, the guessed string w', and the counter c. The strings w and w' each use m bits, and the counter c uses $m + 1$ bits. The procedure also uses the workspace of the current check of cons_S or of R. This check runs in time polynomial in the input length N, so it uses space polynomial in N, and its workspace is erased and reused from one check to the next. The input is only read. Since $w_0 \in \{0,1\}^m$ is written out in the input, $m \le N$. So the space used on every branch is at most $p(N)$ for a polynomial p, which we may take with $p(N) \ge N \ge \log N$. Applying Proposition 2.6 to this space bound therefore yields a deterministic procedure in space $O(p(N)^2)$, which is polynomial. Hence HYPOTHESIS-RECONFIGURATION is in PSPACE. □

HYPOTHESIS-CONNECTIVITY is in PSPACE as well. By Proposition 2.6, the procedure above yields a deterministic procedure $\textsc{Reach}(S, u, v)$ that runs in polynomial space. It decides, given $u, v \in V(S)$, whether they lie in the same component of $G(S)$. Given S, enumerate $\{0,1\}^m$ in lexicographic order and let w^* be the first consistent string. If there is none, $G(S)$ is empty, it is connected by convention, and the procedure accepts. This is the convention of the connectivity problem of Schwerdtfeger (2014), which counts the solution graph of an unsatisfiable formula as connected. Then enumerate $w \in \{0,1\}^m$ again, and reject if $\mathsf{cons}_S(w)$ holds and $\textsc{Reach}(S, w^*, w)$ fails. If no w causes rejection, accept.

The graph is connected exactly when every consistent w is reachable from w^*. If it is connected, then every vertex is reachable from w^*. Conversely, if every consistent w is reachable from w^*, then any two consistent strings are joined through w^*. The space used is $O(m)$ for the two enumerations, polynomial space for each check of cons_S, and the polynomial space of REACH, all reused from one string to the next. This total is polynomial in the input length, because m, given in unary, is at most that length. We will not need this PSPACE membership of HYPOTHESIS-CONNECTIVITY beyond the inherited bounds of the next section.

B A DICHOTOMY FOR CONSTRAINT-REPRESENTABLE CLASSES

We now make precise the sense in which a class can be a constraint satisfaction problem in disguise. For such classes, we observe, the questions of Appendix A are exactly the questions of Proposition 2.9.

Definition B.1 (Γ-representable class)**.** Let Γ be a constraint language. A parameterized concept class $\mathcal{C} = (m, \mathsf{E}, X)$ equipped with the single-bit relation is Γ-*representable* if there are two polynomial-time maps, both receiving m in unary in addition to the sample or formula. The first sends each sample S to a Γ-formula φ_S over the variables $z_1, \dots, z_m$ such that $\mathsf{cons}_S(w) \Leftrightarrow w \models \varphi_S$ for all $w \in \{0,1\}^m$. The second sends each Γ-formula φ over $z_1, \dots, z_m$ to a sample S_φ such that $V(S_\varphi) = \{w \in \{0,1\}^m : w \models \varphi\}$.

The first map says that the version space of S equals the solution set of φ_S, so the two solution graphs are identical as graphs on $\{0,1\}^m$ with hypercube adjacency. The second says that every Γ-

formula arises, effectively, as the consistency condition of some sample. This is what lets hardness flow back from the formula side to the learning side.

Proof of Theorem 1.2. Let $\mathcal{C}$ be Γ-representable, with the maps $S \mapsto \varphi_S$ and $\varphi \mapsto S_\varphi$ of Definition B.1. For the upper-bound directions, given an instance of HYPOTHESIS-RECONFIGURATION or HYPOTHESIS-CONNECTIVITY for $\mathcal{C}$, we compute φ_S in polynomial time. By the first condition, the solution graph $G(S)$ equals the solution graph of φ_S. So the reachability and connectivity answers coincide, the empty graph counting as connected on both sides by the convention recorded after the proof of Theorem 1.1, and the diameters of corresponding components are equal. Thus reconfiguration for $\mathcal{C}$ reduces in polynomial time to the corresponding problem for Γ. The diameter bounds of Proposition 2.9 are stated in the number of variables of φ_S. They apply unchanged, since φ_S is over the m variables $z_1, \ldots, z_m$ and m is the number of parameters.

For the hardness direction, suppose that Γ is not safely tight, and let a Γ-formula φ over $z_1, \ldots, z_m$ be given, together with two of its solutions for the source-to-target problem. For the source-to-target problem with $m = 0$, introduce one unused variable and extend both empty solutions by the bit 0. The endpoints still coincide and satisfy the formula, so this padding preserves the yes-answer and gives $m = 1$. For HYPOTHESIS-RECONFIGURATION, the two explicit solutions bound m by the input length, so we can supply m in unary to the representation map. For HYPOTHESIS-CONNECTIVITY, first drop unused variables and renumber those remaining; if no variable occurs, retain one unused variable so that $m \geq 1$. A variable occurring in no constraint replaces the solution graph by two copies of it joined by a perfect matching, which leaves connectivity unchanged, including when the graph is empty. Thus deleting unused variables preserves connectivity, and a constant-only formula has either an empty solution graph or a connected hypercube both before and after retaining one variable. After this normalization, m is at most the length of the formula plus one and can be written in unary in polynomial time. We then compute S_φ in polynomial time from the formula and unary m.

We next check that this map preserves the reachability and connectivity answers. Since $V(S_\varphi)$ is the solution set of φ, the two solutions are consistent strings and $G(S_\varphi)$ is the solution graph of φ. So their reachability in $G(S_\varphi)$ equals their reachability in the solution graph of φ, and $G(S_\varphi)$ is connected exactly when that solution graph is, both counting the empty graph as connected. This reduces the source-to-target connectivity and the connectivity problems of Γ-formulas to HYPOTHESIS-RECONFIGURATION and HYPOTHESIS-CONNECTIVITY for $\mathcal{C}$, in polynomial time.

It remains to combine this reduction with the complexity of the formula-side problems. These problems for Γ-formulas are PSPACE-complete by Proposition 2.9, so both HYPOTHESIS-RECONFIGURATION and HYPOTHESIS-CONNECTIVITY for $\mathcal{C}$ are PSPACE-hard. Both are in PSPACE by Theorem 1.1 and the paragraph after it. These bounds apply because E is computable in polynomial time by Definition 2.1, and because the single-bit relation is decidable in time $O(m)$ by comparing two strings of $\{0,1\}^m$ bit by bit. So both are PSPACE-complete when Γ is not safely tight.

It remains to show that the diameter can be superpolynomial when Γ is not safely tight. For infinitely many m, Proposition 2.9 gives a Γ-formula ψ on m variables whose solution graph has a component of diameter $2^{\Omega(\sqrt{m})}$. The graph $G(S_\psi)$ is that solution graph, on the same m variables, which are the m parameters of $\mathcal{C}$. So it has a component of the same diameter, superpolynomial in the number of parameters.

When Γ is safely tight, the reductions of the first paragraph place HYPOTHESIS-RECONFIGURATION in P and HYPOTHESIS-CONNECTIVITY in coNP. This is because Proposition 2.9 places the problems they reduce to there, and because both classes are closed under polynomial-time many-one reductions. Such a reduction maps an instance in polynomial time to an instance of polynomial size with the same answer.

When Γ is safely tight, the components of every $G(S)$ also have diameter linear in m. This diameter bound holds because these components are the components of the solution graph of φ_S, a Γ-formula on the m variables $z_1, \ldots, z_m$. □

The boundary used in Theorem 1.2 is not the one first published for Proposition 2.9 but a later correction of it.

Remark B.2 (The tractable side and the safe refinement). The identification-closure definition of safe tightness in Definition 2.8 agrees with the formulation of Schwerdtfeger (2014). The source calls a relation safely componentwise bijunctive, safely OR-free, or safely NAND-free when it and every relation obtained from it by identification of variables have the property, and Γ safely tight when all its relations share one of the three.

An identification of variables in the sense of Schwerdtfeger (2014) may merge the coordinates along any partition of them into blocks at once. Every such merge is reached by identifying two coordinates at a time, block by block, up to a permutation of the surviving coordinates. Coordinate permutations preserve adjacency, majority closure, and the existence of an OR or NAND restriction, so they preserve all three properties used in tightness. If the source convention permits an unused surviving variable, the resulting relation also has a free Boolean coordinate. Adjoining free coordinates replaces every component by its product with a cube, preserving majority closure; a binary OR or NAND restriction cannot use a free coordinate, because both target relations depend on both coordinates. Thus these free coordinates do not change any of the three properties either. Every relation of the closure therefore arises from a single relation of Γ, and all relations of the closure share one of the three properties exactly when every relation of Γ has it together with all its identification minors.

A safely tight language is tight, since every relation of Γ belongs to its identification closure.

The three conditions are what make reachability easy. Within a component of the solution graph of every formula, each of the three conditions of Definition 2.8 simplifies the graph in its own way. Safely componentwise bijunctive relations keep every distance in the component equal to the Hamming distance. Safely OR-free relations give the component a unique locally minimal solution, from which no satisfying single-bit move changes a 1 to a 0. Dually, safely NAND-free relations give the component a unique locally maximal solution, from which no satisfying single-bit move changes a 0 to a 1 (Schwerdtfeger, 2014, Lemmas 4.3 and 4.5).

A language consisting entirely of affine relations, or entirely of bijunctive relations, is tight, because every relation in it is componentwise bijunctive. For a bijunctive relation, this holds because a relation closed under majority has every component closed under majority, by the path argument in the proof of Corollary B.5. For an affine relation, it holds because every connected component of an affine relation is a subcube, by the proof of Corollary 1.4, and a subcube is closed under majority.

The language of clauses of width at most three over literals of either polarity is not tight, as the proof of Corollary 1.3 shows.

Proposition 2.9 was stated by Gopalan et al. (2009) with tight in place of safely tight on both sides. The boundary was later corrected (Schwerdtfeger, 2014). The condition for source-to-target connectivity to lie in P and for the linear-diameter bound is the safe strengthening of tightness. For every language that is not safely tight, both connectivity problems are PSPACE-complete. Theorem 1.2 is stated with the corrected boundary; the proofs of Corollaries 1.3 and 1.4 check the two instantiations against it.

The corrected boundary differs from the original one, as the following example shows.

Remark B.3 (The safe refinement is not vacuous). Some tight languages are not safely tight, so the refinement from tightness to safe tightness changes which languages are on the easy side. Let $R_1 = \{001, 110, 111\}$, which is Example 7 of Schwerdtfeger (2014). Let $R_2 = \{0,1\}^3 \setminus \{110\}$ be the relation of $\overline{z_1} \vee \overline{z_2} \vee z_3$. The language $\{R_1, R_2\}$ is tight, since both relations are OR-free. Fixing one coordinate of R_1 yields $\{01\}$, $\{10, 11\}$, $\{11\}$, or $\{00, 11\}$. The same operation on R_2 yields $\{0,1\}^2$, $\{00, 01, 11\}$, or $\{00, 01, 10\}$.

The language $\{R_1, R_2\}$ is not safely tight. Identifying the first two coordinates of R_1 yields $\{01, 10, 11\}$, which is itself the relation that OR-freeness excludes, obtained by fixing no coordinate. The relation R_2 is not NAND-free either, since fixing $z_3 = 0$ yields $\{00, 01, 10\}$. Nor is R_2 componentwise bijunctive: it is connected while $100, 010, 111 \in R_2$ and $\mathrm{maj}(100, 010, 111) = 110 \notin R_2$. It is connected because its four tuples with last coordinate 1 form the cycle $001, 011, 111, 101$ in $G(R_2)$. Each of the other three tuples, 000, 010 and 100, is adjacent to the tuple obtained by setting its last coordinate to 1. So the identification closure of $\{R_1, R_2\}$ contains a relation that is not OR-free, one that is not NAND-free, and one that is not componentwise bijunctive. Hence it is not tight.

The language's source-to-target connectivity is therefore PSPACE-complete by Proposition 2.9, whereas the boundary as first stated by Gopalan et al. (2009), with tight in place of safely tight, would have placed it in P. For whole-graph connectivity, Schwerdtfeger (2014) gives a finer classification into P, coNP-complete, and PSPACE-complete cases.

We turn to the two instantiations, which show that neither side of the dichotomy is vacuous.

A PSPACE-complete class. Let Γ_3 be the language of the fourteen relations $\{z \in \{0,1\}^k : \ell_1 \vee \cdots \vee \ell_k\}$ with $k \in \{1,2,3\}$ and $\ell_i \in \{z_i, \overline{z_i}\}$. Let $\mathcal{C}_3$ be the class whose instances are clauses $C = \ell_1 \vee \cdots \vee \ell_k$ with $1 \le k \le 3$ over literals of the variables $z_1, \ldots, z_m$, where repeated variables are allowed. Its parameter string $w \in \{0,1\}^m$ is read as an assignment. Its evaluation $\mathsf{E}(w, C)$ is the truth value of C under w. This evaluation is computable in polynomial time. The class shatters the m unit clauses $z_1, \ldots, z_m$, since $\mathsf{E}(b, z_i) = b_i$ for every $b \in \{0,1\}^m$. No set of $r > m$ instances is shattered, since shattering it would need 2^r distinct hypotheses and there are at most 2^m. Its Vapnik–Chervonenkis dimension is therefore m.

Proof of Corollary 1.3. For a sample S, let φ_S be the conjunction of two kinds of constraints. For each example $(C, 1)$ of S, the formula φ_S contains the constraint C. For each example $(C, 0)$ of S, write $C = \ell_1 \vee \cdots \vee \ell_k$. Then φ_S contains the k unit constraints $\overline{\ell_1}, \ldots, \overline{\ell_k}$. Each conjunct is a Γ_3-constraint. A clause with a repeated variable is the relation of its sign pattern applied to a tuple in which that variable repeats, and Section 2.3 allows such repeated entries. For $w \in \{0,1\}^m$, we have $\mathsf{E}(w, C) = 1$ exactly when $w \models C$. Also, $\mathsf{E}(w, C) = 0$ exactly when w falsifies every literal of C, that is, when $w \models \overline{\ell_j}$ for all j. Hence $\mathsf{cons}_S(w) \Leftrightarrow w \models \varphi_S$, and φ_S is computed in polynomial time.

Conversely, let $\varphi = C_1 \wedge \cdots \wedge C_r$ be a Γ_3-formula. We realize each constraint by one example, and S_φ is the sequence of these r examples. A constraint without constants is a clause C_i of width at most three, possibly with repeated variables. It is realized by the example $(C_i, 1)$, which is consistent with exactly the w that satisfy C_i. A constraint carrying constants is first simplified. A literal fixed to true makes the constraint vacuous, and it is realized by the example $(z_1 \vee \overline{z_1}, 1)$. By contrast, a literal fixed to false is deleted. A constraint all of whose literals are fixed to false is realized by the example $(z_1 \vee \overline{z_1}, 0)$, with which no w is consistent. The remaining constraints are clauses of width at most three, realized as above. A string is consistent with S_φ exactly when it is consistent with each example, that is, when it satisfies each constraint. So $V(S_\varphi)$ is the solution set of φ. Each example is computed from its constraint in time linear in the constraint's length, so S_φ is computed in time linear in the length of φ. Thus $\mathcal{C}_3$ is Γ_3-representable.

The language Γ_3 is not tight. The relation of $z_1 \vee z_2 \vee z_3$ is $\{0,1\}^3 \setminus \{000\}$. It has a single connected component, since every tuple with a 1 is joined to 111 by setting its zero coordinates to 1 one at a time, every intermediate tuple still having a 1. It is not closed under majority, as $100, 010, 001$ lie in it while $\mathrm{maj}(100, 010, 001) = 000$ does not. So it is not componentwise bijunctive. Fixing $z_3 = 0$ in it yields $\{01, 10, 11\}$, so it is not OR-free. Similarly, fixing $z_3 = 0$ in the relation of $\overline{z_1} \vee \overline{z_2} \vee z_3$ yields $\{00, 01, 10\}$, so that relation is not NAND-free. So none of the three properties of Definition 2.8 is shared by every relation of Γ_3, and Γ_3 is not tight. Hence Γ_3 is not safely tight, since safe tightness requires the identification closure of Γ_3 to be tight, and a property shared by every relation of that closure is shared by every relation of Γ_3, which lies in it. By Theorem 1.2, HYPOTHESIS-RECONFIGURATION for $\mathcal{C}_3$ is PSPACE-complete. □

A polynomial-time class over $\mathrm{GF}(2)$**.** Let $\mathcal{L}$ be the class of $\mathrm{GF}(2)$-linear concepts. Its instances are vectors $x \in \{0,1\}^m$. Its parameter is $w \in \{0,1\}^m$. Its evaluation is $\mathsf{E}(w, x) = \langle w, x \rangle \bmod 2$, the inner product over the binary field.

Proof of Corollary 1.4. We first describe the graph $G(S)$ explicitly, then use that description to place HYPOTHESIS-RECONFIGURATION for $\mathcal{L}$ in P and to bound the diameter of each component. Let $S = ((x_i, y_i))_{i=1}^t$ and let $U = \{j \in \{1, \ldots, m\} : x_{i,j} = 1 \text{ for some } i\}$ be the set of coordinates used by the sample. Write $\oplus$ for addition modulo two and e_j for the jth coordinate unit vector. For $w \in \{0,1\}^m$ and $j \notin U$, we have $\langle w \oplus e_j, x_i \rangle = \langle w, x_i \rangle$ for every i. This holds because the j-th term $w_j x_{i,j}$ of the inner product is 0 whatever w_j is. So $w \in V(S)$ if and only if $w \oplus e_j \in V(S)$. For $j \in U$, choose i with $x_{i,j} = 1$. Then $\langle w \oplus e_j, x_i \rangle = \langle w, x_i \rangle \oplus 1$, since flipping w_j changes the

term $w_j x_{i,j} = w_j$ by one. So at most one of w and $w \oplus e_j$ is consistent. Hence the edges of $G(S)$ are exactly the pairs $\{w, w \oplus e_j\}$ with $w \in V(S)$ and $j \notin U$.

We next identify the components of $G(S)$. The component of $w \in V(S)$ is the subcube $Q_w = \{w' \in \{0,1\}^m : w'_j = w_j \text{ for all } j \in U\}$. Every edge changes a coordinate outside U, so the component stays inside Q_w. Every element of Q_w is consistent, because $\langle w', x_i \rangle = \bigoplus_{j \in U} w'_j x_{i,j}$ depends only on the coordinates in U, on which w' agrees with w. Any $w' \in Q_w$ is reached from w by flipping the coordinates outside U on which they differ one at a time, every intermediate string lying in Q_w. Therefore $w_0, w_1 \in V(S)$ are connected in $G(S)$ if and only if $w_{0,j} = w_{1,j}$ for all $j \in U$. Computing U and performing this test take time $O((t+1)m)$, including when $t = 0$, so HYPOTHESIS-RECONFIGURATION is in P.

It remains to bound the diameter of each component. Every component is a subcube Q_w of dimension $m - |U|$. Two of its strings that differ in d coordinates are at distance exactly d. Every edge changes one coordinate, so a path between them has at least d edges, and flipping the d coordinates one at a time is a path of d edges inside Q_w. Two strings of Q_w differ only outside U, so in at most $m - |U|$ coordinates. The string w and the string differing from it at every coordinate outside U differ in exactly $m - |U|$ coordinates, so the diameter is $m - |U| \leq m$, which is linear.

The corollary is also an instance of Theorem 1.2 once the instances are restricted to Hamming weight at most three. Let Γ_{aff} be the six relations $\{z \in \{0,1\}^k : z_1 \oplus \cdots \oplus z_k = b\}$ with $k \in \{1,2,3\}$ and $b \in \{0,1\}$. Write $\operatorname{supp}(x_i) = \{j : x_{i,j} = 1\}$ for the support of a binary vector. A sample maps to the conjunction of the constraints $\bigoplus_{j \in \operatorname{supp}(x_i)} z_j = y_i$. Each is a Γ_{aff}-constraint, because $|\operatorname{supp}(x_i)| \leq 3$, and w satisfies the i-th constraint exactly when $\langle w, x_i \rangle = y_i$. An example of empty support, for which $\langle w, x_i \rangle = 0$ for every w, contributes the unary relation $\{z_1 = y_i\}$ applied to the constant 0. This relation holds for every w when $y_i = 0$, and for none when $y_i = 1$. Conversely, after canceling repeated variables in pairs and moving constants to the right-hand side, a Γ_{aff}-constraint is an equation $\bigoplus_{j \in T} z_j = b$ with $|T| \leq 3$. It is realized by the example $(\mathbf{1}_T, b)$, where $\mathbf{1}_T$ is the indicator vector of T. For $T = \emptyset$ this is the example $(\mathbf{0}, b)$, consistent with every w when $b = 0$ and with none when $b = 1$. Writing the indicator vectors takes time $O(rm + |\varphi|)$ for a formula with r constraints, which is polynomial in the input length because the representation map receives m in unary. So the restricted class is Γ_{aff}-representable.

It remains to show that Γ_{aff} is safely tight. Each relation of Γ_{aff} is the solution set of one linear equation over $\mathrm{GF}(2)$, hence a coset of the kernel of a linear form. Fix a coset $a + K$ of a linear subspace K of $\mathrm{GF}(2)^k$, and identify coordinates $i < j$. Let π be the projection dropping coordinate j. The resulting relation is the image under π of the intersection of $a + K$ with the subspace $W = \{a' : a'_i = a'_j\}$. That intersection is empty, or, for every c in it, equal to $c + (K \cap W)$. Since π is linear, the image is empty, or the coset $\pi(c) + L$ of the linear subspace $L = \pi(K \cap W)$ of $\mathrm{GF}(2)^{k-1}$. The identification of an empty relation is empty. By induction on the number of identifications, every relation in the identification closure of Γ_{aff} is empty or a coset of a linear subspace.

It remains to identify the connected components of such a coset, and to check that they are closed under majority. In a coset $a + L$, any two elements u, v have $u \oplus v \in L$. So u and v are adjacent under single-bit adjacency exactly when $u \oplus v = e_i$ for some i with $e_i \in L$. Hence every path from a stays in $a + \operatorname{span}\{e_i : e_i \in L\}$.

We now show that this containment is an equality, so that $a + \operatorname{span}\{e_i : e_i \in L\}$ is exactly the component of a. Every element of that set has the form $a \oplus \bigoplus_{i \in I} e_i$, with $e_i \in L$ for all $i \in I$. Such an element is reached from a by adding the e_i one at a time, and each intermediate string lies in $a + L$ because the span is contained in L. So the connected component of a is $a + \operatorname{span}\{e_i : e_i \in L\}$, the subcube of strings agreeing with a outside the coordinates i with $e_i \in L$. The same description holds at every element of the coset, not only at a. Every element u of $a + L$ satisfies $u + L = a + L$, so it generates the same coset as a does. The argument above therefore applies with u in place of a, and hence the component of u is $u + \operatorname{span}\{e_i : e_i \in L\}$ by the same reasoning. So every connected component of $a + L$ is a subcube.

It remains to check that a subcube is closed under majority, which finishes the safe-tightness claim. A subcube is closed under coordinatewise majority, since the majority of three strings that agree on a coordinate agrees with them there. Thus every relation in the identification closure of Γ_{aff}

is componentwise bijunctive, the empty relation vacuously, since it has no component. So Γ_{aff} is safely tight, and Theorem 1.2 gives the same conclusion for the restricted class. □

Being on the easy side of the dichotomy does not make the version space connected.

Remark B.4. The linear-diameter conclusion of Corollary 1.4 concerns connected components and is compatible with the version space being highly disconnected. Every nonempty affine subspace of $\mathrm{GF}(2)^m$ is the solution set of a system of equations $\langle w, x_i \rangle = y_i$. So it is also the version space of the sample $((x_i, y_i))_i$. An affine subspace of minimum distance at least two contains no two strings at Hamming distance one, so its induced solution graph is a set of isolated vertices. For instance, for $m \geq 2$, consider the one-example sample $S = ((\mathbf{1}, 0))$, with $\mathbf{1}$ the all-ones vector. Its version space is the set of 2^{m-1} strings of even weight, any two of which differ in at least two coordinates. So $G(S)$ has 2^{m-1} components, each a single vertex. The content of the easy side is that reachability is efficiently decidable, not that it usually holds.

Unless $\mathsf{P} = \mathsf{PSPACE}$, the width three of Corollary 1.3 is the least at which the clause class is hard.

Corollary B.5 (Width two is easy)**.** *Let $\mathcal{C}_2$ be the class $\mathcal{C}_3$ with its instances restricted to clauses of width at most two. Then* HYPOTHESIS-RECONFIGURATION *for $\mathcal{C}_2$ under single-bit moves is in* P *and every connected component of every solution graph has linear diameter. Hence, unless* $\mathsf{P} = \mathsf{PSPACE}$*, width three is the least width at which* HYPOTHESIS-RECONFIGURATION *for the clause class is* PSPACE*-complete.*

Proof. Let $\Gamma_2 \subseteq \Gamma_3$ consist of the six relations of clauses of width at most two. The proof of Corollary 1.3, restricted to width two, shows that $\mathcal{C}_2$ is Γ_2-representable. The formula of a sample uses the clauses of the examples labeled 1 and the unit clauses of the negated literals of the examples labeled 0, all of width at most two. The sample of a formula uses the clauses of the formula, together with the width-two instance $z_1 \vee \overline{z_1}$ used for the simplification of constants.

Every relation of Γ_2 is closed under coordinatewise majority. If a, b, c satisfy the clause $\ell_1 \vee \ell_2$ while $\mathrm{maj}(a, b, c)$ falsifies both literals, then each literal is false under at least two of a, b, c. This is because each coordinate of the majority takes the value it takes in at least two of them. Two subsets of the three argument positions with at least two positions each must intersect, as $2 + 2 > 3$. So one of the three falsifies both literals and does not satisfy the clause, a contradiction.

For a unit clause ℓ, the relation is closed as well. If a, b, c all satisfy ℓ, then the coordinate of the variable of ℓ takes the satisfying value in each of a, b, c, and hence in $\mathrm{maj}(a, b, c)$.

It remains to check that identification minors of these relations are closed under majority as well. An identification minor of the relation of $\ell_1 \vee \ell_2$ is the unary relation $\{1\}$ or $\{0\}$ when the two literals have the same polarity. It is the full relation $\{0, 1\}$ when they have opposite polarity. A unary relation admits no identification. Every unary relation is closed under majority, since the majority of three elements of a subset of $\{0, 1\}$ is one of them.

A relation R closed under majority is componentwise bijunctive. Let $a, b, c \in R$, with a and b in one component of $G(R)$. Let $a = a_0, \ldots, a_k = b$ be a path in R. The tuples $\mathrm{maj}(a_j, b, c)$ for $0 \leq j \leq k$ lie in R, since R is closed under majority. Consecutive ones, $\mathrm{maj}(a_j, b, c)$ and $\mathrm{maj}(a_{j+1}, b, c)$, agree on every coordinate where a_j and a_{j+1} agree. So they differ in at most one coordinate, and are equal or adjacent in $G(R)$. The last of these tuples is $\mathrm{maj}(b, b, c) = b$, since on every coordinate two of its three arguments take the value of b. So all of these tuples lie in the component of b, including the first, $\mathrm{maj}(a, b, c)$. Since a and b were arbitrary tuples of one component, and c an arbitrary tuple of R, that component is closed under majority.

Hence every relation in the identification closure of Γ_2 is componentwise bijunctive, Γ_2 is safely tight, and Theorem 1.2 gives the first claim. An instance of width at most one is an instance of $\mathcal{C}_2$. So HYPOTHESIS-RECONFIGURATION for clauses of width at most one is the restriction of HYPOTHESIS-RECONFIGURATION for $\mathcal{C}_2$ to such samples, and is in P as well. If HYPOTHESIS-RECONFIGURATION for width at most one or at most two were PSPACE-complete, every problem in PSPACE would reduce in polynomial time to a problem in P, and so would lie in P itself. This, together with $\mathsf{P} \subseteq \mathsf{PSPACE}$, gives $\mathsf{P} = \mathsf{PSPACE}$. Unless $\mathsf{P} = \mathsf{PSPACE}$, no width below three is PSPACE-complete, while width three is by Corollary 1.3. □

C AGNOSTIC RECONFIGURATION

The realizable problem of the previous sections takes feasibility to be exact consistency with the sample. In practice a hypothesis is accepted when it fits the data up to a loss tolerance, and the natural reconfiguration question is whether two such hypotheses are joined by a path along which the loss never exceeds the tolerance. The constraint-language dictionary of Appendix B does not by itself classify this relaxation: a bound on the total number of misclassified examples introduces a global cardinality constraint. It is also the decision-problem form of mode connectivity, the empirically observed joining of independently trained models by low-loss paths (Garipov et al., 2018; Draxler et al., 2018).

C.1 THE LOSS-THRESHOLD PROBLEM AND THE RECONFIGURATION BOTTLENECK

We first define the error count and the empirical loss of a hypothesis on a sample, and the set of hypotheses whose error count stays within a budget.

Definition C.1 (Empirical loss and feasible set)**.** For a class $\mathcal{C} = (m, \mathsf{E}, X)$ and a sample $S = ((x_i, y_i))_{i=1}^{t}$, the *error count* of $w \in \{0,1\}^m$ is $e_S(w) = |\{i : \mathsf{E}(w, x_i) \neq y_i\}|$ and the *empirical loss*, for $t \geq 1$, is $\widehat{L}_S(w) = e_S(w)/t$. For an integer budget $k \in \{0, \ldots, t\}$ the *feasible set* is $F_k = \{w \in \{0,1\}^m : e_S(w) \leq k\}$.

The reconfiguration question then asks whether two feasible hypotheses are joined inside the feasible set, and how large a budget that needs.

Definition C.2 (Agnostic reconfiguration)**.** Fix a class $\mathcal{C}$. Given a sample S, a budget k, and two feasible hypotheses w_0 and w_1, the problem AGNOSTIC-RECONFIGURATION asks whether w_0 and w_1 lie in the same connected component of F_k under single-bit moves. Here F_k is viewed as an induced subgraph of the hypercube. The *reconfiguration bottleneck* between w_0 and w_1 is

$$\beta_S(w_0, w_1) = \min_{P} \max_{w \in P} e_S(w),$$

where P ranges over single-bit paths from w_0 to w_1 in the hypercube.

The realizable problem HYPOTHESIS-RECONFIGURATION of Definition 2.5 under the single-bit relation is the special case $k = 0$ of AGNOSTIC-RECONFIGURATION. Indeed, $e_S(w) = 0$ exactly when w is consistent with S, so $F_0 = V(S)$. We first locate the agnostic problem in the complexity hierarchy.

Proposition C.3 (Membership and worst-case hardness)**.** AGNOSTIC-RECONFIGURATION *is in* PSPACE *for every concept class with polynomial-time evaluation, and is* PSPACE*-complete for the class $\mathcal{C}_3$ of Corollary 1.3.*

Proof. For membership, run the procedure of the proof of Theorem 1.1, with R the single-bit relation and with the test $\mathsf{cons}_S(w)$ replaced by $e_S(w) \leq k$. This test is decided in polynomial time, by evaluating $\mathsf{E}(w, x_i)$ for each i and counting the disagreements with y_i. The graph searched is then F_k under single-bit moves, on at most 2^m vertices. A simple path in it from w_0 to w_1 therefore has at most $2^m - 1$ edges, so the counter bound of that proof is unchanged. Every accepting computation holds only strings verified to satisfy $e_S(w) \leq k$, each obtained from the previous one by a verified single-bit move, so it traces a path in F_k from w_0 to w_1. Conversely, the branch that guesses the vertices of a simple path in F_k from w_0 to w_1, in order, accepts before the counter reaches 2^m. Besides w, w' and the counter c, the procedure now also stores the number of disagreements counted so far. This count is an integer between 0 and t, held in $O(\log(t+2))$ bits, including when $t = 0$. The space used is therefore still polynomial in the input length, so Proposition 2.6 applies as there, and AGNOSTIC-RECONFIGURATION is in PSPACE.

For hardness, take the class $\mathcal{C}_3$ of Corollary 1.3, for which HYPOTHESIS-RECONFIGURATION under the single-bit relation is PSPACE-complete. Corollary 1.3 is proved through Theorem 1.2, whose move relation is the single-bit one. Map an instance (S, w_0, w_1) of HYPOTHESIS-RECONFIGURATION for $\mathcal{C}_3$ to the instance $(S, 0, w_0, w_1)$ of AGNOSTIC-RECONFIGURATION. This map is computed in polynomial time. This is a legal instance: $e_S(w) = 0$ exactly when $\mathsf{E}(w, x_i) = y_i$ for every i, so $F_0 = V(S)$. The strings w_0, w_1 are consistent with S, so they

are feasible for $k = 0$. The graph F_0 under single-bit moves is then the solution graph $G(S)$ of Definition 2.4 for the single-bit relation, so the two instances have the same answer. Hence AGNOSTIC-RECONFIGURATION for $\mathcal{C}_3$ is PSPACE-hard, and with membership it is PSPACE-complete. □

Thus relaxing feasibility to a loss threshold does not lower the worst-case complexity. The next lemma records the structure that the relaxation does add: connectivity that is monotone in the loss budget, with a threshold given by the bottleneck.

Lemma C.4 (Monotone connectivity threshold)**.** *Fix $\mathcal{C}$, S, and w_0, w_1. For $k \le k'$ one has $F_k \subseteq F_{k'}$ as induced subgraphs, so connectivity of w_0 and w_1 in F_k is monotone in k. Moreover w_0 and w_1 are connected in F_k if and only if $k \ge \beta_S(w_0, w_1)$, and*

$$\max\left(e_S(w_0), e_S(w_1)\right) \le \beta_S(w_0, w_1) \le t.$$

Proof. The inclusion $F_k \subseteq F_{k'}$ holds because $e_S(w) \le k$ implies $e_S(w) \le k'$. Passing to a larger induced subgraph preserves existing paths, so connectivity of w_0 and w_1 in F_k is monotone in k. The single-bit paths from w_0 to w_1 in the hypercube form a nonempty set, since flipping, one at a time, the bits on which w_0 and w_1 differ produces such a path. Each such path P has $\max_{w \in P} e_S(w) \in \{0, \dots, t\}$, because every error count lies in $\{0, \dots, t\}$. A nonempty set of values in the finite set $\{0, \dots, t\}$ has a least element, so the minimum defining $\beta_S(w_0, w_1)$ exists and is attained by some path. If $k \ge \beta_S(w_0, w_1)$, a path attaining the bottleneck has every vertex of error count at most $\beta_S(w_0, w_1) \le k$, and hence lies in F_k. Conversely, any path inside F_k has maximum error count at most k, so $\beta_S(w_0, w_1) \le k$. The lower bound holds because w_0 and w_1 lie on every path P, so $\max_{w \in P} e_S(w) \ge \max\left(e_S(w_0), e_S(w_1)\right)$ for every P, and in particular for a path attaining the minimum. The upper bound holds because the path of bit flips above exists and each of its vertices has error count at most t. □

The bottleneck β_S is a minimax-path objective, and $t - \beta_S(w_0, w_1)$ is the corresponding maxmin objective, the largest number of examples that can be kept correctly classified at every step of a path from w_0 to w_1. For $t \ge 1$, since error counts are integers, $\{w : \widehat{L}_S(w) \le \lambda\} = F_{\lfloor \lambda t \rfloor}$ for $0 \le \lambda \le 1$, and by Lemma C.4 the pair is connected in it exactly when $\lfloor \lambda t \rfloor \ge \beta_S(w_0, w_1)$. Since $\beta_S(w_0, w_1)$ is an integer, this holds exactly when $\lambda t \ge \beta_S(w_0, w_1)$, that is, when $\lambda \ge \beta_S(w_0, w_1)/t$. So $\beta_S(w_0, w_1)/t$ is the least empirical loss at which the pair becomes connected. It is the critical value of the nested family $F_0 \subseteq F_1 \subseteq \cdots \subseteq F_t$ along which feasibility is relaxed. The maxmin objective $t - \beta_S(w_0, w_1)$ is the optimization variant of reconfiguration studied for satisfiability and related problems (Ito et al., 2011; Hirahara & Ohsaka, 2024), here specialized to the empirical loss of a concept class rather than to a fixed constraint language. The upper bound $\beta_S \le t$ expresses the trivial fact that once all hypotheses are admitted the hypercube is connected.

The hardness of Proposition C.3 is exact, and it leaves open whether the bottleneck can nonetheless be *approximated*: whether one can estimate in polynomial time how much loss a pair must tolerate in order to become connected. Unless P = PSPACE, no polynomial-time algorithm gives a finite multiplicative approximation on every instance. The tool is the inapproximability of the same maxmin objective for satisfiability, obtained from the probabilistically checkable reconfiguration proof theorem of Hirahara & Ohsaka (2024) by the gap-preserving reductions of Ohsaka (2026).

Proposition C.5 (Inapproximability of maxmin satisfiability reconfiguration; Hirahara & Ohsaka, 2024, Corollary 1.6 through Ohsaka, 2026, Theorem 3.1)**.** *There is a constant $\epsilon_0 > 0$ for which the following promise problem is* PSPACE*-hard. The input is a 3-CNF formula φ on clauses $c_1, \dots, c_C$, each with exactly three distinct literals, together with two satisfying assignments s and g. The reconfiguration sequences considered are those in which consecutive assignments differ in a single variable and the intermediate assignments are unrestricted. Under the promise that exactly one of the following holds, it is to be decided which. Either some reconfiguration sequence from s to g satisfies every clause of φ at every step, or every such sequence has a step at which more than $\epsilon_0 C$ clauses of φ are violated.*

The bottleneck inherits this gap through the representation already used for the realizable hardness.

Theorem C.6 (Inapproximability of the bottleneck; restatement of Theorem 1.5)**.** *There is a rational constant $\epsilon_0 > 0$ such that, for the class $\mathcal{C}_3$ of Corollary 1.3, the following promise problem is* PSPACE*-hard: given a sample S of size t and two consistent hypotheses w_0 and w_1, decide whether*

$\beta_S(w_0, w_1) = 0$ *or* $\beta_S(w_0, w_1) > \epsilon_0 t$. *Consequently, unless* $\mathsf{P} = \mathsf{PSPACE}$*, no polynomial-time algorithm approximates* $\beta_S(w_0, w_1)$ *within any multiplicative factor.*

Proof. We reduce from Proposition C.5. We may and do take ϵ_0 rational. Indeed, fix a rational $\epsilon_0' \in (0, \epsilon_0]$. If every sequence of an instance has a step violating more than $\epsilon_0 C$ clauses, then every sequence also has a step violating more than $\epsilon_0' C$ clauses. So the promise problem with ϵ_0' in place of ϵ_0 contains that of Proposition C.5, with the same answers, and it is PSPACE-hard as well. From here on we write ϵ_0 for ϵ_0', which is the rational constant of the theorem.

We now build the reduction. Let φ have clauses $c_1, \ldots, c_C$ on m variables, named $z_1, \ldots, z_m$, together with two satisfying assignments $s, g \in \{0,1\}^m$. If $m = 0$, the satisfying assignments coincide and there are no three-literal clauses; map this yes-instance to an empty sample with parameter length one and endpoints $0, 0$. Hence assume $m \geq 1$ below. We work with the class $\mathcal{C}_3$ of parameter length m. Recall that the instances of $\mathcal{C}_3$ are clauses of width at most three and its hypotheses are assignments evaluated on a clause. Form the sample $S = ((c_1, 1), \ldots, (c_C, 1))$. Each c_i has three literals, so it is an instance of $\mathcal{C}_3$, and S has size $t = C$: since a sample is a sequence, repeated clauses are counted separately. This sample is the representation of Corollary 1.3: a hypothesis w has $\mathsf{E}(w, c_i) = 1$ exactly when w satisfies c_i. Since every example of S carries the label 1, the error $e_S(w)$ is the number of clauses of φ violated by w. Hence w is consistent with S exactly when it satisfies φ. In particular, s and g are consistent.

It remains to check that the reduction is correct. The single-bit moves of $\mathcal{C}_3$ are single-variable flips, so every single-bit path from $w_0 = s$ to $w_1 = g$ is a reconfiguration sequence of Proposition C.5. Conversely, every such sequence becomes a single-bit path from s to g, by repeatedly deleting the portion between two visits of the same assignment; the resulting path's assignments are among those of the sequence. For such a path P the quantity $\max_{w \in P} e_S(w)$ is the largest number of clauses violated simultaneously along it. In the first case of Proposition C.5, some reconfiguration sequence violates no clause at any step, and so does the path P obtained from it by deleting repetitions. So $\max_{w \in P} e_S(w) = 0$, hence $0 \leq \beta_S(s, g) \leq 0$, that is, $\beta_S(s, g) = 0$. In the second case every path has a step violating more than $\epsilon_0 C = \epsilon_0 t$ clauses, so $\max_{w \in P} e_S(w) > \epsilon_0 t$ for every P. These values all lie in the finite set $\{0, \ldots, t\}$, so their minimum $\beta_S(s, g)$ is one of them, and $\beta_S(s, g) > \epsilon_0 t$.

The sample lists the clauses of φ with the label 1, and the hypotheses are s and g themselves, so the map is computed in polynomial time. It sends instances of the first kind to instances with $\beta_S(s, g) = 0$, and instances of the second kind to instances with $\beta_S(s, g) > \epsilon_0 t$. The promise problem of the theorem is therefore PSPACE-hard.

For the approximation consequence, suppose a polynomial-time algorithm returns on every instance I a binary rational value b with $\beta_S(w_0, w_1)/\alpha \leq b \leq \alpha \beta_S(w_0, w_1)$ for a finite factor $\alpha = \alpha(I) \geq 1$, which may depend on the input. This covers approximation from above, from below, and from both sides. On instances of the first kind $0 \leq b \leq \alpha \cdot 0 = 0$, so $b = 0$. For instances of the second kind, $b \geq \beta_S(w_0, w_1)/\alpha > \epsilon_0 t/\alpha \geq 0$, so $b > 0$. Testing whether $b = 0$ therefore decides a PSPACE-hard promise problem in polynomial time. Composing this test with the reductions that witness hardness places every problem in PSPACE in P, that is, $\mathsf{P} = \mathsf{PSPACE}$. □

The gap of Theorem C.6 is stated for multiplicative approximation. The next corollary shows that it also rules out additive approximation, which is the form a learner meets in loss terms.

Corollary C.7 (Additive inapproximability)**.** *Unless* $\mathsf{P} = \mathsf{PSPACE}$*, no polynomial-time algorithm computes, for the class* $\mathcal{C}_3$*, a value within additive error* $\epsilon_0 t/2$ *of* $\beta_S(w_0, w_1)$ *on every sample* S *of size* t *and every pair of consistent hypotheses* w_0, w_1*. Equivalently, none estimates the threshold empirical loss* $\beta_S(w_0, w_1)/t$ *within* $\epsilon_0/2$ *on samples of size* $t \geq 1$*.*

Proof. Suppose a polynomial-time algorithm returns a binary rational value b with $|b - \beta_S(w_0, w_1)| \leq \epsilon_0 t/2$. On an instance of the promise problem of Theorem C.6 with $\beta_S(w_0, w_1) = 0$ this gives $b \leq \epsilon_0 t/2$. On one with $\beta_S(w_0, w_1) > \epsilon_0 t$ it gives $b \geq \beta_S(w_0, w_1) - \epsilon_0 t/2 > \epsilon_0 t - \epsilon_0 t/2 = \epsilon_0 t/2$. So answering according to whether $b \leq \epsilon_0 t/2$ decides the promise problem in polynomial time, since ϵ_0 is rational and the comparison is therefore between rational numbers. Hence $\mathsf{P} = \mathsf{PSPACE}$ follows, as in the proof of Theorem C.6. For $t = 0$, the unnormalized bottleneck is zero and can be returned directly. For $t \geq 1$, an estimate of $\beta_S(w_0, w_1)/t$ within $\epsilon_0/2$,

multiplied by t, is an estimate of $\beta_S(w_0, w_1)$ within $\epsilon_0 t/2$. Conversely, an estimate of $\beta_S(w_0, w_1)$ within $\epsilon_0 t/2$, divided by t, is an estimate of $\beta_S(w_0, w_1)/t$ within $\epsilon_0/2$. □

The constant ϵ_0 of Proposition C.5 is not explicit. For clauses of large width an explicit gap is known, by the asymptotically optimal inapproximability of Hirahara & Ohsaka (2025). We import their result with our own notation: q in place of their k, since k here already denotes a loss budget, and s, g in place of their $\alpha_{\mathrm{start}}, \alpha_{\mathrm{end}}$. As before, a reconfiguration sequence is a sequence of assignments in which consecutive ones differ in a single variable and the intermediate ones are unrestricted.

Proposition C.8 (Hirahara & Ohsaka, 2025, Theorem 1.3)**.** *There is an integer q_0 such that for every integer $q \geq q_0$ the following promise problem is* PSPACE*-hard. The input is a satisfiable formula φ in conjunctive normal form each of whose C clauses contains exactly q literals, together with two satisfying assignments s and g. Either some reconfiguration sequence from s to g consists of satisfying assignments of φ, or every reconfiguration sequence from s to g contains an assignment violating more than $\frac{C}{10q}$ clauses of φ.*

Through the same representation as in the proof of Theorem C.6, this gap passes to the bottleneck of the class of wider clauses.

Corollary C.9 (Explicit gap for wide clauses)**.** *For an integer $q \geq 3$ let $\mathcal{C}_q$ be the class $\mathcal{C}_3$ with clauses of width at most q in place of width at most three. For every $q \geq \max(q_0, 3)$, the following promise problem for $\mathcal{C}_q$ is* PSPACE*-hard: given a sample S of size t and two consistent hypotheses w_0 and w_1, decide whether $\beta_S(w_0, w_1) = 0$ or $\beta_S(w_0, w_1) > \frac{t}{10q}$. Consequently, unless* P $=$ PSPACE*, no polynomial-time algorithm estimates the threshold empirical loss $\beta_S(w_0, w_1)/t$ for $\mathcal{C}_q$ within $\frac{1}{20q}$ on every sample with $t \geq 1$.*

Proof. The evaluation of $\mathcal{C}_q$ is the truth value of a clause of at most q literals under an assignment, computable in polynomial time. Let (φ, s, g) be an instance of Proposition C.8, with clauses $c_1, \ldots, c_C$ on m variables $z_1, \ldots, z_m$. If $m = 0$, map the empty-formula yes-instance to an empty sample with parameter length one and endpoints $0, 0$; hence assume $m \geq 1$. We work with the class $\mathcal{C}_q$ of parameter length m. Form the sample $S = ((c_1, 1), \ldots, (c_C, 1))$. Each c_i has exactly q literals, so it is an instance of $\mathcal{C}_q$, and S has size $t = C$. A hypothesis w has $\mathsf{E}(w, c_i) = 1$ exactly when w satisfies c_i. So $e_S(w)$ is the number of clauses of φ violated by w. Since s and g satisfy φ, they are consistent with S.

It remains to check that the reduction is correct. Every single-bit path from s to g is a reconfiguration sequence from s to g. Conversely, repeatedly deleting the portion of a reconfiguration sequence between two visits of the same assignment turns the sequence into such a path, whose assignments are among those of the sequence. Along a path P, the quantity $\max_{w \in P} e_S(w)$ is the largest number of clauses violated at one step. In the first case some reconfiguration sequence consists of satisfying assignments, and so does the path P obtained from it by deleting repetitions. So $\max_{w \in P} e_S(w) = 0$, and hence $\beta_S(s, g) = 0$, since β_S is a minimum of nonnegative values. In the second case every path has $\max_{w \in P} e_S(w) > \frac{C}{10q} = \frac{t}{10q}$. These values all lie in the finite set $\{0, \ldots, t\}$, so their minimum $\beta_S(s, g)$ is one of them, and $\beta_S(s, g) > \frac{t}{10q}$. The map is computed in polynomial time, so the promise problem of the corollary is PSPACE-hard.

For the second statement, answer yes directly when $t = 0$, since then $\beta_S = 0$. For $t \geq 1$, consider an algorithm returning a binary rational estimate of $\beta_S(w_0, w_1)/t$ within $\frac{1}{20q}$; multiplied by t, this estimates $\beta_S(w_0, w_1)$ within $\frac{t}{20q}$. The scaled estimate is therefore at most $\frac{t}{20q}$ when $\beta_S(w_0, w_1) = 0$, and greater than $\frac{t}{10q} - \frac{t}{20q} = \frac{t}{20q}$ when $\beta_S(w_0, w_1) > \frac{t}{10q}$. So comparing it with the rational $\frac{t}{20q}$ decides the promise problem in polynomial time, and P $=$ PSPACE follows, as in the proof of Theorem C.6. □

Corollary C.7 concerns $\mathcal{C}_3$, and its constant ϵ_0 comes from the probabilistically checkable reconfiguration proof theorem, so it is not specified. The gap here is different: it is explicit as a function of the width, although the threshold q_0 beyond which it holds is not. This gap is of the order $\frac{1}{q}$, which Hirahara & Ohsaka (2025) show to be optimal for the satisfiability problem it comes from, by a

polynomial-time $\left(1 - \frac{1}{q-1} - \frac{1}{q}\right)$-approximation of the maxmin fraction of satisfied clauses for every $q \geq 3$. For nonempty positive-labeled samples whose clauses have exactly q distinct literals and two consistent endpoint hypotheses, that algorithm also gives a path with normalized bottleneck at most $\frac{1}{q-1} + \frac{1}{q}$. It does not directly cover every sample of $\mathcal{C}_q$, which allows shorter clauses, repeated literals, and negative labels.

C.2 Collapse on Well-Behaved Landscapes

We turn to the regime in which the relaxation becomes easy. The phenomenon is familiar in continuous form: a convex loss on a convex domain has convex, hence connected, sublevel sets, and certain over-parameterized networks have connected sublevel sets free of bad valleys (Nguyen, 2019). The following is the elementary discrete counterpart, stated for an arbitrary configuration graph and loss function.

Definition C.10 (Configuration graph and good landscapes)**.** Let $G = (V, E)$ be a finite graph with V nonempty and $e\colon V \to \mathbb{R}_{\geq 0}$ a loss. Write $m_\star = \min_{v \in V} e(v)$ and $M = \{v \in V : e(v) = m_\star\}$. The loss e has *no bad local minimum* if every v with $e(v) > m_\star$ has a neighbor u with $e(u) < e(v)$.

When the loss has no bad local minimum and its minimizers induce a connected subgraph, every nonempty sublevel set is connected, and the bottleneck between two hypotheses equals the larger of their two losses.

Theorem C.11 (Sublevel connectivity implies collapse; restatement of Theorem 1.6)**.** *Let $e\colon V \to \mathbb{R}_{\geq 0}$ have no bad local minimum and let the subgraph $G[M]$ on the minimizers be connected. Then for every $k \geq m_\star$ the sublevel set $\{v : e(v) \leq k\}$ induces a connected subgraph of G. In particular, taking G to be the hypercube and $e = e_S$, the sample error count, if e_S has no bad local minimum and its minimizers induce a connected subgraph, then any two feasible hypotheses are connected in F_k, so* Agnostic-Reconfiguration *is a yes-instance and $\beta_S(w_0, w_1) = \max(e_S(w_0), e_S(w_1))$.*

Proof. Fix $k \geq m_\star$ and let u, v be two vertices with $e(u), e(v) \leq k$. Starting from u, repeatedly move to a neighbor of strictly smaller loss. By the absence of bad local minima this is possible as long as the current loss exceeds $m_\star$, and the process terminates because the loss strictly decreases at each step and V is finite. It stops only when the current loss is $m_\star$, since otherwise a neighbor of smaller loss exists. So the resulting walk ends at some $\hat{u} \in M$, and visits only vertices of loss at most $e(u) \leq k$; hence it lies in the sublevel set. The descent from u used nothing about u beyond $e(u) \leq k$, so it applies to v unchanged. Started from v, it ends at some $\hat{v} \in M$ through vertices of loss at most $e(v) \leq k$, so v is joined within the sublevel set to $\hat{v}$.

It remains to connect $\hat{u}$ to $\hat{v}$. The subgraph $G[M]$ is connected, and M is contained in the sublevel set, because $m_\star \leq k$. So there is a path from $\hat{u}$ to $\hat{v}$ lying in M. Concatenate the descent from u, the path from $\hat{u}$ to $\hat{v}$ in M, and the reversed descent from v. This concatenation gives a walk from u to v that stays within the sublevel set. The sublevel set is therefore connected.

We now bound the loss along this walk, to get the bottleneck. This walk has maximum loss $\max(e(u), e(v))$. Its first piece has losses at most $e(u)$, its last piece losses at most $e(v)$, and its middle piece lies in M, where the loss is $m_\star \leq \min(e(u), e(v))$. The maximum is attained, at u or at v, since both lie on the walk. Deleting repeatedly the portion of the walk between two visits of the same vertex leaves a path from u to v whose vertices are among those of the walk, and include u and v. Its maximum loss is therefore also $\max(e(u), e(v))$. Every path from u to v has maximum loss at least $\max(e(u), e(v))$, because u and v lie on it. So the least maximum loss over paths from u to v is $\max(e(u), e(v))$.

For the particular case, let G be the hypercube, $e = e_S$, and $w_0, w_1 \in F_k$. Then $k \geq e_S(w_0) \geq m_\star$, so by the first part $F_k = \{w : e_S(w) \leq k\}$ induces a connected subgraph, w_0 and w_1 are connected in F_k, and the instance of Agnostic-Reconfiguration is a yes-instance. The paths of the hypercube are the single-bit paths of Definition C.2, so the least maximum loss above, with $u = w_0$ and $v = w_1$, is $\beta_S(w_0, w_1) = \max(e_S(w_0), e_S(w_1))$. □

Small examples show what each hypothesis of Theorem C.11 contributes.

Remark C.12 (The two hypotheses). Neither hypothesis of Theorem C.11 can be dropped, and together they are sufficient rather than necessary. The path $v_1v_2v_3$ on three vertices shows all three points. With losses $1, 2, 0$ the only minimizer is v_3, the vertex v_1 has loss above the minimum and its only neighbor has larger loss, and $\{e \leq 1\} = \{v_1, v_3\}$ is disconnected. Take instead losses $0, 1, 0$: every vertex of positive loss has a neighbor of smaller loss, the minimizers v_1, v_3 are not adjacent, and $\{e \leq 0\}$ is disconnected. Now take losses $1, 1, 0$: the only minimizer is v_3, so $G[M]$ is connected, but the vertex v_1 has loss above the minimum and no neighbor of strictly smaller loss, so e has a bad local minimum in the sense of Definition C.10. Yet every sublevel set $\{e \leq k\}$ with $k \geq 0$ is connected: it is $\{v_3\}$ for $k < 1$ and the whole path for $k \geq 1$. The second hypothesis, unlike the first, is necessary for the conclusion on its own: the sublevel set at $k = m_\star$ is M, so it is connected only if $G[M]$ is.

We now turn to the converse direction. A *strict local minimum* is a vertex all of whose neighbors have strictly larger loss. What holds conversely is that connected sublevel sets exclude a strict local minimum above the minimum. If such a vertex v has $e(v) > m_\star$ and $\{e \leq e(v)\}$ is connected, then this set contains v and a minimizer, which differs from v, so a path inside it joins them. The second vertex of that path is a neighbor of v of loss at most $e(v)$, which a strict local minimum does not have. The continuous form of this direction is recalled in Remark C.13.

The theorem has a counterpart in the study of continuous loss landscapes.

Remark C.13. Theorem C.11 is the discrete shadow of a known picture. The continuous form of its descent step is Lemma 2 of Venturi et al. (2019): a nonincreasing path to a global minimizer from every point excludes spurious valleys. The converse direction is Proposition 2.1 of Freeman & Bruna (2017): connected sublevel sets exclude strict spurious minima. A convex loss on a convex domain, when its minimum is attained, satisfies the continuous descent and minimizer-connectivity conditions. Every local minimum of such a loss is global, and its set of minimizers is convex, hence connected. The sublevel sets of the loss are convex as well, hence connected, as the theorem concludes.

Two known results place particular network landscapes in this regime. Under the data, activation and architecture assumptions stated in Section 1.3, Nguyen (2019) proves connected sublevel sets and no bad valleys. Under their hypotheses, Freeman & Bruna (2017) prove connectivity for one-hidden-layer rectified networks up to an energy gap that vanishes as the width grows. Exact path connectivity of sublevel sets makes the continuous analogue of AGNOSTIC-RECONFIGURATION an automatic yes-instance for feasible endpoints. When the sublevel sets are path-connected, two parameters are joined by a path along which the loss never exceeds the larger of their two losses: the continuous form of the mode connectivity observed between independently trained networks (Garipov et al., 2018; Draxler et al., 2018). Where the sublevel sets are connected only up to an energy gap, the same holds with the gap added to that bound.

The opposite regime is equally meaningful. The overlap gap property (Gamarnik, 2021) describes landscapes in which no two near-optimal solutions lie at a distance inside a forbidden intermediate band, so that the near-optimal sets are clustered and mutually distant. In the present language, fix nonnegative integers d_1 and d_2 with $d_1 + 1 < d_2$. Suppose no two hypotheses of error count at most k lie at Hamming distance strictly between d_1 and d_2. Then any two hypotheses u, v of error count at most k at Hamming distance at least d_2 have $\beta_S(u, v) > k$. Indeed, along a single-bit path from u to v, the distance to u changes by one at each step, from 0 to $d_H(u, v) \geq d_2$. So the path passes through a hypothesis at distance $d_1 + 1$ from u. Since u has error count at most k and $d_1 < d_1 + 1 < d_2$, the assumption applied to u and this hypothesis forces the error count of this hypothesis to exceed k. So every single-bit path from u to v has a vertex of error count greater than k, and $\beta_S(u, v)$, the least over these paths of the largest error count, exceeds k. Whether such instances are also computationally hard is a separate question, which Proposition C.3 does not address.

C.3 EXPLICITLY REPRESENTED HYPOTHESIS SPACES

The collapse of Theorem C.11 is a structural sufficient condition. When the hypothesis space is small enough to be written down, the bottleneck and the reconfiguration question become unconditionally tractable, by the minimum-bottleneck path algorithm of Lemma 2.7.

Proposition C.14 (Tractability for polynomial-size graphs)**.** *Let H be a finite graph on a set of hypotheses, given explicitly together with the error count $e_S(w)$ of each vertex, and define $\beta_H(w_0, w_1)$*

and AGNOSTIC-RECONFIGURATION *as in Definition C.2 with paths in H in place of single-bit paths in the hypercube, with $\beta_H(w_0, w_1) = \infty$ when no path of H joins w_0 and w_1. Then $\beta_H(w_0, w_1)$ and* AGNOSTIC-RECONFIGURATION *are computable in time polynomial in the full weighted graph encoding and the budget encoding, and hence in time polynomial in m and t whenever H has a number of vertices polynomial in m and t and $0 \leq k \leq t$.*

Proof. On the explicit graph the error count e_S is a vertex weight, and $\beta_H(w_0, w_1)$ is the minimum over w_0–w_1 paths of H of the maximum vertex weight on the path. This is the quantity of Lemma 2.7, with w_0 and w_1 as its two vertices. Whether a path of H joins w_0 and w_1 is decided by a search of H in polynomial time. If none does, then $\beta_H(w_0, w_1) = \infty$ and AGNOSTIC-RECONFIGURATION rejects for every budget. Otherwise the minimum is attained, since the maxima range over the finite set of error counts of the vertices of H. AGNOSTIC-RECONFIGURATION with budget k then accepts precisely when $\beta_H(w_0, w_1) \leq k$. When $\beta_H(w_0, w_1) \leq k$, a path attaining $\beta_H(w_0, w_1)$ lies in $\{e_S \leq k\}$, so AGNOSTIC-RECONFIGURATION accepts. Conversely, if AGNOSTIC-RECONFIGURATION accepts, a path of H inside $\{e_S \leq k\}$ joins w_0 and w_1 and has maximum error count at most k, so $\beta_H(w_0, w_1) \leq k$. Both $\beta_H(w_0, w_1)$ and AGNOSTIC-RECONFIGURATION are therefore polynomial-time computable, by Lemma 2.7.

Concretely, process the vertices in nondecreasing order of error count, maintaining a disjoint-set structure that holds the connected components of the subgraph induced by the vertices seen so far. Inserting a vertex adds it and its edges to the vertices already inserted, and unites the components those edges join. Once the inserted vertices are exactly those of weight at most x, the structure holds the components of the subgraph induced by $\{e_S \leq x\}$.

We now check that this sweep finds $\beta_H(w_0, w_1)$. If w_0 and w_1 are joined in H, they share a component once every vertex is inserted, so some insertion first places them in a common component. Let x^* be the weight of the vertex that insertion inserts. At that moment every inserted vertex has weight at most x^*, since the order is nondecreasing, so w_0 and w_1 are connected in $\{e_S \leq x^*\}$ and $\beta_H(w_0, w_1) \leq x^*$.

For the reverse inequality, every vertex of weight less than x^* was inserted before that vertex, and just before its insertion w_0 and w_1 were not in a common component of the subgraph induced by the vertices then inserted. That subgraph contains $\{e_S < x^*\}$, so w_0 and w_1 are not connected in $\{e_S < x^*\}$. A path attaining $\beta_H(w_0, w_1)$ lies in $\{e_S \leq \beta_H(w_0, w_1)\}$, so $\beta_H(w_0, w_1) < x^*$ would place it inside $\{e_S < x^*\}$. Hence $\beta_H(w_0, w_1) \geq x^*$, and therefore $\beta_H(w_0, w_1) = x^*$. The sweep uses polynomially many comparisons and graph operations, each taking polynomial time in the full input length. □

This is the regime in which the bottleneck is computed outright, with no condition on the landscape, as for a coarse discretization of a low-dimensional parameter space or a class with polynomially many hypotheses. The existence of low-dimensional continuous curves between trained networks (Garipov et al., 2018; Draxler et al., 2018) alone does not supply such an explicit finite graph. Two kinds of question remain, and both are collected in Section 1.5. One is the complexity and the approximability of the bottleneck for the geometric classes of Appendix D. The other is which landscape conditions separate the tractable regime from the hard one.

D MARGIN, GRANULARITY, AND THE COLLAPSE FOR HALFSPACES

Theorem C.11 is stated for an abstract configuration graph. The weights of a halfspace are continuous, so halfspaces do not come with a canonical such graph. To ask the reconfiguration question for halfspaces, we therefore discretize the weights to a grid and take single-coordinate moves, the finest natural local model. We give a sufficient condition in terms of the functional margin measured against the grid spacing: above a threshold the separators are connected, and the threshold is sharp in the plane. The realizable case and a convex surrogate come first. The section then turns to a positive error budget. There, a three-fold barrier fragments the feasible region, the chamber model admits algorithms in fixed dimension and for fixed budget, and wide two-layer networks admit lattice paths with vanishing loss slack.

D.1 The Grid Model

The weights of a halfspace are placed on a grid, so that a reconfiguration move changes one coordinate by one grid step.

Definition D.1 (Gridded halfspaces)**.** Fix an integer dimension $d \geq 1$ and a resolution $s \geq 1$, and let the parameter space be the grid $\Lambda_s = (\frac{1}{s}\mathbb{Z})^d$ with the move relation that joins w and w' when they differ by $\frac{1}{s}$ in a single coordinate. Data are points $x_i \in \mathbb{R}^d$ with $\|x_i\|_2 \leq R$ and labels $y_i \in \{-1, +1\}$, and $w \in \Lambda_s$ predicts $\operatorname{sign}\langle w, x_i\rangle$, with $\operatorname{sign}(0) = 0$. The *functional margin* of w is $\min_i y_i\langle w, x_i\rangle$, with the minimum of an empty sample taken to be $+\infty$, and w lies in the *zero-error set* when this quantity is positive. The *error count* of w is $e(w) = |\{i : y_i\langle w, x_i\rangle \leq 0\}|$, so a point lying on the hyperplane of w counts as an error and the zero-error set is $\{w \in \Lambda_s : e(w) = 0\}$. A bias term is included by appending a constant feature. The reconfiguration problem and the bottleneck β of Definition C.2 are taken on the graph with vertex set Λ_s and the moves just described, in place of the hypercube.

Since $y_i\langle \lambda w, x_i\rangle = \lambda y_i\langle w, x_i\rangle$ for $\lambda > 0$, one has $e(\lambda w) = e(w)$, so every set $\{w : e(w) \leq k\}$ is invariant under positive scaling. The dilation $w \mapsto sw$ maps Λ_s bijectively onto $\Lambda_1 = \mathbb{Z}^d$, and it carries moves of size $\frac{1}{s}$ to unit moves. It therefore identifies the subgraph that $\{e \leq k\}$ induces on Λ_s with the one it induces on Λ_1. The connectivity of $\{e \leq k\}$ on the grid therefore does not depend on s. Refining the grid changes only which fixed weights are grid points.

D.2 A Routing Lemma

The single fact about the grid that drives the collapse results of this section, here and for two-layer networks, is that two grid points can be joined by a coordinate-move path that hugs the straight segment between them.

Lemma D.2 (Lattice routing)**.** *For every $p, q \in \Lambda_s$ there is a path $p = v_0, v_1, \ldots, v_N = q$, with each v_k obtained from v_{k-1} by a single move of Definition D.1, such that every v_k lies within ℓ_∞-distance $\frac{1}{2s}$ of the segment $[p, q]$ and every coordinate of every v_k lies between the corresponding coordinates of p and q.*

Proof. The dilation $w \mapsto sw$ carries Λ_s onto $\mathbb{Z}^d$. It carries moves of size $\frac{1}{s}$ to unit coordinate moves, and it carries segments to segments. It also multiplies ℓ_∞-distances by s. So we may take $s = 1$ and $p, q \in \mathbb{Z}^d$. Reflecting each coordinate j with $q_j < p_j$ preserves $\mathbb{Z}^d$, unit coordinate moves, segments, and ℓ_∞-distances. So we may assume $q \geq p$ coordinatewise. Put $\delta_j = q_j - p_j \geq 0$ and $z(t) = p + t(q - p)$, so that $z_j(t) = p_j + t\delta_j$.

Fix j with $\delta_j \geq 1$ and $r \in \{1, \ldots, \delta_j\}$. Let $t_{j,r} = \frac{2r-1}{2\delta_j} \in (0, 1)$ be the time at which z_j crosses the half-integer $p_j + r - \frac{1}{2}$. That is, $t_{j,r}$ is the time with $z_j(t_{j,r}) = p_j + \frac{2r-1}{2\delta_j}\delta_j = p_j + r - \frac{1}{2}$. For fixed j, these times are strictly increasing in r. Let $N = \sum_j \delta_j$. List the N pairs (j, r) in nondecreasing order of $t_{j,r}$, breaking ties arbitrarily. Ties occur only between different coordinates. Write (j_k, r_k) for the k-th pair in this list, and $\tau_k = t_{j_k,r_k}$, so that $\tau_1 \leq \cdots \leq \tau_N$. Put $\tau_0 = 0$ and $\tau_{N+1} = 1$. Define $v_0 = p$ and $v_k = v_{k-1} + e_{j_k}$ for $1 \leq k \leq N$. Each step is a unit coordinate move, and $v_{N,j} = p_j + \delta_j = q_j$, so the sequence $v_0, \ldots, v_N$ is a path from p to q.

Fix $k \in \{0, \ldots, N\}$ and any $t \in [\tau_k, \tau_{k+1}]$. This interval is nonempty and contained in $[0, 1]$, because every $t_{j,r}$ lies in $(0, 1)$, so $0 = \tau_0 \leq \tau_1 \leq \cdots \leq \tau_N \leq 1 = \tau_{N+1}$. Hence $z(t)$ is a point of $[p, q]$.

We show $\|v_k - z(t)\|_\infty \leq \frac{1}{2}$. Fix a coordinate j and let $c = |\{l \leq k : j_l = j\}|$, so that $v_{k,j} = p_j + c$. If $\delta_j = 0$ then $c = 0$ and $v_{k,j} = p_j = z_j(t)$. Otherwise, since $t_{j,1} < t_{j,2} < \cdots < t_{j,\delta_j}$, the pairs of coordinate j appear in the sorted list in the order $(j, 1), (j, 2), \ldots, (j, \delta_j)$. Exactly c of them occupy positions at most k, so these are $(j, 1), \ldots, (j, c)$, and the remaining pairs of coordinate j occupy positions at least $k + 1$. Hence, when $c \geq 1$, the pair (j, c) sits at some position $l \leq k$, and $t_{j,c} = \tau_l \leq \tau_k \leq t$, the list being sorted. When $c < \delta_j$, the pair $(j, c + 1)$ sits at some position $l' \geq k + 1$, and $t \leq \tau_{k+1} \leq \tau_{l'} = t_{j,c+1}$.

Since z_j is nondecreasing, the lower bound on $z_j(t)$ follows from the two cases for c. When $c \geq 1$, we have $z_j(t) \geq z_j(t_{j,c}) = p_j + c - \frac{1}{2}$. The case $c = 0$ gives $z_j(t) \geq z_j(0) = p_j = p_j + c$. For

the upper bound, again by monotonicity of z_j, there are two cases for c. When $c < \delta_j$, we have $z_j(t) \le z_j(t_{j,c+1}) = p_j + c + \frac{1}{2}$. The case $c = \delta_j$ gives $z_j(t) \le z_j(1) = p_j + \delta_j = p_j + c$. In every case $|v_{k,j} - z_j(t)| \le \frac{1}{2}$. Moreover $v_{k,j} = p_j + c$ with $0 \le c \le \delta_j$, so $v_{k,j}$ lies between p_j and $q_j = p_j + \delta_j$.

Undoing the reflections and the dilation turns each step $+e_j$ into a step of $\pm\frac{1}{s}$ in the single coordinate j, which is again a move of Definition D.1. It also maps coordinates between p_j and q_j back to coordinates between the original p_j and q_j. It replaces $\frac{1}{2}$ by $\frac{1}{2s}$. □

D.3 Margin Forces the Collapse

Two separators whose functional margins both exceed $\frac{R\sqrt{d}}{2s}$, a threshold that shrinks as the grid is refined, are joined by a path inside the zero-error set.

Theorem D.3 (Margin collapse for halfspaces; restatement of Theorem 1.7)**.** *In the model of Definition D.1, any two weights $p, q \in \Lambda_s$ whose functional margins are both at least μ, with $\mu > \frac{R\sqrt{d}}{2s}$, are joined by a coordinate-move path lying entirely in the zero-error set, every vertex of which has functional margin at least $\mu - \frac{R\sqrt{d}}{2s}$. Consequently, for every $\gamma > 0$, every $\rho > 0$ and every resolution $s > \frac{R\sqrt{d}}{2\gamma\rho}$, any two weights $p, q \in \Lambda_s$ with $\|p\|_2, \|q\|_2 \ge \rho$ and geometric margin $\min_i y_i\langle \cdot, x_i\rangle/\|\cdot\|_2$ at least γ are connected in the zero-error set, and their bottleneck is $\beta(p, q) = 0 = \max(e(p), e(q))$. All such weights lie in a single connected component of the zero-error set. The norm condition cannot be dropped (Proposition D.4).*

Proof. Every point of the segment $[p, q]$ has the form $z = (1-\nu)p + \nu q$ with $\nu \in [0, 1]$. For every i, by linearity of the inner product, $y_i\langle z, x_i\rangle = (1-\nu)y_i\langle p, x_i\rangle + \nu y_i\langle q, x_i\rangle \ge (1-\nu)\mu + \nu\mu = \mu$. Take the path of Lemma D.2. For a vertex v_k, Lemma D.2 gives a segment point z with $\|v_k - z\|_\infty \le \frac{1}{2s}$.

It remains to bound how much replacing z by v_k can change each label's margin. Since each of the d squared coordinates of $v_k - z$ is at most $\|v_k - z\|_\infty^2$, we get $\|v_k - z\|_2 \le \sqrt{d}\|v_k - z\|_\infty \le \frac{\sqrt{d}}{2s}$. By the Cauchy–Schwarz inequality, $|y_i\langle v_k - z, x_i\rangle| \le \|v_k - z\|_2\|x_i\|_2 \le \frac{\sqrt{d}}{2s}R$, so for every i,

$$y_i\langle v_k, x_i\rangle = y_i\langle z, x_i\rangle + y_i\langle v_k - z, x_i\rangle \ge \mu - R \cdot \frac{\sqrt{d}}{2s} > 0,$$

the last inequality being the hypothesis $\mu > \frac{R\sqrt{d}}{2s}$. Thus every v_k classifies all examples correctly, with functional margin at least $\mu - \frac{R\sqrt{d}}{2s}$, and lies in the zero-error set, so p and q are connected there.

For the second statement, a weight w with $\|w\|_2 \ge \rho$ and geometric margin at least γ has functional margin $\min_i y_i\langle w, x_i\rangle = \|w\|_2 \cdot \min_i y_i\langle w, x_i\rangle/\|w\|_2 \ge \gamma\|w\|_2 \ge \gamma\rho$. Moreover, $\gamma\rho > \frac{R\sqrt{d}}{2s}$ is the assumption $s > \frac{R\sqrt{d}}{2\gamma\rho}$ rearranged. The first statement with $\mu = \gamma\rho$ joins any two such weights inside the zero-error set, and every vertex of that path has $e = 0$, so $\beta(p, q) \le 0$. Conversely, $\beta(p, q) \ge 0$, because every error count is a nonnegative integer. Also, $e(p) = e(q) = 0$, because p and q have positive functional margin. Hence $\beta(p, q) = 0 = \max(e(p), e(q))$. Since any two weights as in the second statement are connected in the zero-error set, all of them lie in one connected component of it.

For the last statement, take the instance of Proposition D.4 with an integer $a \ge 1$, and $N = \sqrt{a^2 + (a+1)^2}$ as there. There, $\|u\|_2 = a\sqrt{2}/s$ and $\|v\|_2 = 2a\sqrt{2}/s$. So u and v both have geometric margin $\frac{a/(sN)}{a\sqrt{2}/s} = \frac{2a/(sN)}{2a\sqrt{2}/s} = \frac{1}{\sqrt{2}N}$, a value that does not depend on s. For every $s \ge 1$, they lie in different components of the zero-error set. So for $\gamma = \frac{1}{\sqrt{2}N}$ no resolution s connects all weights of geometric margin at least γ, and the lower bound ρ on the norms cannot be dropped. □

Let two separators of positive geometric margin be points of some grid Λ_{s_0}, let γ be the smaller of their geometric margins, and let ρ be the smaller of their norms. Then every refinement Λ_s with s a multiple of s_0 and $s > \frac{R\sqrt{d}}{2\gamma\rho}$ contains both, and Theorem D.3 places them in one component.

Reconfiguration among well-separated models whose norms are bounded below is then a triviality, with bottleneck equal to the larger endpoint loss. This is the precise sense in which margin forces the collapse for halfspaces.

D.4 The Granularity Obstruction

The threshold is not an artifact of the proof. When the grid fails to resolve the feasible region, connectivity can fail. Moreover, no fixed local move set repairs every instance.

The zero-error set consists of the grid points of an open convex cone, invariant under positive scaling, so the obstruction must be a wedge too thin for the grid at the scale of the weights in question. The following instance shows that the constant of Theorem D.3 cannot be lowered in the plane.

Proposition D.4 (The threshold is sharp in the plane)**.** *Let $d = 2$, let $s \geq 1$, let $a \geq 1$ be an integer, and put $N = \sqrt{a^2 + (a+1)^2}$. The sample $x_1 = (-a, a+1)/N$, $x_2 = (a+1, -a)/N$, $y_1 = y_2 = +1$ has $R = 1$, since $\|x_1\|_2 = \|x_2\|_2 = \sqrt{a^2 + (a+1)^2}/N = 1$, and the weights $u = (a, a)/s$ and $v = (2a, 2a)/s$ of Λ_s have functional margins $\frac{a}{sN}$ and $\frac{2a}{sN}$ and lie in different connected components of the zero-error set. Since $\frac{a}{N} \to \frac{1}{\sqrt{2}}$ as $a \to \infty$, for every $c < 1$ there is an instance with two zero-error weights, both of functional margin at least some $\mu > c \cdot \frac{R\sqrt{d}}{2s}$, that are not connected in the zero-error set.*

Proof. For $w \in \mathbb{R}^2$ one has $N\langle w, x_1\rangle = (a+1)w_2 - aw_1$ and $N\langle w, x_2\rangle = (a+1)w_1 - aw_2$. At u both equal $a/s > 0$, so u lies in the zero-error set with functional margin $a/(sN)$. Both forms are linear in w, so at $v = 2u$ both equal $2a/s > 0$, and v lies in the zero-error set with functional margin $2a/(sN)$. The four neighbors of u are $(a \pm 1, a)/s$ and $(a, a \pm 1)/s$. At $(a+1, a)/s$, the first form equals $((a+1)a - a(a+1))/s = 0$. The second form at $(a-1, a)/s$ equals $\left((a+1)(a-1) - a^2\right)/s = -1/s$. Likewise, at $(a, a+1)/s$ the second form equals $((a+1)a - a(a+1))/s = 0$. And at $(a, a-1)/s$ the first form equals $\left((a+1)(a-1) - a^2\right)/s = -1/s$. None of the four is positive on both forms, so no neighbor of u lies in the zero-error set: u is isolated there, and $v \neq u$ lies in another component. Finally, $a/N = a/\sqrt{2a^2 + 2a + 1} = 1/\sqrt{2 + 2/a + 1/a^2}$, which tends to $1/\sqrt{2}$ as $a \to \infty$. Also, $1/\sqrt{2} = R\sqrt{d}/2$, since $R = 1$ and $d = 2$. Let $c < 1$. If $c \leq 0$, then $cR\sqrt{d}/(2s) \leq 0 < a/(sN)$ for every $a \geq 1$, so every a gives $\mu = a/(sN) > cR\sqrt{d}/(2s)$. If $0 < c < 1$, both sides of $a/N > c/\sqrt{2}$ are positive, so this inequality is equivalent to its square, $a^2/N^2 > c^2/2$. Multiplying by $2N^2 = 2(2a^2 + 2a + 1) > 0$ turns this into $2a^2 > c^2(2a^2 + 2a + 1)$, that is, into $2a^2(1 - c^2) > c^2(2a + 1)$. For every integer $a > \frac{3c^2}{2(1-c^2)}$, this holds. Indeed, then $2a(1 - c^2) > 3c^2$, so $2a^2(1 - c^2) > 3ac^2 \geq c^2(2a + 1)$, the last step being $2a + 1 \leq 3a$ for $a \geq 1$. Every such a gives $\mu = a/(sN) > c/(\sqrt{2}s) = cR\sqrt{d}/(2s)$. □

The same construction defeats any fixed finite enlargement $M \subseteq \mathbb{Z}^2 \setminus \{0\}$ of the move set, the moves being $w \mapsto w + m/s$ for $m \in M$. If M is empty, the two distinct feasible weights of Proposition D.4 are already isolated, so assume M is nonempty. Replacing M by $M \cup (-M)$, which is again finite and has the same undirected move graph, we may assume $M = -M$. Then the neighbors of a weight w are exactly the weights $w + m/s$ with $m \in M$.

Choose an integer direction u_0 parallel to no element of M, which exists because M is finite and the integer directions are infinitely many. Fix an integer $a \geq 1$. For every $m \in M$, the vector $au_0 + m$ is nonzero, since $au_0 + m = 0$ would make $m = -au_0$ parallel to u_0. It is also not parallel to u_0. If $au_0 + m = \lambda u_0$ with $\lambda \neq 0$, then $m = (\lambda - a)u_0$. This equals $0 \notin M$ when $\lambda = a$, and is parallel to u_0 otherwise. So each angle $\angle(au_0 + m, u_0)$ lies in $(0, \pi)$, and $\theta_a = \min_{m \in M} \angle(au_0 + m, u_0)$, a minimum over a finite set, is positive. Let α be the angle of u_0, and choose θ with $0 < \theta < \min(\theta_a, \frac{\pi}{2})$.

Take the sample x_1 at angle $\alpha + \frac{\pi}{2} - \theta$, labeled $+1$. Take x_2 at angle $\alpha - \frac{\pi}{2} + \theta$, also labeled $+1$. Both are unit vectors, so $R = 1$. A weight $w \neq 0$ at angle ψ has $\langle w, x_1\rangle > 0$ exactly when $\psi \in (\alpha - \theta, \alpha + \pi - \theta)$. It has $\langle w, x_2\rangle > 0$ exactly when $\psi \in (\alpha - \pi + \theta, \alpha + \theta)$. Since $0 < \theta < \frac{\pi}{2}$, both arcs lie in $(\alpha - \pi, \alpha + \pi)$, and their intersection is $(\alpha - \theta, \alpha + \theta)$. Also, $w = 0$ has both inner products zero. So the zero-error set consists of the grid points of the open wedge of half-angle θ about u_0.

This wedge contains au_0/s and $2au_0/s$, whose direction is α. Every M-neighbor $(au_0 + m)/s$ of au_0/s has angular distance at least $\theta_a > \theta$ from α, and so lies outside the wedge. So au_0/s is isolated in the zero-error set under the moves of M, and $2au_0/s$ lies in another component.

In dimension $d \geq 3$, consider a diagonal weight $(a, \ldots, a)/s$, with $a \geq 1$ an integer. Whatever the sample, such a weight can be isolated in the zero-error set only when its functional margin is below $R\sqrt{(d-1)/d}/s$. For each a, the construction below isolates $(a, \ldots, a)/s$ at a margin that tends to this value as $a \to \infty$. Dividing by the threshold $\frac{R\sqrt{d}}{2s}$ of Theorem D.3 gives the ratio $\sqrt{(d-1)/d} \cdot \frac{2}{\sqrt{d}} = \frac{2\sqrt{d-1}}{d}$. This ratio is less than 1, because $d^2 - 4(d-1) = (d-2)^2 > 0$ for $d \geq 3$, so beyond the plane an isolated diagonal weight cannot witness that the constant of Theorem D.3 is sharp.

For the upper bound, let $u = (a, \ldots, a)/s$, with $a \geq 1$ an integer, have functional margin $\mu > 0$ on a sample with $\|x_i\|_2 \leq R$, and be isolated in the zero-error set. Then, for every j, its neighbor $u + e_j/s$ makes an error on some example i. Put $b = y_i x_i$ and $\Sigma = \sum_{l=1}^{d} b_l$, so that $\|b\|_2 \leq R$ and $y_i\langle u, x_i\rangle = a\Sigma/s \geq \mu > 0$. The error gives $0 \geq y_i\langle u + e_j/s, x_i\rangle = (a\Sigma + b_j)/s$, so $b_j \leq -a\Sigma \leq -s\mu < 0$. Then $\sum_{l\neq j} b_l = \Sigma - b_j > -b_j = |b_j|$. The Cauchy–Schwarz inequality over the $d-1$ coordinates $l \neq j$ then gives $\sum_{l\neq j} b_l^2 \geq \frac{1}{d-1}\left(\sum_{l\neq j} b_l\right)^2 > \frac{b_j^2}{d-1}$. Hence $R^2 \geq \|b\|_2^2 > b_j^2 + \frac{b_j^2}{d-1} = \frac{d}{d-1}b_j^2$, so $s\mu \leq |b_j| < R\sqrt{(d-1)/d}$, that is, $\mu < R\sqrt{(d-1)/d}/s$.

For the lower bound, let $u = (a, \ldots, a)/s$ with $a \geq 1$ an integer. For each j, take the unit vectors along $\xi_j = -a(d-1)e_j + (a+1)\sum_{l\neq j} e_l$ and $\zeta_j = (a+1)(d-1)e_j - a\sum_{l\neq j} e_l$, both labeled $+1$. All these sample points have $R = 1$. Dividing by a positive norm does not change the sign of an inner product, so for signs it suffices to compute with ξ_j and ζ_j. The coordinates of each of ξ_j and ζ_j sum to $d-1$, so $\langle u, \xi_j\rangle = \langle u, \zeta_j\rangle = a(d-1)/s > 0$. Also, $\langle u + e_j/s, \xi_j\rangle = (a(d-1) - a(d-1))/s = 0$, and $\langle u - e_j/s, \zeta_j\rangle = (a(d-1) - (a+1)(d-1))/s = -(d-1)/s$. So every neighbor of u makes an error, a zero inner product counting as one under Definition D.1, u is isolated in the zero-error set, and the zero-error weight $2u$ lies in another component. The two brackets in $\|\zeta_j\|_2^2 = (d-1)\left((d-1)(a+1)^2 + a^2\right)$ and $\|\xi_j\|_2^2 = (d-1)\left((d-1)a^2 + (a+1)^2\right)$ differ by $(d-2)\left((a+1)^2 - a^2\right) = (d-2)(2a+1) \geq 0$, so $\|\zeta_j\|_2 \geq \|\xi_j\|_2$, and neither norm depends on j. Every point of the sample is a unit vector $x = x'/\|x'\|_2$ with x' one of the ξ_j or ζ_j. It has $\langle u, x\rangle = a(d-1)/(s\|x'\|_2)$, so the smallest of these values is attained at the unit vectors along the ζ_j. The functional margin of u is therefore $\frac{a(d-1)}{s\|\zeta_j\|_2} = \frac{a\sqrt{d-1}}{s\sqrt{(d-1)(a+1)^2+a^2}} = \frac{\sqrt{d-1}}{s\sqrt{(d-1)(1+1/a)^2+1}}$, which tends to $\frac{\sqrt{d-1}}{s\sqrt{d}} = \sqrt{(d-1)/d}/s$ as $a \to \infty$.

For $d = 2$, this is the instance of Proposition D.4 with each of its two points listed twice, since $\xi_1 = \zeta_2 = (-a, a+1)$ and $\zeta_1 = \xi_2 = (a+1, -a)$. Since $\sqrt{(d-1)/d} \geq 1/\sqrt{2}$ for $d \geq 2$, a threshold of order R/s is necessary in every dimension $d \geq 2$, and whether the factor $\sqrt{d}$ is sharp for $d \geq 3$ we leave open.

Remark D.5 (Convex surrogates). The argument extends to a convex surrogate loss $\Phi(w) = \sum_i \ell(y_i\langle w, x_i\rangle)$ with ℓ convex and λ-Lipschitz, where feasibility is $\Phi(w) \leq \eta$. Each term $w \mapsto \ell(y_i\langle w, x_i\rangle)$ is the convex function ℓ composed with a linear function of w, so it is convex, and so is their sum Φ. The sublevel set $\{w : \Phi(w) \leq \eta\}$ is therefore convex, so the segment between two feasible weights is feasible. Moreover, by the Lipschitz property of ℓ and the Cauchy–Schwarz inequality, $|\Phi(w) - \Phi(w')| \leq \lambda\sum_i |\langle w - w', x_i\rangle| \leq \kappa\|w - w'\|_2$ with $\kappa = \lambda\sum_i \|x_i\|_2$. Routing by Lemma D.2 places each vertex within ℓ_∞-distance $\frac{1}{2s}$ of a segment point of Φ-value at most η. Hence each vertex lies within ℓ_2-distance $\frac{\sqrt{d}}{2s}$ of that point. So this routing keeps each vertex within Φ-value $\eta + \kappa\frac{\sqrt{d}}{2s}$. Hence on a grid of resolution s any two feasible weights of Λ_s are joined by a coordinate-move path along which Φ never exceeds $\eta + \kappa\frac{\sqrt{d}}{2s}$. So the bottleneck with Φ in place of e, the minimum over paths read as an infimum since Φ is real-valued, is at most $\eta + \kappa\frac{\sqrt{d}}{2s}$, a near-collapse whose excess over η tends to zero as the grid is refined.

D.5 An Agnostic Barrier

The collapse of Theorem D.3 used convexity of the feasible region in an essential way: the segment between two separators is again a separator. With a positive error budget the feasible region is a union of cones that need not be convex. The next result shows that the disconnection this allows is a feature of the problem itself, not an artifact of the proof of Theorem D.3. Already three data directions in the plane produce a feasible region that is disconnected. The bottleneck between the two weights the next result exhibits, one in each of two pieces, exceeds their error counts by a third of the sample, and no refinement of the grid closes this gap.

Proposition D.6 (Three-fold agnostic barrier; restatement of Proposition 1.8)**.** *For every integer $r \geq 1$ there is a sample of $t = 3r$ points in $\mathbb{R}^2$ with a common label with the following property, for every $s \geq 1$. The grid weights $u = \left(-\frac{2}{s}, 0\right)$ and $v = \left(\frac{2}{s}, \frac{2}{s}\right)$ have $e(u) = e(v) = r$, and every coordinate-move path in Λ_s from u to v, as well as every continuous path in $\mathbb{R}^2$ from u to v, meets a weight of error at least $2r$. Hence the bottleneck satisfies $\beta(u, v) \geq 2r = 2\max(e(u), e(v))$, a gap of $r = t/3$ that no refinement of the grid removes; the bound is attained.*

Proof. We use the error count $e(w) = |\{i : y_i\langle w, x_i\rangle \leq 0\}|$ of Definition D.1 at every $w \in \mathbb{R}^2$, since the statement also concerns continuous paths. Let g_j be the unit vector at angle $\phi_j = 2\pi j/3$, for $j = 0, 1, 2$, with indices taken modulo 3 throughout. Let the sample consist of r copies of each g_j, labeled $+1$. For $w \neq 0$ write $\theta(w)$ for its direction, an angle modulo 2π. The copies of g_j are classified correctly exactly when $\langle w, g_j\rangle > 0$. Since $\langle w, g_j\rangle = \|w\|_2 \cos(\theta(w) - \phi_j)$, this holds exactly when $\theta(w)$ lies in the open arc $A_j = \left(\phi_j - \frac{\pi}{2}, \phi_j + \frac{\pi}{2}\right)$. Consider the six arcs

$$B_j = \left(\phi_j + \tfrac{\pi}{6}, \phi_j + \tfrac{\pi}{2}\right), \qquad D_j = \left[\phi_j - \tfrac{\pi}{6}, \phi_j + \tfrac{\pi}{6}\right], \qquad j = 0, 1, 2,$$

which partition the circle in the cyclic order $D_0, B_0, D_1, B_1, D_2, B_2$ (Figure 1). Since $\phi_{j+1} = \phi_j + \frac{2\pi}{3}$ and $\phi_{j+2} = \phi_j + \frac{4\pi}{3} \equiv \phi_j - \frac{2\pi}{3}$, the arcs of the other two groups are $A_{j+1} = \left(\phi_j + \frac{\pi}{6}, \phi_j + \frac{7\pi}{6}\right)$ and $A_{j+2} = \left(\phi_j - \frac{7\pi}{6}, \phi_j - \frac{\pi}{6}\right)$. To compare these arcs with B_j and D_j, write every direction as $\phi_j + \alpha$. Here α ranges over the window $\left[-\frac{\pi}{6}, \frac{11\pi}{6}\right)$ of length 2π, which represents each direction exactly once. In this window A_{j+1} is $\alpha \in \left(\frac{\pi}{6}, \frac{7\pi}{6}\right)$. Since $\phi_j - \frac{7\pi}{6} \equiv \phi_j + \frac{5\pi}{6}$ and $\phi_j - \frac{\pi}{6} \equiv \phi_j + \frac{11\pi}{6}$, the arc A_{j+2} is $\alpha \in \left(\frac{5\pi}{6}, \frac{11\pi}{6}\right)$ in this window. The arc A_j contains every $\alpha \in \left[-\frac{\pi}{6}, \frac{\pi}{2}\right)$.

We now compute the error e on each type of arc. A direction in B_j has $\alpha \in \left(\frac{\pi}{6}, \frac{\pi}{2}\right)$. This lies in A_j and in A_{j+1} but not in A_{j+2}, since $\frac{\pi}{2} < \frac{5\pi}{6}$. So exactly two groups are correct and $e = r$. A direction in D_j has $\alpha \in \left[-\frac{\pi}{6}, \frac{\pi}{6}\right]$. This lies in A_j, but not in A_{j+1}, whose window interval is open at its left end $\frac{\pi}{6}$, and not in A_{j+2}, whose window interval begins at $\frac{5\pi}{6} > \frac{\pi}{6}$. So exactly one group is correct and $e = 2r$. At $w = 0$ every $\langle 0, g_j\rangle = 0$ counts as an error, so $e(0) = 3r$. Thus e takes only the values $r, 2r, 3r$. The feasible set $\{w : e(w) \leq r\}$ is therefore the union of the three disjoint open sectors $\Sigma_j = \{w \neq 0 : \theta(w) \in B_j\}$.

It remains to bound how far apart two different sectors are, which is used below to rule out a short path between them. For directions θ, θ' their *angular distance* is the length of the shorter of the two arcs of the circle joining them. For nonzero w, w' it equals the angle $\psi \in [0, \pi]$ with $\cos\psi = \langle w, w'\rangle / (\|w\|_2\|w'\|_2)$. This is because $w = \|w\|_2(\cos\theta(w), \sin\theta(w))$ and likewise for w' give $\langle w, w'\rangle = \|w\|_2\|w'\|_2 \cos(\theta(w) - \theta(w'))$. The two arcs joining $\theta(w)$ and $\theta(w')$ have lengths δ and $2\pi - \delta$, where $\delta \in [0, \pi]$ is their angular distance. Moreover, $\theta(w) - \theta(w') \equiv \pm\delta \pmod{2\pi}$. Since cosine is even and 2π-periodic, $\cos(\theta(w) - \theta(w')) = \cos\delta$, and since cosine is injective on $[0, \pi]$, this gives $\delta = \psi$. Two directions in different arcs B_j are at angular distance greater than $\frac{\pi}{3}$. Indeed, in the cyclic order $D_0, B_0, D_1, B_1, D_2, B_2$, each of the two arcs of the circle joining them contains a whole closed arc D_i of length $\frac{\pi}{3}$, and neither endpoint lies in that D_i, since the arcs B_j are disjoint from it. We have $\theta(u) = \pi \in B_1$ and $\theta(v) = \frac{\pi}{4} \in B_0$, so $e(u) = e(v) = r$ and u, v lie in different sectors.

Since the only values of e are $r, 2r, 3r$, it suffices to show that no path from u to v of either kind stays inside the feasible set.

Continuous paths. Let $\omega\colon [0, 1] \to \mathbb{R}^2$ be continuous with $\omega(0) = u$, $\omega(1) = v$, and suppose $e(\omega(\tau)) \leq r$ for all τ. Then ω avoids 0, since $e(0) = 3r > r$, so $\theta \circ \omega$ is a continuous map to the circle. The set $T = \{\tau : \theta(\omega(\tau)) \in B_1\}$ is the preimage of the open arc B_1 under this continuous

map, hence open in $[0,1]$. It contains 0, since $\theta(u) \in B_1$, and does not contain 1, since $\theta(v) \in B_0$. Let $\tau^* = \sup\{\tau : [0,\tau] \subseteq T\}$, so that $[0,\tau^*) \subseteq T$ and $\tau^* \leq 1$. If τ^* were in T, then $\tau^* < 1$. By openness, some interval $[\tau^*, \tau^* + h]$ with $h > 0$ would then lie in T. Then $[0, \tau^* + h] \subseteq T$, contradicting the choice of τ^*. Hence $\tau^* \notin T$. Since $0 \in T$ and $\tau^* \notin T$, we have $\tau^* > 0$. So $[0,\tau^*)$ is a nonempty interval inside T with right end τ^*. Then $\theta(\omega(\tau^*)) = \lim_{\tau \uparrow \tau^*} \theta(\omega(\tau))$ by continuity, a limit of points of B_1, so it lies in the closure $\overline{B_1}$. It does not lie in B_1, since $\tau^* \notin T$, so it lies in $\overline{B_1} \setminus B_1 = \left\{\phi_1 + \frac{\pi}{6}, \phi_1 + \frac{\pi}{2}\right\}$. The first of these is the right endpoint of D_1, and the second, equal to $\phi_2 - \frac{\pi}{6}$, is the left endpoint of D_2. So $e(\omega(\tau^*)) = 2r > r$, contradicting the assumption on ω.

Grid paths. First, suppose $w, w' \in \Lambda_s$ are adjacent, with $w' \neq 0$ and $\|w\|_2 \geq \frac{2}{s}$. Then the angle $\psi \in [0,\pi]$ between w and w' is at most $\frac{\pi}{6}$. Write $w' = w + \delta e_j$ with $|\delta| = \frac{1}{s}$. The distance from w to the line $\mathbb{R}w'$ is then the length of the component of w orthogonal to w'. Its square is $\|w\|_2^2 - \langle w, w'\rangle^2 / \|w'\|_2^2 = \|w\|_2^2 \left(1 - \cos^2\psi\right)$, so the distance equals $\|w\|_2 \sin\psi$, using $\sin\psi \geq 0$ on $[0,\pi]$. Since w' lies on that line, this distance is at most $\|w - w'\|_2 = |\delta| = \frac{1}{s}$. So $\sin\psi \leq \frac{1/s}{\|w\|_2} \leq \frac{1}{2}$, using $\|w\|_2 \geq \frac{2}{s}$. Also $\langle w, w'\rangle = \|w\|_2^2 + \delta w_j \geq \|w\|_2^2 - \frac{1}{s}|w_j| \geq \|w\|_2 \left(\|w\|_2 - \frac{1}{s}\right) > 0$, using $|w_j| \leq \|w\|_2$ and $\|w\|_2 \geq \frac{2}{s} > \frac{1}{s}$. So $\psi < \frac{\pi}{2}$. Hence, since $\sin$ is increasing on $\left[0, \frac{\pi}{2}\right]$, we get $\psi \leq \arcsin\frac{1}{2} = \frac{\pi}{6}$.

Second, we claim that adjacent $w, w' \in \Lambda_s$ with $e(w), e(w') \leq r$ lie in the same sector. Both are nonzero since $e(0) = 3r$.

We first handle the case where one of w, w' has norm at least $\frac{2}{s}$. If one of them has norm at least $\frac{2}{s}$, apply the first fact with that one in the role of w and the other, which is nonzero, in the role of w'. This bounds their angle, which is their angular distance, by $\frac{\pi}{6} < \frac{\pi}{3}$. So their directions are not in different arcs B_j. Each of them is feasible, so its direction lies in some arc B_j. Hence both lie in the same arc B_j, that is, in the same sector Σ_j.

It remains to handle the case where both have norm less than $\frac{2}{s}$. A point $(i,i')/s$ of Λ_s with $|i| \geq 2$ or $|i'| \geq 2$ has norm at least $\frac{2}{s}$. So both lie among the nine points $(i,i')/s$ with $i, i' \in \{-1, 0, 1\}$. Three of these nine points, $(-1,0)/s$, $(1,1)/s$, and $(1,-1)/s$, have directions $\pi \in B_1$, $\frac{\pi}{4} \in B_0$, and $\frac{7\pi}{4} \in B_2$ respectively. So all three are feasible. The origin has $e = 3r$. The remaining five points are $(1,0)/s$, $(0,1)/s$, $(0,-1)/s$, $(-1,1)/s$, and $(-1,-1)/s$. Their directions are 0, $\frac{\pi}{2}$, $\frac{3\pi}{2}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$ respectively. These directions lie in the arcs D_0, D_1, D_2, D_1, D_2 respectively, so each point has $e = 2r$. The three feasible points are pairwise nonadjacent, each pair differing in both coordinates or by $\frac{2}{s}$ in one, so this case does not occur.

Now let $u = w_0, w_1, \ldots, w_L$ be a coordinate-move path in Λ_s with $e(w_l) \leq r$ for every l. We show by induction on l that $w_l \in \Sigma_1$. The base case is $w_0 = u \in \Sigma_1$. If $w_{l-1} \in \Sigma_1$, then w_{l-1} and w_l are adjacent with $e(w_{l-1}), e(w_l) \leq r$, so by the claim they lie in the same sector, and $w_l \in \Sigma_1$. Hence the path never reaches $v \in \Sigma_0$.

In either model, a path from u to v therefore meets a weight with $e > r$. Since $r, 2r, 3r$ are the only values of e, this means $e \geq 2r$. By Definition C.2, with the minimum taken over coordinate-move paths in Λ_s as Definition D.1 specifies, every path in the minimum has maximum error at least $2r$, so $\beta(u,v) \geq 2r$.

We next check that this lower bound survives every refinement of the grid. The sample does not depend on s, so refining the grid does not remove the gap. Indeed, fix $s_0 \geq 1$ and the weights $u_0 = (-2,0)/s_0$ and $v_0 = (2,2)/s_0$, and let $k \geq 1$ be an integer. Then $u_0 = (-2k,0)/(ks_0)$ and $v_0 = (2k,2k)/(ks_0)$ lie in Λ_{ks_0}. Their directions are $\pi \in B_1$ and $\frac{\pi}{4} \in B_0$, so $e(u_0) = e(v_0) = r$. The claim above holds on Λ_{ks_0}, since it was proved for every resolution, and the induction used only that the path starts in Σ_1 and that its last point lies outside Σ_1. So every coordinate-move path in Λ_{ks_0} from u_0 to v_0 meets a weight of error at least $2r$, on every refinement of Λ_{s_0}.

It remains to exhibit a path that attains this bound. The bound is attained by the grid path $(-2,0), (-2,1), (-1,1), (0,1), (1,1), (2,1), (2,2)$, all divided by s, each point of which differs from the previous one by $\frac{1}{s}$ in a single coordinate. Its first three directions are π, $\pi - \arctan\frac{1}{2}$, and $\frac{3\pi}{4}$. The remaining four are $\frac{\pi}{2}$, $\frac{\pi}{4}$, $\arctan\frac{1}{2}$, and $\frac{\pi}{4}$. Also $0 < \arctan\frac{1}{2} < \frac{\pi}{6}$, because $\tan\frac{\pi}{6} = \frac{1}{\sqrt{3}} > \frac{1}{2}$ and $\arctan$ is increasing. Hence the first three of these directions lie, respec-

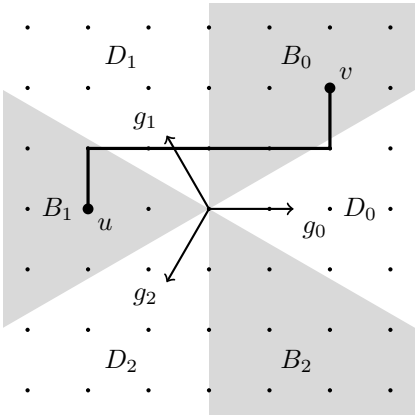

Figure 1: The three-fold barrier of Proposition D.6 in the plane of weights, in units of $\frac{1}{s}$. The arrows are the three data directions g_0, g_1, g_2, each carrying r copies labeled $+1$. A weight in the interior of a shaded sector, with direction in some B_j, misclassifies one group and has $e = r$. In an unshaded sector, where the direction lies in some D_j, a weight misclassifies two groups and has $e = 2r$. The origin has $e = 3r$. The weights $u = (-2, 0)/s$ and $v = (2, 2)/s$ lie in different shaded sectors. Every path between them, on the grid or in the plane, meets an unshaded sector or the origin, and the drawn grid path attains the bottleneck $2r$.

tively, in $B_1 = \left(\frac{5\pi}{6}, \frac{7\pi}{6}\right)$, B_1, and $D_1 = \left[\frac{\pi}{2}, \frac{5\pi}{6}\right]$. The remaining four lie, respectively, in D_1, $B_0 = \left(\frac{\pi}{6}, \frac{\pi}{2}\right)$, $D_0 = \left[-\frac{\pi}{6}, \frac{\pi}{6}\right]$, and B_0. So the errors along the path are $r, r, 2r, 2r, r, 2r, r$, and its maximum error is $2r$. So $\beta(u, v) \leq 2r$, and with the lower bound, $\beta(u, v) = 2r$. □

Remark D.7 (Clustering and the realizable–agnostic divide). Proposition D.6 is a deterministic, two-dimensional instance of the clustering picture behind the overlap gap property (Gamarnik, 2021), inside a concrete learning problem. The low-error halfspaces split into clusters that cannot be joined without crossing a barrier whose height is a constant fraction of the sample. This instance is not an overlap gap in the strict sense, because what separates its clusters is topological rather than a forbidden interval of overlaps.

We now make this precise. The normalized overlap $\langle w, w' \rangle / (\|w\|_2 \|w'\|_2)$ of two weights of error r lies in $\left(\frac{1}{2}, 1\right]$ when they share a sector and in $\left(-1, \frac{1}{2}\right)$ when they do not. For continuous weights, every overlap in these intervals is attained. For an angular separation $\delta \in [0, \pi/3)$, the directions $\pi/3 - \delta/2$ and $\pi/3 + \delta/2$ both lie in B_0. For $\delta \in (\pi/3, \pi)$, the directions $2\pi/3 - \delta/2$ and $2\pi/3 + \delta/2$ lie in B_0 and B_1, respectively. Taking cosine gives the two intervals above. On a fixed grid, their overlaps are dense in these intervals: for a unit vector h inside a sector, round each coordinate of Nh to an integer to obtain z_N, so $\|z_N/N - h\|_2 \leq \sqrt{2}/(2N)$ and $z_N/\|z_N\|_2 \to h$. For sufficiently large N, the grid vector z_N/s remains inside that open sector. Applying this to each member of a pair and using continuity of the inner product approximates its overlap arbitrarily closely. Thus neither the continuous nor the grid model has a forbidden open interval of overlaps, although exact attainment of each value need not hold on the grid. What separates the clusters is instead the arcs D_j of error $2r$ lying between the sectors, which is what we mean by topological. Within a sector the angular distance lies in $\left[0, \frac{\pi}{3}\right)$, since B_j is an open arc of length $\frac{\pi}{3}$. Across sectors it exceeds $\frac{\pi}{3}$, as shown in the proof of Proposition D.6, and it is less than π, since the direction opposite to a point of B_j lies in the interior of D_{j+2}, which meets no arc B_i.

This topological form of the barrier sharpens the contrast with Theorem D.3. In the realizable regime the feasible region is convex, and margin forces the collapse once the grid resolves it. With a positive error budget the region can fragment, and the collapse can then fail outright, independently of resolution, as it does here. The realizable margin argument uses convexity, whereas this agnostic example has a barrier that persists under refinement.

The barrier raises the question of which additional structure can restore the collapse under a positive error budget.

Remark D.8 (Over-parameterized networks). Proposition D.6 shows that a positive error budget alone creates barriers, so any agnostic collapse must draw on additional structure. For networks the natural candidate is over-parameterization. Under the respective activation and architecture assumptions described in Section 1.3, continuous network sublevel sets are connected exactly (Nguyen, 2019) or with a loss allowance that vanishes as the width grows (Freeman & Bruna, 2017). Appendix D.7 carries the mechanism into the discrete model for two-layer networks of width at least twice the sample size, where it reduces once more to convexity of the loss in the output layer (Theorem D.17).

D.6 The Chamber Graph: Fixed Dimension and Fixed Budget

Throughout this subsection the sample points are nonzero rational vectors $x_i \in \mathbb{Q}^d \setminus \{0\}$ given in binary. The natural scale-invariant configuration space for halfspaces is not a grid but the arrangement of the hyperplanes $H_i = \{w : \langle w, x_i\rangle = 0\}$ in weight space. For $\sigma \in \{-1,+1\}^t$ put $C_\sigma = \{w \in \mathbb{R}^d : \sigma_i\langle w, x_i\rangle > 0 \text{ for all } i\}$. Call σ *realizable* when $C_\sigma \neq \emptyset$.

The sets C_σ are open, as finite intersections of open halfspaces. They are convex cones, since each open halfspace $\{w : \sigma_i\langle w, x_i\rangle > 0\}$ is convex and closed under multiplication by a positive scalar. They are pairwise disjoint and cover $\mathbb{R}^d \setminus \bigcup_i H_i$. Indeed, a point w on no H_i lies in C_σ exactly when $\sigma_i = \operatorname{sign}\langle w, x_i\rangle$ for every $i \in [t]$. A connected subset K of that complement has constant sign vector, since each continuous function $\langle \cdot, x_i\rangle$ is nonzero on K and so does not change sign there. Hence K lies inside a single C_σ.

Each nonempty C_σ is convex, hence connected, and so lies inside a connected component of the complement, which by the previous paragraph lies inside some $C_{\sigma'}$. As the sets C_σ are pairwise disjoint, $\sigma' = \sigma$, and the component equals C_σ. So the nonempty C_σ are exactly the connected components of the complement. These are the *chambers*. Every $w \in C_\sigma$ predicts σ_i on x_i, so the error count $e(\sigma) = |\{i : \sigma_i \neq y_i\}|$ is constant on each chamber.

Write $i \sim j$ when x_j is a nonzero scalar multiple of x_i, that is, when $H_i = H_j$. The classes of $\sim$ are the *parallel classes*. For a class P we write σ^P for σ with the coordinates in P negated.

Two chambers are *adjacent* when their closures share a facet, that is, when the convex cone $\overline{C_\sigma} \cap \overline{C_{\sigma'}}$ has dimension $d-1$. Here the dimension of a convex cone is that of its linear span. By Lemma D.10 this happens exactly when $\sigma' = \sigma^P$ for a single parallel class P, so that the two chambers differ by crossing a single hyperplane of the arrangement.

We call the resulting graph the *chamber graph*, and the chamber graph with feasibility $e \leq k$ the *chamber model*. An instance consists of the sample, the budget k, and two feasible chambers given by their sign vectors. The bottleneck β between two chambers is the minimum, over paths of the chamber graph joining them, of the largest error of a chamber on the path. The chamber graph is connected, as the proof of Corollary D.12 shows, so β is finite and attained by a path. Hence the two chambers are connected in the subgraph F_k of chambers of error at most k, the chamber counterpart of the feasible set of Definition C.1, exactly when $\beta \leq k$.

Consider the plane, on the sample of Proposition D.6, with g_1 and g_2 replaced by rational vectors at angles close enough to ϕ_1 and ϕ_2. These angles are chosen so that the six midpoint directions of the arcs B_j and D_j, and the directions of u and v, keep their sign vectors. All these directions have nonzero inner products with the original data, so their finitely many strict sign inequalities persist under sufficiently small perturbations, and rational vectors can be chosen in those neighborhoods. The r copies of each direction g_j then form one parallel class, and the three distinct lines H_i cut the plane into six chambers, each containing exactly one of these six midpoint directions, since their sign vectors are pairwise distinct. The chamber graph is the cycle of these six chambers in the cyclic order $D_0, B_0, D_1, B_1, D_2, B_2$ of the arcs, with errors alternating between $2r$ and r.

The one fact the algorithms of this subsection need from optimization is that the feasibility of a rational linear system is decidable in polynomial time.

Proposition D.9 (Khachiyan, 1979)**.** *Whether a system of linear inequalities with rational coefficients has a solution is decidable in time polynomial in the bit length of the system.*

Adjacency of chambers has a combinatorial description in terms of their sign vectors, and the algorithms below use it.

Lemma D.10 (Adjacency of chambers)**.** *Let σ, σ' be realizable. Then C_σ and $C_{\sigma'}$ are adjacent if and only if $\sigma' = \sigma^P$ for exactly one parallel class P.*

Proof. Suppose $\sigma' = \sigma^P$. Fix $i_0 \in P$ and write $H = H_{i_0}$. Let $A = \{w : \sigma_j \langle w, x_j \rangle > 0 \text{ for all } j \notin P\}$, an intersection of open halfspaces, hence open and convex. For $j \in P$ write $x_j = c_j x_{i_0}$ with $c_j \neq 0$. At any $w \in C_\sigma$, which exists since σ is realizable, both $\sigma_j c_j \langle w, x_{i_0} \rangle = \sigma_j \langle w, x_j \rangle$ and $\sigma_{i_0} \langle w, x_{i_0} \rangle$ are positive. So their product $\sigma_j c_j \sigma_{i_0} \langle w, x_{i_0} \rangle^2$ is positive, and $\sigma_j c_j \sigma_{i_0} > 0$. Hence, for every $w \in \mathbb{R}^d$ and every $j \in P$, using $\sigma_{i_0}^2 = 1$,

$$\sigma_j \langle w, x_j \rangle = \sigma_j c_j \langle w, x_{i_0} \rangle = (\sigma_j c_j \sigma_{i_0})\, \sigma_{i_0} \langle w, x_{i_0} \rangle,$$

which is a positive multiple of $\sigma_{i_0} \langle w, x_{i_0} \rangle$. So the conditions $\sigma_j \langle w, x_j \rangle > 0$ for $j \in P$ are all equivalent to $\sigma_{i_0} \langle w, x_{i_0} \rangle > 0$. For σ' we have $\sigma'_j c_j \sigma'_{i_0} = (-\sigma_j) c_j (-\sigma_{i_0}) = \sigma_j c_j \sigma_{i_0} > 0$ for $j \in P$, and $\sigma'_{i_0} = -\sigma_{i_0}$. The identity displayed above, with σ' in place of σ, gives $\sigma'_j \langle w, x_j \rangle = \left(\sigma'_j c_j \sigma'_{i_0}\right) \sigma'_{i_0} \langle w, x_{i_0} \rangle = -\left(\sigma_j c_j \sigma_{i_0}\right) \sigma_{i_0} \langle w, x_{i_0} \rangle$ for every w and every $j \in P$. So the conditions $\sigma'_j \langle w, x_j \rangle > 0$ for $j \in P$ are all equivalent to $\sigma_{i_0} \langle w, x_{i_0} \rangle < 0$, while $\sigma'_j = \sigma_j$ for $j \notin P$. Hence $C_\sigma = A \cap \{\sigma_{i_0} \langle w, x_{i_0} \rangle > 0\}$ and $C_{\sigma'} = A \cap \{\sigma_{i_0} \langle w, x_{i_0} \rangle < 0\}$, the first nonempty as σ is realizable and the second as σ' is.

We now show that $\overline{C_\sigma} \cap \overline{C_{\sigma'}}$ has dimension exactly $d-1$, which proves the chambers adjacent. Pick $p \in C_\sigma$ and $q \in C_{\sigma'}$. The segment $[p, q]$ lies in A, which is convex. The affine function $\lambda \mapsto \sigma_{i_0} \langle p + \lambda(q-p), x_{i_0} \rangle$ is positive at $\lambda = 0$ and negative at $\lambda = 1$, so it vanishes at some point of the segment, which lies in $A \cap H$. Hence $A \cap H \neq \emptyset$. As A is open, $A \cap H$ is a nonempty relatively open subset of the hyperplane H, hence of dimension $d-1$. Fix $z \in A \cap H$. As A is open, it contains a ball around z. So there is $\delta_0 > 0$ such that the points $z \pm \epsilon \sigma_{i_0} x_{i_0}$ lie in A for every $\epsilon \in (0, \delta_0)$. Since $\langle z, x_{i_0} \rangle = 0$, these points satisfy $\sigma_{i_0} \langle z \pm \epsilon \sigma_{i_0} x_{i_0}, x_{i_0} \rangle = \pm \epsilon \|x_{i_0}\|^2$. So $z + \epsilon \sigma_{i_0} x_{i_0} \in C_\sigma$ and $z - \epsilon \sigma_{i_0} x_{i_0} \in C_{\sigma'}$ for every $\epsilon \in (0, \delta_0)$, and letting $\epsilon \to 0$ gives $z \in \overline{C_\sigma} \cap \overline{C_{\sigma'}}$. Thus $\overline{C_\sigma} \cap \overline{C_{\sigma'}} \supseteq A \cap H$ has dimension at least $d-1$.

The intersection $\overline{C_\sigma} \cap \overline{C_{\sigma'}}$ has dimension at most $d-1$, for the following reasons. The open set C_σ is disjoint from $C_{\sigma'}$, since $\sigma \neq \sigma'$ as $P \neq \emptyset$, hence disjoint from $\overline{C_{\sigma'}}$. So $\overline{C_\sigma} \cap \overline{C_{\sigma'}}$ lies in $\overline{C_\sigma} \setminus C_\sigma$, the boundary of C_σ. A boundary point z of the open cone C_σ has $\sigma_i \langle z, x_i \rangle \geq 0$ for all i, by continuity. Equality holds for some i, since $z \notin C_\sigma$. So the boundary lies in the finite union of hyperplanes $\bigcup_i H_i$. The set $\overline{C_\sigma} \cap \overline{C_{\sigma'}}$ is a convex cone, as an intersection of closures of convex cones. Were its linear span all of $\mathbb{R}^d$, it would contain a basis $v_1, \ldots, v_d$ of $\mathbb{R}^d$ and hence the nonempty open set $\{\sum_l \lambda_l v_l : \lambda_l > 0\}$, which a finite union of hyperplanes cannot contain, each hyperplane being a Lebesgue null set. Hence the chambers are adjacent.

Conversely, let $F = \overline{C_\sigma} \cap \overline{C_{\sigma'}}$ have dimension $d-1$. Then $\sigma \neq \sigma'$, since $\overline{C_\sigma}$ has dimension d. The open set C_σ is disjoint from $C_{\sigma'}$, hence from $\overline{C_{\sigma'}}$, so $F \subseteq \partial C_\sigma \subseteq \bigcup_i H_i$. The set F is a convex cone, as an intersection of closures of convex cones. Let L be its linear span, of dimension $d-1$. Choose $v_1, \ldots, v_{d-1} \in F$ forming a basis of L, which is possible since F spans L. As F is a convex cone, it contains $\{\sum_l \lambda_l v_l : \lambda_l > 0\}$, the image of the open positive orthant of $\mathbb{R}^{d-1}$ under the linear isomorphism $\lambda \mapsto \sum_l \lambda_l v_l$ onto L. So F has nonempty relative interior $\operatorname{relint} F$ in L.

It remains to identify L with one of the hyperplanes of the arrangement. Suppose L were none of the H_i. Then each $H_i \cap L$ would be a proper subspace of L, since L and H_i both have dimension $d-1$. So $\operatorname{relint} F \subseteq F \subseteq \bigcup_i (H_i \cap L)$ would be a nonempty relatively open subset of L covered by finitely many proper subspaces of L. This is impossible, because each proper subspace of L is a null set for Lebesgue measure on L, while a nonempty relatively open subset of L is not. Hence $L = H_i$ for some i, with parallel class P.

We now use this to recover the parallel class P from σ, σ' themselves. Choose z in the relative interior of F lying on no H_j with $j \notin P$. This is possible because for $j \notin P$ the hyperplane H_j differs from $H_i = L$, so $H_j \cap L$ is a proper subspace of L. Finitely many proper subspaces, each a null set for Lebesgue measure on L, do not cover the nonempty relatively open set $\operatorname{relint} F$. Then choose a ball B around z meeting no H_j with $j \notin P$, which exists because z has positive distance from each of these finitely many closed sets. Then $B \setminus H_i$ consists of two open half-balls, each connected and disjoint from every hyperplane, hence each inside a single chamber. Since $z \in \overline{C_\sigma}$, the chamber C_σ meets B and so contains one half-ball. Since $z \in \overline{C_{\sigma'}}$ as well, $C_{\sigma'}$ meets B and contains one half-ball too, and as the two chambers are distinct and disjoint they contain different

half-balls. On B the functionals $\langle \cdot, x_j \rangle$ with $j \notin P$ do not vanish and so keep their sign, while those with $j \in P$ change sign across H_i. Hence $\sigma' = \sigma^P$. □

In fixed dimension the chamber graph is small enough to be built explicitly. Its size is bounded by the Sauer–Shelah lemma, which bounds the number of members of a family of sets that shatters no large set; we state it here, where it is used.

Lemma D.11 (Sauer–Shelah Sauer, 1972; Shelah, 1972; Vapnik & Chervonenkis, 1971; see Kearns & Vazirani, 1994)**.** *Let $\mathcal{F}$ be a nonempty family of subsets of a t-element set, and let D be the largest size of a subset I shattered by $\mathcal{F}$, that is, such that every subset of I is $I \cap F$ for some $F \in \mathcal{F}$. Then $|\mathcal{F}| \le \sum_{i=0}^{D} \binom{t}{i} = O\left((t+1)^D\right)$ for fixed D. In particular, if $\mathcal{C}$ has Vapnik–Chervonenkis dimension D, then on every sample of t instances the number of distinct labelings realized by the hypotheses of $\mathcal{C}$ is at most $\sum_{i=0}^{D} \binom{t}{i}$. This is because the family of sets $\{i : \mathsf{E}(w, x_i) = 1\}$ shatters no set of more than D indices.*

With this count, the chamber graph of a fixed dimension has polynomially many vertices, so it can be written down and Lemma 2.7 applied to it.

Corollary D.12 (Tractability in fixed dimension; restatement of Corollary 1.9)**.** *For every fixed d, agnostic halfspace reconfiguration in the chamber model is solvable in polynomial time, and the bottleneck β is computable in polynomial time.*

Proof. Let N be the number of chambers. If $t = 0$, the empty sign vector defines the sole chamber $\mathbb{R}^d$, whose error is zero. The two given chambers therefore coincide, their bottleneck is zero, and both computations are immediate. Hence assume $t \ge 1$ for the rest of the proof.

Counting. For realizable σ put $M_\sigma = \{i : \sigma_i = +1\} \subseteq [t]$. The map $\sigma \mapsto M_\sigma$ is injective, since $\sigma_i = +1$ exactly when $i \in M_\sigma$ and $\sigma_i = -1$ otherwise. So N is the size of the family $\mathcal{F} = \{M_\sigma\}$. No $I \subseteq [t]$ with $|I| = d+1$ is shattered by $\mathcal{F}$. The $d+1$ vectors x_i for $i \in I$ are linearly dependent in $\mathbb{R}^d$. That is, $\sum_{i \in I} c_i x_i = 0$ for some $c \ne 0$. Put $\tau_i = \operatorname{sign}(c_i)$ when $c_i \ne 0$ and $\tau_i = +1$ otherwise. If a realizable σ agreed with τ on I, then for $w \in C_\sigma$ we would have

$$0 = \left\langle w, \sum_{i \in I} c_i x_i \right\rangle = \sum_{i \in I} c_i \langle w, x_i \rangle = \sum_{i \in I, c_i \ne 0} |c_i| \tau_i \langle w, x_i \rangle = \sum_{i \in I, c_i \ne 0} |c_i| \sigma_i \langle w, x_i \rangle > 0.$$

The equalities use $c_i = |c_i|\tau_i$ for $c_i \ne 0$ and $\tau_i = \sigma_i$ on I. The final sum is positive because each of its terms is positive, as $\sigma_i \langle w, x_i \rangle > 0$ for every i, and it has at least one term, as $c \ne 0$. This is a contradiction, so the labeling τ of I is not realized. In terms of $\mathcal{F}$: the subset $T = \{i \in I : \tau_i = +1\}$ of I equals $I \cap M_\sigma = \{i \in I : \sigma_i = +1\}$ for no realizable σ. This is because that equality would say that σ agrees with τ on I. So I is not shattered. A subset of a shattered set is shattered. Indeed, suppose I is shattered, $J \subseteq I$, and $T' \subseteq J$. Then a member M_σ with $I \cap M_\sigma = T'$ has $J \cap M_\sigma = J \cap I \cap M_\sigma = J \cap T' = T'$. So a shattered set of more than d indices would contain a shattered set of exactly $d+1$ indices, and $\mathcal{F}$ shatters no set of more than d indices. Lemma D.11 applies to the family $\mathcal{F}$ of subsets of the t-element set $[t]$, whose largest shattered set has size $D \le d$. It gives $N = |\mathcal{F}| \le \sum_{l=0}^{D} \binom{t}{l} \le \sum_{l=0}^{d} \binom{t}{l}$, the added terms being nonnegative. For $t \ge 1$ each of these $d+1$ binomial coefficients satisfies $\binom{t}{l} \le t^l \le t^d$, so $N \le (d+1)t^d$.

Realizability test. A sign vector σ is realizable if and only if the linear system $\sigma_i \langle w, x_i \rangle \ge 1$ for every $i \in [t]$ has a solution $w \in \mathbb{R}^d$. Indeed, a solution satisfies $\sigma_i \langle w, x_i \rangle \ge 1 > 0$ for every i and so lies in C_σ. Conversely, if $w \in C_\sigma$ then $\mu = \min_i \sigma_i \langle w, x_i \rangle$ is positive, and w/μ satisfies $\sigma_i \langle w/\mu, x_i \rangle = \sigma_i \langle w, x_i \rangle / \mu \ge 1$ for every i. The system has rational coefficients of bit length polynomial in the input, and its feasibility is decidable in time polynomial in that bit length (Proposition D.9).

Construction. Compute the parallel classes by testing for each pair i, j whether x_j is a scalar multiple of x_i. Pick a coordinate a with $(x_i)_a \ne 0$, which exists since $x_i \ne 0$, and check whether $x_j = ((x_j)_a / (x_i)_a)\, x_i$; this takes $O(d)$ arithmetic operations on rationals of polynomial bit length. Let $\sigma^{(0)}$ be the sign vector of one of the two chambers given in the instance. Run breadth-first search from $\sigma^{(0)}$. For each discovered σ and each parallel class P, test whether σ^P is realizable. If so, record the edge $\{\sigma, \sigma^P\}$ and enqueue σ^P if it is new. By Lemma D.10 the recorded edges are exactly the adjacencies among discovered chambers, and the error of a chamber is $e(\sigma)$.

The search discovers every chamber because the chamber graph is connected. Indeed, given chambers $C_\sigma, C_{\sigma'}$, pick $p \in C_\sigma$ and $q \in C_{\sigma'}$ with $[p, q]$ meeting no intersection $H_i \cap H_j$ of two distinct hyperplanes. Such a q exists, as we now show.

Fix such an intersection $L = H_i \cap H_j$, a linear subspace of dimension $d - 2$. If $[p, q]$ meets L at $z = p + \lambda(q - p)$, then $\lambda \neq 0$, since p lies in a chamber and so on no hyperplane. Hence $q = p + \lambda^{-1}(z - p)$. So the set of q with $[p, q] \cap L \neq \emptyset$ is contained in $\{p + s(l - p) : l \in L, s \geq 1\} = \{(1 - s)p + sl : l \in L, s \geq 1\}$. This set lies in the linear span of $L \cup \{p\}$, a subspace of dimension at most $(d - 2) + 1 = d - 1$ and hence a Lebesgue null set. Finitely many null sets have measure zero and so do not cover the nonempty open set $C_{\sigma'}$, which has positive measure. This gives q.

We now find the walk itself, by following the segment $[p, q]$. Along $[p, q]$, each function $s \mapsto \langle p + s(q - p), x_i \rangle$ is affine. It is nonzero at $s = 0$ and at $s = 1$, because p and q lie on no hyperplane. So it either never vanishes or vanishes at exactly one $s \in (0, 1)$. Between two consecutive zeros of these finitely many functions, and before the first and after the last, no functional vanishes, so that piece of the segment lies inside a single chamber. At a zero $z \in [p, q] \cap H_i$ no hyperplane outside the parallel class of i passes, by the choice of q. The functionals $\langle \cdot, x_j \rangle$ with j in the class of i are nonzero multiples of $\langle \cdot, x_i \rangle$, so they change sign at z. Those outside the class, in contrast, are nonzero at z and keep their sign on a neighborhood of z.

It remains to identify the chambers on either side of each zero and thread them into a path. By the two facts just shown, the chambers containing the pieces of the segment just before and just after z have sign vectors differing exactly on the parallel class of i, and so are adjacent by Lemma D.10. The piece containing $s = 0$ lies in C_σ, since it contains p. Similarly, the piece containing $s = 1$ lies in $C_{\sigma'}$, since it contains q. Consecutive pieces lie in adjacent chambers, so the chambers met by $[p, q]$, in the order of the pieces, form a walk, and hence contain a path, from C_σ to $C_{\sigma'}$ in the chamber graph.

The search processes each of the N chambers once and makes one realizability test per parallel class for each, so it makes at most $Nt \leq (d + 1)t^{d+1}$ realizability tests. Each of these is a linear program in d variables and t constraints, and so is decided in time polynomial in the input length. Storing the discovered sign vectors in a binary trie of depth t makes each test of whether σ^P is new take $O(t)$ steps. The error $e(\sigma)$ of each chamber is likewise computed in $O(t)$ steps. The chamber graph with its vertex errors is therefore built in time $O(t^{d+1}) \cdot \mathrm{poly}(\text{input length})$, polynomial for fixed d. Polynomial-time algorithms for the static problem in fixed dimension are classical (Johnson & Preparata, 1978).

The chamber model is then an instance of agnostic reconfiguration on an explicitly given graph. Lemma 2.7 applies to it, with H the chamber graph, the weight of a vertex σ its error $e(\sigma)$, and the two given chambers as its two distinguished vertices. Its hypotheses hold: H is finite, with $N \leq \max\left(1, (d + 1)t^d\right)$ vertices and at most Nt edges, each vertex having at most one neighbor per parallel class, and the search has written it down together with its weights. Its conclusion, with these objects, is that β, the minimum over paths of the chamber graph between the two given chambers of the largest error on the path, is computable in time polynomial in N and Nt. This is polynomial in the input length for fixed d. The two chambers are connected in F_k exactly when $\beta \leq k$. Indeed, a path attaining β has every error at most β and so lies in F_k when $\beta \leq k$. Conversely, a path inside F_k has every error at most k. This is the computation of Proposition C.14 carried out on the chamber graph. □

The barrier of Proposition D.6 has its counterpart in the chamber model, where the chambers of u and v have error r and are separated on both sides of the six-cycle by chambers of error $2r$, so that $\beta = 2r$ there. Such barriers, though real, are therefore efficiently detectable in the chamber model whenever the dimension is fixed: disconnection of the feasible region does not entail intractability there.

The dimension is one route to tractability. The error budget is another, and it is independent of the dimension. A chamber of error at most k misclassifies few points, and this alone makes the feasible region small.

Proposition D.13 (Tractability for a bounded error budget; restatement of Proposition 1.10)**.** *For every fixed k, agnostic halfspace reconfiguration in the chamber model with error budget k is solv-*

able in polynomial time, uniformly in the dimension d; the bottleneck β is computed in the same time whenever $\beta \leq k$.

Proof. If $t = 0$, the empty sign vector defines the sole chamber $\mathbb{R}^d$, whose error is zero. The two given chambers therefore coincide, their bottleneck is zero, and the conclusion is immediate. Hence assume $t \geq 1$ for the rest of the proof. A chamber of error at most k has a sign vector differing from the label vector y in at most k coordinates, so it is determined by its set $M \subseteq [t]$ of misclassified indices. Its sign vector y^M is y with the coordinates in M negated. The feasible chambers are therefore in bijection with the sets M of size at most k for which y^M is realizable, that is, for which the system $y_i^M \langle w, x_i \rangle > 0$ for every $i \in [t]$ has a solution. This holds if and only if the linear system $y_i^M \langle w, x_i \rangle \geq 1$ for every $i \in [t]$ is feasible. Indeed, a solution of the latter satisfies the strict system, and a solution w of the strict system, divided by $\mu = \min_i y_i^M \langle w, x_i \rangle > 0$, satisfies $y_i^M \langle w/\mu, x_i \rangle \geq 1$ for every i. The feasibility of the linear system is decided in time polynomial in the bit length of the rational data (Proposition D.9). There are $\sum_{j=0}^{k} \binom{t}{j}$ candidate sets, at most $(k+1)t^k$ when $t \geq 1$, since $\binom{t}{j} \leq t^j \leq t^k$ for $j \leq k$. So the vertex set of F_k is produced in time $O(t^k) \cdot \mathrm{poly}(\text{input length})$, with no dependence on d beyond the size of each linear program.

By Lemma D.10 two feasible chambers with index sets M, M' are adjacent exactly when $y^{M'} = (y^M)^P$ for a single parallel class P. Negating the coordinates in M and then those in P negates exactly the coordinates in $M \triangle P$, a coordinate in $M \cap P$ being negated twice, so $(y^M)^P = y^{M \triangle P}$. As $M \mapsto y^M$ is injective, the two chambers are adjacent exactly when $M' = M \triangle P$. The parallel classes are computed by the pairwise scalar-multiple tests of the proof of Corollary D.12.

The edges of F_k are enumerated by forming $M \triangle P$ for each feasible M and each of the at most t classes P. We keep those results that are feasible, that is, those of size at most k that are among the feasible sets already produced. There are at most t such edges per vertex. Thus F_k is an explicitly given graph with $O(t^k)$ vertices and $O(t^{k+1})$ edges, and Proposition C.14 decides whether the two given chambers are connected in it.

If they are, then $\beta \leq k$, since a path inside F_k has every error at most k. A path of the chamber graph attaining β exists, because β is a minimum of values in $\{0, \ldots, t\}$ over a set of paths that is nonempty, the chamber graph being connected by the proof of Corollary D.12. This path has every error at most $\beta \leq k$ and so lies in F_k. Hence the minimum-bottleneck value on F_k (Lemma 2.7) is at most β, and it is at least β because every path in F_k is a path of the chamber graph. So it equals β. If they are not, then $\beta > k$, since a path of the chamber graph attaining a value $\beta \leq k$ would lie in F_k by the same argument and connect them. The algorithm reports so. □

Together with Corollary D.12 this confines any intractability to the regime in which the dimension and the error budget grow at once. Fixing either alone places the problem in polynomial time, the dimension through the $O(t^d)$ count of chambers and the budget through the $O(t^k)$ count of low-error chambers.

Remark D.14 (Growing dimension). When d grows with the input the chamber graph can have exponentially many vertices, all 2^t sign vectors being realizable when the points are linearly independent (see below). So Corollary D.12 no longer applies. The problem remains in PSPACE.

Run the nondeterministic procedure of the proof of Theorem 1.1 with the sign vectors $\sigma \in \{-1, +1\}^t$ as configurations in place of the strings of $\{0,1\}^m$, so that t plays the role of m and the counter has $t + 1$ bits. Replace the test $\mathsf{cons}_S(w)$ by the test that σ is realizable and $e(\sigma) \leq k$, and replace the move relation by the relation $\sigma' = \sigma^P$ for some parallel class P. Both are decided in polynomial time. Realizability is decided by the linear program of the proof of Corollary D.12 and Proposition D.9, and the error is decided by counting. The move relation is decided by computing the parallel classes as there and checking that the set of coordinates on which σ and σ' differ is one of them. This relation is symmetric, and on realizable sign vectors it is exactly adjacency of chambers by Lemma D.10. A simple path between two feasible chambers visits each of the at most 2^t chambers at most once. The sign vectors and counter use $O(t)$ bits, and each polynomial-time realizability test uses polynomial workspace in the full input length. Reusing that workspace keeps the total space polynomial, so Proposition 2.6 applies as there. But the exact complexity of the problem is open.

The chambers' sign vectors are the *topes* of the oriented matroid of the data. After retaining one representative of each parallel class, their graph is a partial cube (Björner et al., 1999): graph distance equals Hamming distance in the reduced sign vectors. The error budget is still measured on the full sample sign vectors, with every repeated example counted. The moves are between topes that differ on a single parallel class, a single sign when the oriented matroid is simple.

The natural reductions do not settle it. An error budget gives one bound on the total number of misclassified examples; turning it into the separate constraints of a clause formula or a constraint-logic instance requires an additional encoding argument. Within an orthant the sign of a non-axis functional can depend on the magnitudes of the weights, so Boolean coordinate signs alone do not in general determine the chamber.

The static decision problem of whether a halfspace meets a given error budget is NP-complete once the dimension grows (Johnson & Preparata, 1978); minimizing that error is NP-hard to approximate within any constant factor. For every $\epsilon, \delta > 0$ it is NP-hard to tell whether some halfspace is correct on a $1 - \epsilon$ fraction of a sample from the hypercube or none is correct on more than a $\frac{1}{2} + \delta$ fraction (Guruswami & Raghavendra, 2009, Theorem 8.1). A halfspace with a threshold, $\{x : \langle w, x\rangle > \theta\}$, is the homogeneous halfspace of weight $(w, -\theta)$ on the lifted point $(x, 1) \in \mathbb{Q}^{d+1} \setminus \{0\}$, the constant feature of Definition D.1, so these statements hold as well for homogeneous halfspaces on nonzero points, the setting of the chamber model.

We now adapt this to the least error measure used here. On the lifted samples just described, a sufficiently small shift of θ preserves all nonzero signs and removes every tie. For a fixed binary tie convention, shift toward the label assigned at a tie; then all predictions are preserved. If instead every tie counts as an error, either direction of a sufficiently small nonzero shift preserves the correctness of every previously correct example and can only improve the tied examples. Conversely, a weight with no ties has the same error under both conventions. Thus the minimum error under either convention equals the minimum over chambers, which exclude ties, and the static hardness transfers to the chamber setting. This gives the inapproximability within every constant factor $c \geq 1$: take $\delta = \frac{1}{4}$ and $\epsilon < \frac{1}{4c}$. On a sample of size t of the first kind the least error is at most ϵt, so an algorithm returning a halfspace within factor c of the least error returns at most $c\epsilon t < \frac{t}{4}$ errors. A sample of the second kind is different: every halfspace on it errs on at least $\left(\frac{1}{2} - \delta\right) t = \frac{t}{4}$ points, so it returns at least $\frac{t}{4}$. The two kinds are therefore told apart.

So the landscape whose sublevel sets we navigate is already intractable to optimize globally. But that problem is to *find* a low-error halfspace, whereas reconfiguration is handed two of them and asks only whether they are connected, and we know no reduction relating the two.

Any hardness moreover needs the data to have rank below t. If the points $x_1, \ldots, x_t$ are linearly independent, so that $t \leq d$, then every sign vector σ is realizable. Indeed, the linear map $w \mapsto (\langle w, x_i\rangle)_{i \in [t]}$ from $\mathbb{R}^d$ to $\mathbb{R}^t$ has rank t and so is onto. Hence there is a w with $\langle w, x_i\rangle = \sigma_i$ for every i, and this w has $\sigma_i \langle w, x_i\rangle = 1 > 0$. No x_j is a scalar multiple of an x_i with $i \neq j$, by linear independence, so the parallel classes are singletons and the hyperplanes are pairwise distinct. By Lemma D.10, two sign vectors are adjacent exactly when they differ in one coordinate: the tope graph is the whole hypercube $\{-1, +1\}^t$.

We now exhibit an explicit walk between any two topes and compute its bottleneck. Between two topes σ_0, σ_1, flip first the coordinates on which σ_0 disagrees with y and σ_1 agrees. Then flip those on which σ_0 agrees and σ_1 disagrees. These are exactly the coordinates on which σ_0 and σ_1 differ, since of two signs that differ from each other exactly one equals y_i. So the walk ends at σ_1, and each of its steps changes one coordinate and is an edge of the hypercube. Each flip of the first kind lowers the error by one and each of the second kind raises it by one. The error along this path decreases from $e(\sigma_0)$ and then increases to $e(\sigma_1)$, so its maximum is $\max(e(\sigma_0), e(\sigma_1))$. This is also a lower bound for every path, so the bottleneck equals it. As both given chambers are feasible, $\max(e(\sigma_0), e(\sigma_1)) \leq k$, so on linearly independent data every instance is a yes-instance.

A barrier, that is, $\beta > \max(e(\sigma_0), e(\sigma_1))$, therefore requires a sign vector within Hamming distance $b = \max(e(\sigma_0), e(\sigma_1))$ of y that is not a tope. Indeed, if every sign vector within distance b of y is a tope, then either $b = 0$ and $\sigma_0 = \sigma_1 = y$, or $b \geq 1$ and the parallel classes are singletons. The parallel classes are singletons when $b \geq 1$ because, for $i \sim j$ with $i \neq j$, every tope has the same product $\sigma_i \sigma_j$, as in the proof of Lemma D.10. But y and y with coordinate i negated, both within

distance b, have opposite products. When $b \geq 1$, then, every vertex of the path above has error at most b, that is, lies within Hamming distance b of y, and so is a tope. Consecutive vertices of the path differ in one coordinate, which is a parallel class since the classes are singletons, so they are adjacent by Lemma D.10. So $\beta = b$ when $b \geq 1$, and when $b = 0$ the two chambers coincide and $\beta = 0 = b$.

We conjecture the problem is PSPACE-complete when the dimension grows with the input. By Proposition D.13, unless P = PSPACE, any such hardness must use an error budget that grows as well, and even NP-hardness, in this doubly growing regime, is open.

D.7 Over-Parameterized Two-Layer Networks

For two-layer networks of width at least twice the sample size, the discrete model admits connectivity with an explicit loss allowance that vanishes under refinement. In this subsection the sample size is written n rather than t, matching the network literature. Fix a nonempty sample $(x_i, y_i)_{i=1}^n$ with $x_i \in \mathbb{R}^d$, $\|x_i\|_2 \leq R$, and real labels y_i. Consider the two-layer network $f_\theta(x) = \sum_{j=1}^m a_j \operatorname{relu}(\langle w_j, x\rangle)$, where $\operatorname{relu}(u) = \max(0, u)$ is the rectified linear unit and the parameters are $\theta = (a, w) \in \left(\frac{1}{s}\mathbb{Z}\right)^m \times \left(\frac{1}{s}\mathbb{Z}\right)^{m\times d}$. The move relation is the one of Definition D.1: a single coordinate changes by $\frac{1}{s}$. The hidden weights are confined to the box $\|w_j\|_\infty \leq W$ for a fixed nonnegative $W \in \frac{1}{s}\mathbb{Z}$, so that

$$|\operatorname{relu}(\langle w_j, x_i\rangle)| \leq |\langle w_j, x_i\rangle| \leq \|w_j\|_2\|x_i\|_2 \leq \sqrt{d}\|w_j\|_\infty\|x_i\|_2 \leq W\sqrt{d}R =: G \qquad \text{for all } i, j.$$

The first inequality holds because $0 \leq \operatorname{relu}(u) \leq |u|$. The second is the Cauchy–Schwarz inequality. The third holds because each of the d coordinates of w_j has absolute value at most $\|w_j\|_\infty$, so that $\|w_j\|_2^2 \leq d\|w_j\|_\infty^2$. The last uses $\|w_j\|_\infty \leq W$ and $\|x_i\|_2 \leq R$.

Next we fix the loss function and record a Lipschitz constant for it, one that is used throughout the section. The loss is $L(\theta) = \frac{1}{n}\sum_i \ell(f_\theta(x_i), y_i)$ with $\ell(\cdot, y_i)$ convex and λ-Lipschitz on $\mathbb{R}$, as the absolute loss is, and as the hinge and logistic losses are for labels in $\{-1, +1\}$. The network bottleneck is the infimum, over finite lattice paths between the endpoints within the hidden-weight box, of the largest loss on a path. For the squared loss $\ell(f, y_i) = (f - y_i)^2$, which is convex but not Lipschitz on $\mathbb{R}$, let U be a bound on the absolute value of the prediction $f_\theta(x_i)$, and take $\lambda = 2(U + \max_i |y_i|)$. This bound must be valid at every network the proofs below evaluate. These are the vertices of the paths they construct, and the points on the segments between output vectors against which those vertices are compared. This λ is a Lipschitz constant of $\ell(\cdot, y_i)$ on $[-U, U]$, since for $f, f' \in [-U, U]$ we have $\left|(f - y_i)^2 - (f' - y_i)^2\right| = |f - f'||f + f' - 2y_i| \leq 2(U + |y_i|)|f - f'|$. Every use of the Lipschitz property below compares two predictions in $[-U, U]$, so this bound applies throughout.

We now introduce notation for a network's predictions, and fix a value of U that is valid for every network the proofs below evaluate. Write $\phi_j = (\operatorname{relu}(\langle w_j, x_i\rangle))_{i=1}^n \in \mathbb{R}^n$ for the activation of neuron j on the sample, and $G_i = (\operatorname{relu}(\langle w_j, x_i\rangle))_{j=1}^m \in \mathbb{R}^m$ for the activations of all neurons at sample i. With this notation, $f_\theta(x_i) = \langle a, G_i\rangle$, $\|\phi_j\|_1 \leq nG$, and $\|G_i\|_1 \leq mG$. The *prediction vector* is $F(\theta) = (f_\theta(x_i))_i = \sum_j a_j\phi_j$. When the two networks to be connected have output weights of ℓ_∞-norm at most K, the value $U = m^2KG$ suffices below. Every output vector a that the proofs below evaluate satisfies $\|a\|_\infty \leq mK$. This holds for the path vertices, by the coordinatewise clause of Theorem D.15 and the output-weight clause of Lemma D.16. It holds for the rounded vector $\hat{a}$, by Step 2 of the proof of Lemma D.16. And it holds for the points of the segments, because each such point is a convex combination of two vectors of ℓ_∞-norm at most mK. Hence $|f_\theta(x_i)| = |\langle a, G_i\rangle| \leq \|a\|_\infty\|G_i\|_1 \leq mK \cdot mG = m^2KG$.

Two facts are used throughout. First, with the hidden weights fixed, $a \mapsto F$ is linear, so L is convex in a. Indeed, each $a \mapsto \ell(\langle a, G_i\rangle, y_i)$ is the convex function $\ell(\cdot, y_i)$ composed with a linear map, hence convex, and an average of convex functions is convex.

Second, $L = \frac{1}{n}\sum_i \ell(F_i, y_i)$ depends on the network only through its prediction vector. For two networks with prediction vectors F, F' and losses L, L', the triangle inequality together with the

λ-Lipschitz property of each $\ell(\cdot, y_i)$ give

$$|L' - L| \le \frac{1}{n}\sum_i |\ell(F_i', y_i) - \ell(F_i, y_i)| \le \frac{\lambda}{n}\sum_i |F_i' - F_i| = \frac{\lambda}{n}\|F' - F\|_1.$$

For output vectors a, a' at the same hidden weights, $F_i' - F_i = \langle a' - a, G_i\rangle$, so Hölder's inequality gives $|F_i' - F_i| \le \|G_i\|_1\|a' - a\|_\infty$. Using $\|G_i\|_1 \le mG$ for each of the n indices i,

$$|L(a', w) - L(a, w)| \le \frac{\lambda}{n}\sum_i \|G_i\|_1\|a' - a\|_\infty \le \lambda mG\|a' - a\|_\infty.$$

Likewise the Cauchy–Schwarz inequality gives $|F_i' - F_i| \le \|G_i\|_2\|a' - a\|_2$. Also $\|G_i\|_2 \le \sqrt{m}G$, since each of the m entries of G_i has absolute value at most G, so

$$|L(a', w) - L(a, w)| \le \frac{\lambda}{n}\sum_i \|G_i\|_2\|a' - a\|_2 \le \lambda\sqrt{m}G\|a' - a\|_2.$$

We write $\kappa := \lambda\sqrt{m}G$ for the ℓ_2-Lipschitz constant of L in the output weights. Note that $\kappa\sqrt{m} = \lambda mG$.

The *active set* of θ is $\{j : a_j \neq 0\}$. We first connect networks already in disjoint-support form, then show that over-parameterization lets any low-loss network be brought to that form.

Theorem D.15 (Discrete connectivity under disjoint support)**.** *Let θ_A and θ_B satisfy $L(\theta_A), L(\theta_B) \le \eta$ and have disjoint active sets. Then there is a single-coordinate-move lattice path from θ_A to θ_B along which the loss never exceeds $\eta + \kappa\frac{\sqrt{m}}{2s} = \eta + \frac{\lambda mG}{2s}$, and along which every coordinate lies between its values at θ_A and at θ_B. If the grid is refined to resolution $s' = ks$ with $k \in \mathbb{Z}_{\ge 1}$, both endpoints remain lattice points and the bound becomes $\eta + \frac{\lambda mG}{2s'}$; in particular it is at most $\eta + \epsilon$, for every $\epsilon > 0$, once $s' \ge \frac{\lambda mG}{2\epsilon}$.*

Proof. Let S_A and S_B be the active sets, so $S_A \cap S_B = \emptyset$, and let $N = \{1, \dots, m\} \setminus (S_A \cup S_B)$. Write a^A, w^A and a^B, w^B for the output and hidden weights of the two networks.

Phase 1. For each $j \in S_B \cup N$, move w_j from w_j^A to w_j^B one coordinate at a time. Each coordinate changes monotonically, in steps of $\frac{1}{s}$, from its value in w_j^A to its value in w_j^B, and both of these values are lattice points of $\frac{1}{s}\mathbb{Z}$. Every coordinate therefore stays between those two values throughout the move. Since $j \notin S_A$, we have $a_j = 0$ throughout. So ϕ_j enters F with coefficient zero, and neither F nor L changes: the loss stays at $L(\theta_A)$. Call the result θ_A''. Its hidden weights are w_j^A for $j \in S_A$ and w_j^B otherwise, and its output vector is a^A.

Phase 2. Hold the hidden weights of θ_A'' fixed, and route the output vector from a^A to a^B by Lemma D.2 in $\left(\frac{1}{s}\mathbb{Z}\right)^m$. The lemma is stated for $\left(\frac{1}{s}\mathbb{Z}\right)^d$, but its statement and proof hold verbatim in any dimension, here m. At a^A only the neurons of S_A contribute, with hidden weights w^A, so $F = F(\theta_A)$ and the loss is $L(\theta_A)$. Likewise, at a^B only the neurons of S_B contribute, with hidden weights w^B, so $F = F(\theta_B)$ and the loss is $L(\theta_B)$. Write w for these fixed hidden weights, and consider a point $z = (1-r)a^A + ra^B$ of the segment $[a^A, a^B]$, for $r \in [0, 1]$. By convexity of L in a, its loss satisfies $L(z, w) \le (1-r)L(\theta_A) + rL(\theta_B) \le \max(L(\theta_A), L(\theta_B)) \le \eta$. Each vertex v of the routed path has $\|v - z\|_\infty \le \frac{1}{2s}$ for some such z. By the bound $|L(v, w) - L(z, w)| \le \lambda mG\|v - z\|_\infty$ established above, $L(v, w) \le \eta + \lambda mG\frac{1}{2s}$. By Lemma D.2 each coordinate of v lies between its values at a^A and a^B.

Phase 3. For each $j \in S_A$, we now have $a_j = a_j^B = 0$, because $j \notin S_B$. Move w_j from w_j^A to w_j^B coordinatewise and monotonically, as in Phase 1. As in Phase 1, the loss does not change. The result is θ_B.

Concatenating the three phases gives the path. Every hidden weight coordinate only ever takes values between its values in w_j^A and w_j^B, both of absolute value at most W. So the box $\|w_j\|_\infty \le W$, and with it the bound G, apply throughout. For the refined grid, $\frac{1}{s}\mathbb{Z} \subseteq \frac{1}{s'}\mathbb{Z}$, so both endpoints are lattice points of resolution s'. Phases 1 and 3 move hidden weights between their endpoint values in steps of $\frac{1}{s'}$, at output weight zero, and so change no loss. Phase 2, on the grid of resolution s',

places each routed vertex within ℓ_∞-distance $\frac{1}{2s'}$ of the segment, by Lemma D.2. So the bound reads $\eta + \frac{\lambda m G}{2s'}$. □

The next lemma supplies the reduction promised above: it moves any network's active set into a prescribed set of at least n neurons, without raising the loss along the way by more than a controlled amount.

Lemma D.16 (Reduction to a prescribed set of neurons)**.** *Let $H \subseteq \{1, \dots, m\}$ with $|H| \geq n$, and let $\theta = (a, w)$ be a network with $\|a\|_\infty \leq K$. There is a single-coordinate-move lattice path from θ to a network θ' whose active set is contained in H such that, along the path, the loss never exceeds $L(\theta) + \frac{\lambda G(m+n)}{2s}$, the output weights satisfy $\|a\|_\infty \leq mK$, and every hidden weight vector is a hidden weight vector of θ or lies coordinatewise between two of them. The endpoint satisfies $L(\theta') \leq L(\theta) + \frac{\lambda n G}{2s}$.*

Proof. Let $F = \sum_j a_j \phi_j$ be the prediction vector of θ. Let r be the dimension of the span of $\phi_1, \dots, \phi_m$. Since these vectors lie in $\mathbb{R}^n$, we get $r \leq n$.

Step 1: a basis with bounded coefficients. Let B be an r-subset such that $\{\phi_j\}_{j \in B}$ is linearly independent. Let Φ_B be the matrix whose columns are ϕ_j for $j \in B$. Define $\mathrm{vol}(B) = \det\left(\Phi_B^\mathsf{T} \Phi_B\right)^{1/2}$. There are finitely many such B, and at least one exists, since the family $\phi_1, \dots, \phi_m$ spans $\mathrm{span}\{\phi_1, \dots, \phi_m\}$ and therefore contains a basis of that span, and a basis has r elements. Each has $\mathrm{vol}(B) > 0$, because the Gram matrix $\Phi_B^\mathsf{T} \Phi_B$ is positive definite: $x^\mathsf{T} \Phi_B^\mathsf{T} \Phi_B x = \|\Phi_B x\|_2^2 > 0$ for $x \neq 0$ by linear independence of the columns. Choose B maximizing $\mathrm{vol}(B)$.

We now show that every ϕ_k is a combination of the basis $\{\phi_j\}_{j \in B}$ whose coefficients are bounded by 1 in absolute value, using the maximality of $\mathrm{vol}(B)$. When $r = 0$, every ϕ_k is 0. The only such B is then the empty set $\emptyset$, and the empty determinant is by convention equal to 1. Everything below then holds with empty sums, giving $F = 0$ and $\hat{a} = 0$. The r independent vectors ϕ_j with $j \in B$ span the r-dimensional space $\mathrm{span}\{\phi_1, \dots, \phi_m\}$. So for every k there are unique coefficients c_{kj} with $\phi_k = \sum_{j \in B} c_{kj} \phi_j$. For $k \in B$ these coefficients are $c_{kj} = \delta_{kj}$, which is one when $k = j$ and zero otherwise.

We show $|c_{kj}| \leq 1$ for $k \notin B$ by comparing $\mathrm{vol}(B)$ with the volume of a basis obtained by swapping j for k. Fix $k \notin B$ and $j \in B$. Index the rows and columns of the $r \times r$ matrix E by B in the same order used for the columns of Φ_B. Let E equal the identity except in column j, whose entries in that column are the coefficients $(c_{ki})_{i \in B}$. Then $\Phi_B E$ is Φ_B with its j-th column replaced by $\sum_{i \in B} c_{ki} \phi_i = \phi_k$. That is, $\Phi_B E$ is the matrix $\Phi_{(B \setminus \{j\}) \cup \{k\}}$, with ϕ_k in the position of ϕ_j. Every column of E other than the j-th is a unit column e_i with $i \neq j$. Each such column therefore has j-th entry equal to 0. So the j-th row of E is zero outside column j, and its entry in column j is c_{kj}. Expanding $\det E$ along that row gives c_{kj} times the minor obtained by deleting row j and column j, which is the identity matrix of size $r - 1$. So $\det E = c_{kj}$.

This determinant is what relates the volume of B to the volume of the set obtained by swapping j for k, which we compare next. By multiplicativity of the determinant, $\det\left((\Phi_B E)^\mathsf{T} \Phi_B E\right) = \det(E^\mathsf{T}) \det\left(\Phi_B^\mathsf{T} \Phi_B\right) \det(E) = c_{kj}^2 \det\left(\Phi_B^\mathsf{T} \Phi_B\right)$. Reordering the columns of a matrix M replaces $M^\mathsf{T} M$ by $P^\mathsf{T} M^\mathsf{T} M P$ for a permutation matrix P, and $(\det P)^2 = 1$, so the Gram determinant does not depend on the order of the columns. Let $B' = (B \setminus \{j\}) \cup \{k\}$. The Gram determinant of $\{\phi_i\}_{i \in B'}$ is therefore $c_{kj}^2 \, \mathrm{vol}(B)^2$. If $c_{kj} = 0$, then $|c_{kj}| \leq 1$. If $c_{kj} \neq 0$, this Gram determinant is positive, so the Gram matrix of $\{\phi_i\}_{i \in B'}$ is nonsingular. Hence these r columns are linearly independent: $\Phi_{B'} x = 0$ would give $\Phi_{B'}^\mathsf{T} \Phi_{B'} x = 0$, and nonsingularity of the Gram matrix then forces $x = 0$. Since $k \notin B$, the set B' has r elements, so it is one of the subsets over which B was chosen, with $\mathrm{vol}(B') = |c_{kj}| \, \mathrm{vol}(B)$. Maximality of $\mathrm{vol}(B)$ then gives $|c_{kj}| \, \mathrm{vol}(B) \leq \mathrm{vol}(B)$, and dividing by $\mathrm{vol}(B) > 0$ gives $|c_{kj}| \leq 1$.

This bound on the coefficients gives the representation of F that Step 2 rounds. Hence

$$F = \sum_k a_k \phi_k = \sum_{j \in B} \left(\sum_k a_k c_{kj} \right) \phi_j =: \sum_{j \in B} a_j^\sharp \phi_j,$$

where

$$|a_j^\sharp| \le \sum_k |a_k||c_{kj}| \le \sum_k |a_k| = \|a\|_1 \le mK.$$

Step 2: rounding. Let $\hat{a}_j$ be a nearest point of $\frac{1}{s}\mathbb{Z}$ to $a_j^\sharp$ for $j \in B$, and let $\hat{a}_j = 0$ otherwise. Then $\hat{a}$ is supported in B, and $|\hat{a}_j - a_j^\sharp| \le \frac{1}{2s}$. Since every $a_k \in \frac{1}{s}\mathbb{Z}$, also $\|a\|_1 = \sum_k |a_k| \in \frac{1}{s}\mathbb{Z}$. And $|a_j^\sharp| \le \|a\|_1$, by Step 1. The nearest points of $\frac{1}{s}\mathbb{Z}$ to a real number u are among $\frac{\lfloor su \rfloor}{s}$ and $\frac{\lceil su \rceil}{s}$. When $u \in [-p, p]$ with $p \in \frac{1}{s}\mathbb{Z}$, both of these lie in $[-p, p]$. Indeed, since $-sp \le su \le sp$ with $\pm sp$ integers, we get $-sp \le \lfloor su \rfloor$ and $\lceil su \rceil \le sp$. With $u = a_j^\sharp$ and $p = \|a\|_1$ this gives $|\hat{a}_j| \le \|a\|_1 \le m\|a\|_\infty \le mK$. Let $\hat{F} = \sum_j \hat{a}_j \phi_j$. Using $|B| = r \le n$ and $\|\phi_j\|_1 \le nG$,

$$\|\hat{F} - F\|_1 = \left\| \sum_{j \in B} (\hat{a}_j - a_j^\sharp)\phi_j \right\|_1 \le \sum_{j \in B} |\hat{a}_j - a_j^\sharp| \|\phi_j\|_1 \le n \cdot \frac{1}{2s} \cdot nG,$$

so that, by the bound $|L' - L| \le \frac{\lambda}{n}\|F' - F\|_1$ above,

$$L(\hat{a}, w) \le L(\theta) + \frac{\lambda}{n} \cdot \frac{n^2 G}{2s} = L(\theta) + \frac{\lambda n G}{2s}.$$

Step 3: routing. With hidden weights fixed, route the output vector from a to $\hat{a}$ by Lemma D.2. By convexity of L in the output vector, every point of the segment has loss at most $\max(L(\theta), L(\hat{a}, w)) \le L(\theta) + \frac{\lambda n G}{2s}$. Each vertex v is within ℓ_∞-distance $\frac{1}{2s}$ of some point z of the segment. By the ℓ_∞ bound $|L(v, w) - L(z, w)| \le \lambda m G \|v - z\|_\infty$, it therefore has loss at most $L(\theta) + \frac{\lambda n G}{2s} + \frac{\lambda m G}{2s}$. By Lemma D.2 each vertex lies coordinatewise between a and $\hat{a}$. Since $\|a\|_\infty \le K \le mK$ and $\|\hat{a}\|_\infty \le mK$, the vertex also has ℓ_∞-norm at most mK. The active set is now contained in B, and the prediction vector is $\hat{F}$.

Step 4: relocation. Let C denote the current active set. At the start and at the end of every round of Step 4, described next, two invariants hold: $|C| \le n$, and the prediction vector equals $\hat{F}$. Both hold at the start of the first round: after Step 3 the output vector is $\hat{a}$, so $C \subseteq B$ with $|B| = r \le n$, and the prediction vector is $\hat{F}$. Since the loss depends on the network only through its prediction vector, every network along the way with prediction vector $\hat{F}$ has loss $L(\hat{a}, w)$.

We now describe one round of the relocation, which moves one neuron's contribution from a source j outside H to a target j' inside H. While $C \not\subseteq H$, pick $j \in C \setminus H$. Since $j \in C \setminus H$, we have $|C \cap H| \le |C| - 1 \le n - 1 < |H|$, so some $j' \in H$ is inactive. Move $w_{j'}$ to w_j one coordinate at a time, each coordinate monotonically in steps of $\frac{1}{s}$. Since $a_{j'} = 0$, this leaves the prediction vector $\hat{F}$ and the loss $L(\hat{a}, w)$ unchanged, and afterwards $\phi_{j'} = \phi_j$. Let $q = s|a_j|$, a natural number since $a_j \in \frac{1}{s}\mathbb{Z}$, and $\nu = \operatorname{sign}(a_j)$. Repeat q times: change a_j by $-\nu\frac{1}{s}$, then change $a_{j'}$ by $+\nu\frac{1}{s}$. Each repetition starts from the prediction vector $\hat{F}$: the first because the move of $w_{j'}$ left it unchanged, and each later one because the previous repetition restored it. After the first move of a repetition, the prediction vector is $\hat{F} - \nu\frac{1}{s}\phi_j$, since a_j has decreased by $\nu\frac{1}{s}$. After the second move it is $\hat{F} - \nu\frac{1}{s}\phi_j + \nu\frac{1}{s}\phi_{j'} = \hat{F}$ again, since $\phi_{j'} = \phi_j$. So at every vertex $\|F - \hat{F}\|_1 \le \frac{1}{s}\|\phi_j\|_1 \le \frac{nG}{s}$, and the loss is at most

$$L(\hat{a}, w) + \frac{\lambda}{n} \cdot \frac{nG}{s} \le L(\theta) + \frac{\lambda n G}{2s} + \frac{\lambda G}{s} = L(\theta) + \frac{\lambda G(n+2)}{2s} \le L(\theta) + \frac{\lambda G(n+m)}{2s},$$

using $m \ge 2$, which holds because j and j' are two distinct neurons. Along the round a_j moves monotonically from its value to 0 and $a_{j'}$ from 0 to that value, so $\|a\|_\infty \le mK$ persists. At the end, j is inactive and j' carries j's former weight. So C is replaced by $(C \setminus \{j\}) \cup \{j'\}$, of the same size, with prediction vector $\hat{F}$ and loss $L(\hat{a}, w) \le L(\theta) + \frac{\lambda n G}{2s}$.

It remains to check that the rounds terminate inside H, and that every hidden weight visited during Step 4 still lies in the box the theorem needs. Each round decreases $|C \setminus H|$ by one, so after at most n rounds $C \subseteq H$. Every hidden weight vector along the path is a hidden weight vector of θ or lies coordinatewise between two of them. Steps 1 to 3 move no hidden weight, and in Step 4 only the

targets j' are moved. A target lies in H, and from the end of its round on it carries a nonzero output weight. No later round changes that weight, since later rounds only empty neurons outside H and fill inactive ones. So no neuron is a target twice, and each target starts its move from its hidden weight in θ. In its round, a target's hidden weight is moved to the hidden weight w_j of the neuron $j \notin H$ whose output weight it takes over. That neuron j is never a target, so w_j is still its hidden weight in θ. Each coordinate moves monotonically between these two values, so every intermediate vector lies coordinatewise between two hidden weight vectors of θ. □

Combining the disjoint-support case with this reduction gives the connectivity theorem for the over-parameterized regime. The idea is to apply the reduction once to each of the two given networks, bringing their active sets into two disjoint halves, and then to connect the results by Theorem D.15.

Theorem D.17 (Discrete mode connectivity, over-parameterized; restatement of Theorem 1.11)**.** *Let $m \geq 2n$ and let θ_A, θ_B be two-layer networks with $L(\theta_A), L(\theta_B) \leq \eta$ and output weights of ℓ_∞-norm at most K. Then they are joined by a single-coordinate-move lattice path along which the loss never exceeds*

$$\eta + \frac{\lambda G(m+n)}{2s} = \eta + \kappa \frac{\sqrt{m}}{2s} + \frac{\lambda n G}{2s}.$$

On the refined grid of resolution $s' = ks$ with $k \in \mathbb{Z}_{\geq 1}$, the same holds with s' in place of s, so the bottleneck is at most $\eta + \epsilon$, for every $\epsilon > 0$, once $s' \geq \frac{\lambda G(m+n)}{2\epsilon}$.

Proof. Let $H_1 = \{1, \ldots, \lfloor m/2 \rfloor\}$ and $H_2 = \{\lfloor m/2 \rfloor + 1, \ldots, m\}$. Since $m \geq 2n$ we have $\frac{m}{2} \geq n$, and since n is an integer, $|H_1| = \lfloor m/2 \rfloor \geq n$ and $|H_2| = \lceil m/2 \rceil \geq \lfloor m/2 \rfloor \geq n$. We apply the reduction lemma to each network separately, to move their active sets into these two disjoint halves. Apply Lemma D.16 to θ_A with $H = H_1$. This gives a path to a network θ'_A with active set in H_1, along which the loss is at most $\eta + \frac{\lambda G(m+n)}{2s}$, and whose endpoint has loss $L(\theta'_A) \leq \eta + \frac{\lambda n G}{2s}$. Likewise, applying it to θ_B with $H = H_2$ gives θ'_B. The active sets of θ'_A and θ'_B lie in the disjoint sets H_1 and H_2, their hidden weights lie in the box by Lemma D.16, and both have loss at most $\eta + \frac{\lambda n G}{2s}$. So Theorem D.15, applied with threshold $\eta + \frac{\lambda n G}{2s}$, joins them by a path of loss at most $\eta + \frac{\lambda n G}{2s} + \frac{\lambda m G}{2s} = \eta + \frac{\lambda G(n+m)}{2s}$.

It remains to bound the hidden and output weights along the concatenated path, so that the constants G and U fixed at the start of the section apply to it. The concatenation of the first reduction path, the connecting path, and the reversed second reduction path is the required path. Every hidden weight coordinate along it has absolute value at most W. Along the two reduction paths this holds because, by Lemma D.16, each hidden weight vector there is a hidden weight vector of θ_A or of θ_B, or lies coordinatewise between two hidden weight vectors of the same network. All of these lie in the box. Along the connecting path it holds by the coordinatewise clause of Theorem D.15, since each coordinate lies between its values at θ'_A and θ'_B, both of absolute value at most W. So the constant G applies throughout. All output weights along it have ℓ_∞-norm at most mK: along the reduction paths by Lemma D.16, and along the connecting path by the coordinatewise clause of Theorem D.15, since its endpoints satisfy that bound. So every prediction satisfies $|f_\theta(x_i)| \leq \|a\|_\infty \|G_i\|_1 \leq m^2 KG$, which is what the squared-loss choice $U = m^2 KG$ accounts for.

For the refinement clause, the model at resolution $s' = ks$ satisfies every hypothesis used above. Indeed $\frac{1}{s}\mathbb{Z} \subseteq \frac{1}{s'}\mathbb{Z}$, so θ_A, θ_B and the box bound W lie on the finer lattice as well. And the constants G, K and λ, together with the endpoint losses $L(\theta_A)$ and $L(\theta_B)$, do not depend on the resolution. So Lemma D.16 and Theorem D.15, applied on that grid, give three paths of the same form, with s' in place of s in every bound. Indeed s enters the bounds only through the routing slack $\frac{1}{2s}$ of Step 3 of the proof of Lemma D.16 and of Theorem D.15, the rounding slack $\frac{1}{2s}$ of Step 2 of that proof, and the relocation step $\frac{1}{s}$ of its Step 4. So the loss along the concatenation is at most $\eta + \frac{\lambda G(m+n)}{2s'}$, which is at most $\eta + \epsilon$, for every $\epsilon > 0$, once $s' \geq \frac{\lambda G(m+n)}{2\epsilon}$. □

We place this theorem against the continuous results it discretizes.

Remark D.18 (Scope of the resolution). Theorem D.17 closes the reduction left open by the disjoint-support case: in the over-parameterized regime $m \geq 2n$, every low-loss two-layer network is connected to every other through low loss, up to the grid slack $\frac{\lambda G(m+n)}{2s}$. The proof adapts the reserve-of-neurons mechanism of Nguyen (2019, Theorem 5.1) to lattice paths; it applies to ReLU even

though that source theorem requires a strictly monotone activation. The disjoint-support step of Theorem D.15 is the discrete form of Lemma 2 of Kuditipudi et al. (2019).

The mechanism throughout is convexity, reached through spare capacity, and it is worth saying plainly what that means and how far it reaches. The prediction vector lives in $\mathbb{R}^n$. So once a network carries at least n neurons in reserve, its predictions on the sample can be re-expressed and relocated, up to the rounding error, without moving an active neuron's hidden weights. The genuine function changes are two: the rounding interpolation in Step 3 of the proof of Lemma D.16, and the interpolation of Theorem D.15. Both route the output vector near a segment on which the loss is convex. The relocation of Step 4 changes the function only transiently, by one grid step of one neuron. Beyond the grid, the argument uses only that the output layer is linear and that the activations are bounded on the box. So it holds for every activation function, with G replaced by a bound on the absolute values that the activation function takes at $\langle w, x_i \rangle$ over the box $\|w\|_\infty \le W$. Such a bound exists whenever the activation function is bounded on bounded intervals, for instance whenever it is continuous.

What remains open is threefold. The first question is deeper networks, where the output-layer convexity no longer covers all the parameters. The second is the exact width threshold, since $m \ge 2n$ is sufficient but need not be necessary. The third is whether the grid slack ϵ can be removed to give exact rather than approximate connectivity.

Over-parameterization cannot be dropped entirely, as a network of width one already shows.

Remark D.19 (Some width is needed). The collapse of Theorem D.17 fails without over-parameterization, already at width one. Take $d = 1$ and the sample $x_1 = 1$ and $x_2 = -1$, with $y_1 = y_2 = 1$, so that $R = 1$. Take one neuron, confined to the box $W = 1$. Take the absolute loss $\ell(f, y) = |f - y|$, for which $\lambda = 1$.

We now compute the bottleneck loss along any path between two particular networks, and show it cannot be brought below 1. The networks $\theta_A = (a, w) = (1, 1)$ and $\theta_B = (1, -1)$ predict $(1, 0)$ and $(0, 1)$ on the sample. So both have loss $\eta = \frac{1}{2}$. Consider a single-coordinate-move lattice path from θ_A to θ_B. It moves the hidden weight w from 1 to -1, in steps of $\frac{1}{s}$, and therefore passes through $w = 0$. There the prediction vector vanishes whatever a is, and the loss is 1. Along the path that keeps $a = 1$, the loss never exceeds 1. Indeed, for $w \in [-1, 1]$ the predictions are $(\max(w, 0), \max(-w, 0))$, one of which equals $|w|$ and the other 0, so the loss is $\frac{1}{2}\left((1 - |w|) + 1\right) = 1 - \frac{|w|}{2} \le 1$. So the bottleneck is exactly $1 = 2\eta$ at every integer resolution $s \ge 1$, for which both endpoints and $W = 1$ lie on the lattice.

This already falsifies the bound of Theorem D.17 once its hypothesis $m \ge 2n$ is dropped. With these constants, $G = W\sqrt{d}R = 1$, $\lambda = 1$, $n = 2$ and $m = 1$. The bound of Theorem D.17 would then read $\frac{1}{2} + \frac{3}{2s}$, which is less than 1 once $s \ge 4$. So without the hypothesis $m \ge 2n$, its conclusion fails at every integer resolution $s \ge 4$.

Two neurons already suffice for approximate connectivity with vanishing grid slack for this particular pair. Give θ_A a second neuron of output weight zero, and set that neuron's hidden weight to -1. This changes no prediction and hence no loss, and gives the network $((1, 0), (1, -1))$, with active set $\{1\}$. The two-neuron network $((0, 1), (1, -1))$ has active set $\{2\}$ and predicts $(0, 1)$, as θ_B does. On the segment between the output vectors $(1, 0)$ and $(0, 1)$, the point $(1 - r, r)$ predicts $(1 - r, r)$ and has loss $\frac{1}{2}(r + (1 - r)) = \frac{1}{2}$. Routing the output vector along this segment by Lemma D.2, as in Phase 2 of the proof of Theorem D.15, gives a path of loss at most $\frac{1}{2} + \frac{\lambda m G}{2s} = \frac{1}{2} + \frac{1}{s}$, by the ℓ_∞ bound with $\lambda = 1$, $m = 2$ and $G = 1$. Where between 1 and $2n$ the threshold lies is the open question of Remark D.18.