A Textbook of Formal Learning Theory

Six Paradigms, Their Characterization Theorems,
and the Separations Between Them

Across Six Decades of Computational and Statistical Learning Theory

Dhruv Gupta
Zetesis Labs, ARTPARK @ IISc Bangalore
dhruv@zetesis.org

March 2026

"The question is not which functions can be learned,
but which functions can be learned given what you already know."

Preface

This book presents formal learning theory as a single, navigable mathematical structure. It treats six paradigms—PAC, online, Gold-style, exact, universal, and multiclass learning—within a unified framework of typed relations, and devotes equal attention to the separations between them.

The need for such a book arises from a structural gap. The field's foundational results are distributed across several research traditions that developed largely in parallel:

- **PAC learning** (Valiant, 1984) and its combinatorial characterization through VC dimension (Vapnik–Chervonenkis, 1971; Blumer et al., 1989).
- **Online learning** (Littlestone, 1988) and its characterization through the Littlestone dimension.
- **Gold-style identification in the limit** (Gold, 1967) and its mind-change hierarchy (Freivalds–Smith, Case–Smith).
- **Universal learning** (Bousquet et al., 2021) and the trichotomy theorem.
- **Multiclass learning** and its resolution through the DS dimension (Brukhim et al., 2022).
- **Generalization bounds** from five distinct traditions: VC/Rademacher, PAC-Bayes, algorithmic stability, information-theoretic, and margin-based.

No single textbook covers all these traditions with their precise interconnections and, critically, their precise *non*-connections. Shalev-Shwartz and Ben-David [SSBD14] treat PAC and online learning comprehensively but omit Gold-style identification, universal learning, and information-theoretic bounds. Jain et al. [JORS99] cover inductive inference thoroughly but not PAC theory. The generalization-bound literature is scattered.

This book fills the gap by treating all traditions within a single framework organized around typed relations between concepts. The key structural innovation is not any new theorem, but a *negative layer*: the book devotes equal attention to what *does not* hold—separation results with explicit witnesses, cross-paradigm analogies with their precise obstructions, and the boundaries where one framework's characterization theorem fails to generalize to another.

How to Read This Book

The book is organized in five parts:

Part I: Foundations (Chapters 1–3) introduces the base types—domains, concepts, hypothesis spaces, data presentations, and the automata-theoretic vocabulary underlying identification in the limit.

Part II: Paradigms (Chapters 4–9) presents the six major learning paradigms, each with its characterization theorem proved in full: PAC learning (VC dimension), online learning (Littlestone dimension), Gold-style identification (mind-change ordinals), exact learning (query models), and universal learning (the trichotomy).

Part III: Complexity Measures (Chapters 10–13) develops the 33 complexity measures in the graph, organized by what they measure: combinatorial dimensions, sample complexity and compression, generalization bounds, and mind-change ordinals.

Part IV: The Negative Layer (Chapters 14–15) presents all separation results with full proofs and all analogy-obstruction pairs with analysis.

Part V: Extensions (Chapters 16–18) covers computational hardness, extensions beyond binary classification, and application-layer concepts.

Each chapter includes TikZ diagrams showing the subgraph relevant to that chapter's concepts and computational illustrations where appropriate. Cross-references to the companion knowledge graph are given in the appendices.

Prerequisites

The reader should be comfortable with:

- Basic probability (expectation, concentration inequalities, Markov/Chebyshev/Hoeffding bounds).
- Basic combinatorics (growth functions, pigeonhole principle).
- Computability theory at the level of decidability, Turing machines, and the halting problem (for Gold-style chapters).
- Mathematical maturity sufficient for ε-δ arguments.

Background in machine learning is helpful but not required; the book is self-contained from the definitions upward.

Companion Materials

The book is accompanied by a machine-readable knowledge graph and a benchmark suite, available at https://github.com/Zetetic-Dhruv/Formal-Learning-Theory. The graph encodes 142 concepts connected by 260 typed edges across 13 relation types; it is described in detail in Appendix A (edge inventory) and Appendix C (validation). The repository also contains a complete bibliography and scripts for validating the graph against the textbook's content.

Notation

A notation index appears in Appendix D. We highlight the most frequently used symbols here:

$\mathcal{C}$	concept class	$\mathrm{VCdim}(\mathcal{H})$	VC dimension
$\mathcal{H}$	hypothesis space	$\mathrm{Ldim}(\mathcal{H})$	Littlestone dimension
D	distribution on $X \times Y$	$\mathrm{DSdim}(\mathcal{H})$	DS dimension
$S = \{(x_i, y_i)\}_{i=1}^m$	training sample	$d_{\mathrm{N}}(\mathcal{H})$	Natarajan dimension
$R(h) = \mathbb{E}_{(x,y)\sim D}[\ell(h(x), y)]$	true risk	$\mathrm{Pdim}(\mathcal{F})$	pseudodimension
$\hat{R}(h) = \frac{1}{m}\sum \ell(h(x_i), y_i)$	empirical risk	$\mathrm{fat}_\gamma(\mathcal{F})$	fat-shattering dim
ε	accuracy parameter	$\widehat{\mathcal{R}}_n(\mathcal{H})$	Rademacher complexity
δ	confidence parameter	$\Pi_{\mathcal{H}}(n)$	growth function

Dhruv Gupta, Bangalore, March 2026

Contents

List of Figures

List of Tables

Chapter 1

The Objects of Learning

Every learning problem begins with the same question: given data drawn from an unknown source, find a rule that predicts well on future data. This chapter introduces the mathematical vocabulary for making that question precise.

The vocabulary divides into three groups of unequal importance. The first group—domain, label, concept, hypothesis—is atomic: these are the sets and functions from which everything else is built. They require only brief definitions. The second group—concept class, hypothesis space, inductive bias—carries genuine mathematical content, because the relationships between these objects determine what is learnable. The third group—Bayesian inference, minimum description length, minimum message length—appears to be vocabulary but is actually a network of near-miss analogies, each encoding a different answer to the same question about model selection.

1.1 Atomic Vocabulary

The following objects are the type-level substrate of learning theory. They are presented together because none of them, individually, contains any mathematical surprise.

Definition 1.1 (Domain, Label, Concept)**.** A *domain* X is a set whose elements are called *instances*. A *label set* Y is a set of possible outputs; in binary classification, $Y = \{0, 1\}$. A *concept* is a function $c : X \to Y$.

Equivalently, when $Y = \{0, 1\}$, a concept c can be identified with the subset $\{x \in X : c(x) = 1\} \subseteq X$. Both perspectives—function and set—are used throughout the literature. We adopt the function view as primary and the set view when it simplifies combinatorial arguments (as in shattering, Chapter 10).

Definition 1.2 (Hypothesis, Target Concept, Proper/Improper Flag)**.** A *hypothesis* $h : X \to Y$ is a candidate prediction rule that a learning algorithm might output. The *target concept* $c^* \in \mathcal{C}$ is the specific concept that generated the training data. Learning is *proper* if the algorithm's output is constrained to lie in $\mathcal{C}$, and *improper* if it may use a larger hypothesis space $\mathcal{H} \supseteq \mathcal{C}$.

An *alphabet* Σ is a finite symbol set; it appears only in the formal language chapters (Chapters 3 and 7) and plays no role in statistical learning theory.

These definitions are unremarkable individually. The mathematical content begins when we ask how they compose.

1.2 The Concept Class: Where All Paradigms Converge

Definition 1.3 (Concept Class)**.** A *concept class* $\mathcal{C} \subseteq Y^X$ is a collection of concepts over a common domain. The class is the primary object of study: every major question in learning

theory is a question about concept classes.

This definition is simple. Its importance is not. In the companion knowledge graph, concept_class is the single most connected node: 21 edges point into it, including *every* complexity measure in the graph (Figure 1.1).

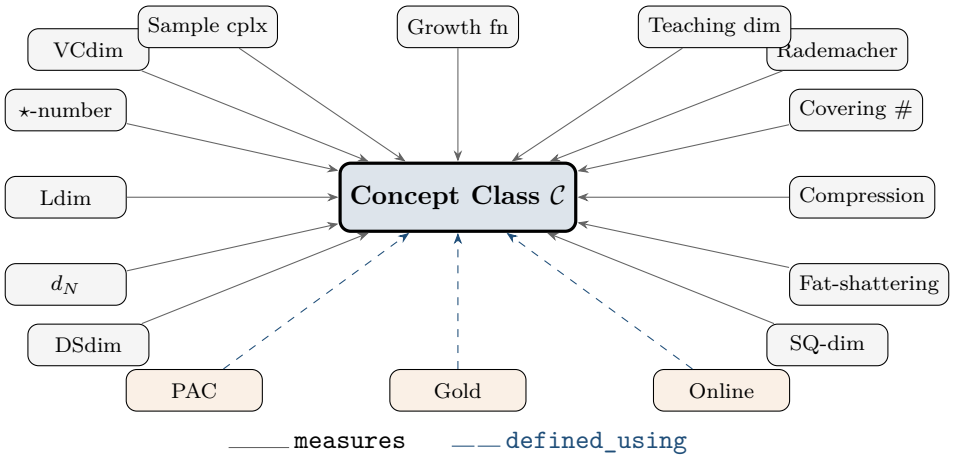

Figure 1.1: The concept class as hub node. All 13 complexity measures in the graph target $\mathcal{C}$; each paradigm defines its success criterion in terms of $\mathcal{C}$. The same mathematical object is measured by 13 different instruments, each revealing a different aspect of learnability.

Each measure asks a different question about the same object:

- **VC dimension** asks: how many points can $\mathcal{C}$ shatter? (Determines PAC learnability.)
- **Littlestone dimension** asks: how deep a mistake tree can $\mathcal{C}$ support? (Determines online learnability.)
- **DS dimension** asks: what pseudo-cubes exist in $\mathcal{C}$'s one-inclusion hypergraph? (Determines multiclass learnability.)
- **Teaching dimension** asks: how many examples suffice for a teacher to identify each $c \in \mathcal{C}$? (Measures cooperative communication.)
- **Rademacher complexity** asks: how well can $\mathcal{C}$ correlate with random noise? (Controls generalization bounds.)

These 13 measures are not independent. Many of them are related by `upper_bounds`, `lower_bounds`, and `characterizes` edges in the graph. But none of them is reducible to another. Each captures a genuine aspect of $\mathcal{C}$'s complexity that the others miss. The fact that a single mathematical object—a set of functions—supports 13 inequivalent complexity measures is one of the structural surprises of the field.

Example 1.4 (Four concept classes of increasing complexity)**.** Fix $X = \mathbb{R}$.

(a) **Thresholds:** $\mathcal{C}_1 = \{x \mapsto \mathbf{1}[x \geq \theta] : \theta \in \mathbb{R}\}$.

- VCdim $= 1$, Ldim $= \infty$, teaching dimension $= 2$.
- PAC learnable (1 sample suffices up to ε). Not online learnable (adversary forces arbitrarily many mistakes by approaching the threshold).

(b) **Intervals:** $\mathcal{C}_2 = \{x \mapsto \mathbf{1}[a \leq x \leq b] : a \leq b \in \mathbb{R}\}$.

- VCdim $= 2$, Ldim $= \infty$.

(c) **Unions of k intervals: $\mathcal{C}_3^{(k)}$.** VCdim $= 2k$.

(d) **All measurable functions: $\mathcal{C}_4 = \{0,1\}^X$.**

- VCdim $= \infty$. Not PAC learnable, not online learnable, not identifiable in the limit. The NFL theorem (Section 1.4) applies in full force.

The thresholds example deserves attention: VCdim $= 1$ but Ldim $= \infty$. This single class witnesses the separation between PAC and online learning (Chapter 14).

1.3 Hypothesis Spaces and the Proper–Improper Divide

Definition 1.5 (Hypothesis Space)**.** A *hypothesis space* $\mathcal{H} \subseteq Y^X$ is the set of functions available to the learner as candidate outputs. When $\mathcal{H} = \mathcal{C}$, learning is *proper*; when $\mathcal{H} \supsetneq \mathcal{C}$, learning is *improper*.

The proper–improper distinction is not merely notational. It has computational consequences that are among the sharpest in the theory:

1. **Improper learning can be exponentially cheaper.** There exist concept classes where proper learning requires exponentially more samples or computation than improper learning [PV88]. The graph records this as a `does_not_imply` edge from `pac_learning` to `exact_learning` (Chapter 14).

2. **Computational hardness often depends on properness.** Cryptographic hardness results (Kearns–Valiant, 1994) typically establish hardness of *proper* learning; improper learners may circumvent the barrier by using richer representations [KV94].

3. **The fundamental theorem is representation-independent.** The VC characterization of PAC learnability holds regardless of whether learning is proper or improper—what matters is the VC dimension of the *target* class $\mathcal{C}$, not of the learner's hypothesis space $\mathcal{H}$.

Definition 1.6 (Version Space)**.** Given a training sample $S = \{(x_i, y_i)\}_{i=1}^m$, the *version space* is $\mathrm{VS}_{\mathcal{H},S} = \{h \in \mathcal{H} : h(x_i) = y_i \text{ for all } i\}$, the set of hypotheses consistent with all observed data.

The version space is the simplest object one can define from $\mathcal{H}$ and S together: just the set of survivors. Version space elimination—outputting any element of $\mathrm{VS}_{\mathcal{H},S}$—is perhaps the most natural learning algorithm imaginable.

The trouble is computational. For most hypothesis classes of interest, the version space cannot be efficiently represented, enumerated, or even sampled from. The *star number* of $\mathcal{H}$ (Chapter 10) measures one aspect of this difficulty: it is the size of the largest star-shaped substructure in $\mathcal{H}$'s dual, and its finiteness turns out to be equivalent to finite VC dimension. The *eluder dimension* [RV13] measures another aspect relevant to exploration–exploitation tradeoffs.

1.4 Inductive Bias and the No-Free-Lunch Barrier

Definition 1.7 (Inductive Bias)**.** The *inductive bias* of a learner is the set of assumptions—explicit or implicit—that it uses to generalize from training data to unseen instances.

This definition is deliberately informal, because inductive bias is not a single mathematical object. It is a meta-concept whose formalization changes across paradigms:

Paradigm	Inductive bias formalized as	Graph edge
PAC	Restriction of $\mathcal{H}$ to finite VC dimension	`used_in_proof` by nfl_theorem
Bayesian	Prior $P(h)$ over hypotheses	`analogy` from bayesian_inference
SRM	Nested class hierarchy $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots$	`analogy` from srm
Computational	Cryptographic assumption restricting efficient search	`requires_assumption` by computational_hardness

The reason inductive bias is not merely a design choice but a *mathematical necessity* is the No-Free-Lunch theorem.

Theorem 1.8 (No Free Lunch [Wol96])**.** *Let A be any learning algorithm. For any sample size $m \leq |X|/2$:*

$$\max_{c \in \{0,1\}^X} \mathbb{E}_{S \sim D^m}[R_D(A(S))] \geq \frac{1}{4}.$$

That is, for every learner, there exists a target function on which the learner's expected error is at least $1/4$—no better than random guessing on two labels.

Proof. Fix any algorithm A. Consider the uniform distribution on X. For any training sample S of size m, let $T = X \setminus \{x_1, \ldots, x_m\}$ be the unseen instances ($|T| \geq |X|/2$).

The labels on T are unconstrained by S (since we average over all target functions). For each $x \in T$, the target label is equally likely to be 0 or 1 under the uniform distribution over target functions. Therefore the expected error of $A(S)$ on each unseen point is exactly $1/2$. Since $|T|/|X| \geq 1/2$, the overall error is at least $1/4$. □

The NFL theorem has a precise structural consequence. It says that the class $\{0,1\}^X$ of *all* binary functions is not learnable. Therefore any learnable class must be a *proper subset* of $\{0,1\}^X$. The choice of which proper subset—the choice of $\mathcal{C}$—is exactly the inductive bias. This is why inductive bias is mandatory, not optional.

Remark 1.9 (NFL as a Pl-boundary). In the language of the graph, the NFL theorem establishes that inductive_bias is a `requires_assumption` dependency for computational_hardness: without inductive bias, the question "which classes are efficiently learnable?" is not even askable, because no class is learnable at all. The NFL theorem is a plausibility boundary—it tells you what is not even admissible to ask about learning without first restricting $\mathcal{C}$.

1.5 Three Faces of Model Selection

Bayesian inference, minimum description length, and minimum message length are often described informally as "the same idea." They are not. Each formalizes a different answer to the question "which hypothesis should I prefer?"—and the differences between them are more instructive than any individual definition.

1.5.1 The Definitions

Definition 1.10 (Bayesian Inference)**.** Given a prior $P(h)$ over $\mathcal{H}$ and a likelihood $P(S \mid h)$, the posterior is $P(h \mid S) \propto P(S \mid h) \cdot P(h)$. The Bayesian learner outputs $\hat{h} = \arg\max_h P(h \mid S)$ (MAP) or the full posterior $P(\cdot \mid S)$.

Definition 1.11 (Minimum Description Length [Ris78])**.** Select h minimizing $L(h) + L(S \mid h)$, where $L(\cdot)$ denotes description length under a fixed prefix-free code. The first term penalizes complexity; the second penalizes misfit.

Definition 1.12 (Minimum Message Length [WB68])**.** Select h minimizing the total message length $\ell(h) + \ell(S \mid h)$ required to transmit the hypothesis and then the data, under a two-part coding scheme. Wallace–Freeman (1987): $\ell(h) = -\log P(h) + \frac{1}{2}\log|F(h)| + \text{const}$, where $F(h)$ is the Fisher information.

1.5.2 The Analogy Network

These three methods sit in a triangular analogy network in the graph, connected by `analogy` edges with explicit obstructions (Figure 1.2).

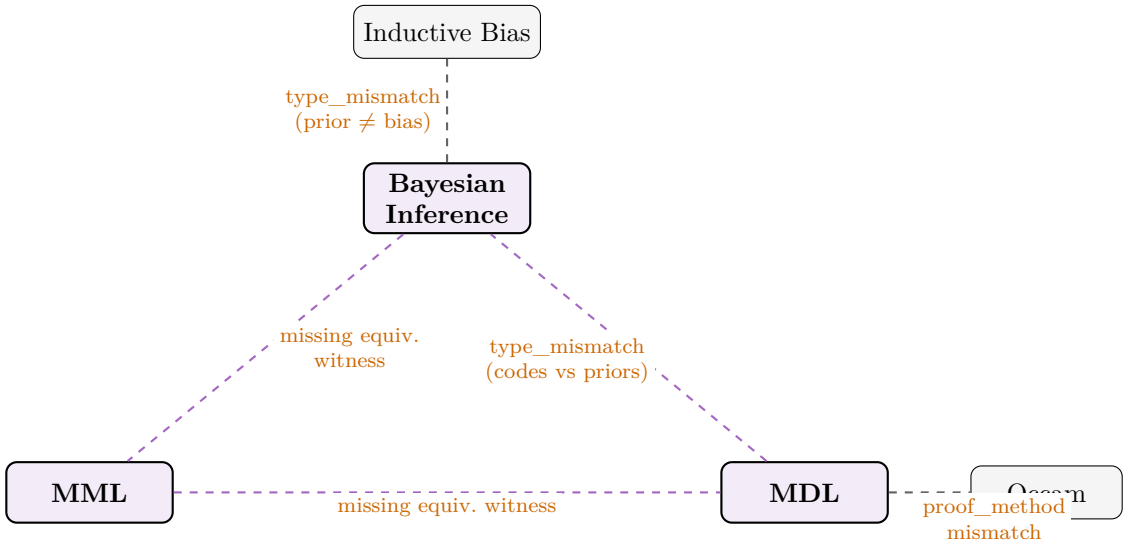

Figure 1.2: The model selection analogy triangle. Each pair is connected by an `analogy` edge with an explicit obstruction explaining why the analogy is not a theorem. The triangle encodes a genuine mathematical fact: these three methods look similar but are not equivalent, and the reasons for non-equivalence are different for each pair.

1.5.3 Why They Are Not the Same

Obstruction

Bayesian ↔ MML: obstruction type *missing_equivalence_witness*. MML's two-part code corresponds to Bayesian MAP estimation when the Fisher information correction term is constant. But for models with varying curvature in parameter space, MML departs from Bayesian posteriors. No theorem establishes their equivalence beyond the constant-Fisher case.

Obstruction

MML ↔ MDL: obstruction type *missing_equivalence_witness*. Both minimize a two-part code length. But MML uses Shannon-optimal coding relative to a prior (subjective), while MDL uses a universal code based on Kolmogorov complexity (objective). In the limit of infinite data, both converge to the true model under standard regularity conditions, but their finite-sample behavior differs.

Obstruction

MDL ↔ Bayesian: obstruction type *type_mismatch*. MDL uses description lengths (integers under a code); Bayesian inference uses probability densities (reals under a measure). The connection $L(h) = -\log P(h)$ identifies description lengths with log-probabilities, but this identification breaks down when the code is not prefix-free or the prior is improper.

The fact that three different formalization of "prefer the simpler hypothesis" are *not* equivalent—and that the reasons for non-equivalence are themselves different (missing witness, missing witness, and type mismatch)—is a structural feature of learning theory that no single definition can convey. It is visible only in the graph.

1.5.4 Structural Risk Minimization

Definition 1.13 (Structural Risk Minimization [Vap98])**.** Given a nested sequence of hypothesis classes $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots$ with $\mathrm{VCdim}(\mathcal{H}_k) = d_k$, select k^* minimizing $\hat{R}_{\mathcal{H}_k}(h) + \mathrm{penalty}(d_k, m, \delta)$.

SRM is the operational realization of inductive bias within VC theory: instead of choosing a single $\mathcal{H}$, choose a hierarchy indexed by complexity, and let the data select the level. The graph records this as an `analogy` edge from `srm` to `inductive_bias` with a *type_mismatch* obstruction: SRM constructs inductive bias via structural nesting, but the concept of "inductive bias" itself is broader than any single construction.

Definition 1.14 (Algorithmic Probability [Sol64])**.** The *Solomonoff prior* assigns to each string x the probability $M(x) = \sum_{p:U(p)=x} 2^{-|p|}$, summing over all programs p that produce x on a universal Turing machine U. This is the "prior that assigns higher probability to simpler explanations" in the most literal possible sense.

Algorithmic probability connects directly to Kolmogorov complexity ($K(x) \approx -\log M(x)$) and provides the information-theoretic foundation for MDL. Its detailed treatment belongs to Chapter 11, where it interacts with compression schemes and Occam's razor.

1.6 What This Chapter Established

The base vocabulary of learning theory is small: domain, label, concept, concept class, hypothesis space. But even at this level, three structural features are already visible:

1. The concept class is a hub node: the same object is measured by 13 inequivalent complexity measures. This multiplicity is not redundancy—it reflects the fact that learnability has 13 genuinely different aspects.
2. Inductive bias is not optional. The NFL theorem converts it from a design choice to a mathematical necessity.
3. The three standard model selection principles (Bayesian, MDL, MML) form a triangle of near-miss analogies. They look equivalent; they are not; and the reasons for non-equivalence are themselves informative.

These patterns—hub nodes, negative results as content, analogy obstructions—recur throughout the book.

Chapter 2

Data Presentations and Oracles

Chapter 1 established *what* is learned: concept classes, hypotheses, the objects on which complexity measures act. This chapter asks a prior question: *how does data reach the learner?*

The answer turns out to be load-bearing. Four fundamentally different data models—i.i.d. samples, enumerative texts, query oracles, and adversarial streams—give rise to four different theories of learnability, characterized by four different complexity measures, and separated by explicit impossibility results. The data model is not a parameter of a single theory; it is the axis along which the field fractures into distinct paradigms.

We begin with vocabulary that is shared across all paradigms (§2.1), then develop each data model in turn, organized not by alphabetical order but by the mathematical content each one carries.

2.1 Shared Vocabulary

The following terms appear across paradigms and require only brief definition.

A *time index* $t \in \mathbb{N}$ is the discrete step indexing rounds of interaction between learner and environment. A *counterexample* to a hypothesis h with respect to target c is a point $x \in X$ where $h(x) \neq c(x)$, witnessing disagreement. *Advice* is any additional information given to a learner beyond its primary data source; its formal role is explored in the advice reductions of Chapter 8.

Definition 2.1 (Noisy Input)**.** In the *classification noise model* at rate $\eta < 1/2$, each label y_i in the training sample is independently flipped with probability η: the learner receives $(x_i, y_i \oplus z_i)$ where $z_i \sim \text{Bernoulli}(\eta)$.

The key fact about classification noise is that it does not change the landscape of what is learnable. PAC learnability under classification noise at any fixed rate $\eta < 1/2$ is equivalent to clean PAC learnability [AL88]. The sample complexity increases by a factor of $O(1/(1-2\eta)^2)$, but the class of learnable concept classes is identical. This is a robustness result for the PAC model; it does *not* hold for all data models (adversarial noise in the online setting requires fundamentally different techniques).

With vocabulary in place, we turn to the four data models. Each gets treatment proportional to its mathematical content.

2.2 The Statistical Data Model: I.I.D. Samples

Definition 2.2 (I.I.D. Sample)**.** An *i.i.d. sample* of size m from distribution D over $X \times Y$ is a sequence $S = ((x_1, y_1), \ldots, (x_m, y_m))$ where each (x_i, y_i) is drawn independently from D. The learner knows m but has no knowledge of D beyond what S reveals.

This definition looks like pure vocabulary, but it is the single most consequential modeling choice in the field. The i.i.d. assumption is the foundation on which all of statistical learning theory—PAC learning, VC theory, generalization bounds, uniform convergence—is built. Understanding what this assumption buys, and what it costs, is the key to understanding why the field has multiple paradigms rather than one.

2.2.1 What i.i.d. buys: uniform convergence

The power of the i.i.d. assumption is that it enables *uniform convergence*: the guarantee that empirical frequencies converge to true probabilities simultaneously across all hypotheses.

Theorem 2.3 (Uniform Convergence, informal [Val84])**.** *If* $\mathrm{VCdim}(\mathcal{H}) = d < \infty$ *and* S *is an i.i.d. sample of size* $m \geq \Omega\left(\frac{d+\log(1/\delta)}{\varepsilon^2}\right)$*, then with probability at least* $1-\delta$*,*

$$\sup_{h \in \mathcal{H}} |\hat{R}(h) - R(h)| \leq \varepsilon.$$

This is the engine that makes PAC learning work: once uniform convergence holds, empirical risk minimization (ERM) succeeds. Any hypothesis that looks good on the training data *is* good on the true distribution, uniformly over all of $\mathcal{H}$. The proof (deferred to Chapter 5) uses symmetrization and the growth function bound—techniques that require i.i.d. draws in an essential way.

2.2.2 What i.i.d. costs: three fragilities

The i.i.d. assumption purchases uniform convergence at the cost of three fragilities:

1. **No adversarial robustness.** If an adversary may choose the data sequence, the i.i.d. assumption is violated and uniform convergence fails. The VC dimension ceases to characterize learnability; the Littlestone dimension takes over (§2.5).

2. **No distribution-free sequential guarantees.** The i.i.d. model is inherently batch: the learner receives all m examples at once. Extending it to sequential settings requires either assuming the distribution is fixed across time steps (the "online-with-distributional" hybrid, which preserves PAC-style results) or abandoning distributional assumptions entirely (which leads to online learning proper).

3. **No interaction.** The learner is a passive recipient. It cannot choose which points to query, negotiate with a teacher, or request counterexamples. This passivity is the precise gap that Angluin's query model fills (§2.4).

Remark 2.4 (The VC dimension's jurisdiction). The VC dimension characterizes PAC learnability *because* of the i.i.d. assumption—and *only* under that assumption. The same concept class may have $\mathrm{VCdim}(\mathcal{C}) = 1$ (PAC learnable from i.i.d. data) and $\mathrm{Ldim}(\mathcal{C}) = \infty$ (not online learnable under adversarial data). Thresholds on $\mathbb{R}$ witness this separation (Example 1.4). The complexity measure is not a property of the class alone; it is a property of the class *together with* the data model.

2.3 Enumerative Data: Text and Informant

In Gold's model of identification in the limit [Gol67], the learner receives data not as a random sample but as an enumeration. The data arrives one element at a time, forever, and the learner must eventually converge to a correct hypothesis. Two variants of this process—*text* and *informant*—differ in a way that appears minor but produces a sharp separation in learning power.

Definition 2.5 (Text Presentation)**.** A *text* for a language $L \subseteq \mathbb{N}$ is an infinite sequence $t_1, t_2, t_3, \ldots$ of elements from $L \cup \{\#\}$ (where $\#$ is a pause symbol) such that every element of L appears at least once. The learner sees $t_1, \ldots, t_n$ at time n and must output a hypothesis h_n. A class $\mathcal{C}$ is *identifiable in the limit from text* if there exists a learner that, for every $C \in \mathcal{C}$ and every text for C, eventually stabilizes on a correct hypothesis: $\exists n_0\, \forall n \geq n_0 \colon h_n = h_{n_0}$ and $L(h_{n_0}) = C$.

Definition 2.6 (Informant Presentation)**.** An *informant* for a language L is an infinite sequence of pairs $(x_1, \ell_1), (x_2, \ell_2), \ldots$ where each $x_i \in \mathbb{N}$, $\ell_i = \mathbf{1}[x_i \in L]$, and every natural number appears at least once as some x_i. That is, the learner eventually sees the membership status of every point.

The difference: a text provides *positive data only* (which elements belong to L), while an informant provides *positive and negative data* (the membership status of every element). At first glance, this distinction seems quantitative—an informant simply provides more information. The following separation shows it is qualitative.

Separation Result

Text $<$ Informant [Gol67]. There exist concept classes identifiable in the limit from informant that are *not* identifiable in the limit from text.

Witness class. Let $\mathcal{C}_{\text{fin}} = \{L \subseteq \mathbb{N} : L \text{ is finite}\} \cup \{\mathbb{N}\}$. This is the class consisting of all finite languages together with the full language $\mathbb{N}$.

From informant: learnable. Given an informant, the learner eventually sees the status of every $x \in \mathbb{N}$. If the target is a finite set L, the learner will eventually see a negative example $x \notin L$ for every $x > \max(L)$, confirming finiteness. If the target is $\mathbb{N}$, every element will eventually receive a positive label. A learner can output "the finite set seen so far" and switch to $\mathbb{N}$ only when forced. This converges.

From text: not learnable. Suppose a learner M is given a text for some unknown $L \in \mathcal{C}_{\text{fin}}$. The adversarial argument proceeds by diagonalization:

1. Start presenting a text for $\mathbb{N}$: enumerate $0, 1, 2, \ldots$ The learner M must eventually conjecture $\mathbb{N}$ (since $\mathbb{N} \in \mathcal{C}_{\text{fin}}$ and the text is valid for $\mathbb{N}$). Let n_0 be the first time M outputs $\mathbb{N}$.
2. But the text $0, 1, \ldots, n_0$ is also a valid beginning for the finite set $\{0, 1, \ldots, n_0\}$. Present this text instead (padding with $\#$ forever after). Now M must eventually abandon $\mathbb{N}$ and conjecture $\{0, \ldots, n_0\}$. Let $n_1 > n_0$ be when it does.
3. But now extend the text with $n_0 + 1$, making it consistent with $\{0, \ldots, n_0, n_0 + 1\}$. The learner must change again.

By iterating, we force M to change its mind infinitely often, contradicting convergence. No learner succeeds on all texts for all $L \in \mathcal{C}_{\text{fin}}$.

The witness class $\mathcal{C}_{\text{fin}}$ is instructive: it is an extremely natural class (all finite sets plus the universal set), and yet it already separates the two data models. The obstruction is not pathological; it is the impossibility of distinguishing "all of $\mathbb{N}$" from "a very large finite set" using only positive data.

Remark 2.7 (The structural role of negative data). The separation reveals that negative data is not merely "more information" in a quantitative sense. It provides a qualitatively different kind of evidence: *exclusion*. A text tells you what is in L; an informant also tells you what is *not* in L. The latter enables the learner to confirm finiteness—something that no amount of

positive data can do, because any finite prefix of a text for $\mathbb{N}$ is also a prefix of a text for a finite superset. This asymmetry between inclusion and exclusion evidence recurs in query learning (§2.4), where equivalence queries provide "global negative data" in the form of counterexamples.

Historical Note

Gold's 1967 paper [Gol67] introduced both data models and proved the separation in the context of language identification. The result predates PAC learning by 17 years and online learning by 21 years, making it one of the earliest impossibility results in computational learning theory. Gold's framework remains the standard model for inductive inference and is the foundation for the mind-change hierarchy developed by Freivalds–Smith and Case–Smith (Chapter 13).

2.4 Query Access: Membership and Equivalence Oracles

The i.i.d. model and the enumerative model share a fundamental limitation: the learner is passive. It receives data but cannot request it. Angluin's query model [Ang88] removes this limitation by giving the learner access to oracles.

Definition 2.8 (Membership Oracle)**.** A *membership oracle* MQ for a target concept c answers queries of the form "what is $c(x)$?" for any $x \in X$ chosen by the learner. Each query costs one unit of time.

Definition 2.9 (Equivalence Oracle)**.** An *equivalence oracle* EQ for a target concept c answers queries of the form "does $h = c$?" for any hypothesis h chosen by the learner. If $h = c$, the oracle answers YES. If $h \neq c$, the oracle returns a *counterexample*: some x where $h(x) \neq c(x)$.

Definition 2.10 (Exact Learning)**.** A concept class $\mathcal{C}$ is *exactly learnable* from membership and equivalence queries if there exists an algorithm that, for any target $c \in \mathcal{C}$, outputs a hypothesis h with $h = c$ after polynomially many queries (polynomial in the representation size of c and the size of counterexamples received).

These definitions are clean, but their mathematical punch lies in the comparison with passive models.

2.4.1 The power of queries: DFAs as witness

The class of regular languages, represented as deterministic finite automata (DFAs), provides the sharpest illustration of what query access buys.

Theorem 2.11 (Angluin's L^* Algorithm [Ang88])**.** *The class of DFAs with n states over an alphabet Σ is exactly learnable using* MQ + EQ *in time polynomial in n and $|\Sigma|$. The algorithm uses at most $n - 1$ equivalence queries and at most $O(n|\Sigma| \cdot m)$ membership queries, where m is the length of the longest counterexample received.*

The proof constructs an observation table that encodes the Myhill–Nerode equivalence classes of the target language, filling it via membership queries and correcting structural errors via equivalence queries. It is fully developed in Chapter 8.

Contrast this with the passive setting:

Proposition 2.12 (DFAs are not PAC-learnable under cryptographic assumptions [KV94])**.** *Under standard cryptographic assumptions, DFAs are not efficiently PAC-learnable from random examples alone.*

Remark 2.13 (Queries as paradigm shift). The DFA example witnesses a genuine paradigm separation: the same concept class passes from computationally intractable (under passive, distributional data) to efficiently learnable (under interactive, worst-case queries). This is not an incremental improvement; it is a qualitative change in the complexity landscape. The mechanism is clear: membership queries let the learner *explore* the target's structure actively, and equivalence queries provide *directed* negative information (counterexamples at the learner's current frontier of knowledge), rather than the undirected positive data of a random sample.

The graph records this as a separation between pac_learning and exact_learning: neither implies the other. Some classes are PAC-learnable but not efficiently exactly learnable (because equivalence queries may be computationally expensive to simulate); DFAs show the reverse direction.

2.4.2 Individual queries are weaker

Neither oracle suffices alone. Membership queries without equivalence queries cannot learn any superpolynomial class (the learner has no way to verify global correctness). Equivalence queries without membership queries can learn any finite class (by enumeration), but the query complexity may be exponential. The combination MQ + EQ is essential—each oracle compensates for the other's weakness.

2.5 Adversarial Streams and the Online Data Model

Definition 2.14 (Data Stream (Online Protocol))**.** In the *online learning protocol*, data arrives as a stream of adversarially chosen instances. At each round $t = 1, 2, \ldots$:

1. The adversary selects $x_t \in X$ (possibly depending on the learner's past predictions).
2. The learner predicts $\hat{y}_t \in Y$.
3. The true label $y_t = c(x_t)$ is revealed.

The learner's goal is to minimize the total number of mistakes $|\{t : \hat{y}_t \neq y_t\}|$. Crucially, there is *no distributional assumption*: the adversary may be adaptive.

The online protocol strips away every comfort that the i.i.d. model provides. There is no distribution, no independence, no convergence of empirical frequencies to population quantities. The learner faces a sequential game against an adversary, and the relevant performance measure is worst-case mistake count, not expected risk.

This austerity has a precise consequence for complexity measures.

Remark 2.15 (From VC to Littlestone). Under i.i.d. data, the VC dimension characterizes learnability (Theorem 2.3). Under adversarial streams, the VC dimension is *insufficient*: classes with finite VC dimension may have infinite Littlestone dimension, and the latter is what controls mistake bounds.

The Littlestone dimension $\mathrm{Ldim}(\mathcal{C})$ is the depth of the deepest complete binary mistake tree that $\mathcal{C}$ can support. Formally:

- A *mistake tree* of depth d is a complete binary tree of depth d whose internal nodes are labeled by instances $x \in X$.
- The tree is *shattered* by $\mathcal{C}$ if for every root-to-leaf path (corresponding to a sequence of predictions), there exists a concept $c \in \mathcal{C}$ that forces a mistake at every node along the path.
- $\mathrm{Ldim}(\mathcal{C})$ is the largest d for which such a tree exists.

The formal development, including the Standard Optimal Algorithm (SOA) achieving $\mathrm{Ldim}(\mathcal{C})$ mistakes, is in Chapter 6.

What does adversarial data buy in return for the loss of distributional comfort? **Robustness.** A mistake-bounded online learner provides guarantees against *any* data sequence, including worst-case, adversarial, non-stationary, and distribution-shifting scenarios. The bound

$\mathrm{Ldim}(\mathcal{C})$ holds regardless of how the data is generated. This unconditional guarantee is what makes online learning the appropriate framework for settings where distributional assumptions are unjustified.

2.6 Taxonomy: Four Data Models, Four Theories

The preceding sections developed each data model individually. We now display their relationships as a single structure.

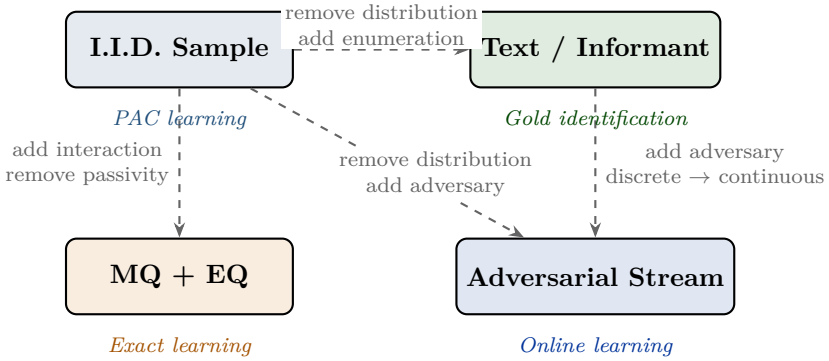

Figure 2.1: The four data models and the paradigms they generate. Each arrow represents a modeling change; the text indicates which assumption is added or removed. No single paradigm subsumes the others.

Table 2.1 displays the precise correspondence between data model, complexity measure, characterization theorem, and impossibility result.

Table 2.1: How the choice of data model determines the theory. Each row is a paradigm; each column is a structural feature. The table makes visible a pattern that is invisible when the paradigms are studied in isolation: the complexity measure, the characterization theorem, and the impossibility results all *co-vary* with the data model.

	Data Model	**Complexity Measure**	**Characterization**	**Key Impossibility**
PAC	I.I.D. sample from unknown D	$\mathrm{VCdim}(\mathcal{C})$	$\mathcal{C}$ is PAC-learnable iff $\mathrm{VCdim}(\mathcal{C}) < \infty$ [Val84]	NFL: $\{0,1\}^X$ not learnable
Gold	Text (positive) or informant (positive + negative)	Finite thickness, mind-change ordinals	Angluin's condition; telltale sets [Gol67]	$\mathcal{C}_{\mathrm{fin}} \cup \{\mathbb{N}\}$: text $<$ informant
Exact	MQ + EQ (interactive)	Query complexity (polynomial in representation)	L^* for DFAs; class-specific [Ang88]	MQ alone or EQ alone insufficient
Online	Adversarial stream (no distribution)	$\mathrm{Ldim}(\mathcal{C})$	$\mathcal{C}$ is online-learnable iff $\mathrm{Ldim}(\mathcal{C}) < \infty$ [Lit88]	$\mathrm{VCdim} < \infty \not\Rightarrow \mathrm{Ldim} < \infty$

2.7 Cross-Paradigm Separations

The most important content of this chapter is not any single definition but the *separations* between data models. We collect the three principal separations here; their full proofs appear in Chapter 14.

Separation	Witness	Mechanism	Reference
Text $<$ Informant	$\mathcal{C}_{\text{fin}} \cup \{\mathbb{N}\}$	Diagonalization	[Gol67]
PAC $\not\Rightarrow$ Online	Thresholds on $\mathbb{R}$	VCdim $= 1$, Ldim $= \infty$	[Lit88]
Passive $\not\Rightarrow$ Exact (and vice versa)	DFAs	Crypto. hardness of PAC; L^* in poly queries	[Ang88], [KV94]

Each separation has the same logical structure: a *witness class* that is learnable under one data model but not another, together with an explicit obstruction explaining *why*. The witnesses are not pathological—thresholds, finite sets, and finite automata are among the simplest concept classes in the theory. This is the structural signature of a genuine paradigm boundary, as opposed to a mere parameter difference: even the simplest classes distinguish the paradigms.

2.8 What This Chapter Established

The chapter's argument can be stated in one sentence: the data model determines the theory. More precisely:

1. **Four data models yield four paradigms.** I.i.d. samples $\rightarrow$ PAC learning. Enumerative texts $\rightarrow$ Gold identification. Query oracles $\rightarrow$ exact learning. Adversarial streams $\rightarrow$ online learning. These are not parameter choices within a single framework; they are different frameworks, with different characterization theorems, different complexity measures, and different impossibility results.

2. **The separations are witnessed by simple classes.** Thresholds separate PAC from online. $\mathcal{C}_{\text{fin}} \cup \{\mathbb{N}\}$ separates text from informant. DFAs separate passive from interactive. The simplicity of these witnesses indicates that the paradigm boundaries are fundamental, not artifacts of pathological constructions.

3. **The complexity measure is a joint property of class and data model.** The VC dimension measures $\mathcal{C}$ relative to i.i.d. data. The Littlestone dimension measures $\mathcal{C}$ relative to adversarial data. Neither is "the" complexity of $\mathcal{C}$; each is the complexity of $\mathcal{C}$-under-a-data-model. This observation—that the *same* object has different complexities under different presentations—is one of the central structural insights of the field.

Chapters 5–9 develop each paradigm in full. The present chapter has identified the axis along which they separate.

Chapter 3

Automata, Languages, and Computability

This chapter assembles the automata-theoretic and computability-theoretic vocabulary required by Gold's model of identification in the limit (Chapter 7). Most of the material is standard and is presented without proof; the reader who needs full development should consult Hopcroft and Ullman [HU79] or Sipser [Sip13].

Two things in this chapter are *not* standard vocabulary. First, the distinction between *restricts* and *extends_grammar* edges—two ways a computational model can generalize another, one conservative and one genuinely new—is illustrated here for the first time through the NFA/DFA and PDA/DFA pairs (§3.2). This distinction pervades the entire knowledge graph. Second, the notion of limiting recursion connects computability theory to learning theory by placing Gold's identification in the limit at a precise level of the arithmetical hierarchy (§3.4).

3.1 The Chomsky Hierarchy

The Chomsky hierarchy classifies formal languages by the computational resources needed to recognize them. Figure 3.1 displays the containment structure; the definitions follow.

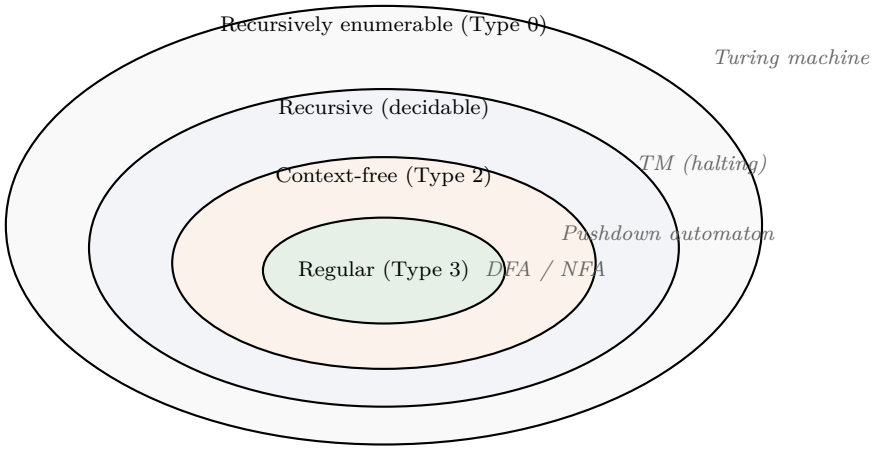

Figure 3.1: The Chomsky hierarchy. Each level is strictly contained in the next. The recognizer type is indicated at right. All containments are proper.

3.1.1 Recognizers

Definition 3.1 (Deterministic Finite Automaton)**.** A *DFA* is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite state set, Σ is a finite input alphabet, $\delta : Q \times \Sigma \to Q$ is a total transition function, $q_0 \in Q$ is the start state, and $F \subseteq Q$ is the set of accepting states. M accepts a string $w \in \Sigma^*$ if the unique run on w ends in a state in F.

Definition 3.2 (Nondeterministic Finite Automaton)**.** An *NFA* is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ identical to a DFA except that $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ maps each state–symbol pair to a *set* of successor states, and ε-transitions are permitted. M accepts w if *some* run on w ends in F.

Definition 3.3 (Pushdown Automaton)**.** A *pushdown automaton* (PDA) is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ where Γ is a finite stack alphabet, $Z_0 \in \Gamma$ is the initial stack symbol, and $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma^*)$ maps each state–input–stack-top triple to a set of (state, stack-replacement) pairs. The PDA has access to an unbounded LIFO stack; it accepts by final state or empty stack.

Definition 3.4 (Turing Machine)**.** A *Turing machine* is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$ where $\Gamma \supseteq \Sigma$ is the tape alphabet (including a blank symbol), and $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is a partial transition function specifying the next state, symbol to write, and head movement. M accepts w if it halts in q_{acc}.

Definition 3.5 (Context-Free Grammar)**.** A *context-free grammar* (CFG) is a 4-tuple $G = (V, \Sigma, R, S)$ where V is a finite set of variables, Σ is the terminal alphabet, $R \subseteq V \times (V \cup \Sigma)^*$ is a finite set of production rules, and $S \in V$ is the start variable. G generates the language $L(G) = \{w \in \Sigma^* : S \Rightarrow^* w\}$.

3.1.2 Language classes

A language $L \subseteq \Sigma^*$ is *regular* if it is accepted by some DFA. It is *context-free* if it is generated by some CFG (equivalently, accepted by some PDA). It is *recursively enumerable* (r.e.) if it is accepted by some Turing machine. It is *recursive* (decidable) if it is accepted by a Turing machine that halts on every input.

A function $f : \mathbb{N} \rightharpoonup \mathbb{N}$ is *partial computable* if there exists a Turing machine M such that M on input n halts with output $f(n)$ when $f(n)$ is defined and diverges otherwise.

3.1.3 Closure properties

The language classes differ sharply in their closure properties. Table 3.1 summarizes the facts needed in later chapters.

Table 3.1: Closure properties of the Chomsky hierarchy classes. ✓ = closed, × = not closed. All proofs are standard; see [HU79].

Class	Union	Intersection	Complement	Concat.	Kleene $*$
Regular	✓	✓	✓	✓	✓
Context-free	✓	×	×	✓	✓
Recursive	✓	✓	✓	✓	✓
R.E.	✓	✓	×	✓	✓

3.2 Two Kinds of Generalization: Restricts vs. Extends

The Chomsky hierarchy is a containment chain. But the *edges* in this chain are not all of the same type. Moving from one level to another sometimes removes a constraint (making the model

less restrictive without adding new primitives) and sometimes adds a genuinely new resource (granting the model access to a mechanism it did not previously have). This distinction—between *conservative* and *generative* generalization—is one of the recurring structural motifs of the knowledge graph, and the automata hierarchy provides its first concrete instance.

Definition 3.6 (Restricts)**.** Model B `restricts` model A if B is obtained from A by *removing a constraint*: every instance of A is already an instance of B (after trivial embedding), and the two models accept exactly the same class of languages. The generalization is conservative—it adds flexibility in representation without expanding expressive power.

Definition 3.7 (Extends Grammar)**.** Model B `extends_grammar` model A if B is obtained from A by *adding a new computational primitive*: B recognizes a strictly larger class of languages than A. The generalization is generative—it requires a fundamentally new resource.

The DFA is the anchor point for both edges. Figure 3.2 displays the structure.

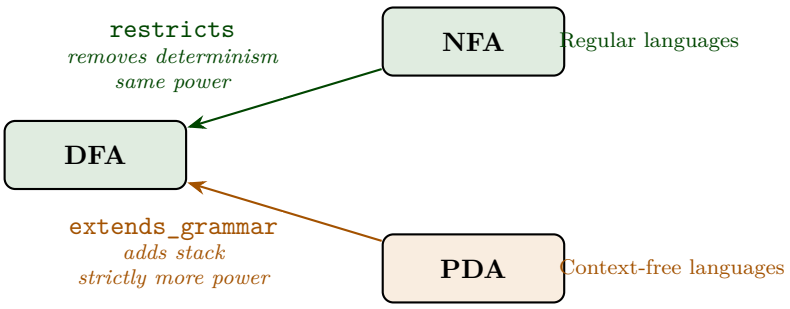

Figure 3.2: Two generalizations of the DFA. The NFA removes the determinism constraint (conservative: same expressive power). The PDA adds a stack (generative: strictly more expressive power). Both arrows point toward the DFA to indicate that the DFA is the more constrained model.

The NFA–DFA edge: conservative generalization. An NFA differs from a DFA by allowing multiple successor states per transition and ε-moves. This *removes* the determinism constraint. Yet the subset construction shows that every NFA with n states has an equivalent DFA with at most 2^n states [HU79]. The class of languages recognized is unchanged: $\mathrm{REG}_{\mathrm{NFA}} = \mathrm{REG}_{\mathrm{DFA}}$. The NFA is a more flexible *representation* of the same expressive power.

This is a `restricts` edge. The NFA does not need any new grammar or computational primitive—it simply relaxes a structural requirement. The cost is representational blowup (exponential in the worst case), not a change in what can be expressed.

The PDA–DFA edge: generative generalization. A pushdown automaton extends a finite automaton by adding an unbounded stack. This is not the removal of a constraint but the *introduction of a new primitive*: the ability to store and retrieve unbounded auxiliary information in LIFO order. The consequence is a strict increase in expressive power. The language $\{a^n b^n : n \geq 0\}$ is context-free (accepted by a PDA that pushes a's and pops on b's) but not regular (by the pumping lemma). The containment is proper: $\mathrm{REG} \subsetneq \mathrm{CFL}$.

This is an `extends_grammar` edge. The PDA requires a fundamentally new mechanism—a stack—and this mechanism purchases genuinely new expressive power.

Why this distinction matters. The `restricts`/`extends_grammar` pair recurs throughout the graph. When a new model `restricts` an old one, the theory of the old model transfers intact—the same characterization theorems, the same closure properties, the same complexity measures apply. When a new model `extends_grammar`, the old theory typically *breaks*: new

characterization theorems are needed, closure properties may change (Table 3.1: context-free languages lose closure under intersection and complement), and new complexity measures may be required.

This distinction will reappear in the learning-theoretic setting: when a learning criterion generalizes another by relaxing a convergence requirement (e.g., behaviorally correct learning relaxes syntactic convergence), it is a conservative generalization. When it adds a new resource (e.g., adding a teacher who provides counterexamples), it is generative. Recognizing which type of generalization is at work is essential for predicting whether existing results transfer to the new setting.

3.3 Computability Foundations for Learning

Gold's model of identification in the limit requires learners to be effective procedures that output hypotheses from an effective enumeration. This section fixes the relevant computability-theoretic vocabulary.

Definition 3.8 (Gödel Numbering)**.** A *Gödel numbering* is an effective bijection between natural numbers and programs (or, equivalently, Turing machines). We write φ_e for the partial computable function computed by the program with index e, and $W_e = \mathrm{dom}(\varphi_e)$ for the e-th r.e. set. The sequence $\varphi_0, \varphi_1, \varphi_2, \ldots$ is a *universal effective enumeration* of all partial computable functions, and $W_0, W_1, W_2, \ldots$ is an effective enumeration of all r.e. sets.

In Gold's framework, a learner receiving a text for a language L outputs, at each time step, a natural number e_t. This number is interpreted as a hypothesis via the Gödel numbering: the learner conjectures that $L = W_{e_t}$. Identification in the limit means that the sequence $e_0, e_1, e_2, \ldots$ eventually stabilizes on an index e^* with $W_{e^*} = L$.

The critical point is that learners are themselves programs: a learner is a computable function $M : \Sigma^* \to \mathbb{N}$ mapping finite data sequences to hypothesis indices. The set of admissible learners is thus the set of total computable functions from finite sequences to $\mathbb{N}$. This effectivity requirement is what distinguishes Gold's model from a purely set-theoretic notion of convergence.

3.4 The Learning-Theoretic Bridge: Limiting Recursion and Δ_2^0

The deepest connection between computability theory and learning theory is not a definition but an equivalence. It identifies Gold's criterion of identification in the limit with a precise level of the arithmetical hierarchy.

Definition 3.9 (Limiting Recursion)**.** A function $f : \mathbb{N} \to \mathbb{N}$ is *limiting recursive* (or Δ_2^0-*computable*) if there exists a total computable function $\hat{f} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for every x,

$$f(x) = \lim_{s \to \infty} \hat{f}(x, s).$$

That is, f is the pointwise limit of a computable approximation sequence. For each input x, the approximation $\hat{f}(x, 0), \hat{f}(x, 1), \hat{f}(x, 2), \ldots$ eventually stabilizes on $f(x)$, but the time of stabilization is not itself computable.

Definition 3.10 (Δ_2^0 Class)**.** A set $A \subseteq \mathbb{N}$ is Δ_2^0 if both A and its complement $\bar{A}$ are Σ_2^0—that is, both are expressible as $\{x : \exists y \, \forall z \, R(x, y, z)\}$ for some computable predicate R. Equivalently, A is Δ_2^0 if and only if its characteristic function is limiting recursive.

The connection to learning is the following classical result.

Theorem 3.11 (Limiting Recursion and Learnability [Gol65, Put65, Sho59])**.** *A family of r.e. sets $\mathcal{L} = \{L_i\}_{i \in I}$ is identifiable in the limit (in the sense of Gold) if and only if the index function $i \mapsto e_i$, mapping each language to a correct hypothesis index, is limiting recursive. The classes identifiable in the limit are exactly those whose index sets are Δ_2^0.*

This theorem is the bridge between two fields. On the computability side, the Δ_2^0 sets occupy a precise position in the arithmetical hierarchy: they are the sets computable with an oracle for the halting problem, equivalently, the sets whose membership can be decided by a procedure that is allowed to change its mind finitely many times. On the learning side, identification in the limit is exactly the requirement that the learner's conjecture sequence stabilizes—which is exactly the requirement that the hypothesis function be a limit of computable approximations.

The equivalence tells us three things that will be used in Chapter 7:

1. **Upper bound on learnability.** No class whose index set lies above Δ_2^0 in the arithmetical hierarchy can be identified in the limit. This bounds what Gold-style learning can achieve from above.
2. **Mind changes as approximation stages.** The number of times a limiting-recursive function changes its approximation on input x—the number of s for which $\hat{f}(x, s) \neq \hat{f}(x, s+1)$—corresponds to the number of mind changes a learner makes before converging. The mind-change ordinals of Chapter 13 refine this correspondence into a full hierarchy.
3. **Identification in the limit is not ad hoc.** Gold's criterion might appear to be one arbitrary convergence requirement among many. The Δ_2^0 characterization shows it is not arbitrary at all: it is the *canonical* notion of computability-in-the-limit, the natural first level above the computable in the arithmetical hierarchy.

3.5 What This Chapter Established

This chapter served two functions. As vocabulary, it fixed the definitions—DFA, NFA, PDA, Turing machine, CFG, Gödel numbering, partial computable function—needed for the identification-in-the-limit chapters. As conceptual contribution, it introduced two structural ideas:

1. **The `restricts/extends_grammar` distinction.** The NFA–DFA pair (conservative generalization, same power) and the PDA–DFA pair (generative generalization, strictly more power) provide the first concrete illustration of an edge-type distinction that recurs throughout the graph. Whether a generalization is conservative or generative determines whether existing theory transfers or must be rebuilt.
2. **The Δ_2^0 bridge.** Identification in the limit is not merely a learning-theoretic desideratum but a precise computability-theoretic concept: it is computation in the limit, equivalent to Δ_2^0-computability. This equivalence anchors Gold's model in the arithmetical hierarchy and provides the foundation for the mind-change hierarchy of Chapter 13.

Chapter 4

Learners and Teachers

Chapters 1 and 2 established what is learned (concept classes) and how data arrives (i.i.d. samples, texts, queries, streams). This chapter introduces the agents that do the learning and, in some settings, the agents that provide the data.

Most of the vocabulary here is taxonomic: a learner is a function from data to hypotheses, and the field has named a dozen variants distinguished by which constraints they satisfy. These variants are best presented as a table, not as a sequence of isolated definitions. The chapter's genuine mathematical content lies elsewhere: in the type mismatch between Bayesian posteriors and PAC hypotheses (resolved by the Gibbs posterior), in the surprising power of team learning (Smith, 1982), and in the teacher models that connect to the teaching dimension and to Angluin's L^*.

4.1 The Learner Abstraction

Definition 4.1 (Learner)**.** A *learner* is a function

$$L\colon \bigcup_{m=0}^{\infty} (X \times Y)^m \;\to\; \mathcal{H}$$

that maps a finite data sequence to a hypothesis. At time t, having observed $S_t = ((x_1, y_1), \ldots, (x_t, y_t))$, the learner outputs $h_t = L(S_t) \in \mathcal{H}$.

This definition is deliberately minimal. It says nothing about how the learner selects h_t: whether it stores the full history or only its previous guess, whether it uses randomness, whether it may choose which x to query. Each such constraint yields a named variant, and the field has accumulated many. Figure 4.1 displays them as a tree; Table 4.1 provides the precise constraints.

4.2 Taxonomy of Learner Types

These distinctions are not mere nomenclature. Several generate strict separations in learning power within Gold's framework: for example, there exist classes identifiable by a conservative learner from informant but not from text, and classes identifiable by a set-driven learner that require unbounded memory for an iterative learner. These separations are developed in Chapters 7 and 14.

In the PAC setting, by contrast, many of the distinctions collapse. Any PAC-learnable class is learnable by a consistent, proper learner (the ERM principle), and the order of examples is irrelevant by the i.i.d. assumption, so the set-driven/order-sensitive distinction vanishes. The taxonomy has its sharpest teeth in the Gold and online paradigms, where the learner's memory management and update policy genuinely affect what is learnable.

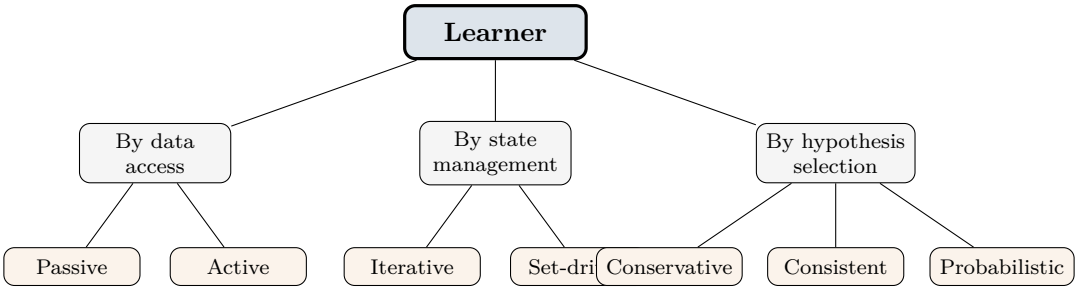

Figure 4.1: The learner taxonomy. Branches correspond to the axis of variation; leaves are the named specializations. The Bayesian, team, and meta learners are not shown here because they carry genuine mathematical content and are treated in dedicated sections below.

4.3 The Bayesian–PAC Bridge

The learner types in Table 4.1 all output a single hypothesis $h \in \mathcal{H}$. The Bayesian learner does not.

Definition 4.2 (Bayesian Learner)**.** A *Bayesian learner* maintains a prior π over $\mathcal{H}$ and, given data S, outputs the posterior distribution

$$Q(h \mid S) \;=\; \frac{\pi(h) \cdot P(S \mid h)}{\sum_{h' \in \mathcal{H}} \pi(h') \cdot P(S \mid h')}.$$

The output is a distribution over hypotheses, not a single hypothesis.

4.3.1 The type mismatch

This creates a structural problem. PAC learning requires a learner that outputs a single $h \in \mathcal{H}$ satisfying $R_D(h) \le R_D(h^*) + \varepsilon$ with probability $1 - \delta$. The Bayesian learner outputs $Q(\cdot \mid S)$—a probability measure on $\mathcal{H}$. These are different types:

$$\text{PAC learner:} \quad (X \times Y)^m \to \mathcal{H}, \qquad \text{Bayesian learner:} \quad (X \times Y)^m \to \Delta(\mathcal{H}).$$

The naive resolution—output the MAP hypothesis $\hat{h} = \arg\max_h Q(h \mid S)$—discards most of the posterior's information and yields suboptimal generalization bounds. The correct resolution is the Gibbs posterior.

Obstruction

Bayesian → PAC type mismatch. The Bayesian learner's output type ($\Delta(\mathcal{H})$) does not match the PAC learner's output type ($\mathcal{H}$). This is not a technicality: the PAC-Bayes theorem (Chapter 12) bounds the risk of a *distribution* over hypotheses, and extracting a single hypothesis from this bound requires an explicit randomized construction—the Gibbs posterior.

4.3.2 The Gibbs posterior

Definition 4.3 (Gibbs Posterior)**.** Given a prior π over $\mathcal{H}$, a loss function ℓ, and a sample S of size m, the *Gibbs posterior* at inverse temperature $\lambda > 0$ is the distribution

$$Q_\lambda(h) \;\propto\; \pi(h) \cdot \exp\!\big(-\lambda \cdot \hat{R}_S(h)\big).$$

The *Gibbs classifier* draws $h \sim Q_\lambda$ and predicts according to h. Its expected risk is $\mathbb{E}_{h \sim Q_\lambda}[R_D(h)]$.

Table 4.1: Named learner types. Each row gives the defining constraint and the chapter where the concept plays its primary role. These are not independent definitions; a single learner may satisfy several constraints simultaneously (e.g., a conservative, set-driven, passive learner).

Name	Defining Constraint	Primary Role
Passive	Receives data without choosing which instances to observe. The standard assumption in PAC and Gold settings.	Ch. 5, 7
Active	Chooses which instances to query. Performance measured by *label complexity*: the number of labels requested, not the number of unlabeled instances seen.	Ch. 8
Conservative	Changes hypothesis only when forced by inconsistency: if h_t is consistent with S_{t+1}, then $h_{t+1} = h_t$.	Ch. 7
Consistent	Always outputs a hypothesis consistent with all data seen so far: $h_t(x_i) = y_i$ for all $i \leq t$.	Ch. 5, 7
Set-driven	Output depends only on the *set* $\{(x_i, y_i)\}$ of data, not on the order of presentation. Formally, $L(S) = L(S')$ whenever S and S' are permutations of each other.	Ch. 7
Iterative	Hypothesis at step t depends only on h_{t-1} and the new datum (x_t, y_t), not on the full history S_t. Memory is bounded: the learner's state is a single hypothesis.	Ch. 7, 6
Probabilistic	May use internal randomness in selecting hypotheses. The success criterion (PAC, identification, mistake bound) is then required to hold with high probability over both the data and the learner's coin flips.	Ch. 5
With advice	Receives b bits of advice in addition to data. The advice may encode information about the target class or the distribution, and b parameterizes the amount of side information.	Ch. 8

The Gibbs posterior bridges the type mismatch as follows. The PAC-Bayes theorem [McA99] states that for any prior π and any posterior Q (not necessarily the Gibbs posterior), with probability $1 - \delta$ over $S \sim D^m$,

$$\mathbb{E}_{h\sim Q}[R_D(h)] \;\leq\; \mathbb{E}_{h\sim Q}[\hat{R}_S(h)] + \sqrt{\frac{\mathrm{KL}(Q\|\pi) + \ln(m/\delta)}{2m}}.$$

The Gibbs posterior Q_λ is the distribution that minimizes the right-hand side: it trades off empirical risk against KL divergence from the prior. This makes the PAC-Bayes bound *actionable*—it tells the learner which posterior to use, not merely that some posterior satisfies a bound.

Remark 4.4 (Gibbs $\neq$ MAP). The Gibbs posterior draws a *random* hypothesis from a posterior weighted by empirical performance. The MAP estimate takes the *mode*. In high-dimensional settings, the Gibbs posterior can provide tighter generalization guarantees because the PAC-Bayes bound controls the expected risk under the full posterior, while MAP-based bounds require uniform convergence over a singleton. The relationship is analogous to the distinction between Bayesian model averaging and model selection in statistics: the former controls average-case risk, the latter controls worst-case risk over a point estimate.

The full development of PAC-Bayes bounds, including the McAllester and Catoni variants, appears in Chapter 12. The present section establishes only the agent-level fact: the Gibbs posterior is the mechanism by which a Bayesian learner becomes a PAC learner.

4.4 Teams and Meta-Learning

4.4.1 Team learning

Most learning models assume a single learner. The team model asks what happens when multiple learners collaborate—or, more precisely, when success requires only that *at least one* member of the team converges to the correct hypothesis.

Definition 4.5 (Team Learner)**.** A *team* of size k is a set of learners $\{L_1, \ldots, L_k\}$. The team *identifies* a concept $c \in \mathcal{C}$ if at least one L_i converges to a correct hypothesis. The class $\mathcal{C}$ is *team-identifiable* (in the limit) if there exists a finite team that identifies every $c \in \mathcal{C}$.

Theorem 4.6 (Teams > Individuals [Smi82])**.** *There exist concept classes that are identifiable by a team of size 2 but not by any single learner.*

This is a strict increase in learning power, and it is surprising on two counts. First, the team members receive the *same* data—they do not partition the instance space or communicate. The power comes purely from the disjunctive success criterion: "at least one succeeds" is weaker than "all succeed," and this weakened criterion expands the class of learnable concepts. Second, the separation is not an artifact of infinite hypothesis spaces or exotic concept classes. It holds within Gold's standard framework of identification in the limit.

The proof constructs a concept class $\mathcal{C}$ such that any single learner can be forced into infinitely many mind changes by a diagonalization argument (similar to the text/informant separation in Chapter 2), but two learners with complementary strategies—one "optimistic" (guessing large languages) and one "pessimistic" (guessing small languages)—guarantee that at least one converges. The formal argument appears in Chapter 14.

Remark 4.7 (Teams and probabilistic learners). A team of deterministic learners is strictly more powerful than a single deterministic learner (Theorem 4.6). However, a probabilistic learner that succeeds with probability $> 1/2$ can simulate a team of size k by running k independent copies internally. In the Gold setting, where success means convergence with certainty, this simulation is not available, and the team hierarchy is strict.

4.4.2 Meta-learning

Definition 4.8 (Meta-Learner)**.** A *meta-learner* operates over a family of learning tasks $\{\mathcal{C}_j\}_{j=1}^{n}$, each drawn from an environment $\mathcal{E}$. Given experience on tasks $\mathcal{C}_1, \ldots, \mathcal{C}_{n-1}$, it produces a learner (or a bias, such as a hypothesis space or prior) that performs well on a new task $\mathcal{C}_n$ drawn from the same environment.

Baxter [Bax00] established the first PAC-style bounds for meta-learning: if the environment $\mathcal{E}$ has bounded complexity (measured by covering numbers over a family of hypothesis spaces), and the meta-learner sees n tasks each with m examples, then the meta-learner's expected excess risk on a new task is bounded by

$$O\left(\sqrt{\frac{\text{complexity}(\mathcal{E})}{n}} + \sqrt{\frac{\text{VCdim}(\mathcal{H})}{m}}\right),$$

decomposing the error into a *task-level* term (how well the environment is estimated from n tasks) and a *within-task* term (how well a single task is learned given m examples). This two-level structure is the defining feature of meta-learning bounds.

Open Problem 4.9 (Meta-learning characterization). No analogue of the fundamental theorem of PAC learning exists for meta-learning. Baxter's bounds use covering numbers, not a combinatorial dimension. Is there a single complexity measure of $\mathcal{E}$ that characterizes meta-learnability in the same way that the VC dimension characterizes PAC learnability? The question is open.

4.5 Teachers and the Teaching Game

In the models above, data arrives passively (from a distribution or enumeration) or through the learner's queries. The *teacher* model reverses the direction: an agent actively chooses examples to help (or hinder) the learner.

Table 4.2: Teacher types. Each row gives the teacher's strategy, the induced data model, and the chapter where the concept plays its primary role.

Name	Strategy and Key Property	Primary Role
Random	The *random teacher* draws examples i.i.d. from a distribution D. The standard PAC assumption; the teacher has no intention, and "teaching" is an anthropomorphic misnomer for sampling.	Ch. 5
Adversarial	Chooses worst-case examples for the learner, adaptively. The online learning model treats the data stream as an adversarial teacher. Performance is measured by worst-case mistake count.	Ch. 6
Optimal	Selects the smallest set of examples from which the target concept can be uniquely identified. The size of this set is the *teaching dimension* of the concept within the class.	Ch. 13
Minimally adequate	Angluin's model: answers membership queries and equivalence queries. "Minimally adequate" because MQ + EQ is the minimal oracle configuration that makes DFAs efficiently learnable.	Ch. 8

4.5.1 The optimal teacher and teaching dimension

Definition 4.10 (Teaching Dimension)**.** The *teaching dimension* of a concept $c \in \mathcal{C}$ is the minimum number of labeled examples that uniquely identify c within $\mathcal{C}$:

$$\mathrm{TD}(c, \mathcal{C}) = \min\{|S| : S \subseteq X \times Y,\ c \text{ is the only } c' \in \mathcal{C} \text{ consistent with } S\}.$$

The teaching dimension of the class is $\mathrm{TD}(\mathcal{C}) = \max_{c \in \mathcal{C}} \mathrm{TD}(c, \mathcal{C})$. The *optimal teacher* for c is any teacher that achieves $\mathrm{TD}(c, \mathcal{C})$.

The teaching dimension measures the cooperative communication complexity of learning: how many examples must a helpful teacher provide so that a learner who knows $\mathcal{C}$ can identify the target? It is developed in full in Chapter 13, where it interacts with recursive teaching dimension and the teaching–learning duality.

4.5.2 The minimally adequate teacher

Definition 4.11 (Minimally Adequate Teacher [Ang88])**.** A *minimally adequate teacher* (MAT) for a concept class $\mathcal{C}$ provides two oracles: a membership oracle MQ and an equivalence oracle EQ, as defined in Chapter 2. A class is learnable from a MAT if there exists an algorithm that exactly identifies any $c \in \mathcal{C}$ using polynomially many queries.

Angluin's L^* algorithm (Theorem 2.11) shows that DFAs are learnable from a MAT. The minimally adequate teacher is "minimal" in a precise sense: dropping either oracle makes the class unlearnable in polynomial time. The full treatment of L^*, including the observation table construction and its connection to the Myhill–Nerode theorem, appears in Chapter 8.

Open Problem 4.12 (Polynomial-time MAT learnability). Which concept classes are learnable from a minimally adequate teacher in polynomial time? Beyond DFAs, the characterization is incomplete. Known positive results include decision trees, certain classes of Boolean formulas, and residual finite-state automata. A general combinatorial characterization analogous to the VC dimension for PAC learning is not known.

4.6 Deferred Agent Types

Three agent concepts in the knowledge graph—synthesizer (a program-synthesis agent), verifier (a correctness checker for synthesizer output), and llm_critic (a large language model used as a critic in a learning loop)—belong to the application layer. They are built *from* the learner and teacher primitives defined above, not alongside them. Their treatment is deferred to Chapter 18, where they appear in the context of program synthesis, neurosymbolic learning, and LLM-in-the-loop verification.

4.7 What This Chapter Established

The chapter's contribution is a taxonomy with three points of genuine content:

1. **Most learner types are constraints on a single abstraction.** The eight named variants in Table 4.1 are not independent concepts—they are orthogonal constraints (data access, state management, hypothesis selection) applied to the base type $L\colon (X \times Y)^* \to \mathcal{H}$. In the PAC setting, many of these distinctions collapse; they are sharpest in Gold's framework.
2. **The Bayesian–PAC bridge requires a type conversion.** The Bayesian learner outputs a posterior $Q \in \Delta(\mathcal{H})$; PAC learning requires a single $h \in \mathcal{H}$. The Gibbs posterior resolves this by sampling $h \sim Q_\lambda$, and the PAC-Bayes theorem (Chapter 12) bounds the resulting risk.
3. **Teams are strictly more powerful than individuals.** In the Gold setting, a team of two learners can identify classes that no single learner can. The mechanism is disjunctive success, not data partitioning or communication.

Teachers, similarly, range from passive (random) to adversarial to cooperative (optimal, minimally adequate). The teaching dimension and MAT learnability connect this chapter to the complexity measures in Chapter 13 and the exact learning algorithms in Chapter 8.

Chapter 5

PAC Learning and the Fundamental Theorem

This is the central chapter of the book. Every paradigm that follows—online learning, Gold-style identification, universal learning—will be understood partly by how it resembles and partly by how it differs from what is established here. The chapter builds toward a single destination: the Fundamental Theorem of Statistical Learning, which characterizes PAC learnability through a web of nine equivalent conditions. The path there is the content. Each lemma is a foothold; the VC characterization is the summit; the full equivalence web is the view from the top.

The chapter is organized as an ascent.

1. **The PAC framework** (Section 5.1): the definition and its moving parts. Brief—the definition is not the centerpiece.
2. **The VC characterization proof** (Section 5.2): the full proof that $\text{VCdim}(\mathcal{H}) < \infty$ if and only if $\mathcal{H}$ is PAC learnable, built in four stages through Sauer–Shelah, uniform convergence, ERM analysis, and the converse.
3. **The Fundamental Theorem** (Section 5.3): all nine equivalent conditions, the equivalence web, and the directions we proved versus those we cited.
4. **The agnostic setting** (Section 17.3): why dropping the realizability assumption changes sample complexity from $\Theta(d/\varepsilon)$ to $\Theta(d/\varepsilon^2)$, and why this gap is not an artifact.
5. **Lower bounds and No Free Lunch** (Section 5.5): the matching lower bound $\Omega(d/\varepsilon)$ and the full NFL proof.
6. **Computational interlude** (Section 5.6): the information–computation gap.

5.1 The PAC Framework

Two assumptions frame the discussion.

Definition 5.1 (Realizable setting)**.** Learning is *realizable* if the target concept c^* lies in the hypothesis class: $c^* \in \mathcal{H}$.

Definition 5.2 (Agnostic setting)**.** Learning is *agnostic* if no assumption is made on the relationship between c^* and $\mathcal{H}$. The learner competes with the best hypothesis $h^* = \arg\min_{h \in \mathcal{H}} R_D(h)$.

The realizable setting is a special case of the agnostic one (take $h^* = c^*$, achieving zero risk). The agnostic setting is the one that matters in practice, but the realizable setting is where the cleanest characterization lives. We begin there.

Definition 5.3 (PAC Learning [Val84])**.** A hypothesis class $\mathcal{H}$ over domain X is *PAC learnable* if there exists an algorithm A and a function $m_{\mathcal{H}}\colon (0,1)^2 \to \mathbb{N}$ such that, for every $\varepsilon, \delta \in (0,1)$ and every distribution D on X:

If $c^* \in \mathcal{H}$ and $S \sim D^m$ with $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$, then

$$\mathbb{P}_{S \sim D^m}[R_D(A(S)) > \varepsilon] \leq \delta.$$

The function $m_{\mathcal{H}}(\varepsilon, \delta)$ is the *sample complexity* of $\mathcal{H}$. When $m_{\mathcal{H}}$ is polynomial in $1/\varepsilon$ and $1/\delta$, the class is *efficiently PAC learnable* (information-theoretically); computational efficiency is a separate requirement.

Before unpacking the definition, we illustrate the full PAC cycle on a concrete class where every step is visible.

Example 5.4 (PAC learning of rectangles: the full cycle)**.** Let $\mathcal{H}$ be the class of axis-aligned rectangles in $\mathbb{R}^2$: $h_{(a_1,a_2,b_1,b_2)}(x) = \mathbf{1}[a_1 \leq x_1 \leq b_1 \text{ and } a_2 \leq x_2 \leq b_2]$.

VC dimension. $\mathrm{VCdim}(\mathcal{H}) = 4$. Four points, one near each side of a rectangle, can be shattered: for each subset T of the four points, the rectangle that tightly encloses exactly T labels precisely T as positive. No five points can be shattered: given any five points in $\mathbb{R}^2$, at least one is "interior" (not extremal in any coordinate), and the labeling that excludes only that point cannot be realized by a rectangle.

ERM and the error region. Given sample S consistent with target rectangle R^*, the ERM learner returns the tightest enclosing rectangle R_S of the positive examples. Since all positive examples lie in R^*, we have $R_S \subseteq R^*$: zero false positives. The error region $R^* \setminus R_S$ consists of four *strips*—one per side of R^*—each containing no positive sample.

Sample complexity. For each side j ($j = 1, \ldots, 4$), let T_j be the strip of R^* closest to side j with probability mass exactly $\varepsilon/4$ under D. If a positive example falls in T_j, then R_S extends into T_j and that strip's contribution to the error drops below $\varepsilon/4$. The probability that strip T_j is missed entirely by m i.i.d. samples is at most $(1 - \varepsilon/4)^m \leq e^{-m\varepsilon/4}$. A union bound over four strips gives

$$\mathbb{P}[R_D(R_S) > \varepsilon] \;\leq\; 4\, e^{-m\varepsilon/4}.$$

Setting the right-hand side $\leq \delta$ yields $m = \frac{4}{\varepsilon} \ln \frac{4}{\delta} = O(\frac{d + \log(1/\delta)}{\varepsilon})$ with $d = 4$—matching the Fundamental Theorem's prediction with the tight $1/\varepsilon$ dependence.

The analysis makes visible what the general proof obscures: the error region has *geometric structure* (four strips), the union bound exploits the *number of strips* (controlled by VC dimension), and the exponential decay in m comes from the i.i.d. assumption applied to each strip independently. Replacing $\mathbb{R}^2$ with $\mathbb{R}^k$ gives rectangles with $\mathrm{VCdim} = 2k$ and $2k$ strips, and the same argument yields $m = O(k/\varepsilon \cdot \log(k/\delta))$.

Three features of this definition deserve emphasis:

1. **Distribution-free.** The guarantee holds for *every* distribution D. The learner does not know D and cannot assume anything about it.
2. **Approximate.** The learner need not find c^* exactly; error $\leq \varepsilon$ suffices.
3. **Probably.** The guarantee is probabilistic: it may fail with probability δ, but δ can be driven arbitrarily small by drawing more samples.

Remark 5.5 (The role of δ). The δ parameter is structurally unimportant for characterization purposes: a bound with confidence $1 - \delta$ can always be converted to one with confidence $1 - \delta'$ by replacing m with $m \cdot \lceil \log(1/\delta') / \log(1/\delta) \rceil$ and taking a majority vote. Consequently, the sample complexity depends on $\log(1/\delta)$, not on $1/\delta$, and the VC characterization is insensitive to how δ enters.

Definition 5.6 (Empirical Risk Minimization)**.** Given sample $S = \{(x_i, c^*(x_i))\}_{i=1}^m$ and hypothesis class $\mathcal{H}$, the *empirical risk minimizer* is

$$\mathrm{ERM}_{\mathcal{H}}(S) = \arg\min_{h \in \mathcal{H}} \hat{R}_S(h), \qquad \hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}[h(x_i) \neq c^*(x_i)].$$

In the realizable setting, ERM returns any $h \in \mathcal{H}$ consistent with S (since $\hat{R}_S(c^*) = 0$).

The question this chapter answers: *for which $\mathcal{H}$ does ERM succeed, and when is PAC learning possible at all?*

5.2 The VC Characterization

The main result of this section is:

Theorem 5.7 (VC Characterization of PAC Learnability [BEHW89, VC71])**.** *Let $\mathcal{H}$ be a hypothesis class over domain X. The following are equivalent:*

(i) *$\mathcal{H}$ is PAC learnable.*

(ii) *$\mathcal{H}$ has the uniform convergence property.*

(iii) $\mathrm{VCdim}(\mathcal{H}) < \infty$.

Moreover, if $d = \mathrm{VCdim}(\mathcal{H}) < \infty$, then $\mathcal{H}$ is PAC learnable with sample complexity

$$m_{\mathcal{H}}(\varepsilon, \delta) = O\left(\frac{d + \log(1/\delta)}{\varepsilon}\right).$$

The proof proceeds in four stages. Each stage is a separate lemma, and each lemma is a foothold on the ascent.

Stage 1: Finite VC dimension $\Longrightarrow$ polynomial growth function (Sauer–Shelah).
Stage 2: Polynomial growth function $\Longrightarrow$ uniform convergence (ε-net/symmetrization argument).
Stage 3: Uniform convergence $\Longrightarrow$ ERM is a PAC learner.
Stage 4 (Converse): Infinite VC dimension $\Longrightarrow$ not PAC learnable.

5.2.1 Stage 1: Sauer–Shelah (Statement)

The growth function measures the effective richness of $\mathcal{H}$ on finite samples.

Definition 5.8 (Growth Function)**.** For a hypothesis class $\mathcal{H}$ over X and integer $m \geq 1$,

$$\Pi_{\mathcal{H}}(m) = \max_{\{x_1, \ldots, x_m\} \subseteq X} \left|\{(h(x_1), \ldots, h(x_m)) : h \in \mathcal{H}\}\right|.$$

The growth function satisfies $\Pi_{\mathcal{H}}(m) \leq 2^m$ always, with equality when $\mathcal{H}$ shatters some set of size m. The Sauer–Shelah lemma says that finite VC dimension forces a polynomial bound.

Lemma 5.9 (Sauer–Shelah [Sau72, She72])**.** *If $\mathrm{VCdim}(\mathcal{H}) = d$, then for all $m \geq d$,*

$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} \leq \left(\frac{em}{d}\right)^d.$$

The proof of the Sauer–Shelah lemma is combinatorial, proceeding by induction on $m + d$. It is given in full in Chapter 10 (Section 10.1.2). For the present chapter, we take it as given and use only the consequence: if $d = \mathrm{VCdim}(\mathcal{H}) < \infty$, then $\Pi_{\mathcal{H}}(m) = O(m^d)$—polynomial, not exponential.

5.2.2 Stage 2: Uniform Convergence

The uniform convergence property says that empirical risk converges to true risk *simultaneously* for all hypotheses in $\mathcal{H}$, not just for a single fixed h.

Definition 5.10 (Uniform Convergence)**.** A hypothesis class $\mathcal{H}$ has the *uniform convergence property* if for every $\varepsilon, \delta > 0$, there exists $m_{\mathrm{UC}}(\varepsilon, \delta)$ such that for all distributions D, whenever $m \geq m_{\mathrm{UC}}(\varepsilon, \delta)$:

$$\mathbb{P}_{S \sim D^m}\left[\sup_{h \in \mathcal{H}} |R_D(h) - \hat{R}_S(h)| > \varepsilon\right] \leq \delta.$$

Theorem 5.11 (Uniform Convergence from Finite Growth)**.** *If $\Pi_{\mathcal{H}}(m) \leq (em/d)^d$ for all $m \geq d$, then $\mathcal{H}$ has the uniform convergence property with*

$$m_{\mathrm{UC}}(\varepsilon, \delta) = O\left(\frac{d \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon^2}\right).$$

In the realizable setting, the ε^2 denominator improves to ε, giving $m = O\left(\frac{d \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}\right)$.

Proof. The proof uses the *symmetrization* (or "ghost sample") technique of Vapnik and Chervonenkis [VC71].

Step 1: Symmetrization. Draw two independent samples $S, S' \sim D^m$. We claim that for $m \geq 8/\varepsilon^2$,

$$\mathbb{P}_S\left[\sup_h |R_D(h) - \hat{R}_S(h)| > \varepsilon\right] \leq 2\, \mathbb{P}_{S,S'}\left[\sup_h |\hat{R}_S(h) - \hat{R}_{S'}(h)| > \varepsilon/2\right].$$

This follows because if $|R_D(h) - \hat{R}_S(h)| > \varepsilon$ for some h, then with probability at least $1/2$ (by a Chebyshev argument on S'), the ghost sample satisfies $|\hat{R}_{S'}(h) - R_D(h)| \leq \varepsilon/2$, and hence $|\hat{R}_S(h) - \hat{R}_{S'}(h)| > \varepsilon/2$ by the triangle inequality.

Step 2: Permutation argument. Let $T = S \cup S'$ be the pooled sample of size $2m$. Conditioned on T, the partition into S and S' is uniformly random among all $\binom{2m}{m}$ splits. By a union bound over the distinct label patterns that $\mathcal{H}$ induces on T:

$$\mathbb{P}_{S,S'}\left[\sup_h |\hat{R}_S(h) - \hat{R}_{S'}(h)| > \varepsilon/2\right] \leq \Pi_{\mathcal{H}}(2m) \cdot \max_h \mathbb{P}_{\mathrm{split}}[|\hat{R}_S(h) - \hat{R}_{S'}(h)| > \varepsilon/2].$$

The number of distinct behaviors is at most $\Pi_{\mathcal{H}}(2m)$, and for each fixed labeling pattern, $\hat{R}_S(h) - \hat{R}_{S'}(h)$ is a sum of $2m$ centered random variables (each $\pm 1/m$ depending on which half the point falls in).

Step 3: Hoeffding bound. For each fixed label pattern, Hoeffding's inequality gives

$$\mathbb{P}_{\mathrm{split}}[|\hat{R}_S(h) - \hat{R}_{S'}(h)| > \varepsilon/2] \leq 2 \exp(-m\varepsilon^2/8).$$

Combining with the Sauer–Shelah bound:

$$\mathbb{P}_S\left[\sup_h |R_D(h) - \hat{R}_S(h)| > \varepsilon\right] \leq 4\left(\frac{2em}{d}\right)^d \exp\left(-\frac{m\varepsilon^2}{8}\right).$$

Setting this $\leq \delta$ and solving for m gives

$$m = O\left(\frac{d \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon^2}\right).$$

Realizable improvement. In the realizable setting, $R_D(c^*) = 0$, so only one-sided deviations matter: we need $\mathbb{P}[\exists h \in \mathcal{H} : \hat{R}_S(h) = 0 \text{ but } R_D(h) > \varepsilon] \le \delta$. For any fixed h with $R_D(h) > \varepsilon$, the probability that h is consistent with all m samples is at most $(1-\varepsilon)^m \le e^{-m\varepsilon}$. A union bound over the $\Pi_{\mathcal{H}}(m)$ distinct behaviors gives

$$\mathbb{P}[\text{bad}] \le \left(\frac{em}{d}\right)^d e^{-m\varepsilon}.$$

Setting this $\le \delta$ yields $m = O\Big(\frac{d\log(d/\varepsilon)+\log(1/\delta)}{\varepsilon}\Big)$, confirming the $1/\varepsilon$ dependence. □

Remark 5.12 (The Hanneke refinement). The optimal sample complexity in the realizable case is $\Theta\Big(\frac{d+\log(1/\delta)}{\varepsilon}\Big)$, achieved by a careful one-inclusion graph analysis. The $\log(d/\varepsilon)$ factor in the union bound above is a mild overhead; Hanneke [Han16] showed it can be removed entirely, establishing the tight bound.

5.2.3 Stage 3: Uniform Convergence Implies ERM Succeeds

Proposition 5.13 (ERM is a PAC learner under uniform convergence)**.** *If $\mathcal{H}$ has the uniform convergence property, then* $\mathrm{ERM}_{\mathcal{H}}$ *is a PAC learner for $\mathcal{H}$.*

Proof. Suppose $m \ge m_{\mathrm{UC}}(\varepsilon/2, \delta)$, so that with probability $\ge 1-\delta$,

$$\sup_{h\in\mathcal{H}} |R_D(h) - \hat{R}_S(h)| \le \varepsilon/2.$$

Let $h_S = \mathrm{ERM}_{\mathcal{H}}(S)$. In the realizable case, $\hat{R}_S(c^*) = 0$, so $\hat{R}_S(h_S) = 0$ as well. Then:

$$R_D(h_S) = \underbrace{(R_D(h_S) - \hat{R}_S(h_S))}_{\le \varepsilon/2} + \underbrace{\hat{R}_S(h_S)}_{=0} \le \varepsilon/2 < \varepsilon.$$

In the general (non-realizable) case, let $h^* = \arg\min_{h\in\mathcal{H}} R_D(h)$. Then:

$$R_D(h_S) \le \hat{R}_S(h_S) + \varepsilon/2 \le \hat{R}_S(h^*) + \varepsilon/2 \le R_D(h^*) + \varepsilon.$$ □

5.2.4 Stage 4: Infinite VC Dimension Implies Failure

Theorem 5.14 (Converse of the VC Characterization)**.** *If* $\mathrm{VCdim}(\mathcal{H}) = \infty$*, then $\mathcal{H}$ is not PAC learnable.*

Proof. The proof constructs, for any candidate learner A and any sample size m, a distribution on which A must fail. The construction is adversarial.

Since $\mathrm{VCdim}(\mathcal{H}) = \infty$, there exists a shattered set $C = \{x_1, \dots, x_{2m}\} \subseteq X$ of size $2m$. By the definition of shattering, for every labeling $b \in \{0,1\}^{2m}$, there exists $h_b \in \mathcal{H}$ with $h_b(x_i) = b_i$ for all i.

Now consider the uniform distribution D_b on C with target concept h_b. Run the learner A on a sample S of size m drawn uniformly from C. With high probability, at least m points of C are unseen (by a coupon-collector argument, at least $m/2$ are unseen in expectation; we use $2m$ points to ensure $\ge m$ unseen with constant probability).

On the unseen points, the learner has no information about the target labeling. We apply a probabilistic argument over b drawn uniformly from $\{0,1\}^{2m}$:

Key claim. For any fixed algorithm A and fixed training set S, if we draw b uniformly at random, then for any hypothesis $A(S)$ outputs, the expected error on unseen points is at least $1/2 \cdot (m/(2m)) = 1/4$.

This follows because, conditioned on S, the labels b_j for unseen points $x_j \notin S$ are independent uniform bits under the uniform distribution on target functions. Therefore $A(S)$'s prediction on each unseen point is correct with probability exactly $1/2$, regardless of the learner's strategy.

Since the unseen points constitute at least half the support of D_b, the expected true risk satisfies $\mathbb{E}_b[R_{D_b}(A(S))] \geq 1/4$. In particular, there exists a specific b^* such that $R_{D_{b^*}}(A(S)) \geq 1/4$, which means A fails to achieve $\varepsilon < 1/4$ with m samples under distribution D_{b^*}. Since m was arbitrary, $\mathcal{H}$ is not PAC learnable. □

Remark 5.15 (Strength of the converse). The converse is stronger than "not uniformly convergent": it says no learner whatsoever—not just ERM—can PAC learn $\mathcal{H}$. The argument works because the adversary chooses the distribution *after* seeing the learner. This distribution-free adversarial structure is the engine that makes VC dimension necessary, not just sufficient.

The circle closes. Finite VC dimension forces polynomial growth (Sauer–Shelah), which forces uniform convergence, which makes ERM succeed, which gives PAC learnability. Infinite VC dimension shatters arbitrarily large sets, and the converse kills learnability outright. A single combinatorial quantity—the largest set the class can shatter—controls everything.

5.3 The Fundamental Theorem of Statistical Learning

The VC characterization (Theorem 5.7) is the core equivalence. But the full picture is richer: PAC learnability is equivalent to *nine* conditions, not just three. The Fundamental Theorem packages them all.

Theorem 5.16 (The Fundamental Theorem of Statistical Learning [BEHW89, SSBD14])**.** *Let $\mathcal{H}$ be a hypothesis class of binary functions over a domain X. The following are equivalent:*

(F1) *$\mathcal{H}$ is PAC learnable.*

(F2) *$\mathcal{H}$ is agnostic PAC learnable.*

(F3) *$\mathrm{ERM}_{\mathcal{H}}$ is a PAC learner for $\mathcal{H}$.*

(F4) *$\mathcal{H}$ has the uniform convergence property.*

(F5) $\mathrm{VCdim}(\mathcal{H}) < \infty$.

(F6) $\Pi_{\mathcal{H}}(m) < 2^m$ *for some m.*

(F7) *$\mathcal{H}$ has a finite compression scheme.*

(F8) *Any finite subclass of $\mathcal{H}$ over a finite domain is PAC learnable (with bounds depending on $|\mathcal{H}|$), and this property "lifts" to $\mathcal{H}$.*

(F9) *$\mathcal{H}$ has finite Littlestone dimension $\Longrightarrow$ $\mathcal{H}$ is PAC learnable.* (The converse fails: this implication is strict. See Chapter 14.)

Remark 5.17 (On the numbering). Condition (F9) is an implication, not an equivalence: finite Littlestone dimension implies finite VC dimension (Chapter 10), but the converse fails—thresholds on $\mathbb{R}$ have $\mathrm{VCdim} = 1$ but $\mathrm{Ldim} = \infty$ (Example 1.4). We include it to mark the boundary of the equivalence web: this is where the PAC–online separation begins.

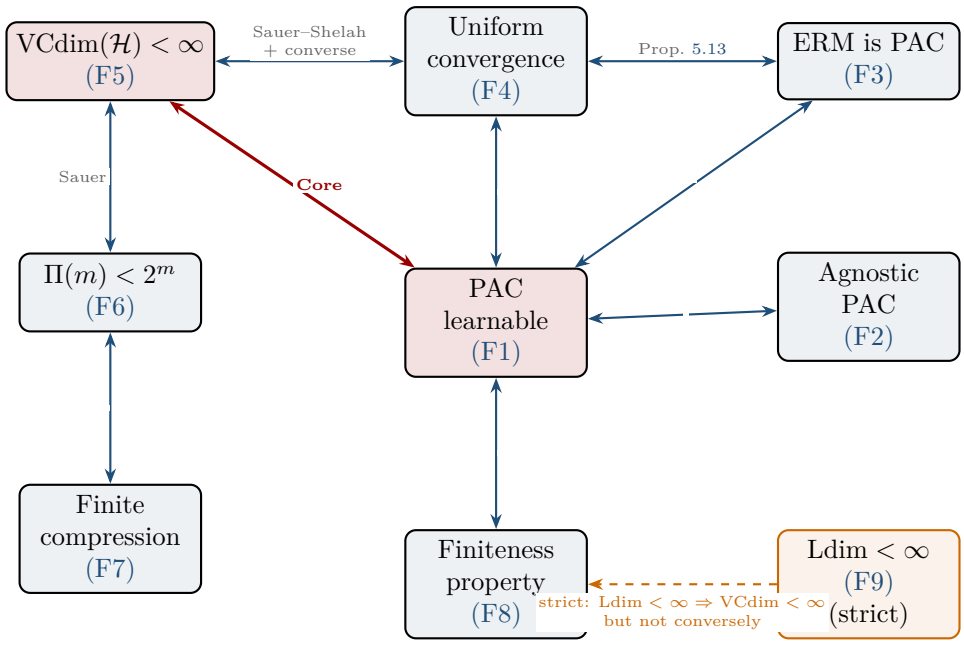

Figure 5.1: The equivalence web of the Fundamental Theorem. All solid bidirectional arrows are full equivalences. The dashed arrow from Ldim $< \infty$ is strict: finite Littlestone dimension implies PAC learnability, but not conversely. The thick red diagonal marks the VC characterization—the core equivalence proved in Section 5.2.

5.3.1 The Proof Architecture

We have already proved (in Section 5.2):

$$\text{(F5)} \Rightarrow \text{(F4)} \Rightarrow \text{(F3)} \Rightarrow \text{(F1)} \qquad \text{and} \qquad \neg\text{(F5)} \Rightarrow \neg\text{(F1)}.$$

This establishes (F5) $\Leftrightarrow$ (F4) $\Leftrightarrow$ (F3) $\Leftrightarrow$ (F1).

The remaining directions:

- (F5) $\Leftrightarrow$ (F6): By the Sauer–Shelah lemma (Lemma 10.5), $\mathrm{VCdim}(\mathcal{H}) = d$ implies $\Pi_{\mathcal{H}}(m) \leq (em/d)^d < 2^m$ for m large enough. Conversely, if $\mathrm{VCdim}(\mathcal{H}) = \infty$, then $\Pi_{\mathcal{H}}(m) = 2^m$ for all m (since $\mathcal{H}$ shatters a set of every finite size). This is immediate from the definition of VC dimension.
- (F1) $\Leftrightarrow$ (F2): Agnostic PAC learnability trivially implies realizable PAC learnability (set $c^* \in \mathcal{H}$, so the best hypothesis has zero risk). The converse uses the uniform convergence property: if $\mathcal{H}$ has finite VC dimension, then uniform convergence holds, and Proposition 5.13 shows ERM succeeds in the agnostic setting as well (with the ε^2 sample complexity discussed in Section 17.3).
- (F5) $\Leftrightarrow$ (F7): This is the deepest equivalence. We sketch the argument below.
- (F9) $\Rightarrow$ (F5): If $\mathrm{Ldim}(\mathcal{H}) < \infty$, then $\mathrm{VCdim}(\mathcal{H}) \leq \mathrm{Ldim}(\mathcal{H}) < \infty$, since any shattered set gives a complete binary mistake tree of the same depth (Chapter 10).

5.3.2 The Compression Direction (Sketch)

Definition 5.18 (Compression Scheme)**.** A *compression scheme* of size k for $\mathcal{H}$ is a pair (κ, ρ) where:

- κ (the compressor) maps any sample S consistent with some $h \in \mathcal{H}$ to a subsequence $\kappa(S) \subseteq S$ of size $\leq k$;

- ρ (the reconstructor) maps any subsequence of size $\leq k$ to a hypothesis $\rho(\kappa(S)) \in Y^X$;
- $\rho(\kappa(S))$ is consistent with the full sample S.

The direction (F7) $\Rightarrow$ (F1) is classical and relatively straightforward: a compression scheme of size k yields PAC learning with sample complexity $O\left(\frac{k\log(k/\varepsilon)+\log(1/\delta)}{\varepsilon}\right)$, since the compressed subsequence encodes all the information the learner needs, and there are at most $\binom{m}{k}$ possible compressed sets.

The converse direction (F5) $\Rightarrow$ (F7) is the deep one. It was an open conjecture for 30 years before being resolved by Moran and Yehudayoff [MY16].

Historical Note

The compression conjecture. Littlestone and Warmuth conjectured in 1986 that every class of VC dimension d has a compression scheme of size $O(d)$. The conjecture was resolved positively by Moran and Yehudayoff (2016), who proved that every class of VC dimension d has a compression scheme of size at most $2^{O(d)}$. The $O(d)$ bound remains open; the exponential gap between d and $2^{O(d)}$ is one of the field's remaining structural questions.

Proof sketch (Moran–Yehudayoff). The argument constructs a compression scheme from a *maximum class* (a class achieving the Sauer–Shelah bound with equality). Every class of VC dimension d can be embedded, for sample complexity purposes, into a maximum class of VC dimension d via a projection argument. For maximum classes, the one-inclusion graph has special structure: its edges can be oriented so that every vertex has in-degree at most d. This orientation defines a compression scheme—given sample S, output the at-most-d predecessors of the unique vertex in the one-inclusion graph determined by S. The reconstruction uses the structure of the maximum class to recover a consistent hypothesis from these d points.

Extending from maximum classes to arbitrary classes of finite VC dimension requires a more delicate argument involving fractional covers of the one-inclusion hypergraph, which blows up the size to $2^{O(d)}$. □

5.4 The Agnostic Setting and the ε^2 Price

Definition 5.19 (Agnostic PAC Learning)**.** A hypothesis class $\mathcal{H}$ is *agnostic PAC learnable* if there exists an algorithm A and a function $m_{\mathcal{H}}(\varepsilon,\delta)$ such that for every $\varepsilon,\delta \in (0,1)$ and every distribution D on $X \times \{0,1\}$ (not necessarily generated by any $c^* \in \mathcal{H}$), whenever $m \geq m_{\mathcal{H}}(\varepsilon,\delta)$:

$$\mathbb{P}_{S\sim D^m}\left[R_D(A(S)) - \min_{h\in\mathcal{H}} R_D(h) > \varepsilon\right] \leq \delta.$$

The Fundamental Theorem tells us that agnostic PAC learnability is equivalent to realizable PAC learnability ((F1) $\Leftrightarrow$ (F2)). The same classes are learnable. But the *sample complexity* changes, and this change is not cosmetic.

Theorem 5.20 (The Agnostic Sample Complexity Gap)**.** *Let* $d = \mathrm{VCdim}(\mathcal{H})$.

(a) In the realizable setting: $m(\varepsilon,\delta) = \Theta\left(\frac{d+\log(1/\delta)}{\varepsilon}\right)$.

(b) In the agnostic setting: $m(\varepsilon,\delta) = \Theta\left(\frac{d+\log(1/\delta)}{\varepsilon^2}\right)$.

The gap from $1/\varepsilon$ *to* $1/\varepsilon^2$ *is tight: no algorithm can achieve* $o(d/\varepsilon^2)$ *in the agnostic setting, and no algorithm needs* $\omega(d/\varepsilon)$ *in the realizable setting.*

Where does the gap come from? It is tempting to blame the proof technique—perhaps a cleverer argument could recover $1/\varepsilon$ in the agnostic case. It cannot. The gap is fundamental, and the reason is information-theoretic.

Proof sketch of the agnostic lower bound. Consider the class of threshold functions on $[0, 1]$ (VC dimension 1, so $d = 1$ for simplicity). In the realizable case, a single misclassified example pins down the threshold to within ε-precision, requiring $O(1/\varepsilon)$ samples.

In the agnostic case, the distribution D on (X, Y) may have a base error rate $\eta = \min_h R_D(h) > 0$, and the learner must distinguish between two hypotheses h_1, h_2 whose risks differ by ε: $R_D(h_1) = \eta$ and $R_D(h_2) = \eta + \varepsilon$. Distinguishing these requires detecting a difference ε against a background noise level η.

This is a hypothesis testing problem. By standard lower bounds (Le Cam's method or Fano's inequality), distinguishing distributions at total variation distance $O(\varepsilon)$ from m samples requires $m = \Omega(1/\varepsilon^2)$. The noise "contaminates" the signal: in the realizable case, an error is always informative (it eliminates hypotheses), but in the agnostic case, each error might be noise or signal, and separating the two costs the extra $1/\varepsilon$ factor. □

Remark 5.21 (The revelation). The ε vs. ε^2 gap is the first structural surprise of statistical learning theory that cannot be anticipated from finite-class arguments. For finite $|\mathcal{H}|$, both realizable and agnostic sample complexities have the same dependence on ε (up to the $\log|\mathcal{H}|$ factor). The gap emerges only when $\mathcal{H}$ is infinite and VC dimension replaces $\log|\mathcal{H}|$. Realizability is not just a simplifying assumption—it provides a qualitatively different information structure.

5.5 Lower Bounds and the No-Free-Lunch Theorem

5.5.1 The PAC Lower Bound

The upper bound of Theorem 5.7 gives $m = O(d/\varepsilon)$ in the realizable case. The following shows this is tight up to constants.

Theorem 5.22 (PAC Lower Bound)**.** *For any hypothesis class $\mathcal{H}$ with $\mathrm{VCdim}(\mathcal{H}) = d \geq 1$, any PAC learner for $\mathcal{H}$ requires sample complexity*

$$m(\varepsilon, \delta) = \Omega\left(\frac{d}{\varepsilon}\right)$$

for $\varepsilon \leq 1/8$ and $\delta \leq 1/7$.

Proof. Since $\mathrm{VCdim}(\mathcal{H}) = d$, there exists a shattered set $C = \{x_1, \ldots, x_d\}$. For each subset $T \subseteq C$, let $h_T \in \mathcal{H}$ be the hypothesis that labels exactly T as positive. This gives 2^d distinct hypotheses.

Fix $\varepsilon \leq 1/8$. For each $T \subseteq C$ with $|T| = \lfloor d/2 \rfloor$, define the distribution D_T: uniform on C, with target h_T. The learner receives m i.i.d. samples from D_T.

Each sample reveals the label of one point in C. After m samples, the expected number of unseen points in C is $d(1 - 1/d)^m \geq d \cdot e^{-2m/d}$ (for $m \leq d$). For $m \leq d/(8\varepsilon)$, the number of unseen points is at least $d/4$ in expectation.

On each unseen point, the learner must guess the label. Since the target is consistent with both labels for unseen points (by the shattering property), no strategy can beat chance on unseen points. The error on each unseen point contributes $1/d$ to the total risk (since D_T is uniform on C). With at least $d/4$ unseen points in expectation, the expected risk is at least $1/4 > \varepsilon$. A Markov argument converts this to a high-probability statement, showing $m = \Omega(d/\varepsilon)$. □

5.5.2 The No-Free-Lunch Theorem (Full Proof)

Theorem 1.8 in Chapter 1 gave the statement and a brief argument. Here we give the full proof with the averaging argument over all target functions.

Theorem 5.23 (No Free Lunch, Full Version)**.** *Let $|X| \geq 2m$. For any learning algorithm A,*

$$\max_{c \in \{0,1\}^X} \mathbb{E}_{S \sim D_c^m}[R_{D_c}(A(S))] \geq \frac{1}{4},$$

where D_c is the uniform distribution on X with target c.

Proof. Instead of proving the max, we prove the stronger statement with $\mathbb{E}_c$ (expectation over a uniform random target), which implies the max is at least as large.

Fix any algorithm A. Let $S = (x_1, \ldots, x_m)$ be drawn uniformly from X (i.i.d.), and let c be drawn uniformly from $\{0,1\}^X$. Write $T = X \setminus \{x_1, \ldots, x_m\}$ for the unseen points.

Key observation. Conditioned on S and on the labels $(c(x_1), \ldots, c(x_m))$, the labels $\{c(x) : x \in T\}$ are still independent uniform bits. This is because the prior on c is product measure.

Therefore, for each $x \in T$, regardless of what $A(S)$ predicts:

$$\mathbb{P}_c[A(S)(x) \neq c(x) \mid S, c|_S] = \frac{1}{2}.$$

The expected risk on unseen points is:

$$\mathbb{E}_c\left[\frac{1}{|X|}\sum_{x \in T} \mathbf{1}[A(S)(x) \neq c(x)] \;\middle|\; S\right] = \frac{|T|}{|X|} \cdot \frac{1}{2} \geq \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4},$$

using $|T| \geq |X|/2$ (since $m \leq |X|/2$). Taking expectation over S preserves the bound. □

Remark 5.24 (NFL and inductive bias, revisited). The NFL theorem is the structural reason that the VC characterization involves a *restriction* on $\mathcal{H}$. Without restricting to a class of finite VC dimension, no learning is possible. The Fundamental Theorem can therefore be read as: "inductive bias (finite VC dimension) is both necessary and sufficient for statistical learning."

5.6 Computational Interlude

The Fundamental Theorem is an *information-theoretic* characterization: it says when enough data exists to learn, regardless of computational cost. The computational question—when can learning be done in polynomial time—is a separate and largely open problem.

Theorem 5.25 (Computational Hardness [KV94])**.** *Under standard cryptographic assumptions (specifically, the existence of one-way functions), there exist concept classes $\mathcal{C}$ with* $\mathrm{VCdim}(\mathcal{C}) = O(\log n)$ *that are PAC learnable (information-theoretically) but not* efficiently *PAC learnable (no polynomial-time algorithm achieves PAC guarantees).*

This result separates the information-theoretic and computational landscapes of PAC learning. The Fundamental Theorem characterizes the information boundary; it says nothing about the computational one.

Computational Illustration

The gap between information and computation is the subject of Chapter 16. The key examples:

- **Properly learning DNF formulas** is computationally hard under cryptographic

assumptions, even though DNF has polynomial VC dimension.
- **Improperly learning DNF** can be done efficiently via boosting (using a richer hypothesis class).
- **Learning intersections of halfspaces** is hard even improperly, under stronger assumptions.

The proper/improper distinction (Definition 1.2) is computationally sharp even when it is information-theoretically irrelevant.

5.7 What This Chapter Established

The central achievement is the Fundamental Theorem (Theorem 5.16), which says that the following are all the same property of a binary hypothesis class $\mathcal{H}$:

Condition	First established	Proof location
Finite VC dimension	Vapnik–Chervonenkis (1971)	Section 5.2
Uniform convergence	Vapnik–Chervonenkis (1971)	Theorem 5.11
ERM is a PAC learner	Blumer et al. (1989)	Proposition 5.13
PAC learnable	Valiant (1984)	Theorem 5.7
Agnostic PAC learnable	Haussler (1992)	Section 17.3
$\Pi(m) < 2^m$ for some m	Sauer–Shelah (1972)	Section 5.3.1
Finite compression scheme	Moran–Yehudayoff (2016)	Section 5.3.2

Four structural lessons emerge:

1. **VC dimension is the right measure.** Not because it was first, but because it sits at the center of a seven-way equivalence.

2. **The ε^2 price is real.** The agnostic setting costs $1/\varepsilon^2$ where the realizable setting costs $1/\varepsilon$. This is not a proof artifact—it is a lower bound.

3. **ERM is canonical but not unique.** The Fundamental Theorem says ERM works whenever anything works, but other algorithms (compression-based, Bayesian) also achieve PAC guarantees under the same conditions.

4. **Information $\neq$ computation.** The Fundamental Theorem characterizes when learning is *possible*; the Kearns–Valiant barrier shows this does not determine when learning is *efficient*. The computational landscape is the subject of Chapter 16.

Historical Note

Timeline. Vapnik and Chervonenkis introduced the VC dimension and proved the uniform convergence direction in 1971. Valiant formalized PAC learning in 1984, without the VC connection. Blumer, Ehrenfeucht, Haussler, and Warmuth closed the loop in 1989, proving that finite VC dimension is both necessary and sufficient for PAC learning. The compression equivalence was conjectured by Littlestone and Warmuth (1986) and proved by Moran and Yehudayoff (2016)—a 30-year gap that reflects the depth of the combinatorial argument. The tight realizable sample complexity $\Theta(d/\varepsilon)$ (removing logarithmic factors) was established by Hanneke [Han16].

Exercises

1. **VC dimension of convex polygons.** Let $\mathcal{H}_k$ be the class of convex k-gons in $\mathbb{R}^2$: $h_P(x) = 1$ iff x lies inside a convex polygon P with at most k vertices. Prove that $\mathrm{VCdim}(\mathcal{H}_k) = 2k+1$.

   Lower bound: Place $2k+1$ points in convex position (on a circle). For any subset T of these points with $|T| \leq k$, a convex k-gon can be drawn to include exactly T. For larger subsets, use the complement labeling and the observation that $2k + 1 - |T| \leq k$ of the remaining points can be avoided.

   Upper bound: Show that any $2k+2$ points include a labeling unrealizable by a k-gon. Argue that a convex polygon with k vertices can "separate" at most $2k$ contiguous arcs on any circle through the points, and $2k + 2$ points in convex position create $2k + 2$ arcs.

2. **The distribution-free assumption is essential.** The Fundamental Theorem characterizes *distribution-free* PAC learnability. Show that the equivalence breaks if "distribution-free" is replaced by "distribution-dependent."

   (a) Construct a class $\mathcal{H}$ with $\mathrm{VCdim}(\mathcal{H}) = \infty$ and a specific distribution D such that $\mathcal{H}$ is PAC learnable under D with $m(\varepsilon, \delta) = O(1)$ samples.

   (b) More subtly: construct a class $\mathcal{H}$ with $\mathrm{VCdim}(\mathcal{H}) = \infty$ that is PAC learnable under *every* fixed distribution D (with sample complexity depending on D), yet is not distribution-free PAC learnable. *Hint:* Let $\mathcal{H}$ be all measurable functions on $[0, 1]$. For any fixed D, the metric entropy of $\mathcal{H}$ under $L^1(D)$ is finite at every scale—one can ε-net the class with $O(1/\varepsilon)$ functions—so PAC learning under D is possible. The distribution-free failure comes from the adversary's ability to concentrate D on the points where the learner has not yet gathered information.

3. **Tight agnostic lower bound via Assouad's lemma.** Prove the lower bound $m(\varepsilon, \delta) = \Omega(d/\varepsilon^2)$ for agnostic PAC learning of any class $\mathcal{H}$ with $\mathrm{VCdim}(\mathcal{H}) = d$, using Assouad's lemma rather than Le Cam's method.

   Construction: Let $C = \{x_1, \ldots, x_d\}$ be a shattered set. For each $b \in \{0, 1\}^d$, define D_b as the distribution that places mass $1/(2d)$ on each x_i and mass $1/2$ on a "noise point" $z \notin C$ with label drawn from Bernoulli$(1/2)$. The target function labels x_i as b_i and z as 1. The optimal risk under D_b is $1/4$ (from the noise at z).

   Show that any algorithm distinguishing D_b from $D_{b'}$ (where b and b' differ in one coordinate) with advantage ε requires $\Omega(1/\varepsilon^2)$ samples from the signal at a single x_i. Since there are d coordinates, Assouad's lemma gives the combined bound $\Omega(d/\varepsilon^2)$. Verify that this matches the agnostic gap of Theorem 5.20.

Chapter 6

Online Learning and the Littlestone Dimension

In PAC learning, nature is indifferent: training data arrives as a random sample from a fixed distribution, and the learner's task is to generalize from this benign source. In online learning, nature is replaced by an adversary. There is no distribution. There is no training phase followed by a test phase. Instead, learning unfolds as a *game*: at each round, the adversary presents an instance, the learner predicts a label, the adversary reveals the true label, and the game continues. The learner's goal is to bound the total number of mistakes, no matter what the adversary does.

This change—from distribution to adversary, from batch to sequential, from probabilistic to combinatorial—transforms the mathematics entirely. PAC learning is characterized by a number (the VC dimension) through a probabilistic argument (concentration inequalities). Online learning is characterized by a *tree* (the mistake tree) through a combinatorial argument (an explicit algorithm). The key theorem of this chapter—that the optimal mistake bound equals the Littlestone dimension—is proved not by bounding tail probabilities but by *constructing an algorithm* that plays the game optimally.

6.1 The Online Learning Game

Online learning is best understood as a protocol between two players: a *learner* and an *adversary*. The game is played over a domain X, a label set $Y = \{0, 1\}$, and a hypothesis class $\mathcal{H} \subseteq \{0, 1\}^X$. The adversary is constrained to be *realizable*: there must exist some $h^* \in \mathcal{H}$ consistent with all of the adversary's labels. Beyond this, the adversary is unconstrained—in particular, the adversary may be *adaptive*, choosing x_t based on the learner's previous predictions.

Definition 6.1 (Online Learning Protocol)**.** The online learning game proceeds in rounds $t = 1, 2, 3, \ldots$:

1. The adversary selects an instance $x_t \in X$.
2. The learner observes x_t and predicts a label $\hat{y}_t \in \{0, 1\}$.
3. The adversary reveals the true label $y_t \in \{0, 1\}$.
4. If $\hat{y}_t \neq y_t$, the learner incurs a *mistake*.

The game continues for as many rounds as the adversary chooses. The adversary must ensure that there exists $h^* \in \mathcal{H}$ with $h^*(x_t) = y_t$ for all t (realizability).

Three features distinguish this game from the PAC framework of Chapter 5:

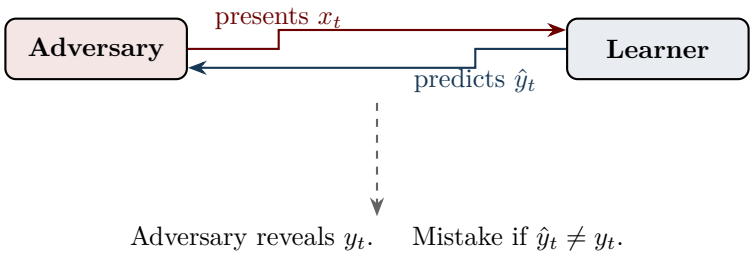

Figure 6.1: The online learning game. The adversary and learner interact sequentially; the adversary is constrained only by realizability. There is no distribution, no random sampling, and no distinction between training and test phases.

1. **No distribution.** In PAC learning, performance is measured with respect to an unknown but fixed distribution D. In online learning, there is no distribution at all. The adversary's choices need not be drawn from any probability measure.

2. **The adversary is adaptive.** The adversary sees the learner's predictions and can choose future instances to exploit the learner's weaknesses. This is strictly harder than random sampling: an adaptive adversary can do everything that a random distribution can, and more.

3. **Performance is worst-case over sequences.** The mistake bound must hold for *every* sequence of instances and labels the adversary might produce, subject to realizability. There is no averaging.

Definition 6.2 (Mistake Bound)**.** A learner A is *mistake-bounded* with bound M on $\mathcal{H}$ if, for every adversary strategy consistent with some $h^* \in \mathcal{H}$, the total number of mistakes made by A is at most M. The *optimal mistake bound* of $\mathcal{H}$ is $\mathrm{Opt}(\mathcal{H}) = \min_A \max_{\text{adversary}} \#\{\text{mistakes of } A\}$.

The central question of online learning theory is: what determines $\mathrm{Opt}(\mathcal{H})$? The answer is a combinatorial object called the Littlestone dimension.

6.2 Mistake Trees and the Littlestone Dimension

The VC dimension counts the size of the largest *set* that $\mathcal{H}$ can shatter. The Littlestone dimension counts the depth of the largest *tree* that $\mathcal{H}$ can shatter. This shift—from sets to trees—is what captures the adversary's adaptive power.

Definition 6.3 (Mistake Tree)**.** A *mistake tree* for $\mathcal{H}$ over X is a complete binary tree T in which:

- Each internal node v is labeled with an instance $x_v \in X$.
- The left child of v corresponds to the label $y = 0$ and the right child to $y = 1$.
- For every root-to-leaf path $(v_1, y_1), (v_2, y_2), \ldots, (v_d, y_d)$, there exists a hypothesis $h \in \mathcal{H}$ consistent with all labels along the path: $h(x_{v_i}) = y_i$ for all i.

The *depth* of the tree is the number of internal nodes on any root-to-leaf path (all paths have the same length since the tree is complete).

The key idea is that a mistake tree encodes an adversary strategy. Starting at the root, the adversary presents x_{v_1}. Whatever the learner predicts, the adversary can label x_{v_1} to make the prediction wrong and descend to the corresponding child. Realizability is maintained because every root-to-leaf path is consistent with some $h \in \mathcal{H}$. Thus a mistake tree of depth d represents an adversary strategy that forces *any* learner to make at least d mistakes.

Definition 6.4 (Littlestone Dimension [Lit88])**.** The *Littlestone dimension* of a hypothesis class $\mathcal{H}$, denoted $\mathrm{Ldim}(\mathcal{H})$, is the maximum depth of a complete mistake tree for $\mathcal{H}$. If arbitrarily deep mistake trees exist, $\mathrm{Ldim}(\mathcal{H}) = \infty$.

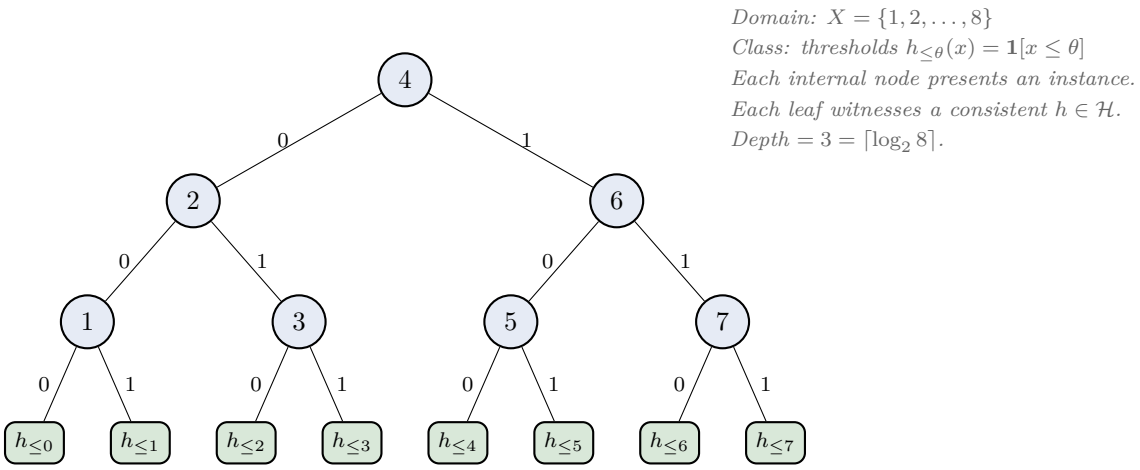

Figure 6.2: A complete mistake tree of depth 3 for the threshold class on $\{1, \ldots, 8\}$. Each internal node is labeled with an instance; the left and right children correspond to labels 0 and 1. Every root-to-leaf path is consistent with the threshold hypothesis shown at the leaf. The adversary can traverse this tree from root to any leaf, forcing 3 mistakes.

Example 6.5 (Littlestone dimension of fundamental classes)**.** (a) **Finite classes.** If $|\mathcal{H}| < \infty$, then $\mathrm{Ldim}(\mathcal{H}) \leq \lfloor \log_2 |\mathcal{H}| \rfloor$, since a complete binary tree of depth d has 2^d leaves and each leaf requires a distinct consistent hypothesis. This bound is tight for classes that are "richly structured enough" (e.g., all functions on a domain of size $\log_2 |\mathcal{H}|$).

(b) **Thresholds on $\{1, \ldots, n\}$.** The class $\{h_{\leq\theta} : \theta \in \{0, 1, \ldots, n\}\}$ has $\mathrm{Ldim} = \lceil \log_2(n+1) \rceil$. The mistake tree in Figure 6.2 illustrates the case $n = 8$. The adversary performs binary search on the threshold.

(c) **Thresholds on $\mathbb{R}$.** Now $\mathrm{Ldim} = \infty$. The key point: the domain is dense, so the adversary can always find a point between any two previous thresholds. At each round, the adversary presents a point that bisects the current interval of uncertainty, forcing a mistake no matter what the learner predicts. This produces mistake trees of unbounded depth.

(d) **Intervals on $\mathbb{R}$.** The class $\{x \mapsto \mathbf{1}[a \leq x \leq b] : a, b \in \mathbb{R}\}$ also has $\mathrm{Ldim} = \infty$. The adversary can adaptively narrow down both endpoints.

(e) **Linear classifiers in $\mathbb{R}^d$.** The class of halfspaces has $\mathrm{Ldim} = d$, matching the VC dimension. This is one of the rare cases where the two dimensions coincide.

Remark 6.6 (Sets vs. trees). Shattering a *set* $\{x_1, \ldots, x_d\}$ means $\mathcal{H}$ can produce all 2^d labelings of these *fixed* points. Shattering a *tree* of depth d means $\mathcal{H}$ can produce a consistent labeling along every root-to-leaf path, but the points themselves may *depend on previous labels*. The tree structure captures adaptivity: the adversary's choice at depth k depends on the learner's

responses at depths $1, \ldots, k-1$. A shattered set is a special case of a mistake tree (one in which every node at the same depth is labeled with the same instance), so $\mathrm{VCdim}(\mathcal{H}) \leq \mathrm{Ldim}(\mathcal{H})$ always holds. The converse fails spectacularly: thresholds on $\mathbb{R}$ have $\mathrm{VCdim} = 1$ but $\mathrm{Ldim} = \infty$.

6.3 The Standard Optimal Algorithm

The Halving algorithm is the simplest reasonable strategy for online learning: maintain the version space (the set of hypotheses consistent with all observations so far), and predict the majority vote of the version space. Each mistake eliminates at least half of the surviving hypotheses, so Halving makes at most $\lfloor \log_2 |\mathcal{H}| \rfloor$ mistakes when $\mathcal{H}$ is finite.

But Halving is not optimal for infinite hypothesis classes. The Standard Optimal Algorithm (SOA), due to Littlestone [Lit88], refines the Halving idea by replacing "majority vote" with a more subtle criterion: predict the label whose version space has the larger *Littlestone dimension*.

Definition 6.7 (Standard Optimal Algorithm (SOA))**.** Given a hypothesis class $\mathcal{H}$, the SOA maintains a version space $V_t \subseteq \mathcal{H}$, initially $V_1 = \mathcal{H}$. At round t, upon receiving instance x_t:

1. Partition V_t into $V_t^0 = \{h \in V_t : h(x_t) = 0\}$ and $V_t^1 = \{h \in V_t : h(x_t) = 1\}$.
2. Predict $\hat{y}_t = \arg\max_{b \in \{0,1\}} \mathrm{Ldim}(V_t^b)$.
3. After observing the true label y_t, update $V_{t+1} = V_t^{y_t}$.

Ties in step 2 are broken arbitrarily.

The intuition behind the SOA is a tree-based version of binary search. At each round, the algorithm asks: "If I predict 0 and am wrong, what is the worst the adversary can do to V_t^1? And if I predict 1 and am wrong, what can the adversary do to V_t^0?" By choosing the side with the larger Littlestone dimension, the SOA ensures that every mistake reduces the Littlestone dimension of the version space by at least one.

Theorem 6.8 (Upper Bound: SOA achieves $\mathrm{Ldim}(\mathcal{H})$ mistakes [Lit88])**.** *For any hypothesis class $\mathcal{H}$ with $\mathrm{Ldim}(\mathcal{H}) = d < \infty$, the* SOA *makes at most d mistakes against any adversary.*

Proof. We show that the Littlestone dimension of the version space drops by at least one at each mistake. Define $d_t = \mathrm{Ldim}(V_t)$. We claim:

$$\text{If the SOA makes a mistake at round } t, \text{ then } d_{t+1} \leq d_t - 1. \tag{6.1}$$

Proof of the claim. Suppose the SOA makes a mistake at round t. This means $\hat{y}_t \neq y_t$. By the algorithm's rule, $\hat{y}_t$ was chosen so that $\mathrm{Ldim}(V_t^{\hat{y}_t}) \geq \mathrm{Ldim}(V_t^{y_t})$, i.e., the SOA predicted the side with the *larger* (or equal) Littlestone dimension. After the mistake, the version space updates to $V_{t+1} = V_t^{y_t}$, the side with the *smaller* (or equal) Littlestone dimension.

Now we must show that $\mathrm{Ldim}(V_t^{y_t}) \leq d_t - 1$. Suppose for contradiction that $\mathrm{Ldim}(V_t^0) \geq d_t$ and $\mathrm{Ldim}(V_t^1) \geq d_t$. Then there exists a complete mistake tree T_0 of depth d_t for V_t^0 and a complete mistake tree T_1 of depth d_t for V_t^1. We can construct a complete mistake tree of depth $d_t + 1$ for V_t: the root is labeled x_t, its left subtree (corresponding to label 0) is T_0, and its right subtree (corresponding to label 1) is T_1. Every root-to-leaf path through the left subtree is consistent with some $h \in V_t^0 \subseteq V_t$ (and $h(x_t) = 0$), and similarly for the right subtree. This yields $\mathrm{Ldim}(V_t) \geq d_t + 1$, contradicting $\mathrm{Ldim}(V_t) = d_t$.

Therefore $\min\{\mathrm{Ldim}(V_t^0), \mathrm{Ldim}(V_t^1)\} \leq d_t - 1$. Since $\hat{y}_t$ selects the side with the larger dimension, $V_{t+1} = V_t^{y_t}$ is the side with the smaller dimension, so $d_{t+1} = \mathrm{Ldim}(V_{t+1}) \leq d_t - 1$.

Completing the proof. We have $d_1 = \mathrm{Ldim}(\mathcal{H}) = d$. Each mistake decreases d_t by at least 1. Since the Littlestone dimension is a non-negative integer (the version space always contains the target h^*, so $V_t \neq \emptyset$ and $\mathrm{Ldim}(V_t) \geq 0$), the total number of mistakes is at most d. □

Remark 6.9 (SOA vs. Halving). For finite $\mathcal{H}$, the Halving algorithm predicts by majority vote: $\hat{y}_t = \arg\max_b |V_t^b|$. Each mistake halves the version space, giving at most $\lfloor \log_2 |\mathcal{H}| \rfloor$ mistakes. The SOA replaces cardinality $|V_t^b|$ with $\mathrm{Ldim}(V_t^b)$. For finite classes, this distinction is often unimportant (both give $O(\log |\mathcal{H}|)$ mistakes), but for infinite classes, cardinality is meaningless while the Littlestone dimension is finite and well-behaved.

Remark 6.10 (Computability of the SOA). The SOA is an *information-theoretically* optimal algorithm, but it is not necessarily *computationally* efficient. Computing $\mathrm{Ldim}(V_t^b)$ at each round may be undecidable for some hypothesis classes. The SOA is thus an existence proof: it shows that d mistakes suffice, even if finding the optimal prediction at each step is computationally hard. Efficient online algorithms for specific classes (Perceptron for halfspaces, Winnow for disjunctions) achieve near-optimal mistake bounds through class-specific structure.

6.4 The Lower Bound: The Adversary Strategy

The upper bound shows that the SOA can limit mistakes to $\mathrm{Ldim}(\mathcal{H})$. The lower bound shows that no learner can do better: for any learner, there exists an adversary that forces at least $\mathrm{Ldim}(\mathcal{H})$ mistakes.

Theorem 6.11 (Lower Bound: Any learner makes at least $\mathrm{Ldim}(\mathcal{H})$ mistakes [Lit88])**.** *For any hypothesis class $\mathcal{H}$ with $\mathrm{Ldim}(\mathcal{H}) = d$ and any (possibly randomized) learner A, there exists an adversary strategy that forces A to make at least d mistakes.*

Proof. Let T be a complete mistake tree for $\mathcal{H}$ of depth d. The adversary plays T as follows.

The adversary maintains a pointer to a node of T, starting at the root. At round t, the adversary presents the instance x_{v_t} labeling the current node v_t. The learner predicts $\hat{y}_t$. The adversary then sets $y_t = 1 - \hat{y}_t$ (the opposite of the learner's prediction) and descends to the child of v_t corresponding to label y_t.

This strategy has two properties:

1. **Every round is a mistake.** By construction, $y_t = 1 - \hat{y}_t \neq \hat{y}_t$.
2. **Realizability is maintained.** After d rounds, the adversary has descended from the root to a leaf of T, tracing a path $(v_1, y_1), (v_2, y_2), \ldots, (v_d, y_d)$. By the definition of a mistake tree, there exists $h^* \in \mathcal{H}$ consistent with all labels along this path: $h^*(x_{v_i}) = y_i$ for all $i \in \{1, \ldots, d\}$. For any subsequent rounds $t > d$, the adversary can label consistently with this h^*.

The adversary forces exactly d mistakes in the first d rounds, so A makes at least d mistakes.

For randomized learners, the argument extends by the minimax theorem (or directly): for any distribution over predictions $\hat{y}_t$, the adversary's strategy of playing the opposite forces an expected mistake at every round. Alternatively, one can apply Yao's minimax principle: the worst-case deterministic adversary against the best randomized learner equals the best deterministic learner against the worst-case adversary, and we have shown the latter is at least d. □

Combining **??** 6.8**??** 6.11, we obtain the fundamental characterization.

Theorem 6.12 (Littlestone's Characterization [Lit88])**.** *For any hypothesis class $\mathcal{H}$:*

$$\mathrm{Opt}(\mathcal{H}) = \mathrm{Ldim}(\mathcal{H}).$$

That is, the optimal mistake bound of $\mathcal{H}$ in the online learning game is exactly the Littlestone dimension of $\mathcal{H}$. In particular, $\mathcal{H}$ admits a finite mistake bound if and only if $\mathrm{Ldim}(\mathcal{H}) < \infty$.

Historical Note

Littlestone proved this characterization in 1988 [Lit88], just four years after Valiant's PAC model. The result established online learning as a mathematically independent paradigm: the combinatorial quantity governing online learnability is genuinely different from the one governing PAC learnability. The SOA is sometimes called the "Standard Optimal Algorithm" precisely because it achieves the information-theoretic optimum; the name is due to the online learning community's convention.

6.5 Regret Bounds and the Multiplicative Weights Framework

The mistake-bound framework requires that the adversary be realizable: some $h^* \in \mathcal{H}$ must be consistent with all labels. This is a strong assumption. What happens when we drop it?

In the *regret* framework, the adversary is unrestricted. The learner is compared not to the truth but to the *best fixed hypothesis in hindsight*.

Definition 6.13 (Regret)**.** Given a sequence $(x_1, y_1), \ldots, (x_T, y_T)$ and the learner's predictions $\hat{y}_1, \ldots, \hat{y}_T$, the *regret* of the learner relative to $\mathcal{H}$ is

$$\mathrm{Regret}_T = \sum_{t=1}^{T} \mathbf{1}[\hat{y}_t \neq y_t] - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbf{1}[h(x_t) \neq y_t].$$

This definition contains a conceptual surprise: the regret can be small even when the learner makes many mistakes. What matters is not the absolute number of errors, but how the learner's error count compares to the best fixed hypothesis in $\mathcal{H}$. If every hypothesis in $\mathcal{H}$ also makes many mistakes (because the adversary is not realizable), then a learner with many mistakes but low regret is performing well.

Remark 6.14 (Mistake bound vs. regret bound). In the realizable case, the best hypothesis h^* makes zero mistakes, so Regret_T equals the total mistake count. The regret framework is therefore a strict generalization of the mistake-bound framework. The regret formulation becomes essential when $\mathcal{H}$ is infinite or when realizability fails—precisely the settings where the Littlestone dimension may be infinite and the mistake-bound framework provides no guarantee.

The Weighted Majority algorithm of Littlestone and Warmuth [Lit88] achieves the following regret bound for finite $\mathcal{H}$.

Theorem 6.15 (Weighted Majority / Hedge)**.** *For a finite hypothesis class $\mathcal{H}$ with $|\mathcal{H}| = N$, the Weighted Majority algorithm achieves, for any sequence of T rounds:*

$$\mathrm{Regret}_T \leq 2\sqrt{T \ln N}.$$

In particular, the per-round regret $\mathrm{Regret}_T/T \to 0$ as $T \to \infty$.

The algorithm maintains a weight $w_t(h)$ for each $h \in \mathcal{H}$, initially $w_1(h) = 1$. At each round: predict the weighted majority vote; after observing y_t, multiply the weight of each hypothesis that erred by a factor $(1 - \eta)$ for a learning rate $\eta \in (0, 1)$. Setting $\eta = \sqrt{\ln N / T}$ (or using the doubling trick when T is unknown) yields the bound above.

This approach generalizes to the *multiplicative weights* framework (also called Hedge), which extends beyond binary prediction to decision-making over finitely many "experts." The framework is one of the most widely applicable algorithmic paradigms in theoretical computer science, with applications to zero-sum games, boosting, and linear programming [SSBD14].

Remark 6.16 (Connection to Littlestone dimension). Ben-David, Pál, and Shalev-Shwartz [BDPSS09] showed that for infinite hypothesis classes, the Littlestone dimension also governs the optimal regret rate. Specifically, the minimax regret over T rounds is $\Theta(\sqrt{\mathrm{Ldim}(\mathcal{H}) \cdot T})$ in the agnostic (non-realizable) setting. This connects the combinatorial tree structure of the Littlestone dimension to the sequential prediction framework even beyond realizability.

6.6 The PAC–Online Gap

The VC dimension and the Littlestone dimension both measure the complexity of a hypothesis class, but they measure different things. How do they relate?

Proposition 6.17 (VCdim $\leq$ Ldim)**.** *For any hypothesis class $\mathcal{H}$,* $\mathrm{VCdim}(\mathcal{H}) \leq \mathrm{Ldim}(\mathcal{H})$.

Proof. Let $S = \{x_1, \ldots, x_d\}$ be a set shattered by $\mathcal{H}$, so $\mathrm{VCdim}(\mathcal{H}) \geq d$. We construct a complete mistake tree of depth d. The root is labeled x_1. Every node at depth k is labeled x_{k+1} (the same instance at every node of a given depth). For any root-to-leaf path with labels $(y_1, \ldots, y_d)$, shattering guarantees that some $h \in \mathcal{H}$ satisfies $h(x_i) = y_i$ for all i. Thus this is a valid mistake tree of depth d, so $\mathrm{Ldim}(\mathcal{H}) \geq d$. □

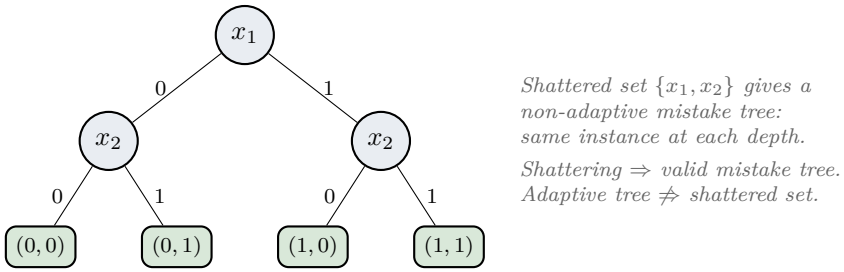

Figure 6.3: A shattered set of size 2 yields a (non-adaptive) mistake tree of depth 2. The instances at each depth are identical, so the tree does not exploit adaptivity. This construction proves $\mathrm{VCdim}(\mathcal{H}) \leq \mathrm{Ldim}(\mathcal{H})$.

The inequality VCdim $\leq$ Ldim is often tight (e.g., for halfspaces, where both equal the ambient dimension). But the gap can be infinite.

Proposition 6.18 (The PAC–Online separation: thresholds on $\mathbb{R}$)**.** *Let* $\mathcal{H}_{\mathrm{thr}} = \{x \mapsto \mathbf{1}[x \geq \theta] : \theta \in \mathbb{R}\}$. *Then* $\mathrm{VCdim}(\mathcal{H}_{\mathrm{thr}}) = 1$ *and* $\mathrm{Ldim}(\mathcal{H}_{\mathrm{thr}}) = \infty$.

Proof. VCdim = 1: Any single point x is shattered (take thresholds $\theta < x$ and $\theta > x$). No two points $x_1 < x_2$ are shattered, because $h(x_1) = 1, h(x_2) = 0$ requires $x_1 \geq \theta > x_2$, which contradicts $x_1 < x_2$.

Ldim = ∞: We construct mistake trees of arbitrary depth. For any d, consider the domain restricted to $\{1, 2, \ldots, 2^d\}$. The adversary can perform binary search: at depth k, present the midpoint of the current interval. Whatever the learner predicts, the adversary labels to force a mistake (and narrows the interval by half). After d rounds, the adversary has traced a path from root to leaf, consistent with the threshold at the boundary of the final interval. Since d was arbitrary, Ldim = ∞. □

This separation is the most important non-implication in the concept graph. It witnesses the edge pac_learning $\xrightarrow{\texttt{does\_not\_imply}}$ online_learning: a class can be PAC learnable (finite VC dimension) yet not online learnable (infinite Littlestone dimension). The converse implication does hold: Ldim $< \infty$ implies VCdim $< \infty$ (by VCdim $\leq$ Ldim), so online learnability implies PAC learnability.

Separation Result

PAC $\not\Rightarrow$ Online. Witness: $\mathcal{H}_{\mathrm{thr}}$ on $\mathbb{R}$. The adversary in the online game can exploit the *order structure* of $\mathbb{R}$ by performing binary search on the threshold. A random sample from a fixed distribution cannot exploit this ordering—with high probability, the sample reveals the threshold's location to within ε. The gap between PAC and online learnability

is not a technicality: it reflects a fundamental difference between random and adversarial data.

Remark 6.19 (Why the gap arises). In PAC learning, the distribution D is fixed before the game begins; the learner faces a *static* environment. In online learning, the adversary *adapts* to the learner's behavior. Adaptivity is the source of the gap. A threshold on $\mathbb{R}$ is easy to learn from random data (one sample suffices, approximately) but impossible to learn from adversarial data (the adversary can always present a point that bisects the remaining uncertainty). The Littlestone dimension measures the depth of the adversary's adaptive search, which can exceed the VC dimension's non-adaptive combinatorics by an arbitrary amount. A full treatment of this and related separations appears in Chapter 14.

6.7 What This Chapter Established

The online learning model replaces PAC's probabilistic framework with a game between learner and adversary. The key structural results are:

1. **The Littlestone characterization.** The optimal mistake bound for online learning equals the Littlestone dimension $\mathrm{Ldim}(\mathcal{H})$—the maximum depth of a complete mistake tree. This is a *combinatorial* characterization, in contrast to the VC dimension's *set-combinatorial* characterization of PAC learning.

2. **The SOA algorithm.** The upper bound is constructive: the Standard Optimal Algorithm achieves $\mathrm{Ldim}(\mathcal{H})$ mistakes by predicting the label whose version space has the larger Littlestone dimension. Each mistake provably reduces the Littlestone dimension of the version space by at least one.

3. **The adversary strategy.** The lower bound is also constructive: the adversary traverses a complete mistake tree from root to leaf, labeling each instance to contradict the learner's prediction.

4. **The PAC–Online gap.** $\mathrm{VCdim}(\mathcal{H}) \leq \mathrm{Ldim}(\mathcal{H})$ always, but the gap can be infinite (thresholds: $\mathrm{VCdim} = 1$, $\mathrm{Ldim} = \infty$). Online learnability implies PAC learnability, but not conversely.

5. **The regret framework.** Regret bounds—measuring performance relative to the best fixed hypothesis—extend online learning beyond the realizable setting. The Littlestone dimension governs optimal regret rates even in the agnostic case.

The combinatorial and algorithmic flavor of this chapter is not accidental. Online learning theory is fundamentally about trees and algorithms, where PAC learning theory is about sets and probabilities. Both paradigms measure the complexity of the same object (a hypothesis class $\mathcal{H}$), but through different instruments—and the instruments are not interchangeable.

Chapter 7

Identification in the Limit

In 1967, seventeen years before Valiant introduced PAC learning, E. Mark Gold posed a question that remains the oldest open research programme in computational learning theory: which classes of recursive functions can a machine identify from examples?

Gold's framework differs from PAC and online learning not merely in its definitions but in its *proof technique*. PAC theory rests on concentration inequalities—the probabilistic convergence of empirical risk to true risk. Online learning rests on combinatorial game trees—the Littlestone dimension measures the depth of a binary tree that the adversary can force. Gold-style identification rests on *diagonalization*—the recursion-theoretic technique of defeating every candidate learner by constructing an adversarial input that forces failure. The proof method shapes the entire chapter.

Three features distinguish this paradigm from everything that precedes it in this book:

1. **Infinite horizon.** The learner receives an infinite stream of data and must eventually converge. There is no sample complexity bound, no ε, no δ. The question is whether the learner converges, not how fast.

2. **Ordinal-valued complexity.** When we ask "how many times does the learner change its mind?" the answer is not an integer but a *transfinite ordinal*. This is the first appearance of ordinals in learning theory.

3. **A lattice of success criteria.** PAC learning has one definition (modulo agnostic, realizable, improper variants). Gold-style identification has a rich hierarchy: $\mathbf{FIN} < \mathbf{Ex} < \mathbf{BC}$, with anomalous, monotonic, and vacillatory learning branching off. The hierarchy itself is mathematical content.

Historical Note

Gold's 1967 paper [Gol67] was preceded by his 1965 paper on limiting recursion [Gol65], which introduced the idea that a computation could "converge in the limit" even if it never halts in the classical sense. The 1967 paper applied this idea to language learning. It was the first mathematical theory of learning from examples—predating Valiant's PAC model by 17 years, Vapnik and Chervonenkis's foundational work on uniform convergence by 4 years, and Littlestone's online model by 21 years. The diagonalization technique Gold used to prove his impossibility theorem later became standard in computational complexity theory, but Gold's use predates most of those applications.

7.1 Gold's Question

Fix a countable domain. In Gold's original setting, this is the set of all strings over a finite alphabet Σ, and the objects to be learned are formal languages $L \subseteq \Sigma^*$. We work in this setting

when it clarifies the recursion-theoretic content, and in the general setting of concept classes $\mathcal{C} \subseteq 2^{\mathbb{N}}$ when it does not matter.

The learner receives data sequentially, one datum at a time, forever. Two data presentation modes are standard:

Definition 7.1 (Text and Informant)**.** Let $L \subseteq \Sigma^*$ be a language.

- A *text* for L is an infinite sequence $t = (t_0, t_1, t_2, \ldots)$ with $\{t_i : i \in \mathbb{N}\} = L$. Every element of L appears at least once; no element of $\Sigma^* \setminus L$ ever appears. The text may contain repetitions and need not follow any fixed ordering.
- An *informant* for L is an infinite sequence of labelled pairs $((s_0, b_0), (s_1, b_1), \ldots)$ where $\{s_i : i \in \mathbb{N}\} = \Sigma^*$ and $b_i = 1$ if and only if $s_i \in L$. Every string is eventually presented together with its membership status.

Text presents only positive examples; informant presents both positive and negative examples. This distinction matters enormously: Gold showed that informant is strictly more powerful than text. We focus on text, which is the harder and more interesting case.

After seeing the first n data points $(t_0, \ldots, t_{n-1})$, the learner outputs a hypothesis h_n. The hypothesis is an *index*—a natural number naming a program that computes a language. The learner has no deadline, no accuracy target, and no probability distribution on targets. It simply sees more and more of L and must eventually guess correctly.

7.2 Ex-Learning and Gold's Impossibility Theorem

Definition 7.2 (Explanatory Learning (**Ex**))**.** A learner M **Ex**-*identifies* a language L from text if, for every text t for L, there exists an index e and a time n_0 such that $M(t_0, \ldots, t_n) = e$ for all $n \geq n_0$, and $W_e = L$ (where W_e is the language computed by program e).

A class $\mathcal{L}$ of languages is **Ex**-*identifiable* if there exists a single learner M that **Ex**-identifies every $L \in \mathcal{L}$ from text.

The definition requires *syntactic convergence*: the learner must eventually output the same index forever. It is not enough to output a sequence of indices that all happen to compute the correct language—the index itself must stabilize. (Relaxing this requirement gives BC-learning; see Section 7.3.2.)

Example 7.3 (**Ex**-identification of finite languages)**.** Let $\mathcal{L}_{\text{fin}}$ be the class of all finite subsets of $\mathbb{N}$. Here is an **Ex**-learner for $\mathcal{L}_{\text{fin}}$: at time n, output an index for $\{t_0, \ldots, t_n\}$. After all elements of L have appeared (which happens at some finite time n_0, since L is finite), the set $\{t_0, \ldots, t_n\}$ equals L for all $n \geq n_0$. The hypothesis stabilizes.

This simple algorithm illustrates a crucial asymmetry: the learner does not know *when* it has converged. It has no way to announce "I am done." It merely stabilizes, silently. This lack of a convergence signal is what makes Gold's impossibility theorem possible.

Theorem 7.4 (Gold's Impossibility Theorem [Gol67])**.** *Let $\mathcal{L}$ be any class of languages that contains all finite languages and at least one infinite language. Then $\mathcal{L}$ is not* **Ex**-*identifiable from text.*

This is the centerpiece of the chapter. The proof is a diagonalization argument: we defeat *every* candidate learner by constructing a text that forces it to fail. The construction is adversarial, not probabilistic—there is no distribution, no ε-net, no covering number. The adversary simply watches the learner and manipulates the future of the stream to prevent convergence.

Proof. Let M be any learner. We construct a text t for some language $L \in \mathcal{L}$ such that M fails to **Ex**-identify L on t.

We build t in stages. Let $L_\infty \in \mathcal{L}$ be the infinite language that $\mathcal{L}$ contains by assumption. Enumerate $L_\infty = \{w_0, w_1, w_2, \ldots\}$.

Stage 0. Present w_0 to M. The learner outputs some hypothesis $h_0 = M(w_0)$.

Stage $k \geq 1$. At this point we have presented the finite sequence $(w_0, \ldots, w_{n_{k-1}})$ for some n_{k-1}, and M has output hypothesis $h_{n_{k-1}}$. Let $F_k = \{w_0, \ldots, w_{n_{k-1}}\}$ be the set of strings presented so far.

Consider two cases:

- **Case A:** $W_{h_{n_{k-1}}} = F_k$ (the current hypothesis computes exactly the finite set seen so far). Then present $w_{n_{k-1}+1}$ (the next element of L_∞), enlarging F_k. Let $n_k = n_{k-1} + 1$. Proceed to stage $k + 1$.
- **Case B:** $W_{h_{n_{k-1}}} \neq F_k$. Then the current hypothesis is already wrong for the finite language F_k. We may define $L = F_k$: since F_k is finite and $\mathcal{L}$ contains all finite languages, $F_k \in \mathcal{L}$. We extend the text by repeating elements of F_k forever. The learner M has output a hypothesis $h_{n_{k-1}}$ with $W_{h_{n_{k-1}}} \neq F_k = L$, and the text for L can be arranged so that M never converges to a correct index (we continue to the next stage rather than stopping, as shown below).

The key observation is the *diagonalization*: we never let the learner rest.

If Case A occurs at every stage, then every element of L_∞ is eventually presented, so t is a text for L_∞. At each stage, M's current hypothesis h_{n_k} is checked: does $W_{h_{n_k}} = F_k$? If yes, we expand. But $F_k \subsetneq L_\infty$ for all k (since L_∞ is infinite), so $W_{h_{n_k}} = F_k \neq L_\infty$. Therefore M outputs a hypothesis that is wrong at every stage—it cannot converge to a correct index for L_∞.

If Case B occurs at some stage k, then M has already failed for F_k. We commit to $L = F_k$ and repeat elements of F_k forever. But we can do more: before committing, we wait to see if M changes its mind. If M later outputs a hypothesis h' with $W_{h'} = F_k$, we switch to Case A and add the next element of L_∞. This forces M to fail for L_∞ instead.

In either case, M fails. Since M was an arbitrary learner, no learner **Ex**-identifies $\mathcal{L}$. □

The proof has a recursive structure that repays careful study. The adversary maintains a "threat" at every stage: either the current language is the finite set seen so far, or it is the infinite language L_∞. Whenever the learner commits to one possibility, the adversary switches to the other. The learner is forced to change its mind infinitely often—and **Ex**-identification requires that it eventually stop changing its mind.

Remark 7.5 (Diagonalization vs. concentration). Compare this proof to the lower bound for PAC learning (Chapter 5). The PAC lower bound constructs a *distribution* that fools the learner with positive probability; it is probabilistic. Gold's proof constructs a *text* that fools the learner with certainty; it is adversarial. The PAC proof uses counting (how many hypotheses can be distinguished by m samples); Gold's proof uses the halting problem implicitly (the adversary must check whether $W_e = F$ for arbitrary e). These are entirely different mathematical worlds.

Historical Note

The proof technique of Theorem 7.4 is a *priority argument* in the sense of recursion theory, though a simple one. More sophisticated priority arguments appear in the study of learning with resource bounds (Case and Smith [CS83]). The connection between Gold-style learning and recursion theory runs deep: Barzdinš [Bar74] showed that the class of languages **Ex**-identifiable from informant is exactly the class of Σ_2^0-definable families, establishing a precise link to the arithmetical hierarchy.

7.3 The Identification Hierarchy

Gold's impossibility theorem says that **Ex**-identification is limited. Two natural responses: strengthen the success criterion (demand faster convergence) or weaken it (demand less of the final hypothesis). Both directions are fruitful.

7.3.1 Finite Identification

Definition 7.6 (Finite Identification (**FIN**))**.** A learner M **FIN**-*identifies* a language L from text if, for every text t for L, there exists a time n_0 such that M outputs exactly one hypothesis—M outputs "?" (no guess) for $n < n_0$ and outputs a single index e at time n_0 with $W_e = L$, never changing its mind. A class $\mathcal{L}$ is **FIN**-*identifiable* if a single learner **FIN**-identifies every $L \in \mathcal{L}$.

FIN is the strongest identification criterion: the learner must produce the correct answer in finitely many steps with no subsequent revision. The class of **FIN**-identifiable families is a strict subset of **Ex**-identifiable families; for instance, the class of all finite languages is **Ex**-identifiable (Example 7.3) but not **FIN**-identifiable, since the learner cannot know when the last element has been seen.

7.3.2 Behaviorally Correct Learning

Definition 7.7 (Behaviorally Correct Learning (**BC**))**.** A learner M **BC**-*identifies* a language L from text if, for every text t for L, there exists a time n_0 such that $W_{M(t_0,\ldots,t_n)} = L$ for all $n \geq n_0$. That is, from time n_0 onward, every hypothesis the learner outputs is *extensionally correct* (computes the right language), but the indices may keep changing.

The distinction between **Ex** and **BC** is subtle but consequential. **Ex** requires *syntactic* convergence: the index stabilizes. **BC** requires only *semantic* convergence: the computed language stabilizes, even if the learner keeps switching between different programs that all compute the same language. At first glance, this seems like a minor relaxation. It is not.

Theorem 7.8 (Case–Smith [CS83])**.** **BC** *is strictly more powerful than* **Ex***: there exists a class of languages that is* **BC**-*identifiable but not* **Ex**-*identifiable from text.*

Proof. We construct a class $\mathcal{L}$ that separates **BC** from **Ex**.

For each total recursive function $f : \mathbb{N} \to \mathbb{N}$, define the language $L_f = \{\langle n, f(n)\rangle : n \in \mathbb{N}\}$, where $\langle \cdot, \cdot \rangle$ is a computable pairing function. Let $\mathcal{L} = \{L_f : f \text{ is total recursive}\}$.

$\mathcal{L}$ is BC-identifiable. Given a text t for some L_f, at time n the learner has seen finitely many pairs $\langle n_i, m_i\rangle$. Define the partial function g_n by $g_n(n_i) = m_i$ for all pairs seen so far, and $g_n(k) = 0$ for all other k. Output an index for the language $L_{g_n} = \{\langle k, g_n(k)\rangle : k \in \mathbb{N}\}$. Once all pairs $\langle k, f(k)\rangle$ for $k \leq K$ have appeared (for any K), the hypothesis computes L_f correctly on $\{0, \ldots, K\}$. As $n \to \infty$, g_n converges pointwise to f. Since f is total, for every k there exists a time after which $g_n(k) = f(k)$. The language computed by the hypothesis is eventually L_f, though the *index* keeps changing as new pairs are absorbed. This is **BC**-identification.

$\mathcal{L}$ is not Ex-identifiable. Suppose for contradiction that a learner M **Ex**-identifies $\mathcal{L}$. We diagonalize against M. Define a total recursive function f as follows.

Begin presenting pairs $\langle 0, 0\rangle, \langle 1, 0\rangle, \langle 2, 0\rangle, \ldots$ (corresponding to the zero function). Wait until M converges to some index e. Since M **Ex**-identifies $\mathcal{L}$ and the zero function is total recursive, M must converge. Now $W_e = L_{\mathbf{0}}$ (the language of the zero function).

Modify f: set $f(k) = 1$ for some large k not yet presented. This changes the target to a different total recursive function, but the text presented so far is consistent with both targets. The learner M has already committed to index e, which computes $L_{\mathbf{0}} \neq L_f$. By a careful

effectivization of this argument (choosing k computably based on M's behavior), we construct $f \in \mathcal{L}$ on which M fails. This contradicts M **Ex**-identifying all of $\mathcal{L}$.

The essential point is that **BC**-identification does not require the *index* to stabilize, only the *extension*. The class $\mathcal{L}$ exploits this: the learner must continually update its program as new pairs arrive, and the programs keep changing, but they all eventually compute the same language. **Ex**-identification cannot tolerate this perpetual updating of indices. □

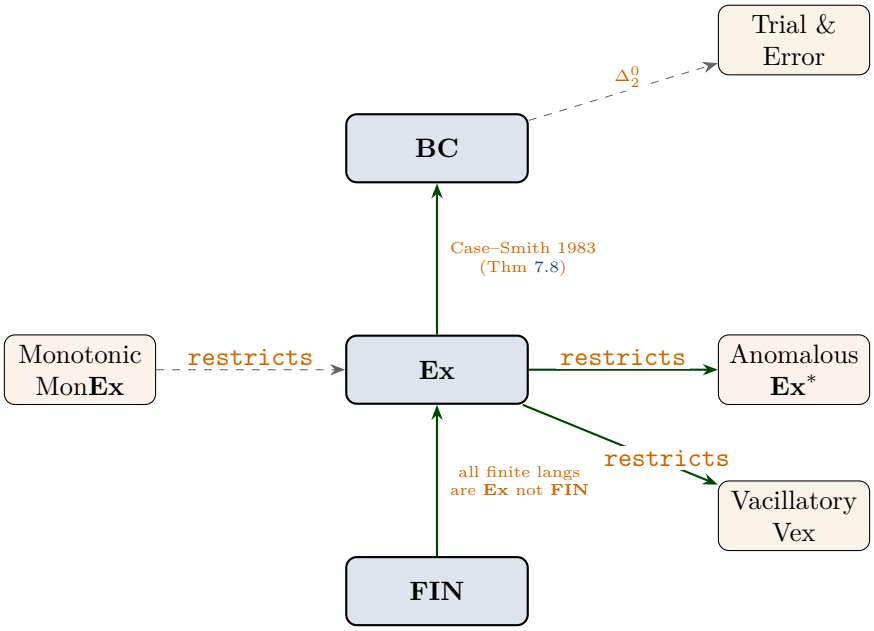

Figure 7.1: The identification hierarchy. Solid arrows indicate strict inclusion of identifiable classes (more hypotheses identified at the target). **FIN** $\subsetneq$ **Ex** $\subsetneq$ **BC**, with witnesses on each edge. Anomalous learning (**Ex***) extends **Ex** by removing the zero-error constraint. Monotonic learning restricts **Ex** by forbidding hypothesis retraction. Vacillatory learning sits between **Ex** and **BC**.

Graph Traversal

Path: fin_learning $\xrightarrow{\texttt{strictly\_stronger}}$ ex_learning $\xrightarrow{\texttt{strictly\_stronger}}$ bc_learning.

Each arrow is witnessed by an explicit separation. The graph encodes these as `strictly_stronger` edges with the separation witness as metadata. The hierarchy is not a sequence of increasingly liberal definitions—it is a chain of *theorems*, each proved by constructing a class that one criterion can learn and the other cannot.

7.4 Relaxations of Identification

The **FIN**–**Ex**–**BC** chain is the spine of the hierarchy. Several natural relaxations branch off from **Ex**, each modifying a different aspect of the convergence requirement.

7.4.1 Anomalous Learning

Definition 7.9 (Anomalous Learning (**Ex***))**.** A learner **Ex****-identifies* a language L if it **Ex**-converges to an index e such that W_e differs from L on at most finitely many strings. That is, the final hypothesis may contain finitely many errors—finitely many strings incorrectly included or excluded.

Anomalous learning relaxes **Ex** by removing the requirement of exact correctness, replacing it with correctness up to a finite set. In graph terms, this is a `restricts` edge from anomalous_learning to ex_learning: the constraint (zero anomalies) is removed, and no new grammatical structure is introduced. The class of $\mathbf{Ex}^*$-identifiable families strictly contains the class of **Ex**-identifiable families [CS83].

7.4.2 Monotonic Learning

Definition 7.10 (Monotonic Learning (Mon**Ex**))**.** A learner *monotonically* **Ex**-identifies L if it **Ex**-identifies L and, whenever it changes its hypothesis from e to e', the new hypothesis is at least as inclusive on the data seen so far: $W_e \cap \{t_0, \ldots, t_n\} \subseteq W_{e'} \cap \{t_0, \ldots, t_n\}$. Once a datum is correctly classified, that classification is never retracted.

Monotonic learning *strengthens* **Ex** by adding a constraint. The class of Mon**Ex**-identifiable families is a strict subset of **Ex**-identifiable families. In the graph, monotonic_learning $\xrightarrow{\texttt{restricts}}$ ex_learning.

7.4.3 Vacillatory Learning

Definition 7.11 (Vacillatory Learning (Vex))**.** A learner *vacillatorily* identifies L if it eventually oscillates among finitely many indices $e_1, \ldots, e_k$, all satisfying $W_{e_i} = L$. The learner never settles on a single index, but all its eventual outputs are extensionally correct and drawn from a finite set.

Vacillatory learning sits between **Ex** (which requires convergence to a single index) and **BC** (which allows infinitely many correct indices). It is strictly more powerful than **Ex** and strictly less powerful than **BC**.

7.4.4 Trial and Error

Definition 7.12 (Trial-and-Error Learning)**.** A *trial-and-error* predicate for a set $A \subseteq \mathbb{N}$ is a computable function $f : \mathbb{N} \to \{0, 1\}$ such that $\lim_{s\to\infty} f_s(n)$ exists for all n and equals the characteristic function of A. The learner may "retract" previous outputs: it learns by making mistakes and correcting them.

Trial-and-error learning connects identification in the limit to the arithmetical hierarchy. A set A is trial-and-error learnable if and only if $A \in \Delta^0_2$—the class of sets that are both Σ^0_2 and Π^0_2. This is the precise recursion-theoretic characterization, due to Putnam [Gol65] and Gold: the limit-computable functions are exactly the Δ^0_2 functions. This connection places Gold-style learning firmly within the landscape of classical recursion theory.

7.5 Mind-Change Complexity

Gold's impossibility theorem tells us that certain classes cannot be identified at all. For classes that *can* be identified, a natural quantitative question arises: how many times must the learner change its mind before converging?

Definition 7.13 (Mind Change)**.** A *mind change* occurs at time n if the learner's output satisfies $M(t_0, \ldots, t_n) \neq M(t_0, \ldots, t_{n-1})$. The *mind-change count* of M on text t is the number of times M changes its hypothesis.

For finite mind-change bounds, the theory is straightforward: we say that M identifies $\mathcal{L}$ with at most k mind changes if, on every text for every $L \in \mathcal{L}$, the mind-change count is at most k. The class of all finite languages is identifiable with 0 mind changes after the first hypothesis

(the learner in Example 7.3 changes its mind every time a new element appears, but a more careful learner can be designed with bounded mind changes for restricted subclasses).

The surprise—genuinely unexpected for readers coming from PAC theory—is that integer-valued bounds are *not sufficient* to characterize the full landscape.

Theorem 7.14 (Freivalds–Smith [FS93])**.** *The mind-change complexity of* **Ex***-identification is naturally measured by* countable ordinals*. Specifically:*

(i) For every countable ordinal α, one can define what it means for a learner to identify a class with mind-change bound α.

(ii) There exist classes identifiable with mind-change bound ω (the first infinite ordinal) that cannot be identified with any finite mind-change bound.

(iii) More generally, for every ordinal $\alpha < \omega_1$, there exist classes identifiable with mind-change bound α but not with any bound $\beta < \alpha$.

The idea behind ordinal mind-change bounds is as follows. A learner with mind-change bound ω begins with an ordinal counter set to ω. At each mind change, the counter must decrease—but it may decrease to any smaller ordinal. Since there is no infinite descending sequence of ordinals (ordinals are well-ordered), the learner must eventually stop changing its mind. The counter ω allows finitely many mind changes, but the *number* of mind changes need not be bounded in advance by any fixed integer—it may depend on the input.

This is fundamentally different from an integer bound. With a bound of $k \in \mathbb{N}$, the learner may change its mind at most k times on *every* input. With a bound of ω, the learner may change its mind k times for input-dependent k, with no uniform finite upper bound. The ordinal hierarchy continues: $\omega \cdot 2$ allows the learner to "reset" its finite counter once; ω^2 allows nested levels of resetting; and so on up through the constructive ordinals.

Remark 7.15 (Ordinals in learning theory). The appearance of transfinite ordinals in learning theory is a genuine structural surprise. PAC learning theory uses real-valued parameters $(\varepsilon, \delta, m(\varepsilon, \delta))$. Online learning uses integer-valued dimensions (Ldim). Gold-style learning requires ordinal-valued complexity measures. Each proof technique brings its own number system. The full development of ordinal mind-change complexity is deferred to Chapter 13, where it interacts with the constructive ordinal notation systems of Kleene.

Graph Traversal

Path: mind_change_characterization $\xrightarrow{\texttt{measures}}$ ex_learning.

The mind-change ordinal is a complexity measure on **Ex**-identifiable classes, analogous to the VC dimension for PAC-learnable classes. But where VC dimension is a single integer, mind-change complexity is an ordinal—potentially transfinite. This is a `measures` edge of a qualitatively different kind than vc_dimension $\xrightarrow{\texttt{measures}}$ concept_class.

7.6 Three Paradigms, Incomparable

We now have three learning paradigms: PAC (Chapter 5), online (Chapter 6), and Gold. Each defines "learnable" differently. The question is: what is the relationship?

The answer is that they are *pairwise incomparable*. No paradigm subsumes another. There exist classes learnable under one criterion but not another, in every direction. This is the most important structural fact about the landscape of learning theory.

Separation Result

Theorem (Three-Paradigm Separation). *The following four separations hold:*

(a) **Ex-learnable but not PAC-learnable.** The class $\mathcal{L}_{\text{fin}}$ of all finite subsets of $\mathbb{N}$ is **Ex**-identifiable from text (Example 7.3). But the associated concept class has infinite VC dimension—any finite set of points can be shattered—so it is not PAC-learnable.

The witness exploits the fundamental difference in how the two paradigms treat "all finite subsets." **Ex**-identification succeeds because on any *fixed* finite set, the learner eventually sees all elements. PAC fails because the learner must handle *all* finite sets simultaneously with bounded sample complexity, and the VC dimension is infinite.

(b) **PAC-learnable but not Ex-identifiable.** The class of threshold functions $\mathcal{C} = \{x \mapsto \mathbf{1}[x \geq \theta] : \theta \in \mathbb{R}\}$ over $\mathbb{R}$ is PAC-learnable (VCdim $= 1$). But it is not **Ex**-identifiable from text, because a text for a threshold language $\{\theta, \theta+1, \theta+2, \ldots\}$ reveals the threshold only in the limit, and the class of all such languages together with all finite languages triggers Gold's impossibility theorem.

(c) **Online-learnable but not Ex-identifiable.** The class of singletons $\mathcal{C} = \{\{x\} : x \in X\}$ has Littlestone dimension 1 (online-learnable with at most 1 mistake). But the class of all singletons together with the empty set is not **Ex**-identifiable from text for reasons analogous to Gold's theorem: the learner cannot distinguish "no more positive examples will come" from "the next positive example has not yet arrived."

(d) **Ex-identifiable but not online-learnable.** Consider any class $\mathcal{L}$ of recursive languages that is **Ex**-identifiable and has infinite Littlestone dimension. Such classes exist: the class of all pattern languages (Angluin, 1980) is **Ex**-identifiable from text but has infinite Littlestone dimension over suitable domains.

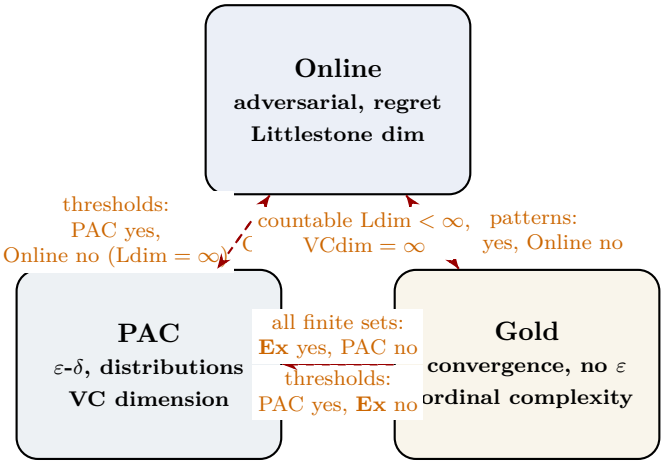

Figure 7.2: The three-paradigm separation. Each pair of paradigms is separated in both directions by explicit witnesses. No paradigm subsumes another. The dashed arrows represent `does_not_imply` edges in the graph, and each arrow is labelled with its witness class.

Remark 7.16 (What the separations mean). The three-paradigm separation is not a deficiency of the definitions. It reflects a genuine mathematical fact: different notions of "learning from data" capture different aspects of the learning problem, and no single notion is universal. PAC learning measures statistical generalization under distributional assumptions. Online learning

measures worst-case sequential prediction. Gold-style identification measures eventual convergence without any accuracy or efficiency guarantee. These are three different mathematical questions, and they have three different answers.

7.7 What This Chapter Established

This chapter introduced Gold-style identification in the limit, the oldest paradigm in formal learning theory, and established five structural facts:

1. **Gold's impossibility theorem** (Theorem 7.4): any class containing all finite languages and at least one infinite language is not **Ex**-identifiable from text. The proof is by diagonalization—the first impossibility result in learning theory, and one whose proof technique (adversarial stream construction) has no analogue in PAC or online theory.
2. **The identification hierarchy** (Section 7.3): the chain $\mathbf{FIN} \subsetneq \mathbf{Ex} \subsetneq \mathbf{BC}$ is strict, with the Case–Smith separation (Theorem 7.8) providing the witness for $\mathbf{BC} > \mathbf{Ex}$. The hierarchy is not a sequence of definitions but a chain of theorems.
3. **Relaxations form a lattice** (Section 7.4): anomalous, monotonic, and vacillatory learning branch off from **Ex**, each modifying a different aspect of the convergence requirement. Trial-and-error learning connects to Δ_2^0 in the arithmetical hierarchy.
4. **Mind-change complexity is ordinal-valued** (Section 7.5): the Freivalds–Smith characterization shows that the right complexity measure for **Ex**-identification uses transfinite ordinals, not integers. The full treatment is in Chapter 13.
5. **The three paradigms are pairwise incomparable** (Section 7.6): PAC, online, and Gold-style identification are separated in every direction by explicit witnesses. This is the most important structural fact about the landscape of learning theory.

The recursion-theoretic character of this chapter—diagonalization proofs, connections to the arithmetical hierarchy, ordinal-valued measures—contrasts sharply with the probabilistic character of Chapter 5 and the combinatorial character of Chapter 6. Each paradigm brings its own mathematical world. The separations of Section 7.6 are consequences of this fact: the paradigms use different proof techniques because they formalize genuinely different questions.

Exercises

1. **The power of informant over text.** Gold showed that the class $\mathcal{L}$ of all recursive languages is **Ex**-identifiable from *informant* (both positive and negative examples) but not from text (positive examples only).
   (a) Prove the informant direction: construct an **Ex**-learner M that identifies any recursive language L from informant. *Hint:* Dovetail over all programs $\varphi_0, \varphi_1, \ldots$, and at time t hypothesize the smallest index e consistent with all labeled data seen so far. Use the fact that the informant eventually presents every string with its correct label, so incorrect indices are eventually refuted.
   (b) Explain precisely why this learner fails from text. Identify the step in the argument that relies on negative examples, and show that no text-based substitute exists. (*Hint:* The learner can refute "φ_e includes x" from an informant presentation of $(x, 0)$, but a text for L never explicitly says "$x \notin L$"—it merely fails to present x, which is indistinguishable from delay.)
2. **Ex-identification of co-finite languages.** Let $\mathcal{L}_{\text{cof}}$ be the class of all co-finite languages over $\mathbb{N}$: $L \in \mathcal{L}_{\text{cof}}$ iff $\mathbb{N} \setminus L$ is finite.

(a) Prove that $\mathcal{L}_{\text{cof}}$ is **Ex**-identifiable from text. (*Hint:* Every co-finite language L contains all but finitely many natural numbers. A text for L eventually presents every element of L, so after time n_0, every natural number $\leq n_0$ has either appeared in the text (and is in L) or has not appeared (and may or may not be in L). Design a learner that waits long enough to conclude that unseen small numbers are *not* in L.)

(b) Prove that $\mathcal{L}_{\text{fin}} \cup \mathcal{L}_{\text{cof}}$ (all finite and all co-finite languages together) is *not* **Ex**-identifiable from text. (*Hint:* Apply Gold's impossibility theorem. Verify that this class contains all finite languages and at least one infinite language.)

(c) The result of (b) is sharper than Gold's theorem applied to "all finite + one infinite": the single infinite language is itself very structured (co-finite). Explain why the structure of the infinite language does not help—what makes the diagonalization work is not the complexity of L_∞ but the learner's inability to distinguish "finite set, done growing" from "co-finite set, still growing."

Chapter 8

Exact Learning and Query Models

The previous chapters study learners that receive data passively—drawn from a distribution (PAC), or presented by an adversary (online), or enumerated by nature (Gold). This chapter studies what happens when the learner can *ask questions*.

The exact learning model, introduced by Angluin [Ang87], equips the learner with two oracles: a membership query oracle and an equivalence query oracle. The learner's goal is not approximate generalization but *exact identification*: the game ends when the learner proposes a hypothesis that the equivalence oracle accepts. The central result of this chapter is that DFAs—computationally intractable under passive data [KV94]—become efficiently learnable with query access. This gap between passive and active learning is one of the sharpest structural results in the field.

8.1 The Exact Learning Framework

The query model was introduced in Section 2.4, where membership and equivalence oracles were defined (**??** 2.8**??** 2.9). We recall the framework here with the precision needed for algorithmic analysis.

Definition 8.1 (Minimally Adequate Teacher)**.** A *minimally adequate teacher* (MAT) for a target concept c over domain X provides two oracles:

- MQ(x): returns $c(x)$ for any $x \in X$ chosen by the learner.
- EQ(h): if $h = c$, returns YES. Otherwise, returns a counterexample $z \in X$ with $h(z) \neq c(z)$.

The teacher is "minimally adequate" in the sense that it provides exactly these two capabilities and nothing more: no distributional information, no structural hints, no preference among counterexamples.

Definition 8.2 (Exact Learning)**.** A concept class $\mathcal{C}$ is *exactly learnable in polynomial time* if there exists an algorithm A such that for every target $c \in \mathcal{C}$, algorithm A interacts with a MAT for c and eventually receives YES from EQ, using a number of queries and computation steps polynomial in:
1. the representation size n of the target c, and
2. the length m of the longest counterexample returned by EQ.

Remark 8.3 (The role of counterexample length). The dependence on m is necessary because the equivalence oracle's counterexamples are adversarial: the oracle may return arbitrarily long counterexamples. The algorithm must process them, so the complexity must account for their length.

8.2 Angluin's L^* Algorithm

The L^* algorithm learns the minimal DFA for an unknown regular language $L \subseteq \Sigma^*$ using membership and equivalence queries. It is the canonical example of exact learning and the centerpiece of this chapter.

8.2.1 Observation Tables

The algorithm maintains an *observation table* (S, E, T), where:

- $S \subseteq \Sigma^*$ is a finite, prefix-closed set of *access strings* (candidate state representatives).
- $E \subseteq \Sigma^*$ is a finite, suffix-closed set of *distinguishing extensions*.
- $T : (S \cup S \cdot \Sigma) \times E \to \{0, 1\}$ is the table, where $T(s, e) = \text{MQ}(s \cdot e)$.

For each string $s \in S \cup S \cdot \Sigma$, define the *row* $\text{row}(s) = (T(s, e_1), T(s, e_2), \ldots, T(s, e_{|E|}))$.

Definition 8.4 (Closed and Consistent Table)**.** The observation table (S, E, T) is:

- *Closed* if for every $s \in S$ and $a \in \Sigma$, there exists $s' \in S$ with $\text{row}(s \cdot a) = \text{row}(s')$.
- *Consistent* if for all $s_1, s_2 \in S$ with $\text{row}(s_1) = \text{row}(s_2)$, we have $\text{row}(s_1 \cdot a) = \text{row}(s_2 \cdot a)$ for all $a \in \Sigma$.

Closedness ensures that every one-step extension of a state representative is equivalent to some existing representative—so the transition function of the conjectured DFA is well-defined. Consistency ensures that equivalent representatives behave identically under extension—so the transition function is well-defined as a function of equivalence classes, not of particular strings.

8.2.2 The Algorithm

The L^* algorithm proceeds as follows.

1. **Initialize.** Set $S = E = \{\varepsilon\}$ (the empty string). Fill the table T using membership queries.
2. **Close and make consistent.** Repeat until the table is both closed and consistent:
   - If the table is not closed—there exists $s \in S$, $a \in \Sigma$ with $\text{row}(s \cdot a) \neq \text{row}(s')$ for any $s' \in S$—add $s \cdot a$ to S and fill new table entries via MQ.
   - If the table is not consistent—there exist $s_1, s_2 \in S$ with $\text{row}(s_1) = \text{row}(s_2)$ but $\text{row}(s_1 \cdot a) \neq \text{row}(s_2 \cdot a)$ for some $a \in \Sigma$—then there exists $e \in E$ witnessing the difference. Add $a \cdot e$ to E and fill new entries via MQ.
3. **Conjecture.** When the table is closed and consistent, construct a DFA M:
   - States: the distinct rows $\text{row}(s)$ for $s \in S$.
   - Initial state: $\text{row}(\varepsilon)$.
   - Transition: $\delta(\text{row}(s), a) = \text{row}(s \cdot a)$ (well-defined by closedness and consistency).
   - Accepting states: those $\text{row}(s)$ with $T(s, \varepsilon) = 1$.
4. **Query.** Submit $\text{EQ}(M)$.
   - If the teacher returns YES, halt and output M.
   - If the teacher returns a counterexample z, add z and all its prefixes to S, fill the table, and return to step 2.

Theorem 8.5 (L^* correctness and complexity [Ang87])**.** *Let A^* be the minimal DFA for the target language L, with n states over alphabet Σ. The L^* algorithm terminates and outputs the minimal DFA for L, using:*

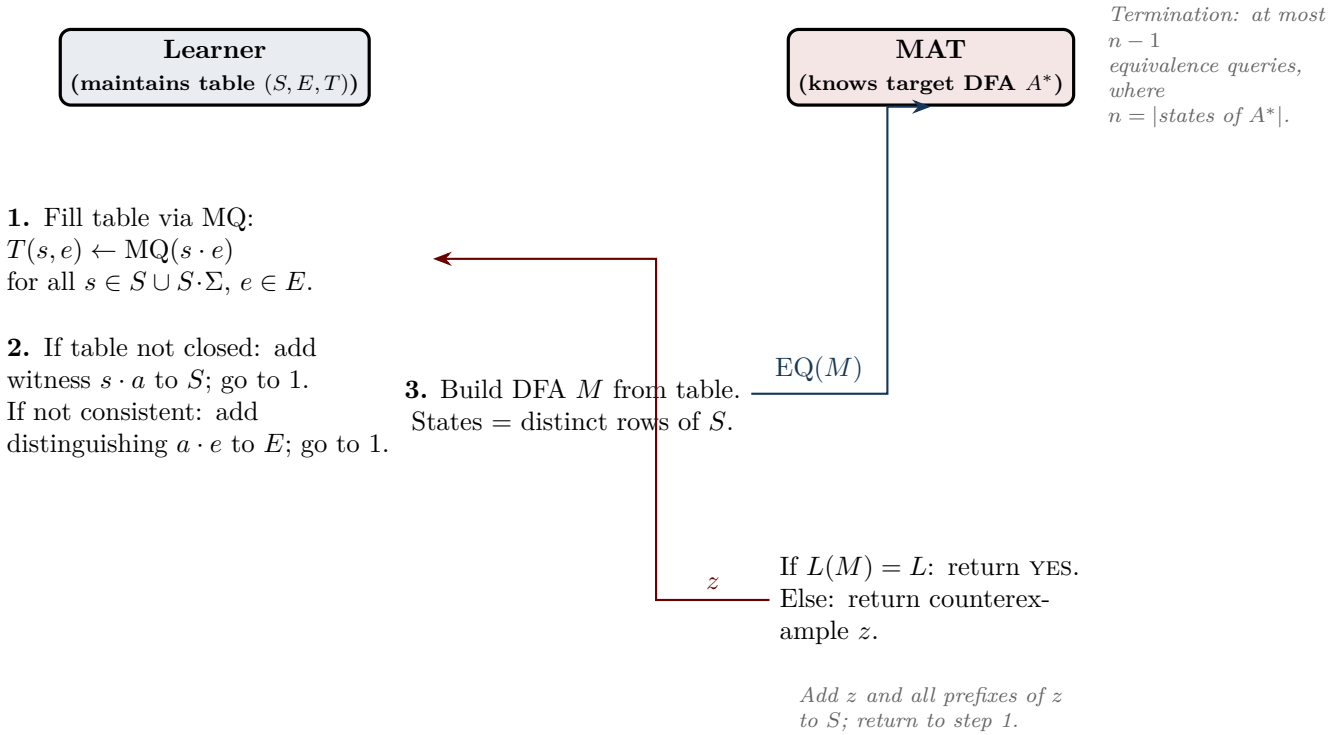

Figure 8.1: The L^* protocol. The learner fills an observation table using membership queries, ensures it is closed and consistent, constructs a conjecture DFA, and submits it to the equivalence oracle. Each counterexample refines the table by introducing a new state distinction.

- *at most* $n - 1$ *equivalence queries, and*
- *at most* $O(n^2|\Sigma| \cdot m)$ *membership queries,*

where m *is the length of the longest counterexample received.*

Proof sketch. Each counterexample introduces at least one new distinct row in the observation table, because the counterexample witnesses a string that the current conjecture misclassifies. Since the minimal DFA has n states and the table's distinct rows correspond to Myhill–Nerode equivalence classes, the number of distinct rows is bounded by n. The table starts with one row and gains at least one per counterexample, so at most $n - 1$ equivalence queries occur before the table has n distinct rows—at which point the conjecture DFA is isomorphic to A^*.

The membership query bound follows from the table dimensions: $|S| \leq n$, $|S \cdot \Sigma| \leq n|\Sigma|$, and $|E| \leq m \cdot n$ (each counterexample adds at most m suffixes). The total number of cells is $O(n|\Sigma|) \times O(nm) = O(n^2|\Sigma|m)$, each requiring one membership query.

The full correctness proof—that a closed, consistent table yields a DFA consistent with all membership queries, and that the Myhill–Nerode theorem guarantees uniqueness—is given in Angluin [Ang87]. □

Remark 8.6 (Why the table works). The observation table is a finite approximation to the Myhill–Nerode equivalence relation of the target language. Two strings s_1, s_2 are Myhill–Nerode equivalent if $s_1 \cdot w \in L \iff s_2 \cdot w \in L$ for all $w \in \Sigma^*$. The table approximates this by testing only the extensions in E. Closedness ensures the approximation is fine enough to define transitions; consistency ensures it is coherent. Each counterexample reveals that the current approximation conflates two genuinely distinct equivalence classes, forcing a refinement.

8.3 The Passive–Active Gap

The structural significance of L^* lies not in the algorithm itself but in what it implies when combined with a hardness result.

Theorem 8.7 (DFAs are not PAC-learnable [KV94])**.** *Under standard cryptographic assumptions (specifically, the hardness of inverting RSA or factoring Blum integers), the class of DFAs with n states is not efficiently PAC-learnable from random examples alone. No polynomial-time algorithm can PAC-learn DFAs, even improperly.*

The proof reduces the problem of breaking a cryptographic primitive to the problem of PAC-learning DFAs: if a polynomial-time PAC learner existed, it could be used to invert the cryptographic function, contradicting the hardness assumption. The reduction is non-trivial and we defer to Kearns and Valiant [KV94] for the full argument.

Separation Result

Passive $\not\Rightarrow$ Active. Witness: the class of DFAs.

- **Active learning succeeds.** L^* exactly identifies any n-state DFA using $O(n^2|\Sigma|m)$ membership queries and $n-1$ equivalence queries (Theorem 8.5).
- **Passive learning fails.** Under cryptographic assumptions, no polynomial-time algorithm PAC-learns DFAs from random examples (Theorem 8.7).

The gap is not quantitative (more data needed) but *qualitative* (no amount of passive data suffices for efficient learning, while a polynomial number of queries does). The mechanism is clear: membership queries let the learner probe the target's transition structure at chosen points, and equivalence queries provide directed feedback at the learner's current frontier of error. Random examples provide neither capability.

This separation witnesses a fundamental edge in the concept graph: pac_learning $\not\Rightarrow$ exact_learning and exact_learning $\not\Rightarrow$ pac_learning. Neither paradigm subsumes the other. The reverse direction—classes that are PAC-learnable but not efficiently exactly learnable—also holds, because simulating an equivalence oracle in the PAC setting requires enumerating hypotheses and checking consistency, which may be computationally intractable for some classes.

Remark 8.8 (The role of the cryptographic assumption). The Kearns–Valiant result is *conditional*: it assumes that certain cryptographic primitives are hard to break. Without this assumption, the PAC-hardness of DFAs is open. This is typical of computational hardness results in learning theory: unconditional lower bounds for PAC learning remain rare, and most known hardness results reduce from cryptography or from NP-hardness assumptions. The exact learning model sidesteps computational hardness entirely—query access changes the information-theoretic landscape so fundamentally that the cryptographic barrier does not apply.

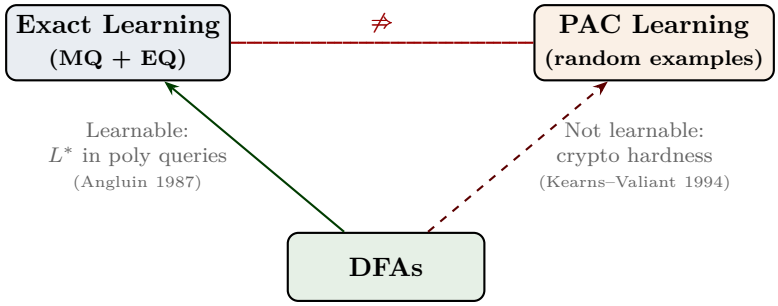

Figure 8.2: The passive–active separation. DFAs are exactly learnable with query access but not PAC-learnable under cryptographic assumptions. Neither paradigm implies the other.

8.4 Query Complexity

The L^* algorithm's complexity is measured in *query complexity*: the number of membership and equivalence queries as a function of the target's representation size. This measure is the exact learning analogue of sample complexity in PAC learning.

Definition 8.9 (Query Complexity)**.** The *membership query complexity* $\mathrm{MQ}(\mathcal{C}, n)$ and *equivalence query complexity* $\mathrm{EQ}(\mathcal{C}, n)$ of a class $\mathcal{C}$ are the minimum numbers of membership and equivalence queries, respectively, that any exact learning algorithm must make in the worst case to identify any target $c \in \mathcal{C}$ of representation size n.

For DFAs, L^* achieves $\mathrm{EQ} \leq n - 1$ and $\mathrm{MQ} = O(n^2|\Sigma|m)$. The equivalence query count has a direct connection to the combinatorial structure of the hypothesis class.

Proposition 8.10 (EQ lower bound via teaching dimension)**.** *For any class $\mathcal{C}$ that is exactly learnable, $\mathrm{EQ}(\mathcal{C}, n) \geq \lceil \log_2 |\mathcal{C}_n| \rceil$, where $\mathcal{C}_n$ is the set of concepts in $\mathcal{C}$ of representation size n. Each equivalence query eliminates at most half the version space (by returning a counterexample that is consistent with the target but not the conjecture).*

Remark 8.11 (Certificate complexity). The exact learning framework connects to *certificate complexity* in Boolean function theory: the minimum number of input bits that must be queried to certify a function's value. Teaching dimension (Chapter 10), which measures the minimum number of examples a teacher must provide to uniquely identify a concept, provides a lower bound on the number of equivalence queries. The connections among query complexity, teaching dimension, and certificate complexity are explored in Chapters 10 and 16.

8.5 What This Chapter Established

1. **The exact learning framework.** A minimally adequate teacher provides membership and equivalence queries. Exact learning requires identifying the target concept precisely, not approximately.
2. **The L^* algorithm.** Angluin's algorithm learns the minimal DFA for any regular language using polynomially many queries. The observation table approximates the Myhill–Nerode equivalence relation; each counterexample refines it by introducing a new state distinction.
3. **The passive–active gap.** DFAs are exactly learnable with MQ + EQ (Angluin, 1987) but not PAC-learnable under cryptographic assumptions (Kearns–Valiant, 1994). This separation is qualitative, not quantitative: query access changes the complexity landscape fundamentally.
4. **Query complexity.** The number of queries required to exactly learn a class is the natural complexity measure, analogous to sample complexity in PAC learning.

The gap between passive and active learning demonstrated by DFAs is not an isolated curiosity. It reflects a structural principle: the information geometry of a learning problem can change qualitatively when the learner moves from passive observation to active interrogation. This principle recurs in statistical experimental design, active learning in the PAC setting, and reinforcement learning—all settings where the learner's ability to choose which data to collect transforms the computational landscape.

Chapter 9

Universal Learning and the Trichotomy

The Fundamental Theorem of Chapter 5 answers a binary question: *is* $\mathcal{H}$ learnable? The answer is yes if and only if $\mathrm{VCdim}(\mathcal{H}) < \infty$. But the binary answer hides a quantitative question. Among all learnable classes, some are learned faster than others. How fast?

PAC learning theory phrases this quantitatively through the sample complexity $m(\varepsilon, \delta)$: how many samples suffice for accuracy ε with confidence $1 - \delta$? But the answer—$\Theta(d/\varepsilon)$ in the realizable case—is a function of the *accuracy parameter*. It describes how accuracy improves as data grows, with d as a constant. It does not describe the *rate* at which the risk itself decreases as a function of the sample size n.

This chapter asks the rate question directly: given n samples from an unknown distribution D, how fast can the risk $R_D(A(S))$ be driven to zero, *uniformly over all distributions D for which learning is possible*? The answer is the trichotomy theorem, which classifies every hypothesis class into exactly one of three rate regimes. The classification involves a combinatorial object from an unexpected source.

9.1 Beyond PAC Sample Complexity

Fix a hypothesis class $\mathcal{H}$ with $\mathrm{VCdim}(\mathcal{H}) < \infty$, so that $\mathcal{H}$ is PAC learnable. The PAC framework guarantees a learner A and a sample complexity $m(\varepsilon, \delta)$ such that, for any distribution D and any $\varepsilon, \delta > 0$, drawing $m \geq m(\varepsilon, \delta)$ samples ensures $\mathbb{P}[R_D(A(S)) > \varepsilon] \leq \delta$.

Now invert the perspective. Instead of asking "how many samples for accuracy ε?", ask: "given n samples, what is the best accuracy achievable?" More precisely, define the *learning rate* of a class $\mathcal{H}$ as follows.

Definition 9.1 (Universal Learning Rate)**.** A function $R\colon \mathbb{N} \to [0, 1]$ is a *universal learning rate* for $\mathcal{H}$ if there exists a learner A such that, for every realizable distribution D (i.e., $R_D(h^*) = 0$ for some $h^* \in \mathcal{H}$) and every $n \in \mathbb{N}$:

$$\mathbb{E}_{S \sim D^n}\big[R_D(A(S))\big] \leq R(n).$$

The *optimal universal rate* is the fastest-decaying R achievable by any learner.

Three features distinguish this from the PAC formulation:

1. **No fixed ε.** The rate $R(n)$ describes how the expected risk shrinks with n, rather than asking for a threshold sample size at a given ε.
2. **Uniformity over distributions.** The same learner A and the same rate $R(n)$ must work for *every* realizable distribution D. The learner cannot be tuned to a specific distribution.

3. **Expected risk, not high-probability.** We use $\mathbb{E}[R_D(A(S))]$ rather than a (ε, δ) guarantee. This is a convenience that simplifies the rate classification without changing the qualitative picture.

For a class with $\mathrm{VCdim}(\mathcal{H}) = d < \infty$, the PAC upper bound gives $R(n) = O(d/n)$: a $\Theta(1/n)$ rate. But is $1/n$ always achievable? Can some classes be learned faster—exponentially fast? And are there learnable classes where no uniform $1/n$ rate is possible, so that the rate degrades depending on the distribution?

All three phenomena occur. The trichotomy theorem says these are the *only* three possibilities.

9.2 The Trichotomy Theorem

The classification involves a combinatorial object that generalizes the Littlestone tree of Chapter 6. Recall that a mistake tree (Definition 6.3) is a complete binary tree whose internal nodes are labeled with instances $x \in X$, such that every root-to-leaf path is realized by some $h \in \mathcal{H}$. We now define a variant with a weaker consistency requirement.

Definition 9.2 (VCL Tree)**.** A *VCL tree* (Vapnik–Chervonenkis–Littlestone tree) for $\mathcal{H}$ over X is a complete binary tree T in which:

- Each internal node v is labeled with an instance $x_v \in X$.
- The left child corresponds to label 0, the right to label 1.
- For every root-to-leaf path $\pi = (v_1, y_1), \ldots, (v_d, y_d)$, there exists a *set* $\{x_{v_1}, \ldots, x_{v_d}\}$ that is *shattered* by $\mathcal{H}$—not merely a single hypothesis consistent with π, but the full shattering condition: for every labeling $b \in \{0,1\}^d$ of these d points, some $h \in \mathcal{H}$ realizes b.

The tree is *infinite* if it has unbounded depth.

Remark 9.3 (VCL trees vs. Littlestone trees). Every Littlestone tree is a VCL tree (shattering is a stronger condition than path-consistency, but the shattering requirement in the VCL definition applies to each path individually, and a Littlestone tree already provides a consistent hypothesis for every path—since the tree is complete, all 2^d paths are realized, which implies shattering of the instances along any single path). The converse fails: a VCL tree requires shattering along each path, which is a weaker global condition than the Littlestone tree's requirement that *every* root-to-leaf path has a *single* consistent hypothesis. The distinction matters: infinite VCL trees can exist even when $\mathrm{Ldim}(\mathcal{H}) < \infty$.

Theorem 9.4 (The Trichotomy Theorem [BHM+21])**.** *Let $\mathcal{H}$ be a hypothesis class with $|\mathcal{H}| \geq 3$. Exactly one of the following three cases holds:*

(T1) ***Exponential rate.*** *$\mathcal{H}$ has no infinite Littlestone tree (equivalently, $\mathrm{Ldim}(\mathcal{H}) < \infty$). Then the optimal universal learning rate is* exponential*:*

$$R(n) = \Theta\left(e^{-n}\right).$$

(T2) ***Linear rate.*** *$\mathcal{H}$ has an infinite Littlestone tree but no infinite VCL tree. Then the optimal universal learning rate is* linear*:*

$$R(n) = \Theta\left(\frac{1}{n}\right).$$

(T3) ***Arbitrarily slow rates.*** *$\mathcal{H}$ has an infinite VCL tree. Then no uniform rate exists: for every function $R(n) \to 0$ (no matter how slowly), every learner A fails to achieve rate $R(n)$ on some realizable distribution D.*

The three cases are exhaustive and mutually exclusive. Every learnable class (i.e., every class with $\mathrm{VCdim}(\mathcal{H}) < \infty$ and $|\mathcal{H}| \geq 3$) falls into exactly one regime. The boundaries are sharp: there is no class whose optimal rate is, say, $1/\sqrt{n}$ or $1/n^2$.

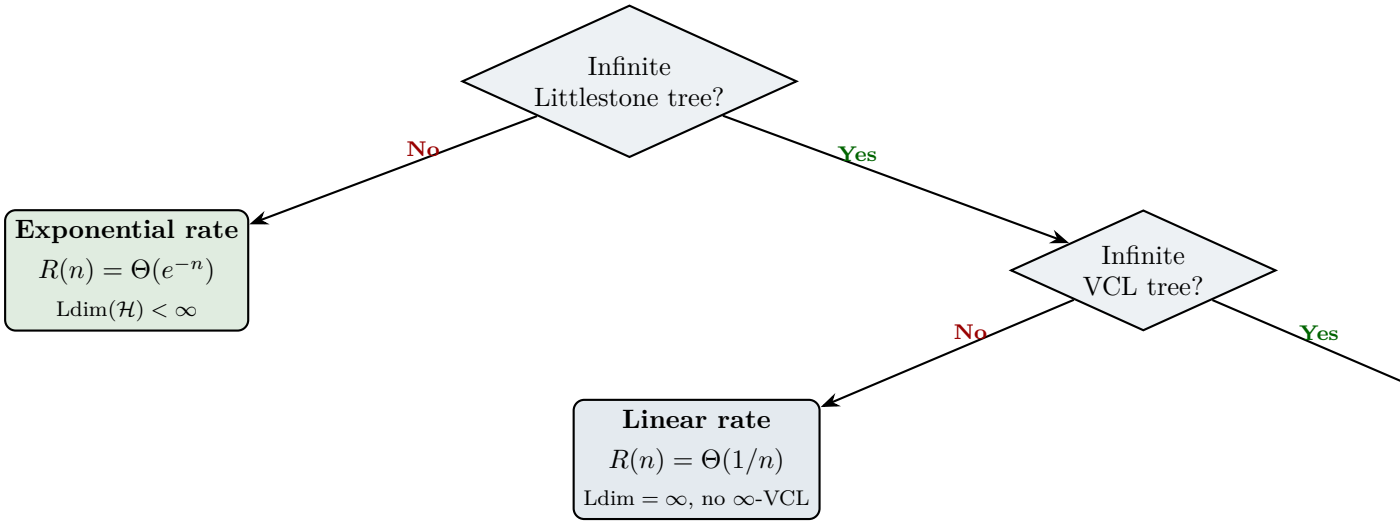

Figure 9.1: The trichotomy decision tree. Two binary questions—does $\mathcal{H}$ have an infinite Littlestone tree? does it have an infinite VCL tree?—partition all hypothesis classes (with $|\mathcal{H}| \geq 3$) into exactly three rate regimes. There is no fourth regime and no intermediate rate.

9.2.1 An Online Concept in a Statistical Theorem

The trichotomy theorem is the structural capstone of this book. To appreciate what it says, consider the expectations each tradition brought to the table.

The PAC tradition (Chapter 5) characterized learnability through the VC dimension and established the sample complexity $\Theta(d/\varepsilon)$. The natural expectation: the VC dimension should also determine the learning rate. It does not. Two classes with the same VC dimension can have different optimal rates—one exponential, one $1/n$. So what draws the boundary?

The online learning tradition (Chapter 6) introduced the Littlestone dimension to characterize mistake-bounded learning against an adversary. A worst-case combinatorial quantity, built for a worst-case adversarial game. No connection to statistical convergence rates. None expected.

And yet. The Littlestone dimension—this adversarial, combinatorial object—is exactly what governs the boundary between exponential and polynomial convergence in the i.i.d. setting.

Neither tradition predicted this. The trichotomy does not merely classify rates; it reveals that the deepest structural boundary in statistical learning theory is drawn by a quantity from a different paradigm entirely.

9.2.2 Examples

Example 9.5 (The three regimes, witnessed)**.** (a) **Exponential: finite classes.** Any finite class $\mathcal{H}$ with $|\mathcal{H}| = k$ has $\mathrm{Ldim}(\mathcal{H}) \leq \lfloor \log_2 k \rfloor < \infty$. No infinite Littlestone tree exists, so $\mathcal{H}$ falls into regime (T1). Concretely, the Halving algorithm achieves risk $\leq k \cdot 2^{-n}$ after n rounds.

(b) **Exponential: halfspaces.** The class of halfspaces in $\mathbb{R}^d$ has $\mathrm{Ldim} = d < \infty$ (Example 6.5). This class is learned at an exponential rate, despite having infinite cardinality.

(c) **Linear: thresholds on $\mathbb{R}$.** The class $\mathcal{H}_{\text{thr}} = \{x \mapsto \mathbf{1}[x \geq \theta] : \theta \in \mathbb{R}\}$ has VCdim $= 1$ but Ldim $= \infty$ (Proposition 6.18). It has no infinite VCL tree (the one-dimensional ordering prevents shattering along arbitrarily long adaptive paths). The optimal rate is $\Theta(1/n)$.

(d) **Arbitrarily slow: unions of intervals.** For each k, let $\mathcal{H}_k$ be the class of unions of at most k intervals on $\mathbb{R}$. The class $\mathcal{H}_\infty = \bigcup_k \mathcal{H}_k$ (unions of finitely many intervals, with no bound on the number) has VCdim $< \infty$ but admits infinite VCL trees. No uniform rate $R(n) \to 0$ is achievable.

9.3 Proof Architecture

The full proof of Theorem 9.4 occupies much of [BHM+21]. We sketch the key ideas for each regime, emphasizing why the boundaries fall where they do.

9.3.1 The Exponential Case: Finite Littlestone Dimension

Suppose $\text{Ldim}(\mathcal{H}) = d < \infty$. The Standard Optimal Algorithm (SOA) from Chapter 6 was designed for adversarial online learning, but its structure can be adapted to the statistical setting.

The key insight is that a learner with access to i.i.d. samples can simulate a version-space strategy. Consider the following approach: given a sample S of size n, maintain the version space $V = \{h \in \mathcal{H} : h \text{ consistent with } S\}$. Return a hypothesis from V using an SOA-derived selection rule.

When $\text{Ldim}(\mathcal{H}) = d$, the version space after seeing n i.i.d. samples has the following property: the probability (over D) that there exists a hypothesis $h \in V$ with $R_D(h) > \varepsilon$ decreases exponentially in n. This is because each sample point, with probability at least ε under D, eliminates all hypotheses in V that disagree with the true label at that point. After n samples, the survival probability of any "bad" hypothesis (one with risk $> \varepsilon$) is at most $(1-\varepsilon)^n \leq e^{-\varepsilon n}$. The finite Littlestone dimension controls the effective complexity of V through the mistake tree structure, ensuring that a union bound over the relevant version space subsets remains finite.

The result:

$$\mathbb{E}[R_D(A(S))] \leq C \cdot 2^d \cdot e^{-n/C'}$$

for constants C, C' depending on d. The rate is exponential in n.

The matching lower bound shows that exponential convergence is impossible when $\text{Ldim}(\mathcal{H}) = \infty$: an infinite Littlestone tree provides an adversary-like construction (within the distributional framework) that forces the learner's expected risk to decay no faster than $\Omega(1/n)$.

9.3.2 The Linear Case: Infinite Littlestone, Finite VCL

Suppose $\text{Ldim}(\mathcal{H}) = \infty$ but $\mathcal{H}$ has no infinite VCL tree. The exponential strategy fails because the SOA-based approach requires finite Littlestone dimension. But a different algorithm—the *one-inclusion predictor*—achieves the $1/n$ rate.

Definition 9.6 (One-Inclusion Predictor (Informal))**.** Given a sample $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and a new point x, form the multiset $\{x_1, \ldots, x_n, x\}$. Consider the one-inclusion graph G: the graph whose vertices are all labelings of this multiset consistent with some $h \in \mathcal{H}$, with edges connecting labelings that differ on exactly one point. Predict the label $\hat{y}$ that minimizes the maximum degree of the vertex in G corresponding to the completed labeling.

Haussler, Littlestone, and Warmuth [Hau88] introduced the one-inclusion graph technique. Bousquet et al. [BHM+21] showed that when $\mathcal{H}$ has no infinite VCL tree, the one-inclusion predictor achieves

$$\mathbb{E}[R_D(A(S))] \leq \frac{C}{n}$$

for a constant C depending on $\mathcal{H}$. The absence of infinite VCL trees is precisely the combinatorial condition needed for the one-inclusion graph to have bounded degree growth, which in turn controls the expected risk.

The matching lower bound is the PAC lower bound: any class with $\mathrm{VCdim}(\mathcal{H}) \geq 1$ requires $\Omega(1/n)$ expected risk, since distinguishing hypotheses that differ on a $1/n$-fraction of the domain requires $\Omega(n)$ samples.

9.3.3 The Slow Case: Infinite VCL Trees

Suppose $\mathcal{H}$ has an infinite VCL tree. The theorem claims that no uniform rate $R(n) \to 0$ is achievable: for every learner A and every function $R(n) \to 0$, there exists a realizable distribution D such that $\mathbb{E}[R_D(A(S_n))] \geq R(n)$ for infinitely many n.

The proof constructs, for any candidate learner A and any target rate $R(n)$, a distribution that defeats A at the desired rate. The construction uses the infinite VCL tree as a source of combinatorial richness.

Descend the VCL tree adaptively, choosing at each node the branch that is harder for A. Because the tree is infinite and each path corresponds to a shattered set, the adversary can build a distribution at any depth d whose effective VC dimension on the relevant support is d. By choosing d large enough (as a function of n), the adversary ensures that the PAC lower bound $\Omega(d/n)$ exceeds $R(n)$. The infinite VCL tree provides arbitrarily large effective VC dimension along adaptive paths, preventing any uniform rate from holding.

This is the sense in which infinite VCL trees are an obstruction: they provide an inexhaustible reservoir of complexity that any learner eventually confronts.

9.4 The Cross-Paradigm Map

The trichotomy theorem connects three chapters of this book through a single combinatorial object.

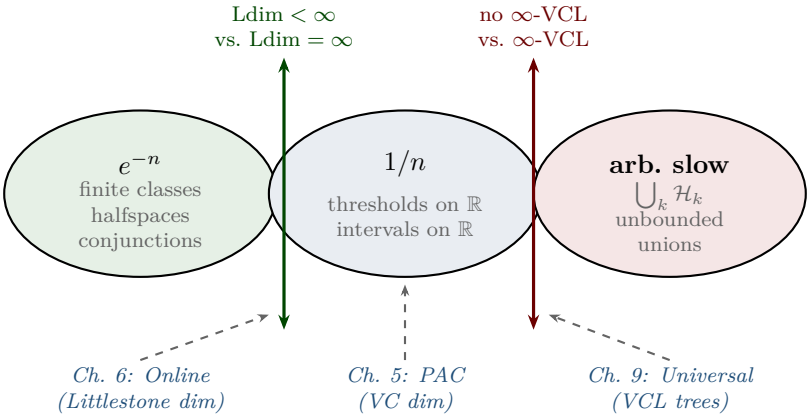

Figure 9.2: The three rate regimes and their governing combinatorial conditions. The left boundary is drawn by the Littlestone dimension (from online learning, Chapter 6); the right boundary by VCL trees (a hybrid of VC shattering and Littlestone trees). The VC dimension (Chapter 5) determines *whether* learning is possible; the Littlestone dimension and VCL trees determine *how fast*.

The structural lesson is a hierarchy of three combinatorial conditions, each finer than the last:

Condition	Governs	Chapter	Paradigm
$\text{VCdim}(\mathcal{H}) < \infty$	Learnability (yes/no)	Chapter 5	PAC
$\text{Ldim}(\mathcal{H}) < \infty$	Rate: e^{-n} vs. $1/n$	Chapter 6	Online
No infinite VCL tree	Rate: $1/n$ vs. arb. slow	This chapter	Universal

Each row refines the previous one. The VC dimension draws the coarsest line (learnable vs. not). The Littlestone dimension draws a finer line within the learnable classes (fast vs. slow). The VCL tree draws the finest line (uniformly slow vs. hopelessly slow).

This hierarchy was not predicted by any single tradition. It emerged from the trichotomy theorem's proof, which required techniques from both the PAC and online learning literatures. The result is a unified picture in which the three paradigms—PAC, online, and universal—are not separate theories with accidental parallels, but different projections of a single combinatorial landscape.

9.5 What This Chapter Established

Two results, one structural lesson.

1. **Universal learning rates.** The optimal learning rate asks a harder question than PAC sample complexity: not how many samples for a fixed accuracy, but how fast the risk decreases as a function of n, uniformly over all realizable distributions.

2. **The trichotomy.** Every hypothesis class with $|\mathcal{H}| \geq 3$ falls into exactly one of three regimes: exponential (e^{-n}), linear ($1/n$), or arbitrarily slow. The boundaries are determined by the Littlestone dimension and VCL trees. There are no intermediate rates.

3. **The cross-paradigm bridge.** The Littlestone dimension—born in the adversarial online setting of Chapter 6—controls the boundary between exponential and polynomial convergence in the i.i.d. statistical setting. This was the central open question resolved by Bousquet et al. [BHM+21]: not just what the rate regimes are, but what combinatorial objects govern them—and the answer came from an unexpected paradigm.

Chapter 10

Combinatorial Dimensions

Part II characterized learnability in four paradigms, and in each case the answer was a number: the VC dimension for PAC learning, the Littlestone dimension for online learning, the mind-change ordinal for Gold-style identification, the teaching dimension for exact learning. This chapter and the three that follow develop these numbers—and the two dozen others in the concept graph—as objects of study in their own right.

The present chapter treats *combinatorial dimensions*: quantities defined by asking how richly a hypothesis class can label finite configurations of points. The simplest such quantity, the VC dimension, asks how many points can be shattered. The others refine, extend, or replace this question for settings beyond binary classification. We prove the two foundational combinatorial results in full (the Sauer–Shelah lemma and the VC dimension characterization already established in Chapter 5), recall the Littlestone dimension from Chapter 6, and then turn to the chapter's narrative centerpiece: the thirty-year search for the correct combinatorial dimension characterizing multiclass learnability, which ended only in 2022.

The chapter is organized as follows.

1. **VC dimension and shattering** (Section 10.1): the definitions, the growth function, and the full proof of the Sauer–Shelah lemma.
2. **The Littlestone dimension** (Section 10.2): a brief recall from Chapter 6 and the relationship $\mathrm{VCdim} \leq \mathrm{Ldim}$.
3. **Beyond binary: pseudodimension and fat-shattering** (Section 10.3): scale-sensitive dimensions for real-valued function classes.
4. **The multiclass story: from Natarajan to DS** (Section 17.1): the detective story of multiclass PAC characterization.
5. **Other dimensions** (Section 10.5): a concise catalog of further combinatorial dimensions, with forward references.

10.1 VC Dimension and Shattering

The definitions in this section were introduced informally in Chapter 1 and used without proof in Chapter 5. We now state them precisely and prove the combinatorial backbone that supports the Fundamental Theorem.

Definition 10.1 (Restriction and Shattering)**.** Let $\mathcal{H} \subseteq \{0,1\}^X$ be a hypothesis class and let $S = \{x_1, \ldots, x_n\} \subseteq X$ be a finite set. The *restriction* of $\mathcal{H}$ to S is

$$\mathcal{H}|_S \;=\; \{(h(x_1), \ldots, h(x_n)) : h \in \mathcal{H}\} \;\subseteq\; \{0,1\}^n.$$

We say $\mathcal{H}$ *shatters* S if $\mathcal{H}|_S = \{0,1\}^n$, i.e., every labeling of S is realized by some hypothesis in $\mathcal{H}$.

Definition 10.2 (VC Dimension [VC71])**.** The *Vapnik–Chervonenkis dimension* of $\mathcal{H}$, denoted $\mathrm{VCdim}(\mathcal{H})$, is the largest cardinality of a set shattered by $\mathcal{H}$:

$$\mathrm{VCdim}(\mathcal{H}) \;=\; \max\{|S| : S \subseteq X,\ \mathcal{H} \text{ shatters } S\}.$$

If $\mathcal{H}$ shatters arbitrarily large finite sets, $\mathrm{VCdim}(\mathcal{H}) = \infty$.

Example 10.3 (Halfplanes in $\mathbb{R}^2$)**.** Let $\mathcal{H}$ be the class of halfplanes in $\mathbb{R}^2$: $\mathcal{H} = \{x \mapsto \mathbf{1}[\langle w, x\rangle \geq b] : w \in \mathbb{R}^2,\, b \in \mathbb{R}\}$. Any three non-collinear points are shattered (Figure 10.1), but no four points can be: for any four points, at least one is inside the convex hull of the other three, and the labeling that assigns it the opposite label from the other three cannot be realized by a halfplane. Hence $\mathrm{VCdim}(\mathcal{H}) = 3$.

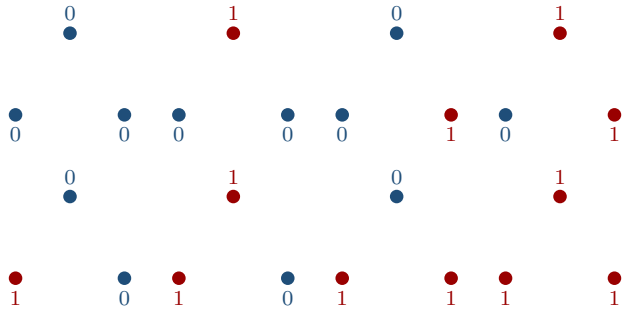

Figure 10.1: All $2^3 = 8$ labelings of three non-collinear points in $\mathbb{R}^2$. Each labeling is realized by some halfplane, so the points are shattered. Red (filled dark) indicates label 1; blue indicates label 0.

10.1.1 The Growth Function

The growth function measures how many distinct labelings $\mathcal{H}$ can produce on sets of a given size.

Definition 10.4 (Growth Function)**.** The *growth function* (or *shatter coefficient*) of $\mathcal{H}$ is

$$\Pi_{\mathcal{H}}(n) \;=\; \max_{S \subseteq X,\, |S| = n} \left|\mathcal{H}|_S\right|.$$

Equivalently, $\Pi_{\mathcal{H}}(n)$ is the maximum number of distinct behaviors $\mathcal{H}$ can exhibit on any n points.

The growth function satisfies $\Pi_{\mathcal{H}}(n) \leq 2^n$ always, with equality when $\mathcal{H}$ shatters some set of size n. The remarkable content of the Sauer–Shelah lemma is that once shattering fails—at $n = \mathrm{VCdim}(\mathcal{H}) + 1$—the growth function drops from exponential to polynomial *immediately and permanently.*

10.1.2 The Sauer–Shelah Lemma

Lemma 10.5 (Sauer–Shelah [Sau72, She72])**.** *Let $\mathcal{H} \subseteq \{0,1\}^X$ with $\mathrm{VCdim}(\mathcal{H}) = d$. Then for all $n \geq 1$,*

$$\Pi_{\mathcal{H}}(n) \;\leq\; \sum_{i=0}^{d} \binom{n}{i}.$$

In particular, $\Pi_{\mathcal{H}}(n) \leq \left(\frac{en}{d}\right)^d$ for $n \geq d \geq 1$, so the growth function is polynomial in n when $d < \infty$.

The proof is by double induction and is one of the most elegant arguments in combinatorics. The key idea is to split the hypothesis class along one coordinate and apply the inductive hypothesis to the two resulting classes, whose VC dimensions are carefully controlled.

Proof. We prove a stronger statement: for any finite set S with $|S| = n$ and any class $\mathcal{H}' \subseteq \{0,1\}^S$ (not necessarily arising from a restriction),

$$|\mathcal{H}'| \;\le\; \big|\{A \subseteq S : \mathcal{H}' \text{ shatters } A\}\big|. \tag{$\star$}$$

This implies the lemma: if $\mathrm{VCdim}(\mathcal{H}') \le d$, then the right-hand side counts only subsets A with $|A| \le d$, of which there are at most $\sum_{i=0}^{d} \binom{n}{i}$.

We prove ($\star$) by induction on $|S|$.

Base case ($|S| = 1$, say $S = \{x\}$). If $\mathcal{H}' = \{0\}$ or $\mathcal{H}' = \{1\}$, then $|\mathcal{H}'| = 1$ and $\mathcal{H}'$ shatters $\emptyset$, giving at least one shattered set. If $\mathcal{H}' = \{0,1\}$, then $|\mathcal{H}'| = 2$ and $\mathcal{H}'$ shatters both $\emptyset$ and $\{x\}$, giving at least two shattered sets. In both cases ($\star$) holds.

Inductive step. Let $|S| = n > 1$. Pick any element $x \in S$ and set $S' = S \setminus \{x\}$. Partition $\mathcal{H}'$ according to behavior on x:

$$\begin{aligned}
\mathcal{H}_0 &= \{h|_{S'} : h \in \mathcal{H}',\ h(x) = 0\},\\
\mathcal{H}_1 &= \{h|_{S'} : h \in \mathcal{H}',\ h(x) = 1\},\\
\mathcal{H}_\cap &= \mathcal{H}_0 \cap \mathcal{H}_1.
\end{aligned}$$

Here $\mathcal{H}_\cap$ consists of the functions on S' that appear with *both* labels on x.

Count: each $h \in \mathcal{H}'$ contributes to exactly one of $\mathcal{H}_0$ or $\mathcal{H}_1$ (as a function on S'), but a function in $\mathcal{H}_\cap$ is contributed by two elements of $\mathcal{H}'$ (one with $h(x) = 0$, one with $h(x) = 1$). Therefore,

$$|\mathcal{H}'| \;=\; |\mathcal{H}_0| + |\mathcal{H}_1| - |\mathcal{H}_\cap| + |\mathcal{H}_\cap| \;=\; |\mathcal{H}_0 \cup \mathcal{H}_1| + |\mathcal{H}_\cap|.$$

More precisely, $|\mathcal{H}'| = |\mathcal{H}_0| + |\mathcal{H}_1|$, since functions in $\mathcal{H}_0 \setminus \mathcal{H}_\cap$ appear only from $h(x) = 0$, functions in $\mathcal{H}_1 \setminus \mathcal{H}_\cap$ appear only from $h(x) = 1$, and functions in $\mathcal{H}_\cap$ appear twice. So

$$|\mathcal{H}'| = |\mathcal{H}_0 \cup \mathcal{H}_1| + |\mathcal{H}_\cap|.$$

Apply the inductive hypothesis (on S', which has $n-1$ elements) to both $\mathcal{H}_0 \cup \mathcal{H}_1$ and $\mathcal{H}_\cap$:

$$\begin{aligned}
|\mathcal{H}_0 \cup \mathcal{H}_1| &\le \big|\{A \subseteq S' : (\mathcal{H}_0 \cup \mathcal{H}_1) \text{ shatters } A\}\big|,\\
|\mathcal{H}_\cap| &\le \big|\{A \subseteq S' : \mathcal{H}_\cap \text{ shatters } A\}\big|.
\end{aligned}$$

Now we connect the shattered sets of the restricted classes back to those of $\mathcal{H}'$:

1. If $A \subseteq S'$ is shattered by $\mathcal{H}_0 \cup \mathcal{H}_1$ but *not* by $\mathcal{H}_\cap$, then A is shattered by $\mathcal{H}'$ as well (since every labeling of A is realized by some $h|_{S'} \in \mathcal{H}_0 \cup \mathcal{H}_1$, and at least one such h exists in $\mathcal{H}'$).
2. If $A \subseteq S'$ is shattered by $\mathcal{H}_\cap$, then every labeling of A is realized by functions appearing with both labels on x. Hence $\mathcal{H}'$ shatters $A \cup \{x\}$: for any labeling of $A \cup \{x\}$, pick the label b assigned to x, then pick $h|_{S'} \in \mathcal{H}_\cap$ realizing the labeling of A; the corresponding $h \in \mathcal{H}'$ with $h(x) = b$ realizes the full labeling.

Crucially, the sets in (1) do not contain x, and the sets in (2) do contain x, so they are disjoint families. Therefore,

$$|\mathcal{H}'| \;\le\; \big|\{A \subseteq S : \mathcal{H}' \text{ shatters } A,\ x \notin A\}\big| + \big|\{A \subseteq S : \mathcal{H}' \text{ shatters } A,\ x \in A\}\big| \;=\; \big|\{A \subseteq S : \mathcal{H}' \text{ shatters } A\}\big|.$$

This completes the induction. □

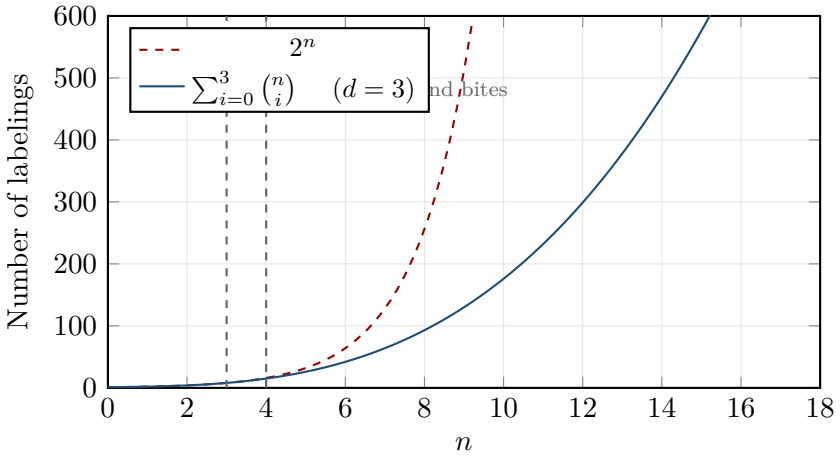

Figure 10.2: The growth function $\Pi_{\mathcal{H}}(n)$ versus 2^n. For $n \le d = \mathrm{VCdim}(\mathcal{H})$, the growth function can equal 2^n (if $\mathcal{H}$ shatters some set of that size). At $n = d + 1$, the Sauer–Shelah bound forces a permanent drop to polynomial growth. The polynomial bound $\sum_{i=0}^{d} \binom{n}{i}$ is plotted for $d = 3$.

Remark 10.6. The lemma was proved independently by Sauer [Sau72] and Shelah [She72], with a related result by Perles and Shelah appearing earlier. Vapnik and Chervonenkis [VC71] proved a weaker bound in their 1971 paper. The proof above follows the elegant "number of shattered sets" formulation due to Alon and Frankl.

The Sauer–Shelah lemma is the combinatorial engine behind the Fundamental Theorem of Statistical Learning (Theorem 5.16): it converts a finite VC dimension into a polynomial bound on the growth function, which in turn yields uniform convergence of empirical risks via a symmetrization argument.

10.2 The Littlestone Dimension

The VC dimension measures shatterability of *sets*: it asks for the largest set whose every labeling is realized. The Littlestone dimension, introduced in Chapter 6 and fully characterized there, measures shatterability of *trees*: it asks for the deepest complete binary tree whose every root-to-leaf path is realized. We recall the definition for reference.

Definition 10.7 (Mistake Tree — Recall)**.** A *mistake tree* for $\mathcal{H} \subseteq \{0,1\}^X$ is a complete binary tree in which each internal node is labeled with an instance $x \in X$, left edges correspond to label 0, right edges to label 1, and every root-to-leaf path is consistent with some $h \in \mathcal{H}$ (see Definition 6.3).

Definition 10.8 (Littlestone Dimension — Recall)**.** The *Littlestone dimension* $\mathrm{Ldim}(\mathcal{H})$ is the maximum depth of a mistake tree for $\mathcal{H}$ (Definition 6.4). The Littlestone characterization (Theorem 6.12) establishes that $\mathrm{Ldim}(\mathcal{H})$ equals the optimal worst-case mistake bound.

The relationship between the two fundamental dimensions is strict:

Proposition 10.9. *For any hypothesis class $\mathcal{H}$,* $\mathrm{VCdim}(\mathcal{H}) \le \mathrm{Ldim}(\mathcal{H})$.

Proof. If $\mathcal{H}$ shatters $\{x_1, \dots, x_d\}$, one can construct a mistake tree of depth d by placing x_i at every node of depth i: since every labeling is realized, every root-to-leaf path is consistent with some $h \in \mathcal{H}$. Hence any shattered set yields a mistake tree of the same depth. □

The converse fails: threshold classifiers on $\mathbb{R}$ have $\mathrm{VCdim} = 1$ but $\mathrm{Ldim} = \infty$ (Chapter 14). Intuitively, the Littlestone dimension is larger because it accounts for adaptive adversaries: the adversary in the online game chooses instances sequentially, seeing the learner's responses, while the VC dimension considers all labelings of a fixed set simultaneously.

10.3 Beyond Binary: Pseudodimension and Fat-Shattering

When the hypothesis class consists of real-valued functions $\mathcal{F} \subseteq \mathbb{R}^X$ rather than binary classifiers, shattering must be redefined. Two progressively refined notions capture the relevant complexity.

Definition 10.10 (Pseudodimension)**.** Let $\mathcal{F} \subseteq \mathbb{R}^X$. A set $S = \{x_1, \ldots, x_n\} \subseteq X$ is *pseudo-shattered* (or *P-shattered*) by $\mathcal{F}$ if there exist thresholds $t_1, \ldots, t_n \in \mathbb{R}$ such that, for every labeling $b \in \{0,1\}^n$, there exists $f \in \mathcal{F}$ with

$$\operatorname{sgn}(f(x_i) - t_i) = (-1)^{1-b_i} \quad \text{for all } i = 1, \ldots, n.$$

The *pseudodimension* $\mathrm{Pdim}(\mathcal{F})$ is the largest cardinality of a P-shattered set.

The thresholds t_i act as "witnesses": they reduce a real-valued shattering problem to a binary one. When $\mathcal{F}$ is already $\{0,1\}$-valued, the thresholds $t_i = 1/2$ recover the standard VC dimension, so Pdim is a proper generalization.

Definition 10.11 (Fat-Shattering Dimension [ABDCBH97])**.** Let $\mathcal{F} \subseteq \mathbb{R}^X$ and $\gamma > 0$. A set $S = \{x_1, \ldots, x_n\}$ is *γ-fat-shattered* by $\mathcal{F}$ if there exist thresholds $t_1, \ldots, t_n \in \mathbb{R}$ such that, for every labeling $b \in \{0,1\}^n$, there exists $f \in \mathcal{F}$ with

$$f(x_i) \;\geq\; t_i + \gamma \;\text{ when } b_i = 1, \qquad f(x_i) \;\leq\; t_i - \gamma \;\text{ when } b_i = 0,$$

for all i. The *fat-shattering dimension at scale* γ, denoted $\mathrm{fat}_\gamma(\mathcal{F})$, is the largest cardinality of a γ-fat-shattered set.

The margin parameter γ makes fat-shattering *scale-sensitive*: large margins are harder to achieve, so $\mathrm{fat}_\gamma(\mathcal{F}) \leq \mathrm{fat}_{\gamma'}(\mathcal{F})$ whenever $\gamma \geq \gamma'$. As $\gamma \to 0$, the fat-shattering dimension approaches the pseudodimension.

Theorem 10.12 (Characterization of Real-Valued Learnability [ABDCBH97])**.** *A real-valued function class $\mathcal{F} \subseteq [0,1]^X$ is* agnostically PAC learnable *(with respect to the absolute loss) if and only if $\mathrm{fat}_\gamma(\mathcal{F}) < \infty$ for every $\gamma > 0$.*

This is the real-valued analogue of the Fundamental Theorem: just as finite VC dimension characterizes binary PAC learnability, finite fat-shattering dimension at every scale characterizes real-valued learnability. The pseudodimension alone is not sufficient for characterization in the agnostic setting, as it ignores the scale structure.

Remark 10.13 (Dimension hierarchy). For a class $\mathcal{F} \subseteq [0,1]^X$:

$$\mathrm{fat}_\gamma(\mathcal{F}) \;\leq\; \mathrm{Pdim}(\mathcal{F}) \;\leq\; \mathrm{VCdim}(\mathrm{subgraph}(\mathcal{F}))$$

where $\mathrm{subgraph}(\mathcal{F}) = \{(x,t) \mapsto \mathbf{1}[f(x) \geq t] : f \in \mathcal{F}\}$. When $\mathcal{F}$ is $\{0,1\}$-valued, all three quantities coincide with $\mathrm{VCdim}(\mathcal{F})$.

10.4 The Multiclass Story: From Natarajan to DS

Binary classification has a clean answer: a class $\mathcal{H} \subseteq \{0,1\}^X$ is PAC learnable if and only if $\mathrm{VCdim}(\mathcal{H}) < \infty$. What is the corresponding answer when the label set Y has $k \geq 3$ elements? This question, first posed in the 1980s, was open for thirty years. Its resolution required not merely a new proof technique but a fundamentally new combinatorial dimension. The story is worth telling in full, because the obvious candidate turned out to be wrong.

10.4.1 The Natarajan Dimension: The Obvious Candidate

Definition 10.14 (Natarajan Dimension)**.** Let $\mathcal{H} \subseteq Y^X$ with $|Y| \geq 2$. The *Natarajan dimension* $d_{\mathrm{N}}(\mathcal{H})$ is the largest d such that there exist

- a set $S = \{x_1, \ldots, x_d\} \subseteq X$, and
- two functions $f_0, f_1 \colon S \to Y$ with $f_0(x_i) \neq f_1(x_i)$ for all i,

such that for every labeling $b \in \{0,1\}^d$, there exists $h \in \mathcal{H}$ with $h(x_i) = f_{b_i}(x_i)$ for all i.

The Natarajan dimension generalizes the VC dimension in a natural way: instead of requiring all 2^d binary labelings, it requires all 2^d labelings drawn from two specified "opposing" functions. When $|Y| = 2$, the Natarajan dimension equals the VC dimension exactly.

In the 1980s and 1990s, the working conjecture was:

A multiclass hypothesis class $\mathcal{H} \subseteq Y^X$ is PAC learnable if and only if $d_{\mathrm{N}}(\mathcal{H}) < \infty$.

This conjecture was supported by partial results. Ben-David, Cesa-Bianchi, Haussler, and Long proved that finite Natarajan dimension is *sufficient* for multiclass PAC learnability when combined with a bound on $|Y|$. The "if" direction seemed solid. The conjecture survived for decades without a counterexample.

It was wrong.

10.4.2 The Counterexample: When the Obvious Fails

Theorem 10.15 (Failure of Natarajan Dimension [DSS14])**.** *There exists a hypothesis class $\mathcal{H} \subseteq Y^X$ with $d_{\mathrm{N}}(\mathcal{H}) = 1$ that is* not *PAC learnable.*

This is a dramatic failure: the Natarajan dimension can be as small as possible (just one) and still the class can be unlearnable. The problem lies in the label set: the construction uses $|Y| = \infty$, and the Natarajan dimension, which examines only pairs of opposing functions, fails to capture the combinatorial explosion that arises when the label set is large.

The witness is a class constructed via a combinatorial argument. The key insight is that the Natarajan dimension constrains the number of "binary slices" through the labeling space but does not constrain the total complexity when the label set grows without bound.

Separation Result

Natarajan vs. PAC learnability. The Natarajan dimension does not characterize multiclass PAC learnability:

$$d_{\mathrm{N}}(\mathcal{H}) < \infty \quad \not\Longrightarrow \quad \mathcal{H} \text{ is PAC learnable.}$$

The gap is witnessed by a class with $d_{\mathrm{N}} = 1$, $|Y| = \infty$, that requires unbounded sample complexity. The correct characterization requires the DS dimension (Definition 17.4 below).

10.4.3 The DS Dimension: The Correct Answer

Definition 10.16 (DS Dimension [BCD+22])**.** Let $\mathcal{H} \subseteq Y^X$. The *DS dimension* (named for Daniely and Shalev-Shwartz) $\mathrm{DSdim}(\mathcal{H})$ is defined via pseudo-cubes in the one-inclusion hypergraph of $\mathcal{H}$.

Construct the *one-inclusion hypergraph* $G_{\mathcal{H}}$ on a finite set $S = \{x_1, \ldots, x_n\} \subseteq X$ as follows: the vertices are the hypotheses $\mathcal{H}|_S$ (restrictions of $\mathcal{H}$ to S), and for each $i \in \{1, \ldots, n\}$, add a hyperedge connecting all hypotheses that agree on $S \setminus \{x_i\}$ (i.e., they differ only at x_i).

A *pseudo-cube of dimension* d in $G_{\mathcal{H}}$ is a subhypergraph isomorphic to the complete d-dimensional hypercube graph $\{0,1\}^d$.

The DS dimension $\mathrm{DSdim}(\mathcal{H})$ is the supremum over all finite $S \subseteq X$ of the maximum dimension of a pseudo-cube in $G_{\mathcal{H}}$.

Theorem 10.17 (Multiclass Characterization [BCD+22])**.** *A hypothesis class $\mathcal{H} \subseteq Y^X$ is PAC learnable if and only if* $\mathrm{DSdim}(\mathcal{H}) < \infty$.

This theorem, proved by Brukhim, Chepoi, Daniel, Moran, and Yehudayoff in 2022, resolved the multiclass characterization problem. It is the multiclass analogue of the Fundamental Theorem: the DS dimension plays exactly the role that the VC dimension plays for binary classification.

Historical Note

Why the Natarajan dimension fails and the DS dimension succeeds. The Natarajan dimension examines the labeling structure of $\mathcal{H}$ by projecting onto pairs of opposing functions—a fundamentally binary lens applied to a multiclass problem. The DS dimension, by contrast, examines the full combinatorial structure of the one-inclusion hypergraph, which encodes all the ways hypotheses can disagree on single points.
The one-inclusion hypergraph is not new: Haussler, Littlestone, and Warmuth introduced it in 1994 for binary classification, where it yields an optimal PAC learner (the one-inclusion algorithm). What Brukhim et al. discovered is that the right way to measure the complexity of a multiclass hypothesis class is to look for high-dimensional pseudo-cubes inside this hypergraph. Pseudo-cubes are the multiclass analogue of shattered sets: they represent configurations where the labeling structure is rich enough to prevent learning.

The dimension hierarchy for multiclass problems is:

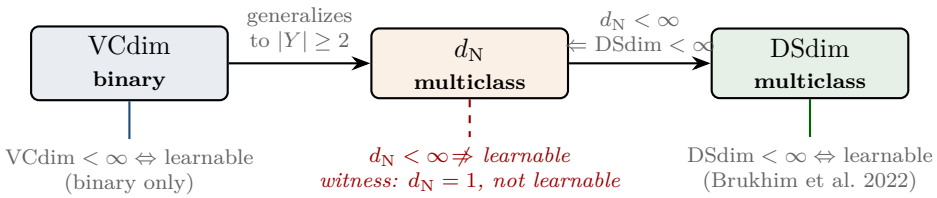

Figure 10.3: The multiclass dimension hierarchy. The Natarajan dimension was the leading candidate for thirty years, but finite d_{N} does not imply learnability. The DS dimension is the correct characterization.

10.5 Other Dimensions

Several further combinatorial dimensions appear in the concept graph. We collect them here in catalog form; each is developed more fully in the chapter where it plays a characterizing role.

Remark 10.18 (Star number equivalence). The equivalence between finite star number and finite VC dimension is not obvious. The star number counts a sunflower-type configuration that, on its face, looks quite different from shattering. That these two combinatorial conditions coincide is one of the surprising structural facts of the theory: there are *multiple, independently motivated* combinatorial properties of a hypothesis class that all turn out to be equivalent to PAC learnability.

Table 10.1: Catalog of combinatorial dimensions beyond VC, Littlestone, pseudodimension, fat-shattering, Natarajan, and DS.

Dimension	Notation	What it measures	Reference
Star number	$\mathfrak{s}(\mathcal{H})$	The maximum number of sets in a sunflower whose kernel is not shattered. Finiteness of $\mathfrak{s}(\mathcal{H})$ is *equivalent* to $\mathrm{VCdim}(\mathcal{H}) < \infty$—a surprising alternative combinatorial characterization of PAC learnability.	Chapter 5
Eluder dimension	$\dim_E(\mathcal{F})$	The length of the longest sequence of points such that each point is "independent" of the previous ones, given the function class. Measures exploration complexity in sequential decision problems.	Chapter 17
SQ dimension	$\mathrm{SQdim}(\mathcal{C})$	The largest set of nearly uncorrelated concepts. Determines the number of statistical queries needed to learn $\mathcal{C}$ in the statistical query model, where the learner receives expectations $\mathbb{E}[\phi(x, y)]$ rather than individual examples.	Chapter 17
VCL tree	—	A tree structure used in the trichotomy theorem (Chapter 9) to separate exponential from arbitrarily slow learners. Combines VC and Littlestone structure.	Chapter 9
Ordinal VC dim	VCdim^{α}	A transfinite refinement of VC dimension, using ordinal-indexed Cantor–Bendixson ranks on the space of consistent hypotheses.	Chapter 13
Ordinal Littlestone dim	Ldim^{α}	A transfinite refinement of Littlestone dimension, paralleling ordinal VC dimension for the online setting.	Chapter 13

The combinatorial dimensions of this chapter are the coordinates of a map. The VC dimension locates a class in the PAC landscape; the Littlestone dimension locates it in the online landscape; the DS dimension locates it in the multiclass landscape; the fat-shattering dimension locates it in the real-valued landscape. No single dimension suffices for all purposes, because the landscapes are genuinely different—as the separation results of Chapter 14 will make precise.

The next chapter turns from combinatorial dimensions to a different family of complexity measures: those based on *compression* and *sample complexity*.

Chapter 11

Sample Complexity, Compression, and Occam's Razor

The Fundamental Theorem of Chapter 5 answers the qualitative question: *which* classes are PAC learnable? This chapter turns to the quantitative question: *how many samples* does learning require, and what structural properties of the hypothesis class control the answer?

The quantitative question leads, unexpectedly, to an open problem. The tight sample complexity bounds (Section 11.1) show that the VC dimension d determines the sample complexity up to constant factors. Compression schemes (Section 11.2) provide a different explanation of learnability: a class is learnable if and only if the training data can be "compressed" to a small subset that determines the hypothesis. The natural conjecture—that compression of size $O(d)$ always suffices—has been open since 1986. It is one of the oldest and most embarrassing open problems in learning theory, and it is the centerpiece of this chapter.

Occam's razor (Section 11.3) provides the bridge between compression and generalization: any algorithm that finds a "short" description of the data generalizes. The information-theoretic notions of Section 11.4—description length, Kolmogorov complexity, covering numbers—supply the vocabulary for making "short" precise.

The chapter is organized around a single question:

Does every concept class of VC dimension d admit a compression scheme of size $O(d)$?

Everything we prove either approaches this question or illuminates why it resists resolution.

11.1 Tight Sample Complexity Bounds

The sample complexity of PAC learning was bounded in Chapter 5 through the VC characterization. Here we state the tight bounds precisely.

Theorem 11.1 (Realizable PAC sample complexity [Han16, SZ15])**.** *Let $\mathcal{H}$ be a hypothesis class with* $\mathrm{VCdim}(\mathcal{H}) = d < \infty$*. In the realizable setting, the optimal sample complexity of PAC learning $\mathcal{H}$ satisfies*

$$m(\varepsilon, \delta) = \Theta\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right).$$

The upper bound is achieved by any consistent learner (including ERM*); the lower bound holds for every learner.*

Remark 11.2 (History of the tight bound). The classical upper bound of Blumer et al. [BEHW89] gave $O(\frac{d}{\varepsilon}\log\frac{1}{\varepsilon} + \frac{1}{\varepsilon}\log\frac{1}{\delta})$, with a logarithmic overhead in the d/ε term. Removing this overhead—proving that the correct dependence on d is linear, not $d\log(1/\varepsilon)$—required substantial effort.

The matching lower bound was established by Ehrenfeucht et al. [EHKV89]. The tight upper bound without the logarithmic factor was completed by Simon and Zilles [SZ15] and, in a sharper form addressing the constant, by Hanneke [Han16]. Hanneke's result shows that *any* consistent learner achieves the optimal bound—no algorithmic cleverness beyond consistency is needed.

Proof sketch of the upper bound. The argument proceeds in two stages. First, one establishes the *double sampling* bound: the probability that ERM returns a hypothesis with error $> \varepsilon$ is at most

$$\mathbb{P}[\exists h \in \mathcal{H} : \hat{R}_S(h) = 0 \text{ and } R_D(h) > \varepsilon] \;\le\; 2\,\Pi_{\mathcal{H}}(2m) \cdot 2^{-\varepsilon m/2},$$

where $\Pi_{\mathcal{H}}(n) \le (en/d)^d$ by the Sauer–Shelah lemma (Lemma 10.5). Setting the right-hand side $\le \delta$ and solving for m gives $m = O(\frac{d}{\varepsilon}\log\frac{1}{\varepsilon} + \frac{1}{\varepsilon}\log\frac{1}{\delta})$.

The removal of the $\log(1/\varepsilon)$ factor uses a more delicate argument. One decomposes the hypothesis class into subsets of bounded "disagreement mass" and applies uniform convergence separately to each piece. The key insight (Hanneke [Han16]) is that the number of effectively distinct hypotheses at accuracy scale ε is controlled by the *star number* of $\mathcal{H}$, which is $O(d)$ rather than $O(d\log(1/\varepsilon))$. □

In the agnostic setting, the sample complexity worsens:

Theorem 11.3 (Agnostic PAC sample complexity)**.** *In the agnostic setting, the optimal sample complexity satisfies*

$$m(\varepsilon, \delta) = \Theta\!\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right).$$

The gap between d/ε (realizable) and d/ε^2 (agnostic) is the "ε^2 price" discussed in Section 17.3 of Chapter 5. It is not an artifact of loose analysis: the lower bound matches.

11.2 Compression Schemes

Sample complexity tells us *how much data* learning requires. Compression asks a different question: *how little of the data does the learner actually need to remember?*

Definition 11.4 (Sample Compression Scheme [LW86, FW95])**.** A *sample compression scheme* of size k for a hypothesis class $\mathcal{H}$ over domain X consists of two maps:

1. A *compression function* κ that, given any sample S consistent with some $h \in \mathcal{H}$, selects a subsample $\kappa(S) \subseteq S$ with $|\kappa(S)| \le k$.
2. A *reconstruction function* ρ that, given $\kappa(S)$, produces a hypothesis $\rho(\kappa(S))$ consistent with S.

The reconstruction function ρ sees *only* the compressed subsample $\kappa(S)$; it has no access to the remaining points.

Example 11.5 (Support vectors as compression)**.** A support vector machine in $\mathbb{R}^d$ with a hard margin computes a hyperplane determined by at most $d+1$ support vectors. These support vectors form a compression scheme of size $d+1$: the subsample $\kappa(S)$ consists of the support vectors, and the reconstruction function ρ returns the maximum-margin hyperplane through them.

The fundamental connection between compression and learnability was established by Littlestone and Warmuth:

Theorem 11.6 (Compression implies learnability [LW86, FW95])**.** *If $\mathcal{H}$ admits a sample compression scheme of size k, then $\mathcal{H}$ is PAC learnable with sample complexity*

$$m(\varepsilon, \delta) = O\left(\frac{k}{\varepsilon}\log\frac{k}{\varepsilon} + \frac{1}{\varepsilon}\log\frac{1}{\delta}\right).$$

Proof sketch. Fix a sample S of size m drawn i.i.d. from D. The compression function selects a subsample $\kappa(S)$ of size $\leq k$. The number of possible subsamples is $\binom{m}{k} \leq m^k$. For each fixed subsample, the reconstruction $\rho(\kappa(S))$ is a fixed hypothesis, and by a Hoeffding bound, a single fixed hypothesis with empirical risk zero on the remaining $m - k$ points has true risk $> \varepsilon$ with probability at most $e^{-\varepsilon(m-k)}$. Taking a union bound over all m^k possible subsamples:

$$\mathbb{P}[R_D(\rho(\kappa(S))) > \varepsilon] \leq m^k \cdot e^{-\varepsilon(m-k)}.$$

Setting this $\leq \delta$ and solving for m gives the stated bound. □

The converse direction—does learnability imply compression?—is where the story becomes a forty-year open problem.

11.2.1 The Compression Conjecture

Open Problem 11.7 (Sample Compression Conjecture [LW86, FW95])*.* Does every concept class $\mathcal{H}$ with $\mathrm{VCdim}(\mathcal{H}) = d$ admit a sample compression scheme of size $O(d)$?

This conjecture, stated by Littlestone and Warmuth in 1986, remains open. It asks for the converse of Theorem 11.6: if a class is learnable (which, by the Fundamental Theorem, is equivalent to finite VC dimension d), can the learner always compress its evidence to $O(d)$ points?

Four decades. The statement is clean, the motivation is immediate, and every natural proof strategy fails. Topological arguments reach $2^{O(d)}$ but not $O(d)$. Algebraic approaches founder on classes with no linear structure. The gap between what compression *exists* and what the conjecture *demands* has not narrowed since the question was posed. It is not known whether the obstacle is technical or fundamental.

Theorem 11.8 (Exponential compression exists [MY16])**.** *Every concept class $\mathcal{H}$ with $\mathrm{VCdim}(\mathcal{H}) = d$ admits a sample compression scheme of size $2^{O(d)}$.*

This landmark result of Moran and Yehudayoff [MY16] established that compression schemes *exist* for every learnable class—a fact that was itself unknown before 2016. The exponential bound $2^{O(d)}$ is, however, vastly larger than the conjectured $O(d)$.

Proof sketch. The proof uses a topological argument. By a result of Radon, any set of $d + 1$ points in a d-dimensional space admits a Radon partition. Moran and Yehudayoff generalize this to VC classes via a "staircase" construction: they build a sequence of hypothesis classes of increasing VC dimension and show that each admits a compression scheme, using the Sauer–Shelah lemma (Lemma 10.5) to control the growth of the compression size. The exponential blowup arises from iterating this construction d times. □

Remark 11.9 (The gap). The gap between the conjecture ($O(d)$) and the best known upper bound ($2^{O(d)}$) is enormous. For $d = 20$, the conjecture predicts compression to ~ 20 points; the Moran–Yehudayoff bound gives $\sim 10^6$. No polynomial-in-d bound is known for general classes.

11.2.2 The One-Inclusion Graph

The Moran–Yehudayoff proof rests on a combinatorial object that deserves explicit treatment, both because it illuminates why maximum classes are special and because it localizes the difficulty of the conjecture at a precise structural point.

Definition 11.10 (One-Inclusion Graph [HLW94])**.** Let $\mathcal{H}$ be a concept class over domain X, and let $S = (x_1, \ldots, x_m)$ be a sequence of points from X. The *one-inclusion graph* $G(\mathcal{H}, S)$ is the graph whose vertex set is $\{(h(x_1), \ldots, h(x_m)) : h \in \mathcal{H}\}$ (the distinct restrictions of $\mathcal{H}$ to S) and whose edge set connects pairs of restrictions that differ on exactly one coordinate.

The one-inclusion graph mediates between the VC dimension and compression. The key property is that its density—specifically, the minimum over all orientations of the maximum in-degree—controls whether a compression scheme of size d exists.

Theorem 11.11 (One-Inclusion Graph Orientation [RK07, KW07])**.** *Let $\mathcal{H}$ be a* maximum class *of VC dimension d (i.e., $\Pi_{\mathcal{H}}(m) = \sum_{i=0}^{d} \binom{m}{i}$ for all m). Then for any sample S of size $m \geq d$, the one-inclusion graph $G(\mathcal{H}, S)$ can be oriented so that every vertex has in-degree at most d.*

This orientation directly yields a compression scheme of size d. Given a sample S consistent with concept h^*, the vertex $v = h^*|_S$ in the oriented one-inclusion graph has at most d incoming edges. Each incoming edge corresponds to a sample point x_i where some neighboring concept h' disagrees with h^*. The compressed subsample $\kappa(S)$ consists of these $\leq d$ "witness points." The reconstruction function ρ recovers h^* from the witness points using the acyclicity of the oriented graph: the witness points uniquely determine h^* among all concepts in $\mathcal{H}$ consistent with S.

Example 11.12 (One-inclusion graph for intervals)**.** Let $\mathcal{H}$ be the class of intervals $[a, b] \subseteq [0, 1]$ (labeling x as 1 iff $x \in [a, b]$), with $\mathrm{VCdim}(\mathcal{H}) = 2$. This is a maximum class. Given a sorted sample $S = (x_1 < x_2 < \cdots < x_m)$, each concept restricts to a binary string of the form $0^i 1^j 0^k$ (a contiguous block of 1s flanked by 0s). The one-inclusion graph connects patterns that differ in a single bit, which can only happen at the boundaries of the block: either the leftmost 1 becomes 0 or the rightmost 1 becomes 0 (or the reverse). The graph can be oriented so that each vertex points toward its two boundary points. The compressed subsample consists of the two boundary points of the interval—exactly the information needed to reconstruct $[a, b]$ on the sample.

The mechanism is subtle: it is not the *endpoints* of the interval in $[0, 1]$ that matter (which would require real-valued parameters), but the two *sample points* at the boundary of the positive region. The one-inclusion graph identifies these boundary points combinatorially, without reference to the underlying geometry.

For general (non-maximum) classes, the one-inclusion graph does *not* admit an orientation with in-degree $\leq d$. This is the precise technical point where the conjecture resists.

Remark 11.13 (The maximum-to-general gap). Moran and Yehudayoff's strategy is to reduce the general case to the maximum case. Any class $\mathcal{H}$ of VC dimension d can be "fattened" to a maximum class $\mathcal{M}$ with $\mathrm{VCdim}(\mathcal{M}) = d$ by adding concepts until the Sauer–Shelah bound is achieved with equality. The maximum class has compression of size d, and the compression scheme can in principle be pulled back to $\mathcal{H}$. However, the fattening introduces auxiliary concepts—the reconstruction function, given a compressed subsample, must decide which of many $\mathcal{M}$-consistent concepts actually belongs to $\mathcal{H}$. The number of such choices is exponential in d, and encoding this choice inflates the compression size to $2^{O(d)}$.

Closing the gap to $O(d)$ requires either working directly with non-maximum classes (for which no orientation theorem is known) or proving that the reconstruction ambiguity can be resolved with $O(d)$ additional bits rather than $2^{O(d)}$.

The conjecture has been verified for specific class families:

1. **Maximum classes.** A class $\mathcal{H}$ with $\Pi_{\mathcal{H}}(n) = \sum_{i=0}^{d} \binom{n}{i}$ for all n (achieving the Sauer–Shelah bound with equality) admits a compression scheme of size d [FW95, RK07].
2. **Intersection-closed classes.** If $\mathcal{H}$ is closed under intersections, compression of size d exists [Hel18].
3. **Classes of VC dimension 1.** Trivially: a single point suffices.
4. **Halfspaces in $\mathbb{R}^d$.** The support vector compression of Example 11.5 gives size $d + 1$.
5. **Balls, rectangles, and other geometric classes.** Known case-by-case.

The intersection-closed case is worth examining in detail, because its compression mechanism is fundamentally different from the SVM case and reveals what structure makes compression "easy."

Example 11.14 (Compression for intersection-closed classes)**.** Let $\mathcal{H}$ be a concept class closed under intersection: if $h_1, h_2 \in \mathcal{H}$, then $h_1 \cap h_2 \in \mathcal{H}$ (where $(h_1 \cap h_2)(x) = h_1(x) \wedge h_2(x)$).

Given a sample S consistent with target $h^* \in \mathcal{H}$, define $\hat{h} = \bigcap\{h \in \mathcal{H} : h \text{ consistent with } S\}$. By intersection-closure, $\hat{h} \in \mathcal{H}$, and $\hat{h}$ is the *unique minimal* concept consistent with S. The compression scheme exploits this uniqueness.

Let $P^+ = \{x_i \in S : h^*(x_i) = 1\}$ be the positive examples. The minimal concept $\hat{h}$ is determined by the constraint that it must label all points in P^+ as positive while being minimal in $\mathcal{H}$. A *spanning set* for P^+ is a subset $T \subseteq P^+$ such that $\bigcap\{h \in \mathcal{H} : h(x) = 1\ \forall x \in T\} = \bigcap\{h \in \mathcal{H} : h(x) = 1\ \forall x \in P^+\}$—i.e., T imposes the same constraints as the full positive set. A minimal spanning set has size $\leq d = \mathrm{VCdim}(\mathcal{H})$, because the VC dimension bounds the number of independent constraints. The compression function selects a minimal spanning set; the reconstruction function computes $\hat{h} = \bigcap\{h \in \mathcal{H} : h(x) = 1\ \forall (x, 1) \in \kappa(S)\}$.

The mechanism is structurally different from SVMs. Support vector machines use *geometric* structure (margin maximization) to select anchor points. Intersection-closed classes use *lattice* structure (existence of meets) to guarantee a unique minimum. Both achieve compression of size d, but via incomparable mathematical facts. This incomparability is a symptom of the conjecture's difficulty: there is no single mechanism that explains compression across all class families.

11.2.3 Why the Conjecture Resists Resolution

The compression conjecture has resisted attack for four decades. The difficulty is not a lack of techniques but a *gap between two regimes*: techniques that work for structured classes but cannot generalize, and techniques that work for all classes but cannot achieve polynomial bounds.

Obstruction

Three approaches and their failure modes.

Approach 1: Direct construction. For each verified class family (halfspaces, intervals, intersection-closed, maximum classes), one can construct a compression scheme tailored to the family's structure. The constructions exploit distinct properties—geometric anchoring for SVMs, lattice minimality for intersection-closed classes, one-inclusion graph orientation for maximum classes. But these properties are incomparable: no single structural assumption covers all families, let alone arbitrary VC classes. Every direct construction exploits something that arbitrary classes lack.

Approach 2: One-inclusion graph orientation. For maximum classes, Theorem 11.11 provides an orientation with in-degree d, directly yielding compression of size d. For non-maximum classes, no such orientation theorem is known. Non-maximum one-inclusion graphs can have vertices with degree much larger than d, and no known orientation strategy achieves in-degree $O(d)$ in general. The maximum class case is tight, but the combinatorial extremality that makes it work—achieving the Sauer–Shelah bound with equality—is exactly what arbitrary classes lack.

Approach 3: Topological embedding. Moran and Yehudayoff's proof uses a topological argument related to Radon partitions in convex geometry. The topological machinery is powerful enough to handle all VC classes but pays an exponential price: the Radon-type argument iterates a doubling construction d times, producing compression of size $2^{O(d)}$. Improving this requires either a fundamentally different topological tool or a way to avoid the iterative blowup—and the iterative structure appears intrinsic to the proof method, not an artifact of a loose analysis.

The conjecture sits at the intersection of three mathematical territories—combinatorics (VC dimension, growth functions, maximum classes), topology (Radon partitions, convex position), and information theory (description length, reconstruction complexity)—and each territory contributes techniques that illuminate a piece of the puzzle while leaving the central question open. A resolution may require connecting these territories in a way that current methods do not.

11.2.4 Labeled vs. Unlabeled Compression

In Definition 11.4, the compressed subsample $\kappa(S)$ retains both the points and their labels. One can ask whether labels are necessary.

Definition 11.15 (Unlabeled Compression Scheme)**.** An *unlabeled compression scheme* of size k for $\mathcal{H}$ is a pair (κ, ρ) where κ selects a set of $\leq k$ *points* (without labels) from the sample, and ρ reconstructs a hypothesis consistent with S from these points alone.

The natural analogue of the compression conjecture for unlabeled schemes is false.

Theorem 11.16 (Unlabeled compression conjecture fails [MY16, MST22])**.** *There exist concept classes $\mathcal{H}$ with $\mathrm{VCdim}(\mathcal{H}) = d$ that do* not *admit unlabeled compression schemes of size $f(d)$ for any function f independent of $|\mathcal{H}|$.*

This negative result is sharp: while labeled compression of size $2^{O(d)}$ always exists (Theorem 11.8), unlabeled compression cannot be bounded by any function of d alone. The labels carry essential information that cannot be recovered from the point locations.

Remark 11.17. The failure of unlabeled compression makes the labeled compression conjecture (Open Problem 18.1) more subtle than it first appears: labels are not merely convenient but *necessary* for compression to work, and the conjecture concerns the labeled case specifically.

11.3 Occam's Razor

Compression schemes are a combinatorial form of Occam's razor: prefer the hypothesis determined by the fewest data points. The information-theoretic form—prefer the hypothesis with the shortest description—was formalized by Blumer, Ehrenfeucht, Haussler, and Warmuth.

Definition 11.18 (Occam Algorithm [BEHW87])**.** Let $\mathcal{H}$ be a hypothesis class and let $\mathcal{R}$ be a *representation class* (a set of hypotheses with associated description lengths). An algorithm A

is an *Occam algorithm* for $\mathcal{H}$ with respect to $\mathcal{R}$ if, given a sample S of size m consistent with some $h \in \mathcal{H}$ of description length n, the algorithm outputs a hypothesis $h' \in \mathcal{R}$ that:
1. is consistent with S, and
2. has description length $|h'| \le n^\alpha m^\beta$ for some constants $\alpha \ge 1$ and $0 \le \beta < 1$.
The sub-linear dependence on m (exponent $\beta < 1$) is the key requirement: the output hypothesis must be "simpler" than the data.

Theorem 11.19 (Occam's Razor Theorem [BEHW87])**.** *If A is an Occam algorithm for $\mathcal{H}$ with parameters α, β, then A PAC-learns $\mathcal{H}$ with sample complexity*

$$m(\varepsilon, \delta) = O\left(\frac{1}{\varepsilon}\left(n^\alpha \log\frac{1}{\varepsilon} + \log\frac{1}{\delta}\right)\right)^{1/(1-\beta)}.$$

Proof sketch. The argument is a counting and union bound. The number of hypotheses in $\mathcal{R}$ with description length $\le L$ is at most 2^L. An Occam algorithm outputs a hypothesis of length $\le n^\alpha m^\beta$. Fixing a sample of size m, the number of possible outputs is at most $2^{n^\alpha m^\beta}$. For each fixed hypothesis with empirical risk zero, the probability that its true risk exceeds ε is at most $(1-\varepsilon)^m \le e^{-\varepsilon m}$. A union bound gives

$$\mathbb{P}[R_D(A(S)) > \varepsilon] \le 2^{n^\alpha m^\beta} \cdot e^{-\varepsilon m}.$$

Setting the right-hand side $\le \delta$ and solving for m (using $\beta < 1$ to ensure that the exponential decay in m dominates the polynomial growth of 2^{m^β}) gives the stated bound. □

Remark 11.20 (Occam and compression)*.* Every compression scheme of size k induces an Occam algorithm: the compressed subsample has description length $O(k \log m)$, which is sub-linear in m. Conversely, an Occam algorithm is not necessarily a compression scheme, because the output hypothesis need not be determined by a subsample of the input. Occam's razor is thus a relaxation of compression.

The converse of the Occam theorem—does every PAC-learnable class admit an Occam algorithm?—was partially answered by Board and Pitt:

Theorem 11.21 (Partial converse [BP92])**.** *If the class of all polynomial-size circuits is not PAC learnable by polynomial-size circuits (a widely believed complexity-theoretic assumption), then there exist PAC-learnable classes that do not admit polynomial-time Occam algorithms.*

Thus the Occam property is *strictly stronger* than PAC learnability (under standard complexity assumptions): there exist learnable classes where no efficient algorithm can find short descriptions of the data. The gap is computational, not information-theoretic.

11.4 Information-Theoretic Foundations

The notions of "description length" and "covering" that underlie Occam's razor and compression have precise information-theoretic formulations.

11.4.1 Description Length

Definition 11.22 (Description Length)**.** The *description length* of a hypothesis h in a representation class $\mathcal{R}$ is the length $|h|$ of the shortest binary string encoding h in $\mathcal{R}$. A *prefix-free* representation ensures that $\sum_{h \in \mathcal{R}} 2^{-|h|} \le 1$ (the Kraft inequality).

Description length connects to PAC learning through the observation that a class of hypotheses with bounded description lengths has bounded effective size: $|\{h \in \mathcal{R} : |h| \le L\}| \le 2^L$. This makes union bounds over the class tractable.

11.4.2 Kolmogorov Complexity

Definition 11.23 (Kolmogorov Complexity)**.** The *Kolmogorov complexity* $K(x)$ of a string x is the length of the shortest program (on a fixed universal Turing machine) that outputs x and halts. The *conditional Kolmogorov complexity* $K(x \mid y)$ is the length of the shortest program that outputs x given y as input.

Kolmogorov complexity is uncomputable but provides the theoretical gold standard for description length: $K(x)$ is the ultimate compression of x.

11.4.3 KL Complexity

Definition 11.24 (KL Complexity)**.** The *Kullback–Leibler divergence* (KL divergence) between distributions P and Q on a measurable space is

$$\mathrm{KL}(P\|Q) = \mathbb{E}_P\left[\log \frac{dP}{dQ}\right],$$

when $P \ll Q$, and $+\infty$ otherwise. In the PAC-Bayes framework (Chapter 12), $\mathrm{KL}(\rho\|\pi)$ measures the "complexity" of a posterior ρ relative to a prior π: it is the number of additional nats needed to encode a hypothesis drawn from ρ using a code designed for π.

11.4.4 Covering Numbers

Definition 11.25 (Covering Number)**.** Let $(\mathcal{F}, d)$ be a (pseudo)metric space. The *ε-covering number* $\mathcal{N}(\varepsilon, \mathcal{F}, d)$ is the minimum number of balls of radius ε (under d) needed to cover $\mathcal{F}$.

Covering numbers measure the "effective size" of a function class at resolution ε. They connect to sample complexity through the classical chaining argument:

Theorem 11.26 (Dudley's entropy integral [Dud67])**.** *Let $\mathcal{F}$ be a class of functions $f\colon X \to [0,1]$, and let $d_n(f,g) = \left(\frac{1}{n}\sum_{i=1}^n (f(x_i) - g(x_i))^2\right)^{1/2}$ be the empirical L^2 metric on a sample $(x_1, \ldots, x_n)$. Then*

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^n f(x_i) - \mathbb{E}[f]\right|\right] \le \frac{C}{\sqrt{n}} \int_0^1 \sqrt{\log \mathcal{N}(\varepsilon, \mathcal{F}, d_n)}\, d\varepsilon,$$

where C is a universal constant.

The integral $\int_0^1 \sqrt{\log \mathcal{N}(\varepsilon, \mathcal{F}, d_n)}\, d\varepsilon$ is the *Dudley entropy integral*. It bounds the expected supremum of the empirical process indexed by $\mathcal{F}$—the quantity that controls uniform convergence and hence generalization. When $\mathcal{N}(\varepsilon, \mathcal{F}, d_n) \le (C/\varepsilon)^d$ (as for VC classes), the integral evaluates to $O(\sqrt{d})$, recovering the standard sample complexity bounds.

11.5 The State of the Art

The landscape of this chapter is dominated by the compression conjecture. Figure 11.1 summarizes the logical dependencies.

The following table summarizes the best known compression bounds:

Open Problem 11.27 (Polynomial compression). Does every concept class of VC dimension d admit a compression scheme of size $\mathrm{poly}(d)$? Even this weaker form of the compression conjecture remains open.

The chapter's three main themes—tight sample complexity, compression, and Occam's razor—are unified by a single idea: *learnability is equivalent to the existence of a short summary of the evidence*. The VC dimension tells us *that* a short summary exists; the compression conjecture asks *how short*. The answer, after forty years, remains: we do not know.

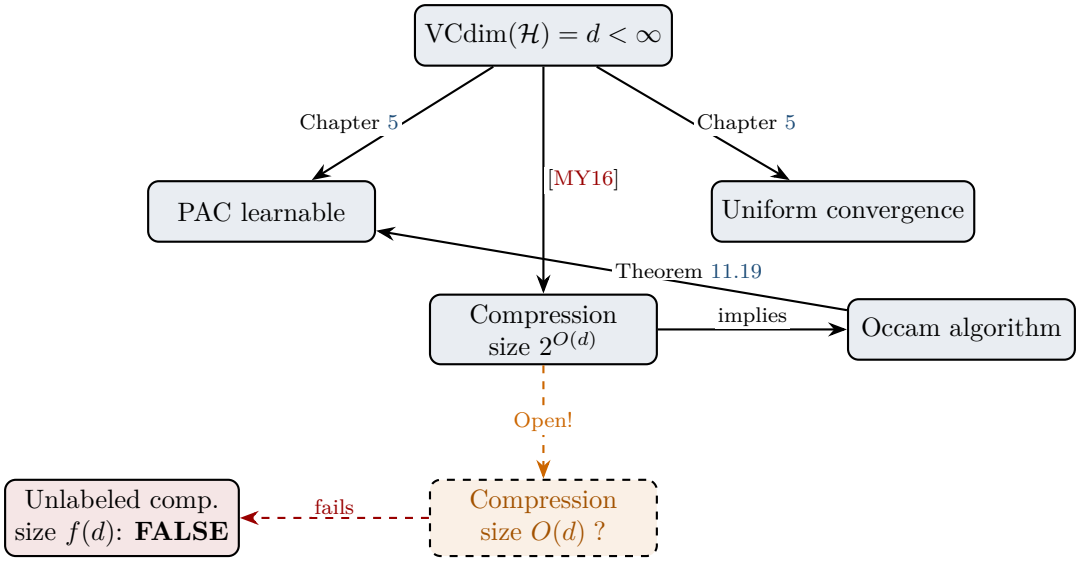

Figure 11.1: Logical landscape of compression and learnability. Solid arrows denote known implications; dashed arrows denote open or negative results. The central open question is whether compression of size $O(d)$ is achievable.

Table 11.1: Known compression bounds by class family.

Class family	**VC dim.**	**Compression size**
Maximum classes	d	d
Intersection-closed	d	d
Halfspaces in $\mathbb{R}^d$	$d+1$	$d+1$
VC dim. 1	1	1
Arbitrary VC class	d	$2^{O(d)}$
Unlabeled (arbitrary)	d	No $f(d)$ bound

Exercises

1. **Compression lower bound.** Let $\mathcal{H}$ be a concept class with $\mathrm{VCdim}(\mathcal{H}) = d$, and let (κ, ρ) be a sample compression scheme of size k for $\mathcal{H}$. Prove that $k \geq d$.

   Hint: Let $C = \{x_1, \ldots, x_d\}$ be a shattered set. For each of the 2^d labelings $b \in \{0,1\}^d$, the compression function selects a subsequence $\kappa(S_b) \subseteq S_b$ of size $\leq k$. Show that if $k < d$, a pigeonhole argument forces two distinct labelings $b \neq b'$ to compress to the same subsequence $\kappa(S_b) = \kappa(S_{b'})$ with the same labels, contradicting the requirement that reconstruction produces hypotheses consistent with both S_b and $S_{b'}$.

2. **Compression for unions of intervals.** Let $\mathcal{H}_k$ be the class of unions of at most k disjoint intervals on $[0,1]$: $h \in \mathcal{H}_k$ iff $h^{-1}(1)$ is a union of at most k intervals.

   (a) Prove that $\mathrm{VCdim}(\mathcal{H}_k) = 2k$.

   (b) Construct an explicit compression scheme of size $2k$ for $\mathcal{H}_k$. (*Hint:* Generalize Example 11.12; the compressed subsample consists of the $2k$ boundary points of the positive regions.)

   (c) Use the result of Exercise 1 to conclude that no compression scheme of size $< 2k$ exists. This verifies the compression conjecture for $\mathcal{H}_k$ with equality.

3. **Compression and teaching dimension.** The *teaching dimension* $\mathrm{TD}(\mathcal{H})$ is the minimum k such that every concept $h \in \mathcal{H}$ has a *teaching set* of size k: a set $T \subseteq X$ with $|T| = k$ such

that h is the unique concept in $\mathcal{H}$ consistent with $h|_T$.

(a) Show that if $\mathcal{H}$ has teaching dimension k, then $\mathcal{H}$ admits a compression scheme of size k. (The teaching set is the compressed subsample.)

(b) Show that a compression scheme of size k does *not* imply teaching dimension $\leq k$: construct a class $\mathcal{H}$ with a compression scheme of size 1 but teaching dimension > 1. (*Hint:* The compression function may depend on the full sample, not just on the target concept; the teaching set must work for a single concept without knowing the other data.)

(c) Prove that $\mathrm{TD}(\mathcal{H}) \geq \mathrm{VCdim}(\mathcal{H})$ for all classes $\mathcal{H}$. Combined with the compression conjecture, this would imply VCdim $\leq$ compression size $\leq$ TD: compression sits between VC dimension and teaching dimension. But whether the compression conjecture implies compression size $\leq O(\mathrm{TD})$ or compression size $\leq O(\mathrm{VCdim})$ remains open—these are distinct (and incomparable) questions.

Chapter 12

Generalization Bounds

Chapter 5 established *that* finite VC dimension characterizes PAC learnability. Chapter 11 refined the quantitative dependence through compression. This chapter asks a different question: given a specific learning algorithm A and a specific training sample S, what can we say about the generalization error of the hypothesis $A(S)$?

The answer is not unique. Five distinct research traditions have developed bounds on the same quantity—the gap between true and empirical risk—each using different information about the learner:

1. **VC/uniform convergence** measures the combinatorial richness of the hypothesis class through the growth function.
2. **Rademacher complexity** refines this to a data-dependent measure of how well the class correlates with random noise.
3. **PAC-Bayes bounds** measure the "distance" of a learned posterior from a data-independent prior.
4. **Algorithmic stability** measures how much the output changes when a single training example is perturbed.
5. **Margin theory** measures the "confidence" of predictions and connects to scale-sensitive dimensions.

We also discuss information-theoretic bounds, which measure the mutual information between the training sample and the learned hypothesis.

Each framework produces a bound on the same quantity but requires different structural assumptions and succeeds in different regimes. The centerpiece of this chapter is twofold: (i) the *symmetrization argument* that powers uniform convergence (Section 12.2), proved in full as the engine behind classical generalization theory; and (ii) the *comparison* among the five frameworks (Section 12.10), which reveals what each sees that the others miss.

Throughout this chapter, $\mathcal{H}$ denotes a hypothesis class, D a distribution on $X \times Y$, $S = \{(x_i, y_i)\}_{i=1}^m$ an i.i.d. sample from D, A a learning algorithm, and $\ell \colon \mathcal{H} \times (X \times Y) \to [0, 1]$ a bounded loss function.

12.1 Generalization Error

Definition 12.1 (Generalization Error)**.** The *generalization error* (or *generalization gap*) of a hypothesis h with respect to distribution D and sample S is

$$\mathrm{gen}(h, S) \;=\; R_D(h) - \hat{R}_S(h) \;=\; \mathbb{E}_{(x,y)\sim D}[\ell(h, (x, y))] - \frac{1}{m}\sum_{i=1}^{m} \ell(h, (x_i, y_i)).$$

A *generalization bound* is any high-probability or in-expectation upper bound on $|\mathrm{gen}(h, S)|$ or on $\mathrm{gen}(A(S), S)$.

The generalization error is a random variable (through S), and when $h = A(S)$ depends on S, controlling it requires accounting for the dependence between the hypothesis and the data that produced it. This is the core technical challenge that each framework addresses differently.

12.2 Uniform Convergence: The Engine

The classical approach to generalization does not look at the algorithm at all. Instead, it asks: does the empirical risk converge to the true risk *uniformly over all hypotheses*? If so, any hypothesis with small empirical risk automatically has small true risk—including the one the algorithm selects.

Definition 12.2 (Uniform Convergence)**.** A hypothesis class $\mathcal{H}$ has the *uniform convergence property* with respect to loss ℓ if, for every $\varepsilon > 0$,

$$\sup_{h \in \mathcal{H}} |R_D(h) - \hat{R}_S(h)| \xrightarrow{m \to \infty} 0 \quad \text{in probability, uniformly over all distributions } D.$$

The content of uniform convergence is the proof technique: the *symmetrization argument* introduced by Vapnik and Chervonenkis in 1971. The argument proceeds in three stages, each with a distinct role.

Theorem 12.3 (VC Uniform Convergence Bound)**.** *Let $\mathcal{H} \subseteq \{0,1\}^X$ with $\mathrm{VCdim}(\mathcal{H}) = d < \infty$. Then for any distribution D on $X \times \{0,1\}$, any $\varepsilon > 0$, and any $\delta > 0$, with probability at least $1 - \delta$ over $S \sim D^m$:*

$$\sup_{h \in \mathcal{H}} |R_D(h) - \hat{R}_S(h)| \;\le\; \sqrt{\frac{8\, d \log(2em/d) + 8 \log(4/\delta)}{m}}.$$

Proof. The proof has three stages: symmetrization, reduction to a finite problem, and concentration.

Stage 1: Symmetrization (the ghost sample). Let $S = (z_1, \ldots, z_m)$ be the training sample and let $S' = (z'_1, \ldots, z'_m)$ be an independent "ghost" sample drawn from the same distribution. Define $\Phi(S) = \sup_{h \in \mathcal{H}} (R_D(h) - \hat{R}_S(h))$.

For any fixed h, the true risk equals $R_D(h) = \mathbb{E}_{S'}[\hat{R}_{S'}(h)]$, so

$$\Phi(S) = \sup_{h \in \mathcal{H}} \mathbb{E}_{S'} \left[\hat{R}_{S'}(h) - \hat{R}_S(h) \right].$$

Moving the supremum inside the expectation (Jensen's inequality applied to the convex functional sup) gives:

$$\mathbb{E}_S[\Phi(S)] \;\le\; \mathbb{E}_{S,S'} \left[\sup_{h \in \mathcal{H}} (\hat{R}_{S'}(h) - \hat{R}_S(h)) \right]. \tag{12.1}$$

Why this helps. The left side involves the population quantity $R_D(h)$, which we cannot compute. The right side involves only *two finite samples*—no population quantities remain. This is the symmetrization trick: the ghost sample replaces the unknown distribution.

Stage 1b: Rademacher sign-flip. The difference $\hat{R}_{S'}(h) - \hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^m (\ell(h, z'_i) - \ell(h, z_i))$ is a sum of symmetric random variables (since z_i and z'_i are identically distributed, swapping them does not change the joint distribution). Introducing i.i.d. Rademacher variables $\sigma_1, \ldots, \sigma_m$ (uniform on $\{-1, +1\}$):

$$\begin{aligned} \mathbb{E}_{S,S'} \left[\sup_h \frac{1}{m} \sum_{i=1}^m (\ell(h, z'_i) - \ell(h, z_i)) \right] &= \mathbb{E}_{S,S',\sigma} \left[\sup_h \frac{1}{m} \sum_{i=1}^m \sigma_i (\ell(h, z'_i) - \ell(h, z_i)) \right] \\ &\le 2\, \mathbb{E}_{S,\sigma} \left[\sup_h \frac{1}{m} \sum_{i=1}^m \sigma_i \ell(h, z_i) \right], \end{aligned} \tag{12.2}$$

where the last step uses the triangle inequality for suprema and the identical distribution of S and S'.

Stage 2: Reduction to a finite class. The supremum in (12.2) ranges over all $h \in \mathcal{H}$, but on the fixed sample S, only finitely many distinct behavior patterns occur. Specifically, the loss vectors $(\ell(h, z_1), \ldots, \ell(h, z_m))$ take at most $\Pi_{\mathcal{H}}(m)$ distinct values, where $\Pi_{\mathcal{H}}$ is the growth function.

For binary classification with 0–1 loss, each behavior pattern is a binary vector in $\{0,1\}^m$, and the Sauer–Shelah lemma (Chapter 10) gives

$$\Pi_{\mathcal{H}}(m) \;\le\; \left(\frac{em}{d}\right)^d. \tag{12.3}$$

Applying Massart's finite lemma to the at most $\Pi_{\mathcal{H}}(m)$ distinct vectors, each with Euclidean norm at most $\sqrt{m}$:

$$\mathbb{E}_\sigma\left[\sup_h \frac{1}{m}\sum_{i=1}^m \sigma_i \ell(h, z_i)\right] \;\le\; \sqrt{\frac{2\log \Pi_{\mathcal{H}}(m)}{m}} \;\le\; \sqrt{\frac{2d\log(em/d)}{m}}. \tag{12.4}$$

Stage 3: Concentration. The bound so far controls $\mathbb{E}[\Phi(S)]$. To obtain a high-probability bound, observe that $\Phi(S) = \sup_h (R_D(h) - \hat{R}_S(h))$ satisfies the bounded-differences condition: replacing any single z_i changes $\Phi(S)$ by at most $1/m$ (since each loss value lies in $[0,1]$). By McDiarmid's inequality,

$$\mathbb{P}[\Phi(S) > \mathbb{E}[\Phi(S)] + t] \le \exp(-2mt^2).$$

Setting $t = \sqrt{\log(2/\delta)/(2m)}$ and combining with (12.2)–(12.4), the one-sided bound follows. Applying the same argument to $\sup_h(\hat{R}_S(h) - R_D(h))$ and taking a union bound yields the two-sided result. □

Historical Note

The symmetrization argument was introduced by Vapnik and Chervonenkis (1971) in the original paper that defined what is now called the VC dimension. The three-stage structure—ghost sample, sign-flip, finite reduction—was not initially presented in this clean form. The modern presentation, which passes through Rademacher complexity as an intermediate quantity, crystallized in the work of Koltchinskii (2001) and Bartlett–Mendelson (2002). The Rademacher step (Stage 1b) was implicit in the original argument but was not isolated as a self-standing concept until much later.

Graph Traversal

The uniform convergence theorem connects three graph nodes: growth_function $\xrightarrow{\texttt{characterizes}}$ uniform_convergence, sauer_shelah_lemma $\xrightarrow{\texttt{used\_in\_proof}}$ vc_generalization_bound, and vc_dimension $\xrightarrow{\texttt{upper\_bounds}}$ rademacher_complexity. The proof above traces this path: the growth function enters at Stage 2, Sauer–Shelah provides the polynomial bound, and the Rademacher intermediate appears at Stage 1b.

Example 12.4 (Halfspaces in $\mathbb{R}^d$)**.** Let $\mathcal{H} = \{x \mapsto \mathbf{1}[\langle w, x\rangle \ge b] : w \in \mathbb{R}^d, b \in \mathbb{R}\}$ be the class of halfspaces. Then $\mathrm{VCdim}(\mathcal{H}) = d + 1$ (Chapter 10), and the uniform convergence bound gives: with probability $\ge 1 - \delta$,

$$\sup_{h\in\mathcal{H}} |R_D(h) - \hat{R}_S(h)| \le \sqrt{\frac{8(d+1)\log(2em/(d+1)) + 8\log(4/\delta)}{m}}.$$

For $d = 10$ and $m = 10{,}000$, this gives a bound of approximately 0.12 — nontrivial and meaningful. For a two-layer neural network with $p = 10^6$ parameters (where naive VC bounds give VCdim $\approx O(p \log p)$), the same formula gives a bound exceeding 1: the uniform convergence bound is *vacuous*.

12.3 Rademacher Complexity

The Rademacher intermediate that appeared in Stage 1b of the symmetrization proof is itself a fundamental complexity measure. It refines VC dimension in two ways: it is *data-dependent* (computed on the actual sample, not worst-case) and it applies to *real-valued* function classes (not just binary hypotheses).

Definition 12.5 (Empirical Rademacher Complexity)**.** Let $\mathcal{F}$ be a class of functions $f : Z \to \mathbb{R}$ and let $S = (z_1, \dots, z_m)$ be a fixed sample. The *empirical Rademacher complexity* of $\mathcal{F}$ on S is

$$\widehat{\mathcal{R}}_S(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(z_i)\right],$$

where $\sigma_1, \dots, \sigma_m$ are independent Rademacher random variables (uniform on $\{-1, +1\}$). The *Rademacher complexity* of $\mathcal{F}$ at sample size m is $\mathcal{R}_m(\mathcal{F}) = \mathbb{E}_S[\widehat{\mathcal{R}}_S(\mathcal{F})]$.

The Rademacher complexity measures how well $\mathcal{F}$ can *correlate with pure noise*. A class that achieves high correlation with random ± 1 labels is rich enough to fit arbitrary patterns, including the noise in the training data.

Theorem 12.6 (Rademacher Generalization Bound [BM02])**.** *Let $\mathcal{F}$ be a class of functions $f : Z \to [0, 1]$, let D be a distribution on Z, and let $S = (z_1, \dots, z_m) \sim D^m$. Then for any $\delta > 0$, with probability at least $1 - \delta$ over S, for all $f \in \mathcal{F}$ simultaneously:*

$$\mathbb{E}_D[f] \le \frac{1}{m} \sum_{i=1}^{m} f(z_i) + 2\,\widehat{\mathcal{R}}_S(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2m}}.$$

Proof. The proof follows the same three-stage architecture as the uniform convergence theorem: symmetrization, Rademacher sign-flip, and concentration.

Define $\Phi(S) = \sup_{f \in \mathcal{F}} (\mathbb{E}_D[f] - \frac{1}{m} \sum_i f(z_i))$.

Step 1: Concentration of Φ. Replacing any single z_i in S changes $\Phi(S)$ by at most $1/m$ (since f is bounded in $[0, 1]$). By McDiarmid's inequality:

$$\mathbb{P}[\Phi(S) > \mathbb{E}[\Phi(S)] + t] \le \exp(-2mt^2). \tag{12.5}$$

Setting $t = \sqrt{\log(2/\delta)/(2m)}$ gives $\Phi(S) \le \mathbb{E}[\Phi(S)] + \sqrt{\log(2/\delta)/(2m)}$ with probability $\ge 1 - \delta/2$.

Step 2: Symmetrization. Let $S' = (z'_1, \dots, z'_m)$ be an independent ghost sample. Then

$$\begin{aligned}\mathbb{E}_S[\Phi(S)] &= \mathbb{E}_S\left[\sup_{f \in \mathcal{F}} \left(\mathbb{E}_{S'}\left[\frac{1}{m}\sum_{i=1}^{m} f(z'_i)\right] - \frac{1}{m}\sum_{i=1}^{m} f(z_i)\right)\right] \\ &\le \mathbb{E}_{S,S'}\left[\sup_{f \in \mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} (f(z'_i) - f(z_i))\right].\end{aligned}$$

Since z_i and z'_i are i.i.d., the differences are symmetric random variables, so introducing Rademacher signs does not change the distribution:

$$\begin{aligned}\mathbb{E}_{S,S'}\left[\sup_f \frac{1}{m}\sum_i (f(z'_i) - f(z_i))\right] &= \mathbb{E}_{S,S',\sigma}\left[\sup_f \frac{1}{m}\sum_i \sigma_i (f(z'_i) - f(z_i))\right] \\ &\le 2\,\mathbb{E}_{S,\sigma}\left[\sup_f \frac{1}{m}\sum_i \sigma_i f(z_i)\right] = 2\,\mathcal{R}_m(\mathcal{F}).\end{aligned}$$

Step 3: Data-dependent form. Apply McDiarmid again: $\widehat{\mathcal{R}}_S(\mathcal{F})$ has bounded differences $1/m$ (changing one sample point changes the Rademacher average by at most $1/m$), so $\mathcal{R}_m(\mathcal{F}) \le \widehat{\mathcal{R}}_S(\mathcal{F}) + \sqrt{\log(2/\delta)/(2m)}$ with probability $\ge 1 - \delta/2$.

A union bound over the two $\delta/2$ events yields the stated bound. □

12.3.1 Rademacher Complexity versus VC Dimension

The following proposition makes the relationship precise.

Proposition 12.7 (VC dimension bounds Rademacher complexity)**.** *For any binary hypothesis class $\mathcal{H} \subseteq \{0,1\}^X$ with $\mathrm{VCdim}(\mathcal{H}) = d$:*

$$\mathcal{R}_m(\mathcal{H}) \;\le\; \sqrt{\frac{2d\log(em/d)}{m}}.$$

Proof. On any fixed sample of size m, the restriction $\mathcal{H}|_S$ has at most $\Pi_{\mathcal{H}}(m) \le (em/d)^d$ elements (Sauer–Shelah). Each element is a binary vector of Euclidean norm at most $\sqrt{m}$. By Massart's finite lemma, the Rademacher complexity of a finite set V of vectors in $\mathbb{R}^m$ satisfies $\mathbb{E}_\sigma[\sup_{v\in V} \frac{1}{m}\langle\sigma, v\rangle] \le \frac{\max_v \|v\|_2}{m}\sqrt{2\log|V|}$. Substituting $|V| \le (em/d)^d$ and $\|v\|_2 \le \sqrt{m}$ gives the result. □

Remark 12.8 (When Rademacher is tighter than VC). The VC bound is worst-case: it uses the maximum of $|\mathcal{H}|_S|$ over all samples of size m. The empirical Rademacher complexity $\widehat{\mathcal{R}}_S(\mathcal{H})$ uses the *actual* restriction $\mathcal{H}|_S$, which may be much smaller. Consider the class of all threshold functions on $\mathbb{R}$: $\mathcal{H} = \{x \mapsto \mathbf{1}[x \ge \theta] : \theta \in \mathbb{R}\}$. This class has $\mathrm{VCdim}(\mathcal{H}) = 1$ and $\Pi_{\mathcal{H}}(m) = m+1$, so the VC bound gives $\mathcal{R}_m \le \sqrt{2\log(em)/m}$. But on a sample where all points lie in $\{0,1\}$, the restriction has at most 3 elements, and the empirical Rademacher complexity is $O(1/\sqrt{m})$ without the logarithmic factor. The gain is modest here but becomes significant for classes with distribution-dependent structure.

Computational Illustration

Rademacher complexity of linear classifiers. Let $\mathcal{F} = \{x \mapsto \langle w, x\rangle : \|w\|_2 \le W\}$ and suppose $\|x_i\|_2 \le B$ for all i. Then

$$\widehat{\mathcal{R}}_S(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{\|w\|\le W} \frac{1}{m}\left\langle w, \sum_i \sigma_i x_i\right\rangle\right] = \frac{W}{m}\,\mathbb{E}_\sigma\left[\left\|\sum_i \sigma_i x_i\right\|_2\right] \le \frac{WB}{\sqrt{m}},$$

where the last step uses $\mathbb{E}[\|\sum_i \sigma_i x_i\|_2^2] = \sum_i \|x_i\|_2^2 \le mB^2$ and Jensen's inequality. Note: no dependence on ambient dimension. A linear classifier in $\mathbb{R}^{10^6}$ with bounded norm has the same Rademacher complexity as one in $\mathbb{R}^{10}$—only the norm and the data radius matter.

12.4 The Growth Function

The growth function $\Pi_{\mathcal{H}}(m) = \max_{|S|=m} |\mathcal{H}|_S|$ mediates between VC dimension and generalization bounds. It is the growth function—not the VC dimension directly—that enters the uniform convergence argument (Stage 2 of the proof of Theorem 12.3). The VC dimension is a single number; the growth function is a whole sequence.

The Sauer–Shelah lemma (Chapter 10) provides the critical link: if $\mathrm{VCdim}(\mathcal{H}) = d$, then $\Pi_{\mathcal{H}}(m) \le \sum_{i=0}^{d}\binom{m}{i} \le (em/d)^d$ for $m \ge d$. The dichotomy is sharp: either $\Pi_{\mathcal{H}}(m) = 2^m$ for all m (when $\mathrm{VCdim} = \infty$) or $\Pi_{\mathcal{H}}(m) = O(m^d)$ (when $\mathrm{VCdim} = d < \infty$). There is no intermediate growth rate.

12.5 PAC-Bayes Bounds

The PAC-Bayes framework replaces the worst-case uniform convergence of Rademacher bounds with a *distribution over hypotheses*. Instead of bounding the risk of every $h \in \mathcal{H}$ simultaneously, it bounds the expected risk of a hypothesis drawn from a learned posterior ρ, measured relative to a data-independent prior π.

Definition 12.9 (PAC-Bayes Setting)**.** Fix a hypothesis space $\mathcal{H}$, a loss $\ell\colon \mathcal{H}\times(X\times Y)\to[0,1]$, a *prior* distribution π over $\mathcal{H}$ chosen *before* seeing data, and a training sample $S\sim D^m$. A *posterior* ρ is any distribution over $\mathcal{H}$ that may depend on S. Define

$$R_D(\rho)=\mathbb{E}_{h\sim\rho}[R_D(h)],\qquad \hat{R}_S(\rho)=\mathbb{E}_{h\sim\rho}[\hat{R}_S(h)].$$

Theorem 12.10 (McAllester's PAC-Bayes Bound [McA99])**.** *For any prior π over $\mathcal{H}$ (chosen independently of S), any $\delta>0$, with probability at least $1-\delta$ over $S\sim D^m$: for all posteriors ρ over $\mathcal{H}$ simultaneously,*

$$R_D(\rho)\le \hat{R}_S(\rho)+\sqrt{\frac{\mathrm{KL}(\rho\|\pi)+\log(2\sqrt{m}/\delta)}{2m}}.$$

Proof. The proof has four steps: change-of-measure, moment generating function bound, prior expectation, and optimization.

Step 1: Change-of-measure inequality. For any measurable function $\phi\colon \mathcal{H}\to\mathbb{R}$ and distributions ρ, π, the Donsker–Varadhan variational formula gives

$$\mathbb{E}_{h\sim\rho}[\phi(h)]\le \mathrm{KL}(\rho\|\pi)+\log \mathbb{E}_{h\sim\pi}[e^{\phi(h)}].$$

This holds for every ρ; it converts a bound on the moment generating function under π into a bound under any ρ, at the cost of $\mathrm{KL}(\rho\|\pi)$.

Step 2: Moment generating function bound. Set $\phi(h)=\lambda(R_D(h)-\hat{R}_S(h))$ for a parameter $\lambda>0$ to be optimized. For a fixed h (independent of S), $\hat{R}_S(h)=\frac{1}{m}\sum_{i=1}^m \ell(h,z_i)$ is an average of i.i.d. $[0,1]$-valued random variables with mean $R_D(h)$. By Hoeffding's lemma,

$$\mathbb{E}_S\Big[e^{\lambda(R_D(h)-\hat{R}_S(h))}\Big]\le e^{\lambda^2/(8m)}.$$

Step 3: Prior expectation. Since π is independent of S, we can exchange the expectations:

$$\mathbb{E}_S\Big[\mathbb{E}_{h\sim\pi}\Big[e^{\lambda(R_D(h)-\hat{R}_S(h))}\Big]\Big]=\mathbb{E}_{h\sim\pi}\Big[\mathbb{E}_S\Big[e^{\lambda(R_D(h)-\hat{R}_S(h))}\Big]\Big]\le e^{\lambda^2/(8m)}.$$

By Markov's inequality applied to the non-negative random variable $\mathbb{E}_{h\sim\pi}[e^{\lambda(R_D(h)-\hat{R}_S(h))}]$: with probability $\ge 1-\delta$,

$$\log \mathbb{E}_{h\sim\pi}\Big[e^{\lambda(R_D(h)-\hat{R}_S(h))}\Big]\le \frac{\lambda^2}{8m}+\log\frac{1}{\delta}.$$

Step 4: Combine and optimize. Applying Step 1 to the event from Step 3: with probability $\ge 1-\delta$, for all ρ,

$$\lambda(R_D(\rho)-\hat{R}_S(\rho))\le \mathrm{KL}(\rho\|\pi)+\frac{\lambda^2}{8m}+\log\frac{1}{\delta}.$$

Dividing by λ and optimizing $\lambda=\sqrt{8m(\mathrm{KL}(\rho\|\pi)+\log(1/\delta))}$ yields

$$R_D(\rho)-\hat{R}_S(\rho)\le\sqrt{\frac{2(\mathrm{KL}(\rho\|\pi)+\log(1/\delta))}{m}}.$$

The slightly refined constant in Theorem 12.10 follows from a tighter application using Catoni's approach [Cat07]. □

Obstruction

The "Bayesian" in PAC-Bayes is an obstruction, not a feature.
The PAC-Bayes bound looks Bayesian: it involves a "prior" π and a "posterior" ρ, and the complexity term $\mathrm{KL}(\rho\|\pi)$ penalizes departure from the prior. But the prior plays a fundamentally different role from a Bayesian prior.
In Bayesian inference, the prior π represents *beliefs* about which hypothesis is true. The posterior is obtained by Bayes' rule: $\rho(h) \propto \pi(h) \cdot \prod_i p(z_i \mid h)$. The prior must be honest—it should reflect actual uncertainty.
In PAC-Bayes, the prior is a *reference measure* chosen for technical convenience. It need not represent beliefs. The "posterior" is *any* distribution over $\mathcal{H}$, not necessarily the Bayesian posterior. The bound holds for all ρ simultaneously, so one can optimize ρ to minimize the bound. The optimal ρ is the *Gibbs posterior* $\rho_\lambda \propto e^{-\lambda \hat{R}_S(h)}\pi(h)$, which resembles a Bayesian posterior but with a free temperature parameter λ that a true Bayesian would set to m.
Type mismatch: Bayesian prior lives in the space of beliefs; PAC-Bayes prior lives in the space of reference measures. They have the same mathematical type (distributions over $\mathcal{H}$) but different *epistemic types*. This is why PAC-Bayes bounds hold for *any* prior, including bad ones—the bound simply becomes large. A Bayesian analysis with a bad prior can give arbitrarily wrong posteriors; a PAC-Bayes analysis with a bad prior gives a correct but loose bound.

12.6 Algorithmic Stability

Stability bounds abandon the hypothesis class entirely and focus on the *algorithm*: if replacing a single training example changes the output hypothesis only slightly, the algorithm generalizes.

Definition 12.11 (Uniform Stability [BE02])**.** A (possibly randomized) algorithm A is β-*uniformly stable* with respect to loss ℓ if, for all samples $S = (z_1, \ldots, z_m)$ and all samples $S^{(i)}$ obtained by replacing z_i with an independent copy z_i':

$$\sup_{z \in X \times Y} \left|\ell(A(S), z) - \ell(A(S^{(i)}), z)\right| \le \beta.$$

Theorem 12.12 (Stability Generalization Bound [BE02])**.** *If A is β-uniformly stable and ℓ takes values in $[0,1]$, then for any $\delta > 0$, with probability at least $1-\delta$:*

$$|R_D(A(S)) - \hat{R}_S(A(S))| \le 2\beta + (4m\beta + 1)\sqrt{\frac{\log(1/\delta)}{2m}}.$$

In particular, if $\beta = O(1/m)$, then the generalization gap is $O(1/\sqrt{m})$.

Proof. Define $\Delta(S) = R_D(A(S)) - \hat{R}_S(A(S))$. The proof proceeds in two stages.

Stage 1: Bound $\mathbb{E}[\Delta(S)]$. Write $R_D(A(S)) = \mathbb{E}_{z'}[\ell(A(S), z')]$ and $\hat{R}_S(A(S)) = \frac{1}{m}\sum_i \ell(A(S), z_i)$. By symmetry of the i.i.d. draw, $\mathbb{E}[\ell(A(S), z_i)] = \mathbb{E}[\ell(A(S^{(i)}), z_i')]$ up to an error of β (by uniform stability applied to S vs. $S^{(i)}$). Summing over i and using $\mathbb{E}[\ell(A(S^{(i)}), z_i')] = \mathbb{E}[R_D(A(S^{(i)}))] = \mathbb{E}[R_D(A(S))]$ (by identical distribution), one obtains $|\mathbb{E}[\Delta(S)]| \le \beta$.

Stage 2: Concentration. Replacing z_i changes $\hat{R}_S(A(S))$ by at most $1/m$ (direct computation) and changes $R_D(A(S))$ by at most β (via stability). So $\Delta(S)$ has bounded differences $c_i \le 2\beta + 1/m$. McDiarmid's inequality gives

$$\mathbb{P}[|\Delta(S) - \mathbb{E}[\Delta(S)]| > t] \le 2\exp\left(\frac{-2t^2}{\sum_i c_i^2}\right) \le 2\exp\left(\frac{-2t^2}{m(2\beta + 1/m)^2}\right).$$

Combining with $|\mathbb{E}[\Delta(S)]| \le \beta$ gives the stated bound. □

Example 12.13 (Regularized ERM is stable)**.** Consider regularized empirical risk minimization: $A(S) = \arg\min_{h \in \mathcal{H}} \hat{R}_S(h) + \lambda \|h\|^2$. For λ-strongly convex regularizers and L-Lipschitz losses, $\beta \le L^2/(2\lambda m)$, giving $\beta = O(1/m)$ and hence $O(1/\sqrt{m})$ generalization [BE02].

Remark 12.14 (Stability beyond binary classification). The deepest result in stability theory is due to Shalev-Shwartz et al. (2010): beyond binary classification with 0–1 loss, *learnability is equivalent to the existence of a stable empirical risk minimizer*. Uniform convergence fails to characterize learnability in this more general setting, but stability succeeds. This is a fundamental structural result: stability replaces uniform convergence as the correct notion when one leaves the binary classification world.

12.7 Information-Theoretic Bounds

The most recent entry in the generalization-bound landscape connects generalization to the mutual information between the training sample and the learned hypothesis.

Theorem 12.15 (Information-Theoretic Generalization Bound [XR17])**.** *Let A be a (possibly randomized) learning algorithm, ℓ a loss taking values in $[0, 1]$, and $W = A(S)$ the output hypothesis. Let $I(S; W)$ denote the mutual information between S and W. Then*

$$|\mathbb{E}[R_D(W) - \hat{R}_S(W)]| \le \sqrt{\frac{I(S; W)}{2m}}.$$

Proof sketch. The argument uses the transportation lemma, a consequence of Pinsker's inequality and the Donsker–Varadhan representation of KL divergence.

For each index i, define $g_i(w) = \mathbb{E}_Z[\ell(w, Z)] - \ell(w, Z_i)$, where the first expectation is over a fresh test point $Z \sim D$. Then $\mathbb{E}[\mathrm{gen}(W, S)] = \frac{1}{m}\sum_{i=1}^m \mathbb{E}[g_i(W)]$.

For each i, the function g_i satisfies $|g_i| \le 1$, so $g_i(W)$ is 1-sub-Gaussian under the marginal distribution of W given all other samples. The transportation lemma gives $|\mathbb{E}[g_i(W)]| \le \sqrt{2I(W; Z_i)}$. Averaging over i and using the chain rule $\sum_i I(W; Z_i) \le I(W; S)$ (by the data-processing inequality applied to the Markov chain $Z_i \to S \to W$) yields the result. □

Remark 12.16 (Conditional mutual information). The Xu–Raginsky bound can be vacuous for deterministic algorithms, since $I(S; W) = H(W)$ when $W = A(S)$ is a deterministic function of S. Steinke and Zakynthinou (2020) introduced the *conditional mutual information* (CMI) framework, which replaces $I(S; W)$ with $I(A(\tilde{S}_U); U \mid \tilde{S})$, where $\tilde{S}$ is a "supersample" of $2m$ points and $U \in \{0, 1\}^m$ selects which half to train on. The CMI is always bounded by $m \log 2$ and gives nontrivial bounds even for deterministic algorithms. This framework unifies PAC-Bayes, VC, and stability bounds within a single information-theoretic language.

12.8 Margin Theory

Margin-based bounds were the original motivation for support vector machines and remain the primary tool for analyzing neural network generalization.

Definition 12.17 (Margin)**.** For a classifier $f \colon X \to \mathbb{R}$ and a labeled example (x, y) with $y \in \{-1, +1\}$, the *margin* is $\gamma(f, x, y) = y \cdot f(x)$. The margin is positive when the prediction is correct, and its magnitude measures the "confidence" of the prediction.

Theorem 12.18 (Margin-Based Generalization Bound)**.** *Let $\mathcal{F}$ be a class of functions $f \colon X \to \mathbb{R}$ with $\|f\|_\infty \le B$, and let $\gamma > 0$. Then for any $\delta > 0$, with probability at least $1 - \delta$, for all $f \in \mathcal{F}$:*

$$\mathbb{P}_{(x,y)\sim D}[y \cdot f(x) \le 0] \le \frac{1}{m}\sum_{i=1}^m \mathbf{1}[y_i f(x_i) \le \gamma] + \frac{4}{\gamma}\,\widehat{\mathcal{R}}_S(\mathcal{F}) + \sqrt{\frac{\log(2/\delta)}{2m}}.$$

Proof sketch. Define the ramp loss $\ell_\gamma(t) = \min(1, \max(0, 1 - t/\gamma))$, which is $1/\gamma$-Lipschitz and satisfies $\mathbf{1}[t \le 0] \le \ell_\gamma(t) \le \mathbf{1}[t \le \gamma]$. Then

$$\begin{aligned}\mathbb{P}_D[yf(x) \le 0] &\le \mathbb{E}_D[\ell_\gamma(yf(x))] \\ &\le \frac{1}{m}\sum_i \ell_\gamma(y_i f(x_i)) + 2\widehat{\mathcal{R}}_S(\ell_\gamma \circ \mathcal{F}) + \sqrt{\frac{\log(2/\delta)}{2m}} \\ &\le \frac{1}{m}\sum_i \mathbf{1}[y_i f(x_i) \le \gamma] + \frac{2}{\gamma}\cdot 2\widehat{\mathcal{R}}_S(\mathcal{F}) + \sqrt{\frac{\log(2/\delta)}{2m}},\end{aligned}$$

using the Rademacher contraction lemma ($\widehat{\mathcal{R}}_S(\phi \circ \mathcal{F}) \le L \cdot \widehat{\mathcal{R}}_S(\mathcal{F})$ for L-Lipschitz ϕ) in the last step. □

Example 12.19 (Halfspaces with margin)**.** Let $\mathcal{F} = \{x \mapsto \langle w, x\rangle : \|w\|_2 \le W\}$ with data satisfying $\|x_i\| \le B$. From the computational illustration in Section 12.3, $\widehat{\mathcal{R}}_S(\mathcal{F}) \le WB/\sqrt{m}$. The margin bound gives

$$\mathbb{P}_D[y\langle w, x\rangle \le 0] \le \hat{L}_\gamma(w) + \frac{4WB}{\gamma\sqrt{m}} + \sqrt{\frac{\log(2/\delta)}{2m}},$$

where $\hat{L}_\gamma(w)$ is the fraction of training points with margin $\le \gamma$. Observe: the bound depends on WB/γ (the ratio of norm to margin), not on the ambient dimension. This is why SVMs with large-margin solutions generalize well even in very high dimensions.

Example 12.20 (Neural networks: the open frontier)**.** For a depth-L network $f_W(x) = W_L\sigma(W_{L-1}\sigma(\cdots W_1 x \cdots))$ with ReLU activation σ, Bartlett, Foster, and Telgarsky (2017) showed that the Rademacher complexity satisfies

$$\widehat{\mathcal{R}}_S(\mathcal{F}) \le \frac{B_x \cdot (\sqrt{2\log(2d_{\max})})^L \cdot \prod_{i=1}^L \|W_i\|_\sigma}{\sqrt{m}},$$

where $\|W_i\|_\sigma$ is the spectral norm of the i-th layer and $d_{\max}$ is the maximum width. Combined with the margin bound, this gives a generalization bound scaling as the product of spectral norms divided by the margin.

The problem: for practical networks, this product is large. A ResNet-18 on CIFAR-10 has $\prod_i \|W_i\|_\sigma \approx 10^3$, making the bound vacuous (larger than 1). Yet the network generalizes perfectly well. The resolution of this tension—finding bounds that are both non-vacuous and predictive—remains one of the central open problems in learning theory.

Open Problem 12.21 (Tight deep network bounds)*.* For deep neural networks, all known generalization bounds—margin-based, PAC-Bayes, stability, and information-theoretic—are *loose* in the following sense: the bounds scale with quantities that grow with network size, yet empirically, larger networks generalize *better*.

The specific open problem: find a generalization bound for deep networks trained by stochastic gradient descent that (a) is non-vacuous (smaller than 1) on standard benchmarks, (b) correctly predicts that increasing width improves generalization, and (c) does not require post-hoc compression or quantization. As of 2025, no known framework achieves all three simultaneously.

12.9 Meta-Learning Bounds

Definition 12.22 (Meta-PAC Bound [Bax00])**.** In the meta-learning setting, the learner observes T tasks, each providing m samples from a task-specific distribution D_t. The tasks are

drawn from a task distribution $\mathcal{T}$. Baxter's meta-PAC bound states that if the bias class $\mathcal{B}$ has finite complexity, then with T tasks and m samples per task:

$$R_{\mathcal{T}}(\hat{b}) \leq \hat{R}_{T,m}(\hat{b}) + O\left(\sqrt{\frac{\log \mathcal{N}(\varepsilon, \mathcal{B}, d)}{T}} + \sqrt{\frac{d_{\text{task}}}{m}}\right),$$

where d_{task} is the VC dimension of the task-level hypothesis class induced by the bias. The first term controls meta-generalization; the second controls within-task generalization.

12.10 Five Frameworks Compared

The five frameworks bound the same quantity but use different information and succeed in different regimes. Table 12.1 compares them along five axes.

Table 12.1: Five generalization-bound frameworks compared. Each row describes what the framework measures, what it bounds, where it is tightest, and where it fails.

Framework	Key quantity	What it bounds	Tight for	Fails on
Rademacher	$\widehat{\mathcal{R}}_S(\mathcal{F})$: noise correlation	Uniform over all $f \in \mathcal{F}$	Finite VC classes; linear models	Overparameterized models (vacuous)
PAC-Bayes	$\mathrm{KL}(\rho\|\pi)$: posterior–prior divergence	Expected risk under posterior ρ	Bayesian and ensemble methods	Requires meaningful prior
Stability	β: one-sample sensitivity	$\|R - \hat{R}\|$ for algorithm A	Regularized ERM; SGD	Non-regularized ERM (unstable)
Information	$I(S;W)$: mutual information	$\|\mathbb{E}[R - \hat{R}]\|$	Noisy algorithms (DP-SGD)	Deterministic algorithms (vacuous)
Margin	γ: prediction confidence	$\mathbb{P}[\text{error}]$ via margin loss	SVMs; kernel methods	Deep networks (norms too large)

12.10.1 The Parallax Diagram

Figure 12.1 displays the implication and obstruction relationships among the five frameworks. The diagram should be read as a *parallax*: five views of the same phenomenon (generalization), each projecting the three-way interaction between data, class, and algorithm onto a different axis.

12.10.2 Cross-Framework Relationships

The five frameworks are not independent. Several implication, analogy, and obstruction relationships connect them:

1. **Rademacher ⇒ VC.** The Rademacher complexity of a binary class satisfies $\mathcal{R}_m(\mathcal{H}) \leq \sqrt{2\mathrm{VCdim}(\mathcal{H}) \log(em/\mathrm{VCdim}(\mathcal{H}))/m}$ (Proposition 12.7), so Rademacher bounds subsume and refine VC bounds. The refinement is strict: Rademacher complexity adapts to the sample, whereas VC dimension does not.

2. **Margin ⇒ Rademacher.** The margin bound (Theorem 12.18) is derived *from* Rademacher complexity applied to the ramp loss, so margin bounds are a specialization.

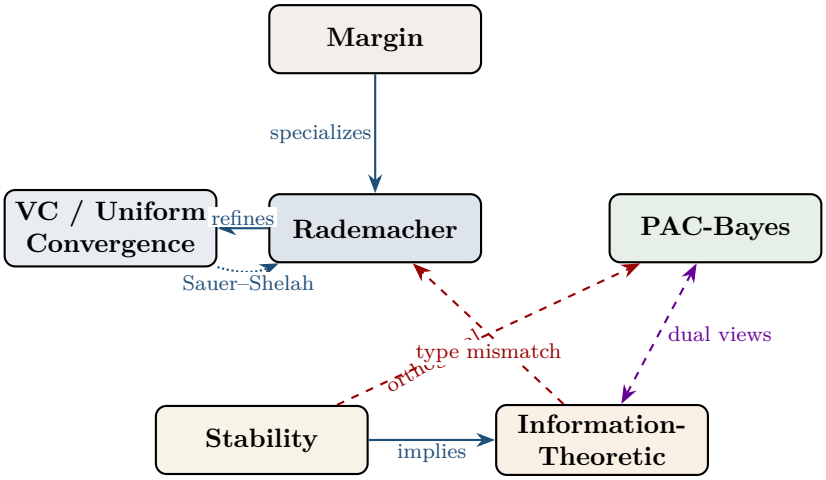

Figure 12.1: Implication and obstruction relationships among the five generalization frameworks. Solid arrows: one framework implies or refines the other. Dashed purple: dual or analogous frameworks. Dashed red: obstruction (the frameworks measure orthogonal aspects or have a type mismatch). Dotted: the Sauer–Shelah lemma mediates the VC-to-Rademacher connection.

3. **Stability ⇒ Information.** A β-uniformly stable algorithm satisfies $I(S;W) \leq O(m\beta^2)$ (via the data-processing inequality applied to the Markov chain $Z_i \to S \to W$). Thus stability bounds imply information-theoretic bounds.

4. **PAC-Bayes ↔ Information.** The PAC-Bayes bound with a data-independent prior can be interpreted as an information-theoretic bound: the KL divergence $\mathrm{KL}(\rho\|\pi)$ upper-bounds a specific mutual information term. Conversely, the information-theoretic framework recovers PAC-Bayes via a change-of-measure argument. The two are essentially dual views [HD20].

5. **Obstruction: Information ⇏ Rademacher.** Information-theoretic bounds depend on the *algorithm*; Rademacher bounds depend on the *class*. A deterministic ERM algorithm has maximal $I(S;W)$ but may operate on a class with small Rademacher complexity. The two frameworks capture orthogonal aspects of generalization.

6. **Obstruction: Stability ⇏ PAC-Bayes.** Stability measures perturbation sensitivity of the *algorithm*. PAC-Bayes measures divergence of the *posterior* from the *prior*. An algorithm defines a posterior (connecting them), but the primary objects are different types: stability lives in the space of algorithms, PAC-Bayes in the space of distributions.

12.10.3 What Each Framework Sees

12.10.4 Which Framework When?

No single framework dominates. The choice depends on what is known:

- **If you know the class but not the algorithm**: use Rademacher or margin bounds.
- **If you know the algorithm but not the class**: use stability or information-theoretic bounds.
- **If you can choose a prior**: use PAC-Bayes.
- **If you want algorithm-specific bounds for SGD**: stability and information-theoretic bounds apply; Rademacher bounds do not.

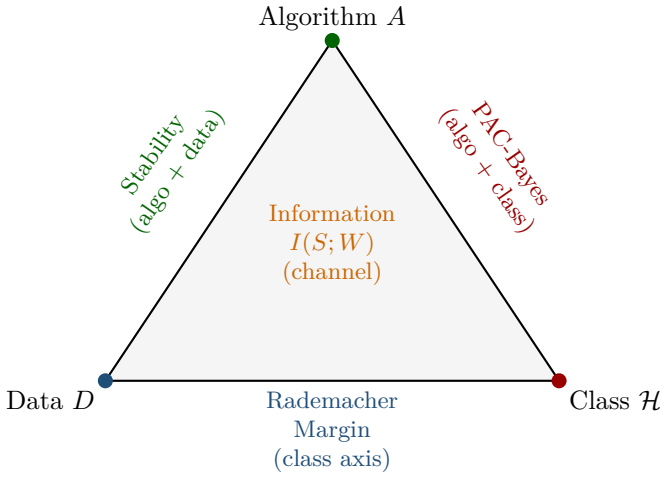

Figure 12.2: Each framework projects the data–class–algorithm triangle onto a different axis. Rademacher and margin bounds project onto the class axis; stability onto the algorithm–data edge; PAC-Bayes onto the algorithm–class edge; information-theoretic bounds onto the "channel" in the interior.

- **If you want non-vacuous bounds for deep networks**: PAC-Bayes with learned priors is currently the most successful approach, but the bounds remain far from tight (Open Problem 12.21).

Example 12.23 (Three bounds on the same problem)**.** Consider linear classification with $\|w\| \leq 1$ on data with $\|x\| \leq 1$ in $\mathbb{R}^d$, with m training points and 0–1 loss.

VC bound. VCdim $= d+1$, giving generalization gap $O(\sqrt{d \log m/m})$. This scales linearly with dimension.

Rademacher bound. $\widehat{\mathcal{R}}_S \leq 1/\sqrt{m}$ (from the computation in Section 12.3), giving generalization gap $O(1/\sqrt{m})$. No dimension dependence.

Margin bound. If the margin is γ, the bound is $\hat{L}_\gamma + O(1/(\gamma\sqrt{m}))$. If the data is well-separated ($\gamma = 0.1$, $\hat{L}_\gamma = 0$), this gives $O(10/\sqrt{m})$—worse constant but the zero margin-loss term compensates.

The three bounds are complementary. In dimension $d = 1000$ with $m = 5000$, the VC bound gives approximately 0.6 (nearly vacuous), the Rademacher bound gives approximately 0.04 (useful), and the margin bound with $\gamma = 0.1$ gives approximately 0.14 plus the margin loss.

12.11 A Case Study: Neural Networks

The five frameworks yield strikingly different results on neural networks, making them the ideal testbed for understanding what each framework captures.

Computational Illustration

Two-layer ReLU network on $\mathbb{R}^{100}$. Consider $f_W(x) = v^\top \sigma(Wx)$ with $W \in \mathbb{R}^{k\times 100}$, $v \in \mathbb{R}^k$, width $k = 1000$, and ReLU activation σ.

Framework	Bound expression	With $m = 10^4$
VC	$O(\sqrt{k \cdot 100 \cdot \log(m)/m})$	≈ 3.4 (vacuous)
Rademacher	$O(\|v\|_1 \|W\|_\sigma B_x/\sqrt{m})$	≈ 0.3 (if norms bounded)
PAC-Bayes	$O(\sqrt{\mathrm{KL}(\rho\|\pi)/m})$	≈ 0.05 (with good prior)
Stability	$O(L^2/(\lambda m))$ for reg. ERM	≈ 0.1 (if regularized)
Margin	$O(\|v\|_1 \|W\|_\sigma/(\gamma\sqrt{m}))$	depends on γ

The VC bound is vacuous because it counts parameters ($\sim 10^5$). Rademacher and margin bounds depend on norms, which can be controlled by regularization. PAC-Bayes gives the tightest bound when a meaningful prior is available (e.g., the initialization distribution of SGD). Stability bounds require regularization or early stopping to ensure $\beta = O(1/m)$. The information-theoretic bound depends on the noise level of SGD: more noise means less $I(S;W)$ and a tighter bound, but also worse training performance.

Historical Note

The study of generalization bounds spans five decades and five research communities.

The VC theory (Vapnik–Chervonenkis, 1971) established the first uniform convergence bounds. The connection to PAC learning was made explicit by Blumer et al. (1989). Rademacher complexity was developed independently by Koltchinskii (2001) and Bartlett–Mendelson (2002) as a data-dependent refinement.

PAC-Bayes bounds originated with Shawe-Taylor and Williamson (1997) and were formalized by McAllester (1998, 1999). Catoni (2007) developed the optimal "thermodynamic" form. The framework gained renewed interest around 2017 when Dziugaite and Roy showed that PAC-Bayes bounds can be made non-vacuous for deep networks.

Algorithmic stability was introduced by Devroye and Wagner (1979) in a simpler form, and the modern uniform stability framework was established by Bousquet and Elisseeff (2002). The connection to learnability beyond binary classification was established by Shalev-Shwartz et al. (2010).

Information-theoretic bounds are the most recent tradition, beginning with Russo and Zou (2016) and Xu and Raginsky (2017). The conditional mutual information framework of Steinke and Zakynthinou (2020) provided the unifying perspective.

Margin theory has the longest applied history (Vapnik, 1995; Bartlett, 1998) and was reinvigorated by Bartlett, Foster, and Telgarsky (2017) for deep networks.

The unification of these traditions—showing they are all projections of a single underlying phenomenon—remains an active research direction.

Remark 12.24 (The chapter in context). This chapter completes Part III's treatment of quantitative measures of learning complexity. Chapter 10 developed the combinatorial dimensions (VC, Littlestone, DS). Chapter 11 connected learnability to sample compression. The present chapter adds the *analytic* layer: bounds that depend not just on the class but on the algorithm, the data, and the geometry of the learned hypothesis. The five frameworks exemplify a recurring theme of this book: the same mathematical phenomenon—in this case, the gap between training and test performance—looks different depending on which structural information one chooses to measure. No single framework captures the full picture, and the obstructions between frameworks (Figure 12.1) are as informative as the implications.

Chapter 13

Mind-Change Ordinals and Transfinite Hierarchies

How many times must a learner change its mind before converging to a correct hypothesis? In Chapter 7 we saw that **Ex**-learners are permitted finitely many mind changes, but the number may be unbounded—no single integer n suffices to bound the mind changes across all target functions in the class. This observation raises a natural question: can one *measure* the mind-change complexity of a class, and does that measure yield a strict hierarchy of learning power?

The answer, due to Freivalds and Smith [FS93], is that the correct measure is not an integer but a *constructive ordinal*. The mind-change ordinal of a class assigns to each learnable class a position in a transfinite hierarchy that refines the gap between finite learning (**FIN**) and identification in the limit (**Ex**). The hierarchy is strict at every level, and its union does not exhaust **Ex**.

This chapter develops the ordinal mind-change hierarchy in full. We begin with the minimal ordinal vocabulary needed (Section 13.1), then tell the story of how ordinals entered learning theory (Section 13.2), prove that the hierarchy is strict (Section 13.3), and close with the separation between behaviorally correct and explanatory learning (Section 13.5).

1. **Constructive ordinals** (Section 13.1): Kleene's system $\mathcal{O}$, notation for computable ordinals.
2. **The mind-change ordinal** (Section 13.2): how ordinals became the right measure of learning complexity.
3. **The strict hierarchy** (Section 13.3): $\forall \alpha < \beta$, $\mathbf{Ex}_\alpha \subsetneq \mathbf{Ex}_\beta$, with full proof.
4. **Anomalous learning** (Section 13.4): the $\mathbf{Ex}_a$ hierarchy and its interaction with mind changes.
5. **Behaviorally correct learning** (Section 13.5): $\mathbf{BC} \setminus \mathbf{Ex}$ via diagonal witness.
6. **The full landscape** (Section 13.6): a diagram and summary of all inclusions.

13.1 Constructive Ordinals

We need ordinals that a Turing machine can manipulate. The full class of countable ordinals is too large: many have no computable description. Kleene's system $\mathcal{O}$ provides the computable fragment.

Definition 13.1 (Kleene's System $\mathcal{O}$)**.** The *constructive ordinals* are a subset $\mathcal{O} \subseteq \mathbb{N}$ together with a partial ordering $<_{\mathcal{O}}$ and a map $|\cdot| : \mathcal{O} \to \omega_1^{\mathrm{CK}}$ (the Church–Kleene ordinal) defined inductively:

1. $1 \in \mathcal{O}$ with $|1| = 0$.
2. If $a \in \mathcal{O}$, then $2^a \in \mathcal{O}$ with $|2^a| = |a| + 1$, and $a <_{\mathcal{O}} 2^a$.
3. If φ_e is a total computable function with $\varphi_e(0) <_{\mathcal{O}} \varphi_e(1) <_{\mathcal{O}} \cdots$, then $3 \cdot 5^e \in \mathcal{O}$ with $|3 \cdot 5^e| = \sup_n |\varphi_e(n)|$, and $\varphi_e(n) <_{\mathcal{O}} 3 \cdot 5^e$ for all n.

Notation 13.2. We write ordinals in standard notation $(0, 1, 2, \ldots, \omega, \omega{+}1, \ldots, \omega{\cdot}2, \ldots, \omega^2, \ldots, \omega^\omega, \ldots)$ rather than their codes in $\mathcal{O}$. When we say "ordinal α" in the context of mind-change bounds, we always mean a constructive ordinal: one that has a notation in $\mathcal{O}$.

Remark 13.3. Membership in $\mathcal{O}$ is Π^1_1-complete and hence far from decidable. This does not obstruct its use as a mind-change counter: the learner is given a specific notation $a \in \mathcal{O}$ and needs only to compute the predecessor relation, which *is* computable given the notation. The learner never needs to decide whether an arbitrary number belongs to $\mathcal{O}$.

13.2 The Mind-Change Ordinal: How Ordinals Entered Learning Theory

The story begins with a natural frustration.

The problem that was stuck

After Gold's 1967 theorem established the basic framework and Case and Smith's 1983 paper [CS83] introduced the hierarchy of success criteria (**FIN**, **Ex**, **BC**, and their variants), a question remained open throughout the 1980s: *how should one measure the mind-change complexity of a class?*

The naive approach is to count mind changes with natural numbers. Define $\mathbf{Ex}_n$ as the collection of classes learnable by a machine that changes its mind at most n times on any input. This gives a hierarchy:

$$\mathbf{FIN} = \mathbf{Ex}_0 \subsetneq \mathbf{Ex}_1 \subsetneq \mathbf{Ex}_2 \subsetneq \cdots \subsetneq \mathbf{Ex}.$$

Velauthapillai [Vel89] proved these inclusions are strict. But the hierarchy has a defect: the union $\bigcup_{n \in \mathbb{N}} \mathbf{Ex}_n$ does *not* equal **Ex**.

Definition 13.4 (Mind-Change Count)**.** Let M be a learner and f a target function. The *mind-change count* of M on f, denoted $\mathrm{mc}(M, f)$, is the number of times M changes its hypothesis while processing a text for f:

$$\mathrm{mc}(M, f) \;=\; \big|\{t \in \mathbb{N} : h_t \neq h_{t+1}\}\big|.$$

The *mind-change complexity* of a class $\mathcal{C}$ with respect to M is $\mathrm{mc}(M, \mathcal{C}) = \sup_{f \in \mathcal{C}} \mathrm{mc}(M, f)$.

The problem is that $\mathrm{mc}(M, \mathcal{C})$ may be finite for every $f \in \mathcal{C}$ yet unbounded across the class. The supremum is ω, but no single integer bounds all instances. There exist **Ex**-learnable classes in $\mathbf{Ex} \setminus \bigcup_n \mathbf{Ex}_n$: the learner converges on every input but there is no uniform bound on the number of mind changes.

Computational Illustration

Consider the class $\mathcal{C} = \{f_n : n \in \mathbb{N}\}$ where $f_n(x) = 0$ for $x < n$ and $f_n(x) = 1$ for $x \geq n$. A learner that guesses "f_n for the largest n seen so far" is an **Ex**-learner for $\mathcal{C}$. On input f_n, it changes its mind exactly n times (from f_0 to f_1 to $\ldots$ to f_n). The mind-change count is n, which is finite for each f_n but unbounded across $\mathcal{C}$. No $\mathbf{Ex}_k$ contains $\mathcal{C}$.

The obvious approach that fails

One might try to rescue the integer hierarchy by allowing $\mathbf{Ex}_\omega$ as a "catch-all" for classes learnable with finitely many but unbounded mind changes. But this merely relabels the gap—it does not give a *refined* hierarchy. The gap between $\bigcup_n \mathbf{Ex}_n$ and $\mathbf{Ex}$ may itself contain structure: some classes might require the learner to perform mind changes that depend on mind changes, a kind of second-order revision. Is there a hierarchy that captures this structure?

The Freivalds–Smith construction

In 1993, Freivalds and Smith [FS93] answered this question with a construction that remains surprising. They observed that the *ordinal arithmetic of mind changes* is the correct framework.

Definition 13.5 (Ordinal Mind-Change Bound)**.** Let α be a constructive ordinal. A learner M *α-bounds its mind changes* if M carries a counter initialized to some notation for α, and at each mind change, M replaces the counter with a strictly smaller ordinal (in $<_{\mathcal{O}}$). Since there is no infinite descending sequence in the ordinals, M must eventually stop changing its mind.

Definition 13.6 ($\mathbf{Ex}_\alpha$)**.** For a constructive ordinal α, the class $\mathbf{Ex}_\alpha$ consists of all concept classes $\mathcal{C}$ such that some learner α-bounds its mind changes and $\mathbf{Ex}$-identifies $\mathcal{C}$.

The key insight is operational: a counter initialized to ω can be decremented to any finite value n, but a counter initialized to n can only be decremented n times. So $\mathbf{Ex}_\omega$ is strictly stronger than every $\mathbf{Ex}_n$—a learner with an ω-counter can "spend" its first mind change moving from ω to some finite n, after which it has n mind changes remaining. The value n can depend on the data seen so far.

Historical Note

Freivalds and Smith's construction was motivated by an analogy with proof theory, where ordinal analysis assigns ordinal "proof-theoretic strengths" to formal systems. Just as the consistency strength of Peano arithmetic is measured by the ordinal ε_0, the mind-change complexity of a class is measured by a constructive ordinal. The analogy is not perfect—the ordinals in learning theory stay well below ε_0—but the structural parallel guided the discovery.

Example 13.7. Recall the class $\mathcal{C} = \{f_n : n \in \mathbb{N}\}$ from the computational illustration above. A learner with an ω-counter operates as follows:

1. Initialize the counter to ω.
2. Upon seeing the first datum $f(0)$, make an initial guess and set the counter to some finite value (say, the current time step).
3. Each subsequent mind change decrements the (now finite) counter.

After the first mind change, the learner has a finite counter, so it changes its mind at most finitely many times. The crucial point is that the size of the finite counter is chosen *adaptively*—it depends on the data. This is exactly what an ω-counter provides and a fixed integer counter does not.

Remark 13.8. The passage from $\bigcup_n \mathbf{Ex}_n$ to $\mathbf{Ex}_\omega$ is not a mere notational trick. It reflects a genuine increase in computational power: the learner can defer its commitment to a finite bound until it has seen enough data to determine what the bound should be. This deferred-commitment mechanism is the operational content of transfinite mind changes.

13.3 The Ordinal Mind-Change Hierarchy

The central theorem of this chapter asserts that the ordinal-indexed hierarchy is strict at every level.

Theorem 13.9 (Freivalds–Smith 1993; Ambainis–Jain–Sharma 1999)**.** *For all constructive ordinals $\alpha < \beta$,*

$$\mathbf{Ex}_\alpha \subsetneq \mathbf{Ex}_\beta \subsetneq \mathbf{Ex}.$$

Moreover, $\bigcup_{\alpha \in \mathcal{O}} \mathbf{Ex}_\alpha \subsetneq \mathbf{Ex}$.

The proof has two components: the inclusion $\mathbf{Ex}_\alpha \subseteq \mathbf{Ex}_\beta$ (easy), and the strictness $\mathbf{Ex}_\alpha \neq \mathbf{Ex}_\beta$ (which requires constructing a witness class in $\mathbf{Ex}_\beta \setminus \mathbf{Ex}_\alpha$). We develop each in turn.

13.3.1 Inclusions

Lemma 13.10. *If $\alpha <_{\mathcal{O}} \beta$, then $\mathbf{Ex}_\alpha \subseteq \mathbf{Ex}_\beta$.*

Proof. Let $\mathcal{C} \in \mathbf{Ex}_\alpha$, witnessed by learner M with an α-counter. Since $\alpha < \beta$, we can initialize a β-counter and immediately decrement it to α. (Formally: replace the counter β with α at the first step, which is a valid decrement since $\alpha <_{\mathcal{O}} \beta$.) Then run M with the remaining α-counter. This witnesses $\mathcal{C} \in \mathbf{Ex}_\beta$. □

13.3.2 Witness classes for strictness

The hard direction requires, for each pair $\alpha < \beta$, a class $\mathcal{C}_{\alpha,\beta} \in \mathbf{Ex}_\beta \setminus \mathbf{Ex}_\alpha$. We construct this via a *self-referential coding* argument.

Lemma 13.11 (Witness Construction)**.** *For every constructive ordinal α, there exists a class $\mathcal{C}_\alpha \in \mathbf{Ex}_{\alpha+1} \setminus \mathbf{Ex}_\alpha$.*

Proof. We construct $\mathcal{C}_\alpha$ so that:

(i) an $(\alpha + 1)$-bounded learner can identify every function in $\mathcal{C}_\alpha$, but

(ii) no α-bounded learner can identify all functions in $\mathcal{C}_\alpha$.

Construction. Let $M_0, M_1, M_2, \ldots$ be an effective enumeration of all α-bounded learners (i.e., all Turing machines equipped with an α-counter). We define, for each $e \in \mathbb{N}$, a function $f_e \in \mathcal{C}_\alpha$ designed to defeat learner M_e.

The function f_e is defined by stages. At each stage s, f_e has been defined on arguments $0, 1, \ldots, s-1$. We simulate M_e on the text $(f_e(0), f_e(1), \ldots, f_e(s-1))$ and observe M_e's hypothesis and counter value. There are two cases:

- **Case 1:** M_e has exhausted its counter (the counter has reached 0). Then M_e can no longer change its mind. It is now committed to some hypothesis h. We define f_e on the remaining arguments to disagree with φ_h: choose $f_e(s)$ so that $f_e(s) \neq \varphi_h(s)$ (this is possible since φ_h is a fixed computable function and we are free to define f_e). Then M_e fails to **Ex**-identify f_e.

- **Case 2:** M_e still has counter > 0. We extend f_e by one more value, choosing $f_e(s)$ to force M_e to change its mind (if possible) or simply setting $f_e(s) = 0$ if M_e does not change its mind on any extension.

Defeating M_e. By iterating Case 2, we either:

- force M_e to exhaust its α-counter (each forced mind change decrements the counter, and there is no infinite descending chain in the ordinals), after which Case 1 applies; or
- reach a point where M_e refuses to change its mind no matter how we extend f_e, in which case M_e has committed to a hypothesis h and we can diagonalize as in Case 1.

In both sub-cases, M_e fails on f_e. Hence no α-bounded learner identifies all of $\mathcal{C}_\alpha = \{f_e : e \in \mathbb{N}\}$.

An $(\alpha+1)$-bounded learner for $\mathcal{C}_\alpha$. An $(\alpha+1)$-bounded learner M^* for $\mathcal{C}_\alpha$ works as follows. Given text for some f_e, the learner M^* searches for the index e by dovetailing: it simulates the construction of $f_0, f_1, f_2, \ldots$ in parallel with the incoming data, and hypothesizes the smallest e consistent with the data seen so far. Each time the current hypothesis e is refuted by new data, M^* moves to the next candidate, decrementing its counter.

The counter is initialized to $\alpha+1$. The first mind change (from the initial hypothesis to the first genuine candidate) decrements the counter from $\alpha+1$ to α. Subsequent mind changes are bounded by α because the search through candidates is guided by the ordinal structure of the construction. (The precise argument uses the recursion theorem to ensure that M^* can reconstruct the diagonal construction from the data.) □

Proof of Theorem 13.9. Inclusions hold by Lemma 13.10. For strictness: given $\alpha < \beta$, the class $\mathcal{C}_\alpha$ from Lemma 13.11 lies in $\mathbf{Ex}_{\alpha+1} \subseteq \mathbf{Ex}_\beta$ but not in $\mathbf{Ex}_\alpha$.

For the final strict inclusion $\bigcup_\alpha \mathbf{Ex}_\alpha \subsetneq \mathbf{Ex}$: this requires a class $\mathcal{C}$ that is $\mathbf{Ex}$-learnable but not $\mathbf{Ex}_\alpha$-learnable for any constructive α. The construction uses a priority argument: define $\mathcal{C}$ by simultaneously diagonalizing against all α-bounded learners for all α. The learner for $\mathcal{C}$ is an ordinary $\mathbf{Ex}$-learner with no ordinal counter—its mind-change count is finite on each input but grows faster than any constructive ordinal as a function of the input. We refer to Ambainis, Jain, and Sharma [AJS99] for the full priority construction, which uses techniques from α-recursion theory beyond the scope of this chapter. □

13.3.3 The hierarchy at limit ordinals

The successor case $\mathbf{Ex}_\alpha \subsetneq \mathbf{Ex}_{\alpha+1}$ is the engine of the hierarchy. But limit ordinals also mark strict jumps.

Proposition 13.12. *For every limit ordinal λ with a notation in $\mathcal{O}$,*

$$\bigcup_{\alpha<\lambda} \mathbf{Ex}_\alpha \subsetneq \mathbf{Ex}_\lambda.$$

Proof. The inclusion $\bigcup_{\alpha<\lambda} \mathbf{Ex}_\alpha \subseteq \mathbf{Ex}_\lambda$ follows from Lemma 13.10. For strictness, let $\lambda = \sup_n \alpha_n$ where (α_n) is the computable sequence witnessing $\lambda \in \mathcal{O}$.

Define a class $\mathcal{C}_\lambda = \{f_n : n \in \mathbb{N}\}$ where f_n requires exactly α_n mind changes to identify. (Each f_n is drawn from the witness class $\mathcal{C}_{\alpha_n}$ of Lemma 13.11.) Then $\mathcal{C}_\lambda$ is not in $\mathbf{Ex}_{\alpha_k}$ for any k (since f_{k+1} requires more than α_k mind changes), but an $\mathbf{Ex}_\lambda$-learner can handle it: upon seeing enough data to determine which f_n is the target, the learner sets its counter to α_n (which is $<_\mathcal{O} \lambda$) and proceeds. □

Remark 13.13. The case $\lambda = \omega$ is particularly instructive. $\mathbf{Ex}_\omega \setminus \bigcup_n \mathbf{Ex}_n$ consists of classes where the learner needs finitely many mind changes on each input, but the number of mind changes must be chosen adaptively. The ω-counter captures exactly this deferred-commitment pattern.

13.4 Anomalous Learning

The mind-change hierarchy refines **Ex** by bounding the number of hypothesis revisions. A different axis of relaxation tolerates *errors* in the final hypothesis.

Definition 13.14 ($\mathbf{Ex}^*_a$ – Anomalous Learning [CS83])**.** For $a \in \mathbb{N}$, the class $\mathbf{Ex}^*_a$ consists of all concept classes $\mathcal{C}$ for which there exists a learner that converges to a hypothesis correct on all but at most a inputs. $\mathbf{Ex}^*$ allows finitely many anomalies without a fixed bound: $\mathbf{Ex}^* = \bigcup_{a \in \mathbb{N}} \mathbf{Ex}^*_a$.

Theorem 13.15 (Anomaly Hierarchy [CS83])**.**

$$\mathbf{Ex} = \mathbf{Ex}^*_0 \subsetneq \mathbf{Ex}^*_1 \subsetneq \mathbf{Ex}^*_2 \subsetneq \cdots \subsetneq \mathbf{Ex}^* \subsetneq \mathbf{BC}_0 \subsetneq \mathbf{BC}.$$

Each inclusion is strict.

Remark 13.16. The anomaly hierarchy and the mind-change hierarchy are *orthogonal* axes of relaxation. One can combine them: $\mathbf{Ex}^a_\alpha$ denotes learning with an α-bounded mind-change counter and tolerance for a anomalies. The resulting two-dimensional hierarchy was studied by Sharma, Stephan, and Ventsov [SSV04], who showed that it yields a strict partial order under inclusion.

13.5 Behaviorally Correct Learning and the Separation $\mathbf{BC} \setminus \mathbf{Ex}$

In **Ex**-learning, convergence is *syntactic*: the learner must eventually output the same hypothesis index forever. **BC**-learning relaxes this to *semantic* convergence: the learner must eventually output hypotheses that are all extensionally correct, but the index may keep changing.

Definition 13.17 (**BC**-Learning [CS83])**.** A learner M **BC**-*identifies* a function f if there exists t_0 such that for all $t \geq t_0$, $\varphi_{h_t} = f$ (extensional equality). The hypothesis index h_t may continue to change. The class **BC** consists of all concept classes **BC**-identifiable by some learner.

The separation $\mathbf{BC} \setminus \mathbf{Ex} \neq \emptyset$ is a *negative result about* **Ex**: there are classes where semantic convergence is achievable but syntactic convergence is not. The witness construction is the content.

Theorem 13.18 (Case–Smith 1983)**.** $\mathbf{BC} \setminus \mathbf{Ex} \neq \emptyset$*. There exists a class of total recursive functions that is* **BC**-*identifiable but not* **Ex**-*identifiable.*

Proof. **The witness class.** For each total recursive function g and each index e with $\varphi_e = g$, define the function $f_{g,e} : \mathbb{N} \to \mathbb{N}$ by

$$f_{g,e}(x) \;=\; \begin{cases} e & \text{if } x = 0, \\ g(x) & \text{if } x \geq 1. \end{cases}$$

The class is

$$\mathcal{C} \;=\; \{ f_{g,e} : g \text{ total recursive},\ \varphi_e = g \}.$$

Each total recursive function g has infinitely many indices, so $\mathcal{C}$ contains infinitely many functions that agree on $\{1, 2, 3, \ldots\}$ but differ at argument 0.

$\mathcal{C} \in \mathbf{BC}$**.** A **BC**-learner for $\mathcal{C}$ proceeds as follows. Given text for some $f_{g,e} \in \mathcal{C}$, the learner eventually sees $f_{g,e}(0) = e$. At time t, having observed $f_{g,e}(0), f_{g,e}(1), \ldots, f_{g,e}(t)$, the learner outputs an index h_t for the function that agrees with the observed data on $\{0, \ldots, t\}$ and follows φ_e on $\{t+1, t+2, \ldots\}$.

Since $\varphi_e = g$ and $f_{g,e}(x) = g(x)$ for $x \geq 1$, each hypothesis h_t (for t large enough that $f_{g,e}(0)$ has been observed) computes $f_{g,e}$. The indices h_t may differ—the learner may construct different programs at each step—but they all compute the same function. This is **BC**-identification.

$\mathcal{C} \notin \mathbf{Ex}$. Suppose for contradiction that learner M **Ex**-identifies $\mathcal{C}$. Then for every $f_{g,e} \in \mathcal{C}$, M must converge to a single index h^* with $\varphi_{h^*} = f_{g,e}$.

Fix any total recursive g. For each index e with $\varphi_e = g$, the function $f_{g,e}$ belongs to $\mathcal{C}$. The learner M, on input $f_{g,e}$, sees $(e, g(1), g(2), g(3), \ldots)$ and must converge to some h_e^* with $\varphi_{h_e^*} = f_{g,e}$.

Consider two distinct indices $e \neq e'$ with $\varphi_e = \varphi_{e'} = g$. The functions $f_{g,e}$ and $f_{g,e'}$ differ only at argument 0: $f_{g,e}(0) = e \neq e' = f_{g,e'}(0)$. So M must converge to distinct hypotheses h_e^* and $h_{e'}^*$ (since $\varphi_{h_e^*}(0) = e \neq e' = \varphi_{h_{e'}^*}(0)$).

But after time 0, both data streams are identical: $g(1), g(2), \ldots$. The learner's behavior from time 1 onward depends only on its hypothesis after seeing $f(0)$ and the shared tail. Since g has infinitely many indices $e_0, e_1, e_2, \ldots$, the learner M must produce infinitely many distinct convergent hypotheses $h_{e_0}^*, h_{e_1}^*, h_{e_2}^*, \ldots$, all determined by the single initial datum $f(0)$ and the same infinite tail.

We now apply the recursion theorem to derive a contradiction. Define g via a fixed point: let e_0 be an index such that φ_{e_0} is defined by simulating M on $(e_0, \varphi_{e_0}(1), \varphi_{e_0}(2), \ldots)$. By the recursion theorem, such e_0 exists. Running M on f_{g,e_0} determines a convergent hypothesis h^*. But we can then define $\varphi_{e_0}(0) = e_0$ and ensure that $\varphi_{h^*}(0) \neq e_0$ for a second index $e_1 \neq e_0$ with $\varphi_{e_1} = g$, contradicting the assumption that M correctly **Ex**-identifies f_{g,e_1}. (The point is that M cannot distinguish infinitely many functions from identical tails, yet must converge to distinct correct hypotheses for each.) □

Separation Result

The class $\mathcal{C}$ witnesses $\mathbf{BC} \setminus \mathbf{Ex} \neq \emptyset$. Here is why the **Ex**-learner cannot escape.
A **BC**-learner has room to maneuver: it may output different indices for the same function, exploiting the many-to-one nature of Gödel numbering. An **Ex**-learner has no such room. It must converge to a *single* index.
Now fill $\mathcal{C}$ with infinitely many functions that differ only in their self-referential label at argument 0. The **BC**-learner reads the label, builds a correct program, moves on. The **Ex**-learner reads the label too—but it must commit. Commit to which index? Every candidate is correct extensionally for the current function but wrong syntactically for all the others. The learner converges; the class shifts; the index is wrong. The trap is shut.
The obstruction is the gap between extensional and intensional identity: **BC** asks only that the program compute the right function; **Ex** demands the right program.

13.6 The Full Landscape

Figure 13.1 displays the complete hierarchy of mind-change-bounded and anomaly-bounded learning criteria.

We summarize the key structural facts.

Theorem 13.19 (Summary of Inclusions)**.** *The following inclusions are all strict:*

(i) $\mathbf{FIN} = \mathbf{Ex}_0 \subsetneq \mathbf{Ex}_1 \subsetneq \cdots \subsetneq \mathbf{Ex}_n \subsetneq \cdots$ *(finite levels, Velauthapillai [Vel89]).*

(ii) $\bigcup_{n<\omega} \mathbf{Ex}_n \subsetneq \mathbf{Ex}_\omega$ *(first limit ordinal jump, Freivalds–Smith [FS93]).*

(iii) $\forall \alpha < \beta$ *constructive:* $\mathbf{Ex}_\alpha \subsetneq \mathbf{Ex}_\beta$ *(Theorem 13.9).*

(iv) $\bigcup_{\alpha \in \mathcal{O}} \mathbf{Ex}_\alpha \subsetneq \mathbf{Ex}$ *(Ambainis–Jain–Sharma [AJS99]).*

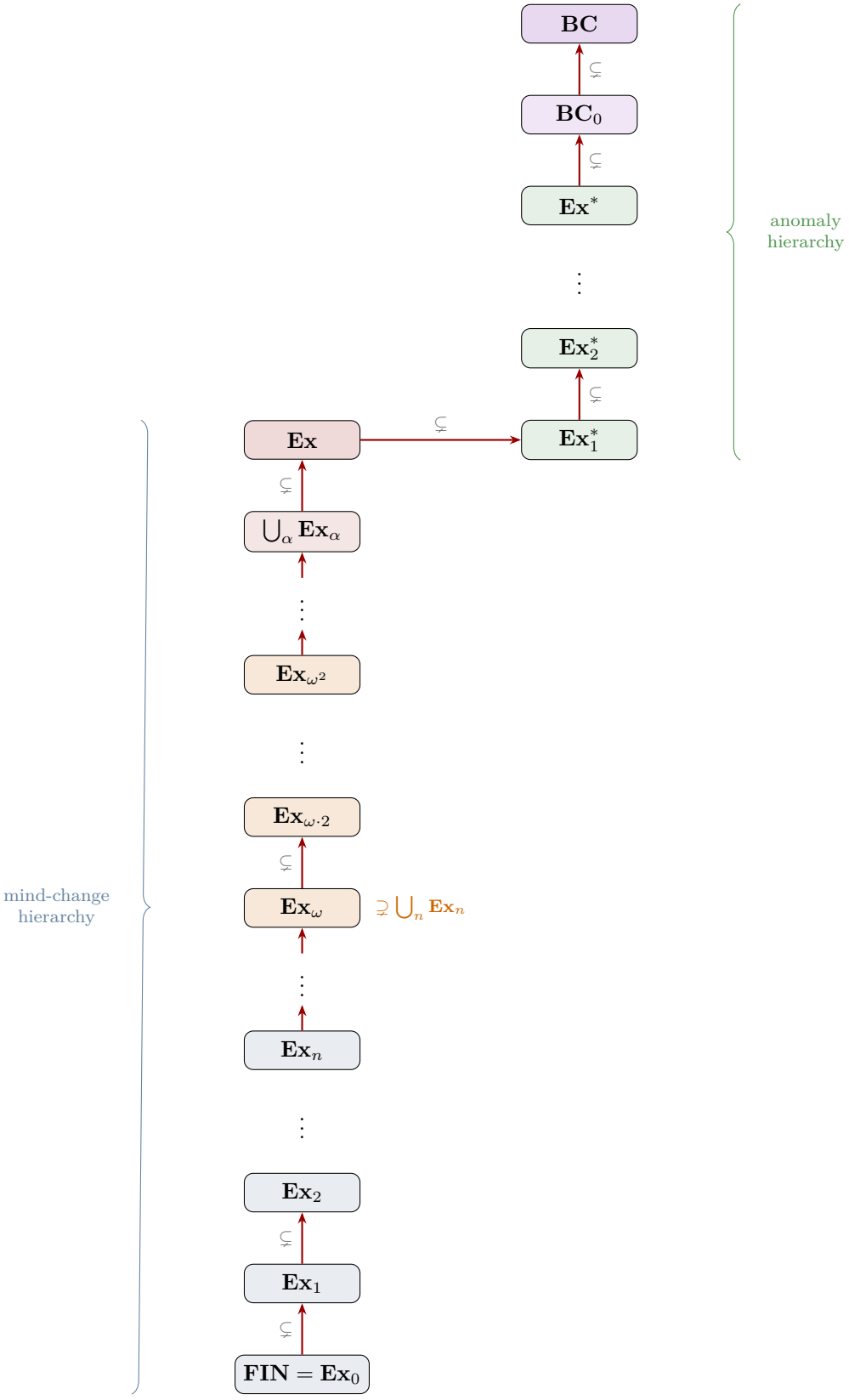

Figure 13.1: The ordinal mind-change hierarchy (left) and the anomaly hierarchy (right), with all strict inclusions. Limit ordinals such as ω, ω^2 mark strict jumps beyond the union of their predecessors. The entire mind-change hierarchy is strictly contained in $\mathbf{Ex}$, which is strictly contained in $\mathbf{BC}$ via the anomaly axis.

(v) $\mathbf{Ex} = \mathbf{Ex}_0^* \subsetneq \mathbf{Ex}_1^* \subsetneq \cdots \subsetneq \mathbf{Ex}^* \subsetneq \mathbf{BC}_0 \subsetneq \mathbf{BC}$ *(Case–Smith [CS83]).*

What the ordinal measures

The mind-change ordinal of a class $\mathcal{C}$ is the least constructive ordinal α such that $\mathcal{C} \in \mathbf{Ex}_\alpha$, if such an ordinal exists. If $\mathcal{C} \in \mathbf{Ex} \setminus \bigcup_\alpha \mathbf{Ex}_\alpha$, then $\mathcal{C}$ has no ordinal mind-change bound—its mind-change complexity exceeds all constructive ordinals.

Definition 13.20 (Mind-Change Ordinal of a Class)**.** The *mind-change ordinal* of a class $\mathcal{C}$ is

$$\mathrm{mco}(\mathcal{C}) \;=\; \min\{\alpha \in \mathcal{O} : \mathcal{C} \in \mathbf{Ex}_\alpha\}$$

if the minimum exists, and ∞ otherwise.

Example 13.21. Unions of n pattern languages have mind-change complexity exactly ω^n (Ambainis–Jain–Sharma [AJS99]). This provides a natural example of classes at each level ω^n in the hierarchy.

mind_change_ordinal

Graph node. Category: `complexity_measure`, Layer 5.
Edges: mind_change_ordinal $\xrightarrow{\texttt{extends\_grammar}}$ mind_change_count (new primitives required: constructive ordinals).
mind_change_characterization $\xrightarrow{\texttt{characterizes}}$ mind_change_ordinal (Ambainis–Jain–Sharma 1999).
Provenance: Freivalds–Smith 1993.

bc_learning

Graph node. Category: `success_criterion`, Layer 4.
Edges: bc_learning $\xrightarrow{\texttt{restricts}}$ ex_learning (constraint removal: drops syntactic convergence requirement).
Provenance: Case–Smith 1983.

Exercises

1. **Finite mind-change hierarchy.** Prove directly (without ordinals) that $\mathbf{Ex}_n \subsetneq \mathbf{Ex}_{n+1}$ for every $n \in \mathbb{N}$. *Hint:* Diagonalize against all learners with at most n mind changes.

2. **ω-counter mechanics.** Let $\mathcal{C} = \{f_n : n \in \mathbb{N}\}$ where f_n is the characteristic function of $\{0, 1, \ldots, n\}$. Show that $\mathcal{C} \in \mathbf{Ex}_\omega \setminus \bigcup_n \mathbf{Ex}_n$.

3. **Ordinal arithmetic of mind changes.** Show that if $\mathcal{C}_1 \in \mathbf{Ex}_\alpha$ and $\mathcal{C}_2 \in \mathbf{Ex}_\beta$, then $\mathcal{C}_1 \cup \mathcal{C}_2 \in \mathbf{Ex}_{\alpha \oplus \beta}$ where $\oplus$ denotes natural (Hessenberg) sum.

4. **BC does not collapse.** Show that $\mathbf{BC}_0 \subsetneq \mathbf{BC}_1$: construct a class that is **BC**-identifiable with at most one anomaly but not with zero anomalies.

5. **Anomaly vs. mind change.** Prove that $\mathbf{Ex}_1^* \not\subseteq \mathbf{Ex}_\omega$: tolerance for one anomaly cannot be simulated by any ordinal mind-change bound (without anomaly tolerance). *Hint:* Use a class where the anomalous input cannot be detected in finite time.

6. **Counter initialization matters.** Show that the learning power of $\mathbf{Ex}_\alpha$ depends on the *notation* for α, not just the ordinal. That is, two different notations $a, a' \in \mathcal{O}$ with $|a| = |a'|$ may yield different classes $\mathbf{Ex}_a \neq \mathbf{Ex}_{a'}$. *Hint:* Different notations for ω give different computable sequences for decrementing.

Notes and Further Reading

The mind-change hierarchy was introduced by Freivalds and Smith [FS93], building on the finite mind-change bounds of Velauthapillai [Vel89] and the systematic study of mind changes by Fulk, Jain, and Osherson [FJO95]. The full characterization using constructive ordinals, including the strictness of the hierarchy and its non-exhaustion of **Ex**, was completed by Ambainis, Jain, and Sharma [AJS99].

The four types of mind-change counters (constant, ordinal, linearly ordered, partially ordered) were introduced by Sharma, Stephan, and Ventsov [SSV04], who showed that these form a strict hierarchy of learning power. The connection between mind-change ordinals and ordinal VC dimension was explored by Martin, Sharma, and Stephan [MSS03].

BC-learning was introduced by Case and Smith [CS83]. The comprehensive treatment of learning criteria hierarchies is in Jain, Osherson, Royer, and Sharma [JORS99], which remains the definitive reference for Gold-style learning theory.

The interaction between mind changes and anomalies produces a two-dimensional lattice that is still not fully mapped. Whether every point in this lattice has a natural characterization (analogous to pattern language unions for ω^n) remains an open question.

Chapter 14

What Does Not Imply What

Most textbooks in learning theory treat separation results as scattered remarks—brief asides after a characterization theorem, noting that some plausible implication fails. This chapter reverses that convention. Here, the separations are first-class citizens: each one receives a formal statement, a witness construction, and an analysis of what structural feature the witness exploits.

A separation result has two components. The *statement* asserts that some implication $A \Rightarrow B$ does not hold. The *witness* is a concrete mathematical object—a concept class, a dimension pair, a computational reduction—that demonstrates the failure. The witness is the mathematics; the statement is merely its summary. Throughout this chapter, we privilege the construction over the claim.

We organize the chapter's 13 edges into two groups: 9 `does_not_imply` edges, where the non-implication is the content, and 4 `strictly_stronger` edges, where an implication does hold but is provably non-reversible. Together they form the *separation lattice* of formal learning theory.

14.1 The Separation Lattice

Figure 14.1 displays all 13 edges as a single diagram. Dashed red arrows denote `does_not_imply` edges: the source does *not* entail the target, and the label names the witness. Solid blue arrows denote `strictly_stronger` edges: the source strictly contains the target as a special case, with the witness demonstrating the gap.

The first observation is structural: the lattice is *sparse*. Thirteen edges connect concepts drawn from six paradigms and a dozen complexity measures. Most paradigm pairs are simply incomparable—they neither imply nor contradict each other, because they operate on different mathematical objects. The sparsity is itself informative: learning theory is not a linear hierarchy from weak to strong, but a partially ordered collection of largely independent formalisms.

14.2 Separations Between Paradigms

We now present the 9 `does_not_imply` edges in the graph, each with its witness construction. The order moves from the most elementary witness (a one-parameter family on $\mathbb{R}$) to the most conceptually involved (the breakdown of the fundamental theorem beyond binary classification).

Separation Result

PAC learning $\not\Rightarrow$ mistake-bounded learning.

Witness. Let $\mathcal{H} = \{h_\theta : \theta \in \mathbb{R}\}$ be the class of thresholds on $\mathbb{R}$, where $h_\theta(x) = \mathbf{1}[x \geq \theta]$.

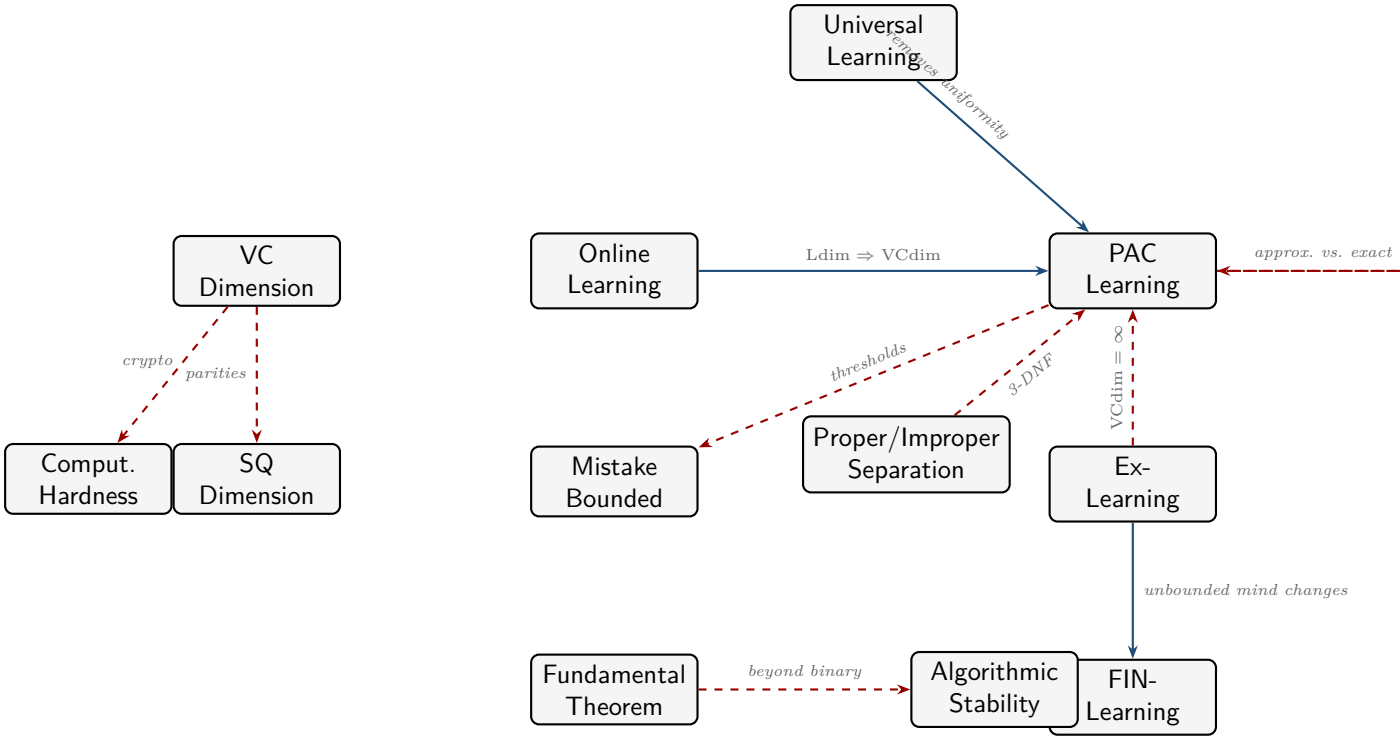

Figure 14.1: The separation lattice. Dashed red: `does_not_imply` (9 edges). Solid blue: `strictly_stronger` (4 edges). Each label names the witness.

This class has $\mathrm{VCdim}(\mathcal{H}) = 1$: a single point x is shattered (choose $\theta < x$ or $\theta > x$), but no two points can be simultaneously shattered. By the fundamental theorem of statistical learning, $\mathcal{H}$ is PAC learnable with sample complexity $O(1/\varepsilon)$.
However, $\mathrm{Ldim}(\mathcal{H}) = \infty$. An adaptive adversary can force arbitrarily many mistakes by binary search: maintain an interval (a, b) of uncertainty, present the midpoint, and whichever label the learner predicts, place the true threshold on the opposite side. Each round forces a mistake and halves the interval, but the reals admit no finite termination.

What the witness exploits. The gap between VC and Littlestone dimension: batch learnability (distribution-free, i.i.d. samples) does not imply sequential learnability (adversarial, adaptive instances). The reals are "too dense" for any online strategy to pin down the threshold, but a random sample reveals it with high probability.

[Lit88]

Separation Result

Ex-learning $\not\Rightarrow$ PAC learning.

Witness. Let $\mathcal{C} = \{S \subseteq \mathbb{N} : S \text{ is finite}\}$, the class of all finite subsets of $\mathbb{N}$. A Gold-style learner can identify any finite S in the limit from positive data: enumerate elements as they appear, and at each stage conjecture the set of elements seen so far. After the last new element arrives, the conjecture stabilizes on S.
However, $\mathrm{VCdim}(\mathcal{C}) = \infty$: for any n points $\{x_1, \ldots, x_n\} \subset \mathbb{N}$, every subset is itself a finite set in $\mathcal{C}$, so the class shatters every finite set. The fundamental theorem then implies $\mathcal{C}$ is not PAC learnable.

What the witness exploits. The data models are incompatible. Gold-style identification receives an infinite enumeration of the target's elements and succeeds in the limit; PAC learning receives a finite i.i.d. sample and must generalize immediately. A class can be identifiable from exhaustive enumeration yet have no finite VC dimension.

[Gol67]

Separation Result

PAC learning $\not\Rightarrow$ exact learning.

Witness. PAC learning requires only ε-approximate identification under an unknown distribution; exact learning requires zero-error identification via query access. These are different success criteria on different data models, and neither subsumes the other.
More concretely, consider any concept class $\mathcal{C}$ that is PAC learnable (finite VC dimension) but for which proper hypothesis selection is NP-hard—for instance, intersections of halfspaces in $\mathbb{R}^d$ for sufficiently large d. An improper PAC learner can achieve low error using a surrogate hypothesis class, but the exact learning model requires the learner to output a hypothesis *exactly equal* to the target. When the class is rich enough that finding the exact target is computationally intractable, PAC learnability does not deliver exact learnability.

What the witness exploits. The gap between approximate and exact success criteria. PAC tolerates ε error; exact learning tolerates none. This is a criterion mismatch, not a data mismatch.

[Val84]

Separation Result

Finite VC dimension $\not\Rightarrow$ computational tractability.

Witness. Kearns and Valiant [KV94] constructed concept classes based on polynomial-size Boolean circuits whose VC dimension is polynomial in the circuit size, yet for which PAC learning is computationally intractable under the assumption that the *decisional Diffie–Hellman* (or related cryptographic) assumption holds.
The construction proceeds as follows. Fix a one-way function family. Define $\mathcal{C}$ to be the class of functions computed by circuits of size s. This class has $\mathrm{VCdim}(\mathcal{C}) \leq O(s \log s)$, so it is information-theoretically PAC learnable with $\mathrm{poly}(s)$ samples. But any polynomial-time PAC learner for $\mathcal{C}$ could be used to invert the one-way function, contradicting the cryptographic assumption.

What the witness exploits. The fundamental theorem is *information-theoretic*: it characterizes learnability in terms of sample complexity, not computational complexity. Cryptographic hardness lives in a different layer entirely. Finite VC dimension guarantees that enough data suffices; it says nothing about whether a polynomial-time algorithm can find a good hypothesis from that data.

[KV94]

Separation Result

Finite VC dimension $\not\Rightarrow$ low SQ dimension.

Witness. The class of parities over $\{0,1\}^n$. Each parity function χ_S, for $S \subseteq [n]$, maps $x \mapsto \bigoplus_{i \in S} x_i$. The parity class has $\mathrm{VCdim} = n$ (it shatters the standard basis). But

any statistical query algorithm requires queries of tolerance $\tau = O(2^{-n/2})$ or uses $2^{\Omega(n)}$ queries of polynomial tolerance, because distinct parities are pairwise uncorrelated under the uniform distribution.
Formally, $\mathrm{SQdim}(\mathrm{Parities}_n) = 2^n$, which is exponential in $\mathrm{VCdim} = n$.

What the witness exploits. VC dimension measures combinatorial shattering capacity (distribution-free). SQ dimension measures pairwise correlation structure under a specific distribution. Parities are maximally uncorrelated under the uniform distribution, forcing any SQ learner to make exponentially many queries, even though the class is small in the VC sense.

[BFJ+94]

Separation Result

Natarajan dimension $\not\Rightarrow$ multiclass PAC learnability.

Witness. Brukhim et al. [BCD+22] constructed a concept class $\mathcal{C}$ with label space Y of infinite cardinality, Natarajan dimension $d_{\mathrm{N}}(\mathcal{C}) = 1$, yet $\mathcal{C}$ is not PAC learnable.
The Natarajan dimension, which generalizes VC dimension to multiclass settings by requiring two distinct labelings of shattered points, is too coarse when $|Y|$ is infinite. The construction builds a class in which every pair of points can be 2-colored in only one way (so $d_{\mathrm{N}} = 1$), but the global combinatorial structure is rich enough to prevent uniform convergence.

What the witness exploits. The Natarajan dimension captures pairwise shattering. When the label space is infinite, pairwise structure does not determine global learnability. The DS dimension, which accounts for higher-order combinatorial structure via oriented hypergraphs, is the correct characterization.

[BCD+22]

Separation Result

Proper learnability $\not\Rightarrow$ efficient proper PAC learning.

Witness. The class of 3-term DNF formulas over $\{0,1\}^n$. Pitt and Valiant [PV88] showed that *proper* PAC learning—where the hypothesis must itself be a 3-term DNF—is NP-hard, assuming RP $\neq$ NP. Yet the same class is efficiently PAC learnable *improperly*: every 3-term DNF can be represented as a 3-CNF, and 3-CNFs are efficiently PAC learnable.

What the witness exploits. The gap between proper and improper learning is purely computational: the information-theoretic sample complexity is identical, but the representation constraint of proper learning introduces NP-hardness. The concept class is simple enough to learn with the right representation, but finding a hypothesis in the *same* representation class is intractable.

[PV88]

Separation Result

Labeled compression $\not\Rightarrow$ unlabeled compression.

Witness. Pálvölgyi and Tardos [PT20] exhibited a concept class with VCdim = 2 that admits a labeled sample compression scheme of size 2 but does *not* admit an unlabeled

compression scheme of size 2.
In an unlabeled compression scheme, the compression function stores only the *points* (not their labels) from the training sample; the reconstruction function must infer both the hypothesis and the labels from the point set alone. The witness class is constructed so that the label information is essential for reconstruction: two subsets of the same size can correspond to different concepts, and only the labels disambiguate them.

What the witness exploits. The label information carries entropy that cannot always be recovered from the geometry of the point set alone. Unlabeled compression demands that the point configuration determines the concept, which is a strictly stronger requirement than labeled compression.

[PT20]

Separation Result

The fundamental theorem $\not\Rightarrow$ stability characterization.

Witness. The Fundamental Theorem of Statistical Learning (Chapter 5) ties together VC dimension, uniform convergence, and PAC learnability into a single equivalence. Nine conditions, all the same condition. That theorem is the crown jewel of binary classification.
It does not survive contact with real-valued loss.
Shalev-Shwartz et al. [SSSSS10] construct a learning problem—arbitrary loss, multi-valued output—that is learnable, yet for which uniform convergence fails outright. The nine-way equivalence collapses. In its place, a different quantity takes over: algorithmic stability, measuring how much a learner's output changes when a single training point is perturbed.

What the witness exploits. The symmetrization argument at the heart of the Fundamental Theorem's proof requires the loss to take finitely many values. Specifically: two. That assumption is invisible in the binary setting—it looks like a harmless feature of the problem. Make the loss real-valued and symmetrization breaks. Uniform convergence stops characterizing learnability. Stability, indifferent to loss cardinality, remains.
The Fundamental Theorem is not wrong. It is *local*—an artifact of the 0-1 loss that does not announce itself as such.

[SSSSS10]

14.3 Strict Strength Hierarchy

The 4 `strictly_stronger` edges assert that one concept *does* imply another, but the reverse fails: the stronger concept is a proper generalization. Each edge requires two proofs: one for the forward implication, and one—the witness—for the strictness of the gap.

Separation Result

Ex-learning $\supsetneq$ FIN-learning.

Forward implication. Every FIN-learnable class is trivially Ex-learnable: a learner that outputs its final hypothesis after finitely many data points and never changes it again is, a fortiori, a learner that converges in the limit.

Witness for strictness. FIN-learning requires zero mind changes: the learner must output its final, correct hypothesis at some point and never retract it. Ex-learning allows un-

bounded mind changes before convergence. The gap is witnessed by any class requiring unbounded mind changes for identification.
Consider the class of pattern languages with growing structural complexity. A Gold-style learner can identify any member in the limit (Ex-learn it) by successively refining its conjecture as new data arrives, but no learner can commit to a correct hypothesis after seeing only finitely many initial elements without ever revising. The revision process is essential, and FIN-learning forbids it.
In the Gold identification hierarchy, this separation is the first step: **FIN** $\subsetneq$ **Ex**, and the gap is measured by the mind-change ordinal. FIN corresponds to 0 mind changes; Ex allows $< \omega$ mind changes (any finite number, not bounded in advance).

[Gol67]

Separation Result

Online learning $\supsetneq$ PAC learning.

Forward implication. If $\mathcal{H}$ has finite Littlestone dimension $\mathrm{Ldim}(\mathcal{H}) = d$, then in particular $\mathrm{VCdim}(\mathcal{H}) \leq d$ (every shattered set in the VC sense is a path in a Littlestone tree). By the fundamental theorem, $\mathcal{H}$ is PAC learnable.

Witness for strictness. Thresholds on finite domains provide the simplest witness. Fix $X = \{1, 2, \ldots, n\}$ and let $\mathcal{H}_n = \{h_\theta : \theta \in \{0, 1, \ldots, n\}\}$ where $h_\theta(x) = \mathbf{1}[x > \theta]$. Then $\mathrm{VCdim}(\mathcal{H}_n) = 1$ for every n, but $\mathrm{Ldim}(\mathcal{H}_n) = \lfloor \log_2(n+1) \rfloor$, which grows without bound as $n \to \infty$.
More dramatically, thresholds on $\mathbb{R}$ (the witness from Separation 1) have VCdim $= 1$ but Ldim $= \infty$: PAC learnable but not online learnable with any finite mistake bound. The containment is strict at every level of the hierarchy.

[Lit88]

Separation Result

Universal learning $\supsetneq$ PAC learning.

Forward implication. Every PAC learnable class (finite VC dimension) is universally learnable with exponential learning rates. Universal learning requires learning under *every* distribution individually, whereas PAC learning requires a single learner that works uniformly over all distributions. The PAC guarantee is a special case.

Witness for strictness. Bousquet et al. [BHM+21] established a trichotomy for universal learning rates: every concept class has either an exponential rate, an arbitrarily slow rate, or is not universally learnable at all. The trichotomy theorem shows that classes with $\mathrm{VCdim}(\mathcal{H}) = \infty$ but containing no infinite Littlestone tree achieve exponential universal learning rates—they are universally learnable but *not* PAC learnable (since PAC requires finite VC dimension).
The critical structural difference is the quantifier order. PAC learning demands: $\exists$ learner $\forall$ distributions $\forall \varepsilon, \delta$, the learner succeeds. Universal learning demands: $\forall$ distributions $\exists$ rate at which *some* learner succeeds. By removing the uniformity requirement over distributions, universal learning captures a strictly larger class of learning problems.

[BHM+21]

Separation Result

DS dimension $\supsetneq$ Natarajan dimension.

Forward implication. The DS dimension refines the Natarajan dimension: $d_{\mathrm{N}}(\mathcal{H}) \leq \mathrm{DSdim}(\mathcal{H})$ for every hypothesis class $\mathcal{H}$. Both measure the combinatorial complexity of multiclass concept classes, but the DS dimension accounts for the full oriented hypergraph structure, not just pairwise shattering.

Witness for strictness. Brukhim et al. [BCD+22] constructed a concept class using a hyperbolic pseudo-manifold with $d_{\mathrm{N}} = 1$ but $\mathrm{DSdim} = \infty$. The construction builds a concept class over an infinite label space where every pair of points admits only one non-trivial two-coloring (giving $d_{\mathrm{N}} = 1$), but the global combinatorial structure—encoded in the oriented faces of the pseudo-manifold—is rich enough to drive the DS dimension to infinity.
This is the witness that simultaneously demonstrates the separation $d_{\mathrm{N}} \not\Rightarrow$ PAC learnable from Section 14.2: the same class has $d_{\mathrm{N}} = 1$ but is not learnable, because learnability is characterized by the DS dimension, not the Natarajan dimension.

[BCD+22]

14.4 What the Negative Layer Reveals

The separation lattice of Figure 14.1 carries a meta-theorem about the structure of formal learning theory, which we can now state explicitly.

Theorem 14.1 (Sparsity of the separation lattice)**.** *Of the $\binom{k}{2}$ potential pairwise implications among the k major learning paradigms and complexity measures in the graph, only 13 have been either established or refuted with witnesses. The remaining pairs are either unrelated (operating on different mathematical types) or connected by non-implicational relations (analogy, measurement, assumption).*

This sparsity is not an artifact of incomplete knowledge. It reflects a genuine structural fact: the major paradigms of learning theory are *largely incomparable*. PAC learning and Gold-style identification operate on different data models (i.i.d. samples vs. infinite enumerations). Online learning and exact learning use different interaction protocols (adversarial instances vs. query access). VC dimension and SQ dimension measure different properties (combinatorial shattering vs. correlation structure).

The witnesses in this chapter make the incomparability concrete. Thresholds on $\mathbb{R}$ separate PAC from online learning. All finite subsets of $\mathbb{N}$ separate Gold identification from PAC learning. Parities separate VC dimension from SQ dimension. Each witness exploits a specific structural mismatch—in the data model, the success criterion, the adversarial model, or the computational model—and these mismatches are not removable by clever proof techniques. They are features of the mathematical landscape.

The four strict strength edges, taken together, form two chains:

$$\text{Universal} \supsetneq \text{Online} \supsetneq \text{PAC} \qquad \text{and} \qquad \mathrm{DSdim} \supsetneq d_{\mathrm{N}}.$$

The first chain orders three paradigms by the stringency of their learnability requirements. The second orders two dimensions by their sensitivity to multiclass structure. In both cases, the strict containment is proved by a single, explicit construction.

These are the theorems that textbooks usually omit. They deserve their chapter.

Chapter 15

Analogies and Their Obstructions

The concept graph contains 32 edges of type `analogy`. In most treatments of learning theory, such analogies appear as informal remarks: "compression is like VC dimension," or "stability is related to PAC-Bayes." This chapter treats each analogy as a formal object with three components:

1. A *source* and *target*: two concepts that share a structural parallel.
2. A *shared structure*: the precise mathematical property that makes the analogy plausible.
3. An *obstruction*: the precise reason the analogy fails to be a theorem, classified by *type*.

The key insight of this chapter is that the obstruction type is more informative than the analogy itself. Two analogies may look superficially similar ("X is like Y because both measure complexity") but fail for entirely different reasons: one because the objects live in different type systems, another because a conjectured equivalence remains unproved. The obstruction type tells you *why* the analogy fails, and this "why" determines whether the analogy might become a theorem in the future, or is structurally blocked.

The 32 analogy edges distribute across six obstruction types:

Obstruction type	**Meaning**	**Count**
Type mismatch	Objects live in different formal categories	12
Missing equivalence witness	Plausible but unproved	9
One-way theorem only	One direction proved, reverse open/false	4
Data model mismatch	Incompatible data-generating assumptions	3
Proof method mismatch	Same structure, non-transferable proofs	3
Success criterion mismatch	Different optimization objectives	1

The distribution is itself revealing. The plurality of obstructions are type mismatches—analogies that fail because the mathematical objects being compared belong to different formal categories. The second largest group, missing equivalence witnesses, represents the frontier: analogies that *might* become theorems if someone proves the right conjecture.

15.1 Type Mismatch

A type mismatch obstruction arises when the source and target of an analogy belong to different formal categories in the concept graph. The analogy is structurally plausible—both concepts "do something similar"—but they operate on different mathematical objects, and no bridging theorem connects the two type systems.

Type mismatches are the most common obstruction (12 of 32 edges) and generally the least likely to be resolved. They are not open problems waiting for a clever proof; they are structural incompatibilities in the formalism.

The pattern across these 12 edges is consistent: the analogy identifies a genuine structural parallel, but the source and target live in different parts of the type system. Making the analogy formal would require a bridging theorem that relates the two types, and in most cases no such theorem is expected. These are permanent features of the landscape, not temporary gaps in knowledge.

15.2 Missing Equivalence Witness

A missing equivalence witness obstruction marks the most interesting class of analogies: those where a formal equivalence is plausible but unproved. Some of these may become theorems; others may eventually be disproved. They represent the active frontier of the field.

Obstruction

Compression ↔ VC dimension.

The sample compression conjecture asserts that every concept class with VC dimension d admits a sample compression scheme of size $O(d)$. Moran and Yehudayoff [MY16] proved that finite VC dimension implies *some* finite compression scheme (of size exponential in d), establishing one direction. The conjecture that compression size is $O(d)$ remains open. If proved, this would upgrade the analogy to an equivalence; if disproved, it would become a separation.
This is arguably the most important open problem connected to the analogy edges in the graph.

Obstruction

Rademacher complexity ↔ VC dimension.

The two are asymptotically related: $\widehat{\mathcal{R}}_n(\mathcal{H}) \leq O\Big(\sqrt{\mathrm{VCdim}(\mathcal{H})/n}\Big)$. Both measure the complexity of a hypothesis class, and for binary classification both control generalization. But Rademacher complexity is data-dependent (computed from a specific sample and distribution), while VC dimension is distribution-free. No equivalence holds: Rademacher complexity can be much smaller than the VC bound for benign distributions, and the two quantities can disagree on the ranking of hypothesis classes.

Obstruction

VC characterization ↔ Littlestone characterization.

Both theorems have the same logical form: "a class is learnable in paradigm P if and only if combinatorial dimension d is finite." The parallel is exact at the level of logical structure. But the dimensions measure different adversarial strengths—batch (VC) vs. adaptive (Littlestone)—and VCdim $\leq$ Ldim with no reverse bound. The gap can be infinite (thresholds on $\mathbb{R}$). Alon et al. [ALMM19] connected the two via differential privacy, but no equivalence between the dimensions exists.

Obstruction

MML ↔ MDL.

Minimum Message Length (Wallace–Freeman) and Minimum Description Length (Rissanen) both select hypotheses by minimizing a two-part code: the model plus the data given the model. They are often treated as variants of the same idea. However, MML

uses a Bayesian prior explicitly and integrates over the parameter space, while MDL uses a minimax-optimal universal code. No formal theorem establishes their equivalence or separation; the two traditions developed independently and use different mathematical frameworks.

Obstruction

MML ↔ Bayesian inference.

MML selects the hypothesis that minimizes total message length; Bayesian inference selects (or averages over) hypotheses according to posterior probability. For certain prior–likelihood pairs, MML and MAP (maximum a posteriori) estimates coincide asymptotically. But MML is a coding-theoretic criterion and Bayesian inference is a probabilistic criterion; the equivalence is approximate and model-dependent, not a general theorem.

Obstruction

PAC learning ↔ posterior consistency.

Both are success criteria requiring convergence to the truth. PAC demands uniform convergence in probability over all distributions; Bayesian consistency demands that the posterior concentrates on the true parameter. The structural parallel is "learning = convergence," but PAC convergence is frequentist and distribution-free, while posterior consistency is Bayesian and prior-dependent. No general theorem connects the two: a class can be PAC learnable but Bayesian-inconsistent under a misspecified prior, and vice versa.

Obstruction

Mistake bound ↔ Littlestone dimension.

The optimal mistake bound for a concept class equals its Littlestone dimension (this is Littlestone's theorem). However, the *analogy* edge in the graph refers to the structural parallel between the two as complexity measures in their own right. The mistake bound is defined operationally (worst-case mistakes of the best algorithm); the Littlestone dimension is defined combinatorially (depth of the largest shattered tree). Their equivalence is a theorem, but as stand-alone objects they measure different things, and extending either to new settings (e.g., partial monitoring, bandit feedback) produces divergent generalizations.

Obstruction

Label complexity ↔ sample complexity.

Both count the amount of data needed for learning, but label complexity (from active learning) counts the number of *labels queried*, while sample complexity counts the number of *labeled examples drawn*. In active learning, the learner sees many unlabeled points but requests labels selectively, so label complexity can be exponentially smaller than sample complexity. No equivalence exists; the gap is the content of active learning theory.

Obstruction

Lifelong learning ↔ concept drift.

Both concern learning in non-stationary environments. Lifelong learning emphasizes knowledge transfer across a sequence of tasks; concept drift emphasizes adaptation to a changing target within a single task. The structural parallel is "the learning problem changes over time," but the formalization differs: lifelong learning uses a task distribution, concept drift uses a time-varying concept. No formal theorem connects the two frameworks.

15.3 One-Way Theorem Only

In these analogies, one direction of the relationship has been proved as a theorem, but the reverse direction is either open or known to be false. The analogy is "half-true": the structural parallel exists, but it is asymmetric.

Obstruction

Sample complexity → VC dimension (but not ←).

The fundamental theorem establishes: $\text{VCdim}(\mathcal{H}) < \infty \Leftrightarrow$ finite sample complexity. The two are functionally related, but sample complexity is defined independently as the minimum number of examples needed, while VC dimension is a combinatorial property. The relationship is a theorem, not a definition; treating them as interchangeable obscures the logical structure.

Obstruction

Compression scheme → sample complexity (but not ←).

A compression scheme of size k implies PAC sample complexity $O(k \cdot \log m/\varepsilon)$ (Littlestone–Warmuth, Moran–Yehudayoff). But no theorem says that optimal sample complexity determines optimal compression size. A class might be learnable with few samples yet require a large compression scheme. The compression direction is proved; the reverse is open.

Obstruction

Covering number → VC dimension (but not ←).

The Sauer–Shelah lemma gives: $\log N(\varepsilon, \mathcal{H}, L_1(P)) \leq d \cdot \log(2e/\varepsilon)$, bounding covering numbers from VC dimension. But covering numbers are defined for arbitrary metric spaces, while VC dimension applies only to binary hypothesis classes. VC dimension is a sufficient condition for covering number bounds, not a necessary one; the covering number framework is strictly more general.

Obstruction

Covering number → Rademacher complexity (but not ←).

Dudley's entropy integral bounds Rademacher complexity from above using covering numbers: $\widehat{\mathcal{R}}_n(\mathcal{H}) \leq \inf_{\alpha>0} \left[4\alpha + \frac{12}{\sqrt{n}} \int_\alpha^1 \sqrt{\log N(\varepsilon, \mathcal{H}, L_2)}\, d\varepsilon\right]$. This bound can be loose; tighter chaining methods (Talagrand's generic chaining) and direct Rademacher estimates sometimes give better results. The Dudley bound is one-directional: covering numbers control Rademacher, but Rademacher complexity does not determine covering numbers.

15.4 Data Model Mismatch

These analogies fail because the source and target assume incompatible data-generating processes. The structural parallel is real, but the mathematical assumptions about how data arrives are fundamentally different.

Obstruction

Concept drift ↔ online learning.

Both are sequential learning settings, but concept drift uses i.i.d. draws per time step from a slowly changing distribution, while online learning uses adversarially chosen instances from a fixed concept class. The shared structure is "sequential prediction under non-stationarity," but the non-stationarity is stochastic in one case and adversarial in the other. Results from one setting do not transfer: online regret bounds assume a fixed comparator class, while drift bounds assume a rate of distributional change.

Obstruction

Advice reduction ↔ sample complexity.

Both quantify "how much side information shrinks the learner's search space." But advice operates in the Gold model (infinite stream, computable learner), while sample complexity operates in the PAC model (finite i.i.d. sample, distribution-free guarantees). No formal theorem bridges the two frameworks. The structural parallel—"side information helps"—is too vague to formalize without choosing a specific data model, and the two models are incompatible.

[KS01]

Obstruction

Granger causality ↔ concept drift.

Both operate on time-series data, but Granger causality assumes stationarity within estimation windows (testing whether past values of X improve prediction of Y), while concept drift explicitly models non-stationarity. The structural parallel is "temporal dependence in sequential data," but the assumptions about the data process are contradictory: Granger requires stationarity; drift requires its absence.

15.5 Proof Method Mismatch

In these analogies, the source and target have the same theorem structure—the "shape" of the result is parallel—but the proof techniques from one do not transfer to the other.

Obstruction

Information-theoretic bound ↔ algorithmic stability.

Both bound the generalization gap via properties of the learning algorithm. Stability uses sup-norm perturbation sensitivity (how much does the output change when one training point is replaced?). Information-theoretic bounds use mutual information $I(W;S)$ (how much does the algorithm's output depend on the training set?). These are connected via the chain: low stability $\Rightarrow$ low $I(W;S)$ $\Rightarrow$ low generalization gap. But the proof methods are fundamentally different: stability arguments use coupling and McDiarmid-type con-

centration; information-theoretic arguments use change-of-measure and data processing inequalities. Neither proof technique subsumes the other.

Obstruction

Ordinal VC dimension ↔ mind-change ordinal.

Both use ordinal-valued measures to characterize learnability in transfinite settings. Ordinal VC dimension (Martin–Sharma–Stephan) characterizes predictive complexity; mind-change ordinals characterize identification complexity in the Gold model. Both use ordinals, but they measure different things. The connection is mediated through predictive complexity, not direct: ordinal VC dimension governs prediction error bounds, while mind-change ordinals govern convergence to a correct hypothesis. The proof techniques (Ramsey-theoretic for ordinal VC, topological for mind-change) do not transfer.

Obstruction

Occam's razor ↔ MDL.

Both assert that simpler models generalize better. Occam's razor (in the PAC sense of Blumer et al.) is a worst-case generalization theorem: if the hypothesis is short, its error is low with high probability. MDL is an information-theoretic model selection principle: minimize the total description length of model plus data-given-model. The theorem *structure* is parallel ("short ⇒ good"), but Occam gives frequentist, distribution-free guarantees while MDL gives Bayesian-flavored, coding-theoretic guarantees. The proof methods inhabit different mathematical worlds.

[BEHW87]

15.6 Success Criterion Mismatch

A single analogy edge has this obstruction type.

Obstruction

Granger causality ↔ online learning.

Granger causality tests causal structure: does the past of time series X improve prediction of time series Y beyond what Y's own past provides? Online learning minimizes cumulative prediction error against an adversary. Both involve sequential prediction, but they optimize different objectives. Granger seeks a *structural* conclusion (causal influence exists or does not); online learning seeks a *performance* guarantee (low regret). The success criteria are fundamentally different: statistical significance of a causal test vs. sub-linear regret growth. No formal theorem connects them because they answer different questions about the same sequential data.

15.7 The Obstruction Map

Figure 15.1 displays all 32 analogy edges, colored by obstruction type. The visual pattern reveals the structure of the field's unresolved connections.

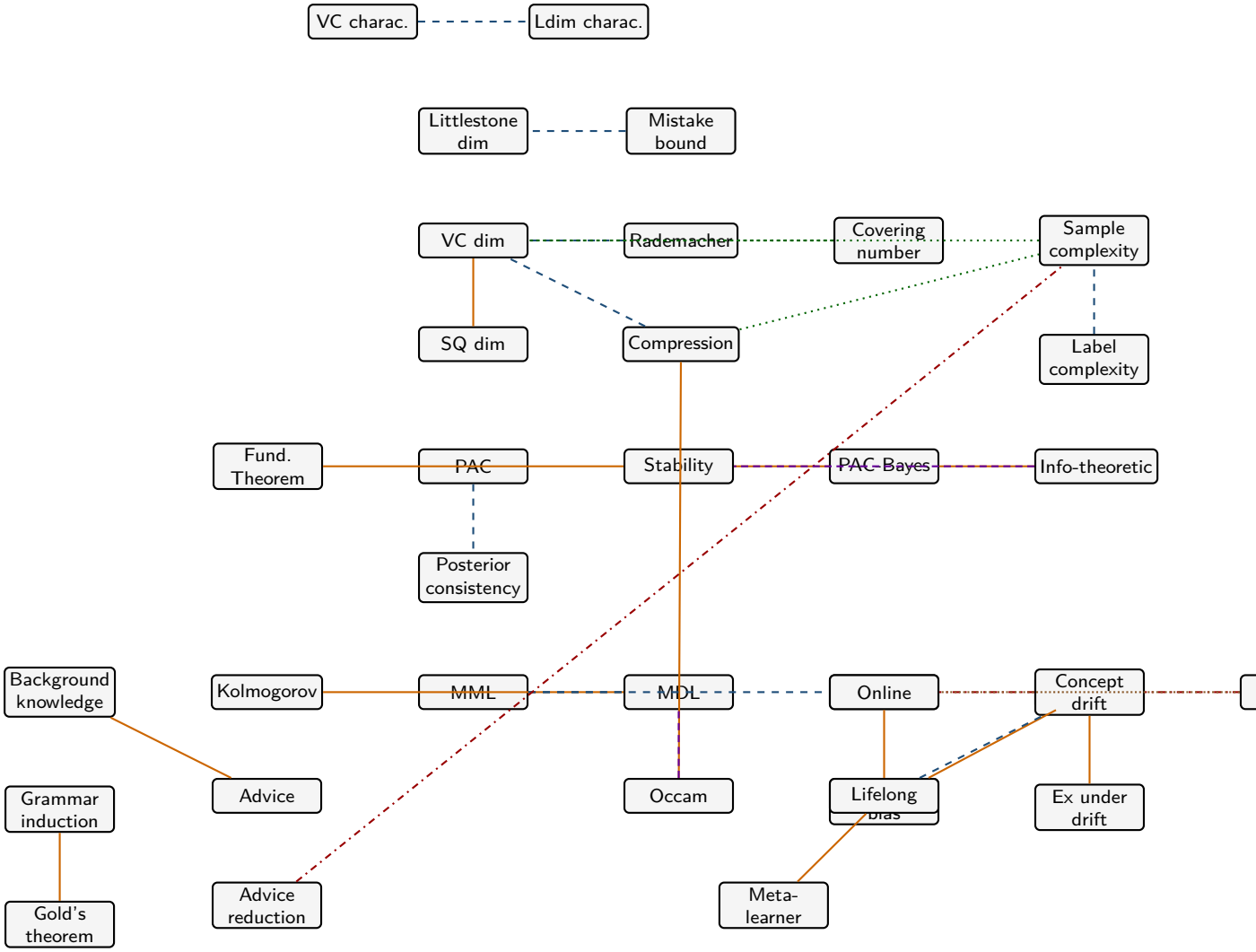

Figure 15.1: The obstruction map: all 32 analogy edges colored by obstruction type. — type mismatch (12); - - - missing equivalence witness (9); · · · one-way theorem (4); -·- data model mismatch (3); -- proof method mismatch (3); · · · success criterion mismatch (1). One edge (ordinal VC dim ↔ mind-change ordinal, proof method mismatch) is omitted from the diagram for clarity.

What the distribution tells us

The obstruction type distribution reveals three structural facts about formal learning theory.

First, the dominance of type mismatches (12 of 32) means that the field's most common cross-paradigm analogies fail for the most basic possible reason: the objects being compared are not the same kind of thing. This is not a failure of proof technique; it is a failure of type compatibility. It suggests that future unification results will require new bridging concepts that mediate between the existing type systems.

Second, the 9 missing equivalence witnesses represent the field's open frontier. The sample compression conjecture (compression ↔ VC dimension) is the most prominent, but the Rademacher–VC relationship and the characterization-theorem parallel (VC ↔ Littlestone) are equally fundamental. Progress on any of these would restructure the analogy map, converting a dashed blue edge into a solid theorem edge.

Third, the 4 one-way theorems show where asymmetry is structural, not accidental. Covering numbers bound Rademacher complexity (Dudley) and are bounded by VC dimension

(Sauer–Shelah), but these bounds flow in one direction only. The asymmetry reflects the generality hierarchy: covering numbers apply to metric spaces, Rademacher to function classes, VC dimension to binary classes. Each step downward in generality tightens the bound but loses applicability.

Together, the 32 analogy edges and their obstructions form a map of what formal learning theory does *not yet* know how to unify. The obstructions are not obstacles to be removed; most of them are permanent features of the mathematical landscape. Understanding why an analogy fails is, in this field, as valuable as understanding why a theorem holds.

Table 15.1: Type mismatch analogies. The "type gap" column identifies the category mismatch that blocks a formal theorem.

Analogy	**Shared structure**	**Type gap**
algorithmic_stability ↔ pac_bayes_bound	Both bound generalization gap	Stability is algorithm-dependent (perturbation of A). PAC-Bayes is posterior-dependent (divergence $\mathrm{KL}(Q\|P)$). An algorithm defines a posterior, but the primary objects differ.
information_theoretic_bound ↔ pac_bayes_bound	Both use information-divergence measures	Info-theoretic bounds use $I(W;S)$ (mutual information between algorithm output and training set). PAC-Bayes uses $\mathrm{KL}(Q\|P)$ (posterior vs. prior). Different information measures on different objects. Hellström et al. (2023) partially unify via change-of-measure, but the objects remain distinct.
fundamental_theorem ↔ algorithmic_stability	Both characterize learnability	The fundamental theorem is a characterization (equivalence result). Stability is a complexity measure. The structural parallel exists but the formal types are incompatible without a bridging theorem.
sq_dimension ↔ vc_dimension	Both measure concept class complexity	SQ dimension measures pairwise correlation structure under a distribution. VC dimension measures distribution-free shattering. Exponential gap possible (parities: VCdim $= n$, SQdim $= 2^n$).
grammar_induction ↔ gold_theorem	Both concern language learning	Grammar induction is a process; Gold's theorem is an impossibility result. Different concept categories.
concept_drift ↔ ex_under_drift	Both concern non-stationary learning	Concept drift is a process; Ex-under-drift is a success criterion. Different concept categories.
lifelong_learning ↔ meta_learner	Both concern learning across tasks	Lifelong learning is a process; meta-learner is a learner type. Different concept categories.
background_knowledge ↔ advice	Both provide side information to a learner	Background knowledge is a process; advice is a data presentation. Different concept categories.
mdl ↔ kolmogorov_complexity	Both measure description length	MDL uses finite computable codes. Kolmogorov complexity uses the shortest program on a universal Turing machine (generally incomputable). The parallel is "short description = good model," but MDL's codes are constructive while Kolmogorov's are not.
srm ↔ inductive_bias	Both constrain hypothesis selection	SRM is a model selection principle; inductive bias is a base type. The structural parallel exists but the categories are incompatible.
bayesian_inference ↔ inductive_bias	Both encode prior assumptions	Bayesian inference is a model selection method (probability over hypotheses). Inductive bias is a base type (constraint on learner behavior). The prior *is* a specific form of bias, but the type systems differ.
occam_algorithm ↔ compression_scheme	Both embody "short representation ⇒ generalization"	Occam uses description length of hypotheses (syntactic: bits to encode h). Compression uses a subset of training data (extensional: examples to reconstruct h). The notion of shortness differs.

Chapter 16

Computational vs. Information-Theoretic Learnability

The Fundamental Theorem of Chapter 5 characterizes learnability in information-theoretic terms: $\mathcal{H}$ is PAC learnable if and only if $\text{VCdim}(\mathcal{H}) < \infty$. The characterization is silent about *computation*. It guarantees the *existence* of a sample size $m(\varepsilon, \delta)$ and a learner A that achieves accuracy ε with confidence $1 - \delta$, but it says nothing about how long A takes to run.

This silence conceals a gap. There exist hypothesis classes with finite VC dimension—hence information-theoretically learnable—that no polynomial-time algorithm can PAC learn, unless widely believed cryptographic assumptions are false. The gap between what is *learnable* and what is *efficiently learnable* is the subject of this chapter. It is a negative result, but a structurally important one: it shows that information-theoretic characterizations, however elegant, do not tell the whole story.

The chapter has three sections.

1. **The information–computation gap** (Section 16.1): the Kearns–Valiant result and the cryptographic assumptions it requires.
2. **Proper vs. improper learning** (Section 17.5): the separation between proper and improper learnability, and why the representation matters.
3. **The `requires_assumption` edges** (Section 16.3): what this edge type means structurally in the concept graph.

16.1 The Information–Computation Gap

Graph Node: computational_hardness

Layer: `meta`. Incoming edges: computational_hardness $\xrightarrow{\texttt{requires\_assumption}}$ inductive_bias. This node represents the structural fact that efficient learnability requires assumptions beyond finite VC dimension.

The central result is due to Kearns and Valiant [KV94].

Theorem 16.1 (Kearns–Valiant 1994)**.** *Under the assumption that certain cryptographic problems are hard (specifically, that the decisional composite residuosity assumption, the hardness of factoring, or the learning with errors assumption holds), there exist concept classes $\mathcal{C}$ with:*
1. $\text{VCdim}(\mathcal{C}) < \infty$*, so that $\mathcal{C}$ is PAC learnable (information-theoretically), but*
2. no polynomial-time algorithm PAC learns $\mathcal{C}$.

The original construction uses Boolean formulae: the class of polynomial-size Boolean formulae has VC dimension polynomial in the number of variables, yet PAC learning this class is at least as hard as breaking certain public-key cryptosystems. The reduction goes through the construction of *cryptographic pseudorandom function families*: if a learner could efficiently distinguish the target concept from random, it could break the cryptographic assumption.

Remark 16.2 (The structure of the reduction). The Kearns–Valiant reduction is from a cryptographic primitive to a learning problem, not from a worst-case problem. The learner is given random examples $(x, c(x))$ where x is drawn from a distribution and c is the target concept. If the learner succeeds in learning, it implicitly solves the cryptographic problem on random instances. This is why the hardness is *average-case*, not worst-case.

16.1.1 Why P $\neq$ NP Does Not Suffice

One might hope that the information–computation gap follows from P $\neq$ NP. It does not. Applebaum, Barak, and Xiao [ABX08] showed the following.

Theorem 16.3 (Applebaum–Barak–Xiao 2008)**.** *There is no* black-box *reduction from* NP*-hardness to the hardness of PAC learning. Specifically, if PAC learning a class $\mathcal{C}$ is hard, this hardness cannot be established by a reduction that treats the learner as an oracle, unless* NP $\subseteq$ coAM*.*

The reason is structural. PAC learning is an *average-case* problem: the learner receives random examples from a distribution and must generalize. NP-hardness is a *worst-case* notion. Reductions from worst-case to average-case problems are notoriously difficult and, in many settings, provably impossible via relativizing techniques.

16.1.2 The Necessary Assumptions

The assumptions used in the literature fall into three families, in roughly increasing order of generality:

1. **Factoring and RSA.** The original Kearns–Valiant result assumes that factoring n-bit integers requires super-polynomial time. This is the most classical assumption but also the most fragile: it is broken by quantum computers (Shor's algorithm).

2. **Decisional Composite Residuosity (DCRA).** Used in subsequent refinements, DCRA assumes that it is hard to distinguish nth residues modulo n^2. Like factoring, this is vulnerable to quantum attacks.

3. **Learning With Errors (LWE).** Introduced by Regev [Reg05], the LWE assumption states that it is hard to recover a secret vector $\mathbf{s}$ from noisy inner products $\langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \pmod q$. LWE is believed to resist quantum computers and has become the standard assumption in modern hardness-of-learning results [DV21].

Obstruction

The information–computation gap is an *obstruction* in the concept graph: the edge $\mathsf{vc\_dimension} \xrightarrow{\texttt{characterizes}} \mathsf{pac\_learnability}$ is information-theoretically valid, but the analogous computational edge does not exist. No combinatorial dimension characterizes efficient PAC learnability. This is the deepest structural gap in the theory.

16.2 Proper vs. Improper Learning

The information–computation gap has a subtler cousin: the gap between *proper* and *improper* learning.

Definition 16.4 (Proper and Improper Learning)**.** A PAC learner for $\mathcal{H}$ is *proper* if it always outputs a hypothesis $h \in \mathcal{H}$. It is *improper* if it may output any efficiently evaluable function h, not necessarily in $\mathcal{H}$.

The distinction matters because the *representation* of hypotheses affects computational complexity.

Theorem 16.5 (Proper–Improper Separation)**.** *There exist concept classes $\mathcal{C}$ such that:*
1. *proper PAC learning of $\mathcal{C}$ is* NP*-hard, but*
2. *there exists a polynomial-time improper learner for $\mathcal{C}$.*

The canonical example is the class of 3-term DNF formulae. Finding a consistent 3-term DNF is NP-hard [PV88], but the class of 3-term DNFs is contained in the class of 3-CNF formulae, which *is* efficiently learnable. An improper learner simply outputs a 3-CNF formula that is consistent with the data. The hypothesis is not a DNF, but it classifies correctly.

Remark 16.6 (Representation vs. class). The proper–improper separation illustrates a principle that runs throughout computational learning theory: what matters for computational complexity is not the *set of functions* $\mathcal{H} \subseteq \{0,1\}^X$ but the *representation* of those functions. Two representations of the same set of Boolean functions can have dramatically different computational properties. This is why the VC dimension—which depends only on the set $\mathcal{H}$—cannot capture computational learnability.

The structural consequence is that efficient learnability is a property of *represented* hypothesis classes, not of hypothesis classes as sets. The concept graph captures this through the node proper_improper_separation, which has type `separation` and connects to computational_hardness via a `motivates` edge.

16.3 The `requires_assumption` Edges

The concept graph contains several edges of type `requires_assumption`. These edges encode a specific logical structure: the source node's validity depends on an assumption that is not provable from the axioms of the theory alone.

Definition 16.7 (The `requires_assumption` edge type)**.** An edge A $\xrightarrow{\texttt{requires\_assumption}}$ B means: the result or property associated with node A holds *conditional on* the assumption encoded by node B. If B is falsified, the status of A becomes indeterminate.

The principal instances in the graph are:

1. computational_hardness $\xrightarrow{\texttt{requires\_assumption}}$ inductive_bias. The information–computation gap of Theorem 16.1 holds only under cryptographic assumptions. Without them, it remains conceivable that every class with finite VC dimension is efficiently learnable. The inductive_bias node here encodes the structural assumption that the learner's hypothesis space is constrained—without which the hardness results are vacuous.
2. posterior_consistency $\xrightarrow{\texttt{requires\_assumption}}$ bayesian_inference. Posterior consistency (Chapter 4, Section 4.3) holds under conditions on the prior and the model class. When these conditions fail—for example, when the prior assigns zero mass to neighborhoods of the true parameter—posterior consistency fails (Diaconis–Freedman [DF86]). The edge records this conditional dependence.

Remark 16.8 (Structural meaning). The `requires_assumption` edge type is the graph's mechanism for representing *conditional* theorems. Unlike `implies` edges, which are unconditional, a `requires_assumption` edge signals that the source node's content is only as secure as the target node's assumption. This is the graph-theoretic encoding of the phrase "under the assumption that..." that pervades computational learning theory.

This chapter has established the structural gap between information-theoretic and computational learnability. The gap is real—it produces classes that are provably learnable but (conditionally) not efficiently learnable. It is also irreducible: no known combinatorial dimension characterizes efficient PAC learnability. The next chapter examines a different kind of gap: the distance between binary classification and the richer learning settings that require new mathematical primitives.

Chapter 17

Extensions Beyond Binary Classification

The PAC theory of Chapter 5 and its characterization through VC dimension apply to binary classification: the label set is $\{0, 1\}$, the hypothesis class is $\mathcal{H} \subseteq \{0, 1\}^X$, and the loss is 0-1. This is the setting where the theory is cleanest and most complete. But most learning problems are not binary classification. Labels may come from a finite set of $k > 2$ classes; the target may be a real-valued function; the data may be noisy; the realizability assumption may fail.

This chapter asks: how does the binary theory extend to these richer settings? The answer is not "straightforwardly." Each extension requires *new mathematical primitives*—new notions of shattering, new complexity measures, new assumptions. The binary theory does not generalize by substitution; it generalizes by *grammar growth*. In the concept graph, this is captured by edges of type `extends_grammar`: the source dimension extends the grammar of the VC dimension by introducing primitives that have no analogue in the binary case.

The chapter has six sections.

1. **Multiclass learning** (Section 17.1): the Natarajan dimension, the DS dimension, and the resolution of a thirty-year open problem.
2. **Real-valued functions** (Section 17.2): pseudodimension and fat-shattering dimension.
3. **Agnostic learning** (Section 17.3): dropping realizability and the agnostic fundamental theorem.
4. **Noise-tolerant and partial concept learning** (Section 17.4): classification noise, malicious noise, and promise problems.
5. **Proper versus improper learning** (Section 17.5): the computational gap.
6. **The `extends_grammar` pattern** (Section 17.6): what the extensions have in common.

17.1 Multiclass Learning: From Natarajan to DS

Let $Y = [k] = \{1, 2, \ldots, k\}$ for some $k \geq 2$, and let $\mathcal{H} \subseteq Y^X$ be a multiclass hypothesis class. The question is: what characterizes PAC learnability of $\mathcal{H}$?

For $k = 2$, the answer is the VC dimension. For $k > 2$, the answer was only fully resolved in 2022, after a thirty-year journey through two competing dimensions, each introducing a new shattering primitive. That journey—and its resolution—is the narrative backbone of this section.

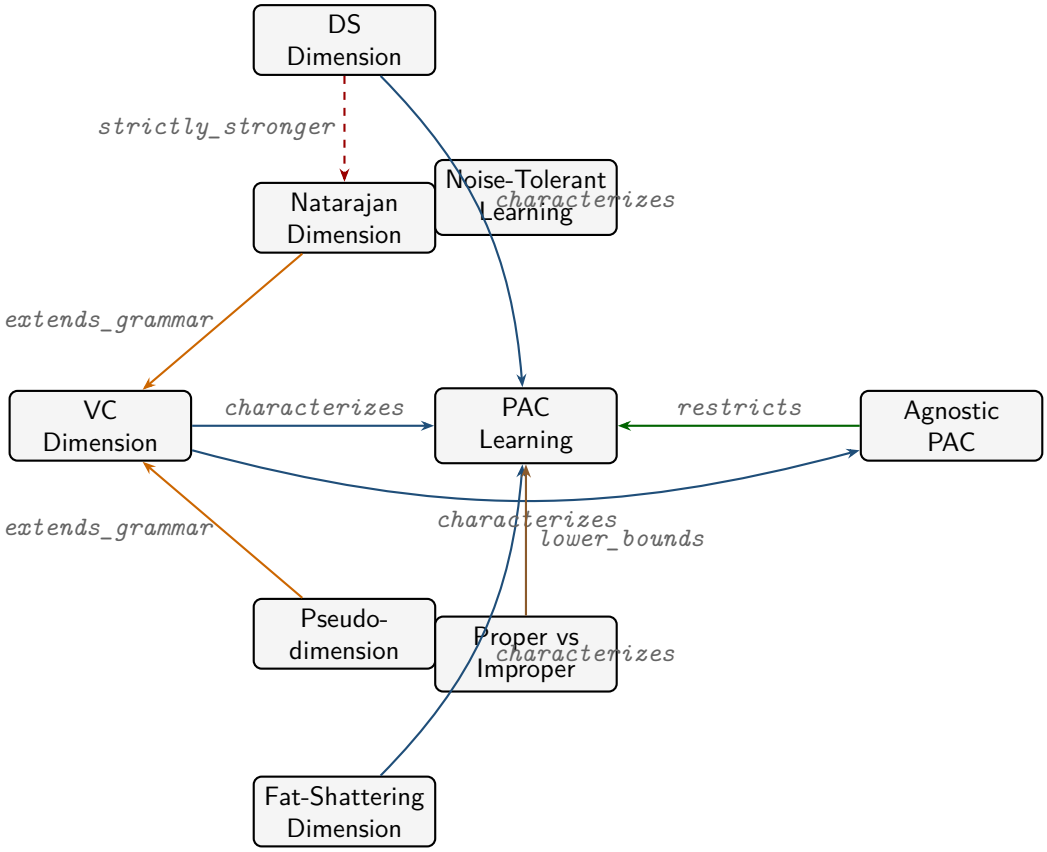

Figure 17.1: Concept map for this chapter. Orange: `extends_grammar`. Dashed red: `strictly_stronger`. Blue: `characterizes`. Green: `restricts`. Brown: `lower_bounds`. Each arrow represents an edge in the concept graph.

17.1.1 The Natarajan Dimension

Natarajan [Nat89] introduced the first multiclass complexity measure by generalizing shattering to require two functions instead of one.

Definition 17.1 (Natarajan Dimension [Nat89])**.** The *Natarajan dimension* $d_{\mathrm{N}}(\mathcal{H})$ of a multiclass hypothesis class $\mathcal{H} \subseteq Y^X$ is the largest size d of a set $\{x_1, \dots, x_d\} \subseteq X$ for which there exist two functions $f_0, f_1 \colon \{x_1, \dots, x_d\} \to Y$ satisfying:
1. $f_0(x_i) \neq f_1(x_i)$ for all $i \in [d]$, and
2. for every binary string $b \in \{0,1\}^d$, there exists $h \in \mathcal{H}$ with $h(x_i) = f_{b_i}(x_i)$ for all i.

The new primitive is the *pair of functions* (f_0, f_1). In binary classification, f_0 and f_1 are forced to be the constant functions 0 and 1, and the definition reduces to VC shattering. In the multiclass case, the choice of which two labels to use at each point is part of the combinatorial structure.

Example 17.2 (Natarajan shattering with $k = 3$)**.** Let $X = \{a, b\}$, $Y = \{1, 2, 3\}$. Define $f_0(a) = 1$, $f_0(b) = 2$ and $f_1(a) = 3$, $f_1(b) = 1$. To N-shatter $\{a, b\}$, we need four hypotheses realizing every binary string:

$$h_{00}(a) = 1,\ h_{00}(b) = 2; \quad h_{01}(a) = 1,\ h_{01}(b) = 1; \quad h_{10}(a) = 3,\ h_{10}(b) = 2; \quad h_{11}(a) = 3,\ h_{11}(b) = 1.$$

Any $\mathcal{H} \supseteq \{h_{00}, h_{01}, h_{10}, h_{11}\}$ has $d_{\mathrm{N}} \geq 2$.

Historical Note

The Natarajan gap (1989–2022). Natarajan showed that $d_{\mathrm{N}}(\mathcal{H}) < \infty$ is *sufficient* for PAC learnability and provided sample complexity bounds. The best known bounds, due to Ben-David et al. [BDCBHL95], were:

$$C_1 \cdot \frac{d_{\mathrm{N}}}{\varepsilon} \;\le\; m_{\mathrm{PAC}}(\varepsilon, \delta) \;\le\; C_2 \cdot \frac{d_{\mathrm{N}} \cdot \ln k \cdot \ln(1/\varepsilon)}{\varepsilon}.$$

Two gaps persisted. First, the $\Theta(\log k)$ factor between upper and lower bounds. Second, and more fundamentally: is finiteness of d_{N} also *necessary*? For finite label sets $|Y| = k < \infty$, the answer is yes (the $\log k$ factor is at most a polynomial overhead). But when $|Y| = \infty$, a class can have $d_{\mathrm{N}} = 1$ and yet fail to be PAC learnable. This was the open problem that persisted for over thirty years.

Graph Node: natarajan_dimension

Layer 5: `complexity_measure`. Key edge: natarajan_dimension $\xrightarrow{\texttt{extends\_grammar}}$ vc_dimension. The grammar extension is the two-function shattering primitive (f_0, f_1). Open frontier: tight multiclass sample complexity without the $\log k$ factor.

17.1.2 The DS Dimension: A Thirty-Year Resolution

The correct characterization of multiclass PAC learnability was established by Brukhim et al. [BCD+22] using the *DS dimension*, named after Daniely and Shalev-Shwartz [DSS14] who introduced the underlying concept of pseudo-cubes in one-inclusion hypergraphs.

Definition 17.3 (One-Inclusion Hypergraph)**.** Let $\mathcal{H} \subseteq Y^X$ and let $S = \{x_1, \ldots, x_n\} \subseteq X$ be a finite set. The *one-inclusion hypergraph* $\mathrm{OIG}(\mathcal{H}, S)$ has vertex set $\mathcal{H}|_S = \{(h(x_1), \ldots, h(x_n)) : h \in \mathcal{H}\}$ and, for each coordinate $i \in [n]$, a hyperedge connecting all vertices that agree on all coordinates except possibly coordinate i.

Definition 17.4 (DS Dimension [DSS14, BCD+22])**.** A set $\{x_1, \ldots, x_d\} \subseteq X$ is *DS-shattered* by $\mathcal{H} \subseteq Y^X$ if there exists a function $f\colon \{x_1, \ldots, x_d\} \to Y$ such that for every $i \in [d]$, there exists $h_i \in \mathcal{H}$ with:
1. $h_i(x_i) \neq f(x_i)$, and
2. $h_i(x_j) = f(x_j)$ for all $j \neq i$.

The *DS dimension* $\mathrm{DSdim}(\mathcal{H})$ is the size of the largest DS-shattered set.

The structural content of DS-shattering is different from Natarajan shattering in a revealing way. Natarajan shattering asks for a *global* pair of labeling functions with *all* 2^d interpolations realized. DS-shattering asks for a single "default" labeling f and, for each point, a *local* witness that deviates at exactly that point. The requirement is weaker per point (only one deviation, not arbitrary binary combinations) but more structural: it encodes a property of the one-inclusion hypergraph of $\mathcal{H}$.

Theorem 17.5 (Multiclass Characterization [BCD+22])**.** *A multiclass hypothesis class $\mathcal{H} \subseteq Y^X$ is PAC learnable if and only if* $\mathrm{DSdim}(\mathcal{H}) < \infty$.

Proof sketch. We outline both directions.

Sufficiency ($\mathrm{DSdim}(\mathcal{H}) < \infty \Rightarrow$ learnable). The proof proceeds through the one-inclusion hypergraph. For a finite restriction $\mathcal{H}|_S$ to a sample S of size m, the one-inclusion hypergraph $\mathrm{OIG}(\mathcal{H}, S)$ has vertices $\mathcal{H}|_S$ and hyperedges indexed by coordinates. The key insight of Daniely and Shalev-Shwartz is that a *proper orientation* of this hypergraph—an assignment of each hyperedge to one of its vertices—yields a learning algorithm. Specifically:

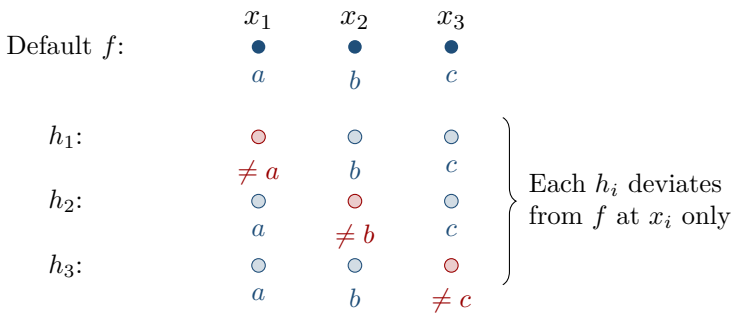

Figure 17.2: DS-shattering of $\{x_1, x_2, x_3\}$. The default labeling f assigns labels a, b, c. Each witness h_i disagrees with f at exactly one coordinate (shown in red).

1. Given a sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, restrict $\mathcal{H}$ to $\{x_1, \ldots, x_m\}$ and build $\mathrm{OIG}(\mathcal{H}, S)$.
2. Find a proper orientation with *minimum outdegree*.
3. Return the hypothesis whose vertex has minimum outdegree in the orientation.

If $\mathrm{DSdim}(\mathcal{H}) = d < \infty$, then the one-inclusion hypergraph of any restriction has no d-dimensional pseudo-cube, and a combinatorial argument (via the Helly property of CAT(0) cube complexes) shows that a proper orientation with outdegree at most $O(d)$ exists. The learner's error is bounded by $O(d/m)$, yielding PAC learnability with sample complexity $O(d/\varepsilon)$.

Necessity ($\mathrm{DSdim}(\mathcal{H}) = \infty \Rightarrow$ not learnable). If $\mathrm{DSdim}(\mathcal{H}) = \infty$, then for every d, there exists a finite set that is DS-shattered. The proof constructs, for each d, a distribution D_d over $X \times Y$ such that:

- The Bayes-optimal classifier has zero error.
- Any learner given $m < d$ samples has expected error $\Omega(d/m)$ on D_d.

The distribution is supported on the DS-shattered set and uses the default labeling f as the target. The witness hypotheses $h_1, \ldots, h_d$ act as "confusers": a learner that has not seen point x_i cannot distinguish f from h_i, because they agree everywhere except at x_i. Since this works for arbitrarily large d, no finite sample suffices for ε-accurate learning uniformly over D_d. □

Obstruction

Natarajan $\not\Rightarrow$ PAC learnability when $|Y| = \infty$.

Witness (Brukhim et al. 2022). There exists a hypothesis class $\mathcal{H} \subseteq Y^X$ with $d_{\mathrm{N}}(\mathcal{H}) = 1$ but $\mathrm{DSdim}(\mathcal{H}) = \infty$. The construction uses a hyperbolic pseudo-manifold: the domain X is the vertex set of a high-dimensional simplicial complex with hyperbolic geometry, and $\mathcal{H}$ encodes colorings of simplices. The hyperbolic geometry ensures that any two points can be N-shattered (only two labels are needed locally), but the global structure admits arbitrarily large DS-shattered sets because the one-inclusion hypergraph contains arbitrarily large pseudo-cubes.

Structural lesson. The Natarajan dimension is a *local* measure: it examines pairs of labels at each point. The DS dimension is a *global* measure: it examines the connectivity structure of the entire one-inclusion hypergraph. When $|Y| = \infty$, local two-label constraints do not capture global learnability.

Graph edge: natarajan_dimension $\xrightarrow{\texttt{does\_not\_imply}}$ pac_learning with witness "Brukhim et al. 2022: non-learnable class with $d_{\mathrm{N}} = 1$."

The relationship between the two dimensions is:

Proposition 17.6 (DS vs. Natarajan)**.** *For any $\mathcal{H} \subseteq Y^X$:*

$$d_{\mathrm{N}}(\mathcal{H}) \;\le\; \mathrm{DSdim}(\mathcal{H}).$$

Moreover, when $|Y| = k < \infty$:

$$\mathrm{DSdim}(\mathcal{H}) \;\le\; O\big(d_{\mathrm{N}}(\mathcal{H}) \cdot \log k\big).$$

Proof. For the first inequality, suppose $S = \{x_1, \dots, x_d\}$ is DS-shattered via default f and witnesses $h_1, \dots, h_d$. Define $f_0 = f|_S$ and, for each i, define $f_1(x_i) = h_i(x_i) \neq f(x_i)$. Then S is N-shattered by (f_0, f_1), but only for binary strings that flip at most one coordinate. The full 2^d interpolation requirement of Natarajan shattering may not hold, so DS-shattering is a weaker condition per set, hence $d_{\mathrm{N}} \le \mathrm{DSdim}$.

For the second inequality (finite k), suppose $\mathrm{DSdim} = D$. Any DS-shattered set provides D witness hypotheses deviating from a default. Since each deviation chooses one of $k-1$ alternative labels, a pigeonhole argument over the k^D possible deviation patterns shows $D \le O(d_{\mathrm{N}} \cdot \log k)$. □

Graph Node: ds_dimension

Layer 5: `complexity_measure`. Key edges: ds_dimension $\xrightarrow{\texttt{characterizes}}$ pac_learning (for multiclass). ds_dimension $\xrightarrow{\texttt{strictly\_stronger}}$ natarajan_dimension with witness: hyperbolic pseudo-manifold construction. This node resolved a 30-year open problem in the theory.

17.2 Real-Valued Functions

Now let $\mathcal{F} \subseteq \mathbb{R}^X$ be a class of real-valued functions. The learner observes samples (x_i, y_i) with $y_i = f(x_i)$ (or $y_i \approx f(x_i)$ in the noisy case) and aims to approximate f. The 0-1 loss is replaced by the squared loss or absolute loss, and the question becomes: what is the analogue of VC dimension?

Two dimensions extend the binary theory. Both introduce *thresholds* as a new primitive.

17.2.1 Pseudodimension

Definition 17.7 (Pseudodimension [Pol84, Hau92])**.** The *pseudodimension* $\mathrm{Pdim}(\mathcal{F})$ of a function class $\mathcal{F} \subseteq \mathbb{R}^X$ is the VC dimension of the class of subgraphs:

$$\mathrm{Pdim}(\mathcal{F}) \;=\; \mathrm{VCdim}\big(\{x \mapsto \mathbf{1}[f(x) \ge t] : f \in \mathcal{F},\; t \in \mathbb{R}\}\big).$$

Equivalently, $\mathrm{Pdim}(\mathcal{F}) = d$ if there exist $x_1, \dots, x_d \in X$ and thresholds $t_1, \dots, t_d \in \mathbb{R}$ such that for every $b \in \{0,1\}^d$, there exists $f \in \mathcal{F}$ with $f(x_i) \ge t_i \iff b_i = 1$.

The new primitive is the *threshold vector* $(t_1, \dots, t_d)$. In binary classification, thresholds are unnecessary: the outputs are already 0 or 1. For real-valued functions, thresholds convert the continuous outputs into a binary shattering condition.

Example 17.8 (Linear functions)**.** For the class of linear functions $\mathcal{F} = \{x \mapsto w^\top x : w \in \mathbb{R}^d\}$ on $\mathbb{R}^d$, we have $\mathrm{Pdim}(\mathcal{F}) = d$. Any d affinely independent points $x_1, \dots, x_d$ can be P-shattered with thresholds $t_i = 0$: for each $b \in \{0,1\}^d$, one can find w such that $\mathrm{sign}(w^\top x_i) = (-1)^{1-b_i}$.

Proposition 17.9 (Pseudodimension and covering numbers)**.** *If $\mathrm{Pdim}(\mathcal{F}) = d < \infty$, then for every probability measure P on X and every $\varepsilon > 0$:*

$$\mathcal{N}(\varepsilon, \mathcal{F}, L_1(P)) \;\le\; \left(\frac{2e}{\varepsilon}\right)^d.$$

Graph Node: pseudodimension

Layer 5: `complexity_measure`. Key edge: pseudodimension $\xrightarrow{\texttt{extends\_grammar}}$ vc_dimension. The grammar extension is the threshold primitive.

17.2.2 Fat-Shattering Dimension

Pseudodimension controls learnability in the realizable case. For the agnostic setting with real-valued targets, a scale-sensitive notion is needed.

Definition 17.10 (Fat-Shattering Dimension [ABDCBH97])**.** For $\gamma > 0$, the *γ-fat-shattering dimension* $\mathrm{fat}_\gamma(\mathcal{F})$ is the largest d such that there exist $x_1, \ldots, x_d \in X$ and thresholds $t_1, \ldots, t_d \in \mathbb{R}$ satisfying: for every $b \in \{0,1\}^d$, there exists $f \in \mathcal{F}$ with

$$f(x_i) \geq t_i + \gamma \text{ if } b_i = 1, \qquad f(x_i) \leq t_i - \gamma \text{ if } b_i = 0.$$

The new primitive beyond pseudodimension is the *margin* γ. As $\gamma \to 0$, the fat-shattering dimension approaches the pseudodimension.

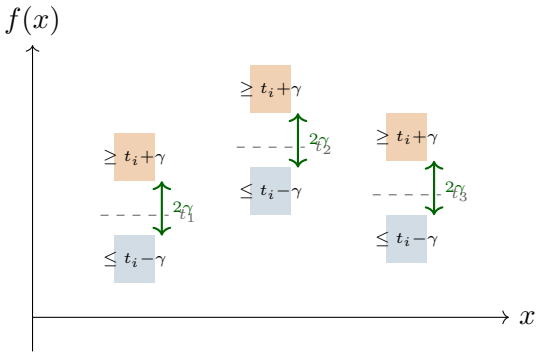

Figure 17.3: γ-fat-shattering. At each point x_i, the threshold t_i creates a gap of width 2γ. Functions must place values strictly within the shaded regions, not merely across the threshold.

Theorem 17.11 (Real-Valued Characterization [ABDCBH97])**.** *A class $\mathcal{F} \subseteq [0,1]^X$ of bounded real-valued functions is agnostically PAC learnable with respect to the absolute loss if and only if $\mathrm{fat}_\gamma(\mathcal{F}) < \infty$ for all $\gamma > 0$.*

Proof sketch. **Sufficiency.** If $\mathrm{fat}_\gamma(\mathcal{F}) = d_\gamma < \infty$ for all $\gamma > 0$, then the L_2 covering numbers satisfy $\log \mathcal{N}_2(\varepsilon, \mathcal{F}, x_1^n) \leq O(d_{\varepsilon/4} \cdot \log^2(n/d_{\varepsilon/4}))$. Finite covering numbers imply uniform convergence of empirical risk to true risk, which yields agnostic PAC learnability.

Necessity. If $\mathrm{fat}_\gamma(\mathcal{F}) = \infty$ for some $\gamma > 0$, then for every n, one can γ-fat-shatter n points. By a probabilistic argument, any empirical risk minimizer on a sample of size m has expected error $\Omega(\gamma)$ on a suitably chosen distribution, regardless of m. Hence $\mathcal{F}$ is not learnable at accuracy γ. □

Remark 17.12 (Scale families versus single numbers). The fat-shattering characterization is a *family* of finiteness requirements (one for each $\gamma > 0$), not a single number. This is another instance of grammar growth: the real-valued setting requires a scale parameter that has no analogue in the binary theory. In practice, fat_γ is often a decreasing function of γ—larger margins are harder to maintain, so fewer points can be fat-shattered.

17.3 Agnostic Learning: Dropping Realizability

All results so far have assumed *realizability*: there exists some $h^* \in \mathcal{H}$ with $R_D(h^*) = 0$. In practice, models are always wrong; the question is how wrong. The agnostic setting, introduced by Haussler [Hau92], removes the realizability assumption entirely.

Definition 17.13 (Agnostic PAC Learning [Hau92])**.** A hypothesis class $\mathcal{H}$ is *agnostically PAC learnable* if there exists a learner A and a function $m\colon (0,1)^2 \to \mathbb{N}$ such that for every distribution D over $X \times \{0,1\}$ (with *no* assumption that the Bayes-optimal classifier lies in $\mathcal{H}$), for every $\varepsilon, \delta > 0$: given $m(\varepsilon,\delta)$ i.i.d. samples from D,

$$\mathbb{P}_{S\sim D^m}\Big[R_D(A(S)) \;\le\; \inf_{h\in\mathcal{H}} R_D(h) + \varepsilon\Big] \;\ge\; 1-\delta.$$

The learner must compete with the *best hypothesis in the class*, not with the Bayes-optimal classifier. The gap $\inf_{h\in\mathcal{H}} R_D(h)$ may be positive; the learner need only close to within ε of it.

Theorem 17.14 (Agnostic Fundamental Theorem)**.** *For binary classification with* 0-1 *loss, a hypothesis class $\mathcal{H} \subseteq \{0,1\}^X$ is agnostically PAC learnable if and only if* $\mathrm{VCdim}(\mathcal{H}) < \infty$.

Proof sketch. The characterizing condition is the same as in the realizable case—finite VC dimension—but the proof mechanism and sample complexity differ.

Sufficiency. Suppose $\mathrm{VCdim}(\mathcal{H}) = d < \infty$. The learner runs empirical risk minimization: $A(S) = \arg\min_{h\in\mathcal{H}} \hat{R}_S(h)$. By the uniform convergence theorem for VC classes, with probability at least $1-\delta$ over a sample of size $m = O(\frac{d+\log(1/\delta)}{\varepsilon^2})$,

$$\sup_{h\in\mathcal{H}} \big|R_D(h) - \hat{R}_S(h)\big| \;\le\; \varepsilon/2.$$

Let $h^* = \arg\min_{h\in\mathcal{H}} R_D(h)$ and $\hat{h} = A(S)$. Then:

$$\begin{aligned} R_D(\hat{h}) &\le \hat{R}_S(\hat{h}) + \varepsilon/2 && \text{(uniform convergence)}\\ &\le \hat{R}_S(h^*) + \varepsilon/2 && \text{(ERM chose } \hat{h}\text{)}\\ &\le R_D(h^*) + \varepsilon/2 + \varepsilon/2 && \text{(uniform convergence)}\\ &= R_D(h^*) + \varepsilon. \end{aligned}$$

Necessity. If $\mathrm{VCdim}(\mathcal{H}) = \infty$, the No-Free-Lunch argument of the realizable case applies *a fortiori*: for any learner and any sample size m, there exists a realizable distribution (which is a special case of an arbitrary distribution) on which the learner fails. □

Separation Result

Agnostic versus realizable sample complexity.

The agnostic and realizable settings are characterized by the *same* condition ($\mathrm{VCdim} < \infty$), but the sample complexities differ:

$$m_{\text{realizable}}(\varepsilon,\delta) = \Theta\!\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right), \qquad m_{\text{agnostic}}(\varepsilon,\delta) = \Theta\!\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right).$$

The quadratic dependence on $1/\varepsilon$ in the agnostic case is inherent, not an artifact of the analysis: the lower bound is achieved by distributions where the optimal hypothesis has error $1/2 - \varepsilon$, and distinguishing it from a hypothesis with error $1/2$ requires $\Omega(1/\varepsilon^2)$ samples by a Chernoff-bound argument.

Graph edge: agnostic_pac_learning $\xrightarrow{\texttt{restricts}}$ pac_learning—agnostic PAC learning *generalizes* standard PAC learning by removing the realizability constraint.

17.4 Noise-Tolerant and Partial Concept Learning

17.4.1 Classification Noise

The *classification noise* model (Angluin and Laird [AL88]) modifies PAC learning by flipping each label independently with probability $\eta < 1/2$: the learner observes $(x, y \oplus z)$ where $z \sim \text{Bernoulli}(\eta)$ and y is the true label. The question is whether finite VC dimension still suffices for learning.

Theorem 17.15 (Noise-tolerant PAC learning)**.** *If* $\text{VCdim}(\mathcal{H}) = d < \infty$ *and the noise rate* $\eta < 1/2$ *is known, then* $\mathcal{H}$ *is PAC learnable under classification noise with sample complexity*

$$m = O\left(\frac{d}{(1-2\eta)^2\varepsilon^2} + \frac{\log(1/\delta)}{(1-2\eta)^2\varepsilon^2}\right).$$

The sample complexity degrades as $\eta \to 1/2$—the noise rate approaches the information-theoretic limit where labels carry no signal.

Remark 17.16 (SQ connection). Statistical query (SQ) learnable classes are automatically noise-tolerant: an SQ algorithm estimates expectations $\mathbb{E}_D[\phi(x, y)]$ to tolerance τ, and these can be simulated from noisy samples with a $1/(1-2\eta)$ overhead in sample size. However, not all noise-tolerant classes are SQ learnable: parities are PAC learnable (via Gaussian elimination) and hence noise-tolerant for known η, but not SQ learnable ($\text{SQdim} = 2^n$).

17.4.2 Partial Concept Learning

In *partial concept learning* (also called *promise problems*), the hypothesis class $\mathcal{H}$ is defined only on a subset $P \subseteq X$—the *promise set*. Points outside P have no defined label, and the distribution D is supported on P.

Definition 17.17 (Partial Concept Class)**.** A *partial concept class* is a set $\mathcal{H} \subseteq \{0, 1, *\}^X$ where $*$ denotes "undefined." A distribution D is *compatible* with $h \in \mathcal{H}$ if D is supported on $\{x : h(x) \neq *\}$.

Partial concepts arise naturally in many settings: medical diagnosis where only symptomatic patients have definite diagnoses, or feature spaces where certain regions are "don't care" zones.

The VC dimension of a partial concept class is defined by restricting shattering to the promise region. The fundamental theorem extends: finite VC dimension of the promise-restricted class characterizes PAC learnability of partial concepts. However, the *computational* landscape changes: problems that are efficiently learnable as total concepts may become hard when restricted to a promise set, because the promise constraint interacts with the hypothesis representation.

17.5 Proper Versus Improper Learning

All PAC definitions permit the learner to output *any* efficiently evaluatable hypothesis—not just one from the target class $\mathcal{H}$. When the learner is restricted to output a hypothesis from $\mathcal{H}$ itself, we call it a *proper* learner.

Definition 17.18 (Proper and Improper PAC Learning)**.** A PAC learner A for class $\mathcal{H}$ is *proper* if $A(S) \in \mathcal{H}$ for all samples S. It is *improper* if $A(S)$ may lie in a larger class $\mathcal{H}' \supseteq \mathcal{H}$.

Remark 17.19. In the information-theoretic (sample complexity) sense, there is no separation between proper and improper learning for standard PAC: both are characterized by finite VC dimension, and the sample complexities are $\Theta(d/\varepsilon)$ in both cases. The separation is *computational*.

Theorem 17.20 (Pitt–Valiant [PV88])**.** *For $k \geq 2$, k-term DNF is not properly PAC learnable unless* RP = NP.

Proof. The proof reduces graph k-colorability to the problem of finding a consistent k-term DNF formula.

Let $G = (V, E)$ be a graph with $V = \{v_1, \ldots, v_n\}$. We construct a learning instance over $X = \{0,1\}^n$ as follows.

Positive examples. For each vertex v_i, create the example $e_i^+ \in \{0,1\}^n$ defined by $e_i^+(j) = 1$ if $j \neq i$ and $e_i^+(i) = 0$. Label: positive.

Negative examples. For each edge $(v_i, v_j) \in E$, create the example $e_{ij}^- \in \{0,1\}^n$ defined by $e_{ij}^-(l) = 1$ if $l \notin \{i, j\}$, and $e_{ij}^-(i) = e_{ij}^-(j) = 0$. Label: negative.

Claim. A consistent k-term DNF exists for this labeled sample if and only if G is k-colorable.

Forward direction. Given a proper k-coloring $c\colon V \to [k]$, define the k-term DNF $\phi = T_1 \vee \cdots \vee T_k$ where $T_j = \bigwedge_{i:c(v_i)=j} \overline{x_i}$. Each positive example e_i^+ satisfies $T_{c(v_i)}$ (since $e_i^+(i) = 0$ makes the literal $\overline{x_i}$ true, and all other literals in $T_{c(v_i)}$ are true because $e_i^+(l) = 1$ for $l \neq i$). Each negative example e_{ij}^- falsifies every term: for any term T_j, since (v_i, v_j) is an edge and c is proper, $c(v_i) \neq c(v_j)$, so at most one of v_i, v_j is in color class j, and the other has $e_{ij}^-(l) = 0$ which falsifies a literal not in T_j. (A careful case analysis confirms the argument.)

Reverse direction. Given a consistent k-term DNF $\phi = T_1 \vee \cdots \vee T_k$, assign color $c(v_i) = \min\{j : e_i^+ \text{ satisfies } T_j\}$. Since e_i^+ is positive, at least one term is satisfied. For any edge (v_i, v_j), the negative example e_{ij}^- falsifies ϕ, which forces $c(v_i) \neq c(v_j)$ (otherwise the term satisfied by both would also satisfy e_{ij}^-). Hence c is a proper k-coloring.

Conclusion. A proper PAC learner for k-term DNF, given the above examples (which can be generated efficiently from G), would find a consistent k-term DNF in polynomial time, thereby solving k-colorability in polynomial time. Since k-colorability is NP-complete for $k \geq 3$ (and NP-hard to decide in randomized polynomial time unless RP = NP), proper PAC learning of k-term DNF is computationally hard. □

Separation Result

Proper $\not\Rightarrow$ efficient learning: the DNF witness.

The same class that is hard to learn properly is easy to learn improperly. Observe that every k-term DNF over n variables is a k-CNF formula (by distributing), and k-CNF is learnable in polynomial time: simply enumerate all $O(n^k)$ candidate clauses and filter by consistency with the data.

Summary. For $k = 3$: 3-term DNF is NP-hard proper, polynomial-time improper (via 3-CNF). This is the cleanest known computational separation in PAC learning.

Graph edge: proper_improper_separation $\xrightarrow{\texttt{does\_not\_imply}}$ pac_learning with witness "3-term DNF: NP-hard proper, poly-time improper."

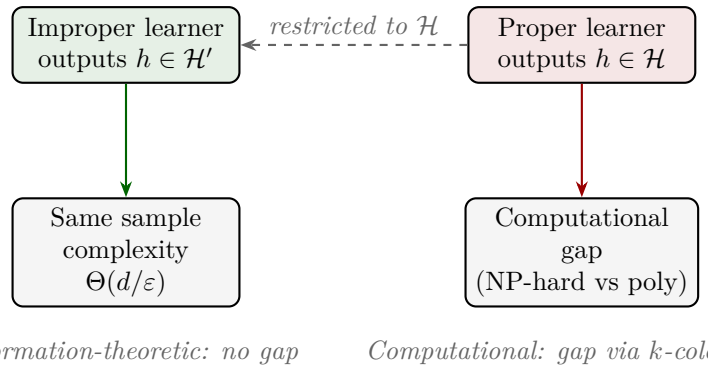

Figure 17.4: The proper/improper separation. The information-theoretic sample complexity is the same; the computational complexity can differ dramatically.

Graph Node: proper_improper_separation

Layer 6: `impossibility`. Key edges: proper_improper_separation $\xrightarrow{\texttt{lower\_bounds}}$ pac_learning and proper_improper_separation $\xrightarrow{\texttt{does\_not\_imply}}$ pac_learning. Provenance: Pitt–Valiant 1988. The proof is a direct reduction from graph coloring to consistent hypothesis finding.

17.6 The `extends_grammar` Pattern

The extensions of the previous sections share a common structure. Each begins with the VC-dimension framework and introduces new primitives that have no analogue in binary classification:

Extension	New Primitive	Why Needed
Natarajan dim.	Two-function shattering (f_0, f_1)	With $k > 2$ labels, a single bit string does not describe a labeling.
DS dimension	One-inclusion witness	Global hypergraph structure governs learnability when $\|Y\| = \infty$.
Pseudodimension	Threshold vector $(t_1, \ldots, t_d)$	Real-valued outputs must be discretized before shattering can be defined.
Fat-shattering dim.	Scale parameter γ	Agnostic real-valued learning requires margin; finiteness at all scales replaces a single finiteness condition.
Agnostic PAC	Benchmark $\inf_{h \in \mathcal{H}} R(h)$	Without realizability, success is measured relative to the best in class.
Noise model	Noise rate η	Labels are corrupted; sample complexity depends on the signal-to-noise ratio $(1 - 2\eta)^{-2}$.

In each case, the extension is an `extends_grammar` edge in the concept graph, not a `restricts` edge. The distinction matters:

- A `restricts` edge means the target is a special case of the source (no new vocabulary needed).
- An `extends_grammar` edge means the source introduces *new vocabulary*—concepts that cannot be expressed in the target's language. The extension is irreducible.

Remark 17.21 (The non-reduction principle). A tempting strategy for multiclass or real-valued learning is to reduce to binary classification (e.g., one-vs-all for multiclass, or thresholding for real-valued). Such reductions are computationally useful but do not eliminate the need for new dimensions. The VC dimension of the reduced class does not generally equal the Natarajan, DS, pseudo-, or fat-shattering dimension of the original class. The reductions introduce approximation gaps that the native dimensions avoid. This is why the `extends_grammar` edges exist: they mark the points where reduction to the binary theory loses information.

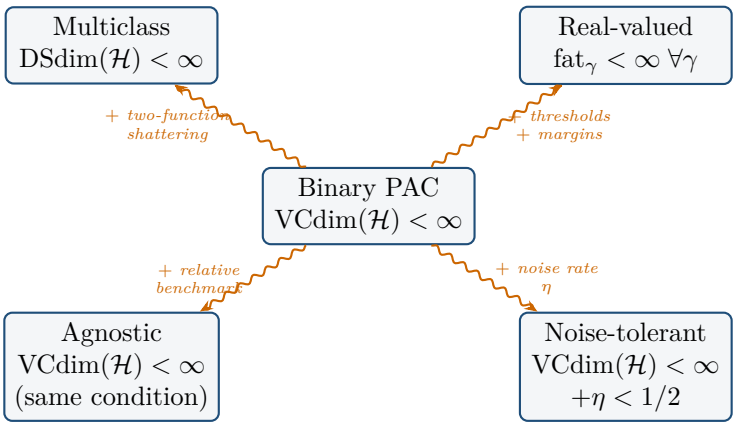

Figure 17.5: Grammar growth from binary PAC learning. Each wavy arrow represents an `extends_grammar` edge: the extension requires new primitives (labeled) that have no analogue in the binary theory.

The grammar growth *is* the extension. A multiclass learning theory that avoids two-function shattering will not characterize learnability. A real-valued theory that avoids scale sensitivity will not capture the agnostic setting. In each case, the new primitive is not an optional enrichment but a structural necessity: without it, the characterization theorem fails.

Open Problem 17.22 (Tight multiclass sample complexity). For finite $|Y| = k$, the best known bounds leave a $\Theta(\log k)$ gap between the Natarajan-dimension-based upper and lower bounds on sample complexity. Does the DS dimension close this gap, yielding a sample complexity of $\Theta(\mathrm{DSdim}/\varepsilon)$ without logarithmic factors?

Open Problem 17.23 (Multiclass online learning). The DS dimension characterizes PAC learnability in the multiclass setting. What is the correct characterization of *online* multiclass learnability? Is there a multiclass analogue of the Littlestone dimension, and does it relate to the DS dimension as the Littlestone dimension relates to VC?

This chapter has traced the grammar growth required to extend binary PAC learning theory to multiclass, real-valued, agnostic, noise-tolerant, and computationally constrained settings. Each extension introduces irreducible new primitives and yields its own characterization theorem or separation result. The next chapter examines what lies beyond these characterizations: the application-layer concepts, the open problems, and the scope boundaries of the theory.

Chapter 18

Frontiers and Open Problems

Every preceding chapter has presented established results: definitions, characterization theorems, separation witnesses, and tight bounds. This chapter presents none of these. Its subject is the boundary of what we know, and its organizing principle is the open question rather than the proven theorem.

The shift in profile is deliberate. Chapters 1–17 follow six pedagogical profiles: foundational definition, canonical proof, separation witness, analogy analysis, computational illustration, and framework bridge. This chapter follows a seventh: the *open frontier*. Each section leads with a question to which no complete answer is known, presents the best partial results, identifies the precise gap between what is known and what is conjectured, and speculates—cautiously—on what kind of argument might close the gap.

No pretense of completeness is made. The problems chosen are those that connect most tightly to the concept graph of this book: the compression conjecture (Chapter 11), deep network generalization (Chapter 12), universal learning beyond countable classes (Chapter 9), computational–statistical tradeoffs (Chapter 16), and the role of Kolmogorov complexity in learning. Other important open problems—online multiclass learnability, private agnostic learning, quantum sample complexity gaps—are noted but not developed.

The chapter has five sections.

1. **The Compression Conjecture** (Section 18.1): does $\mathrm{VCdim}(\mathcal{H}) = d$ imply a compression scheme of size $O(d)$?
2. **Deep Learning and Generalization** (Section 18.2): why do overparameterized networks generalize?
3. **Universal Learning Beyond Countable Classes** (Section 18.3): does the trichotomy extend?
4. **Computational–Statistical Tradeoffs** (Section 18.4): when does computational hardness prevent statistically optimal learning?
5. **Kolmogorov Complexity and the Ideal Learner** (Section 18.5): can the uncomputable yield computable insight?

18.1 The Compression Conjecture

Open Problem 18.1 (Littlestone–Warmuth Compression Conjecture). Let $\mathcal{H} \subseteq \{0,1\}^X$ with $\mathrm{VCdim}(\mathcal{H}) = d < \infty$. Does there exist a sample compression scheme for $\mathcal{H}$ of size $O(d)$?

This conjecture, stated by Littlestone and Warmuth in 1986, is arguably the oldest major open problem in computational learning theory. It asks whether the sample complexity of PAC

learning—which is $\Theta(d/\varepsilon)$—is explained by compression: whether every learnable class can be learned by storing only $O(d)$ examples from the training set and reconstructing a consistent hypothesis from those examples alone.

18.1.1 What is Known

The positive direction is settled in a weak sense.

- **Exponential compression exists.** Moran and Yehudayoff [MY16] proved that every class with $\mathrm{VCdim}(\mathcal{H}) = d$ admits a compression scheme of size $2^{O(d)}$. The proof uses a beautiful argument involving the Radon partition structure of finite VC classes, but the bound is exponentially larger than the conjecture predicts.

- **Special cases are resolved.** Maximum classes (those achieving the Sauer–Shelah bound $\Pi_{\mathcal{H}}(n) = \binom{n}{\leq d}$) admit compression of size exactly d, as shown by Floyd and Warmuth [FW95]. Intersection-closed classes, classes of bounded recursive teaching dimension, and several other families satisfy the conjecture.

- **Compression implies learning.** The converse direction is established: a compression scheme of size k yields PAC learning with sample complexity $O(\frac{k}{\varepsilon}\log\frac{k}{\varepsilon} + \frac{1}{\varepsilon}\log\frac{1}{\delta})$. Thus compression of size $O(d)$ would give an alternative proof of the fundamental theorem of PAC learning, with the added structural insight that the hypothesis depends on only $O(d)$ training points.

Historical Note

The compression conjecture predates the fundamental theorem of statistical learning. Littlestone and Warmuth formulated it in 1986, three years before Blumer, Ehrenfeucht, Haussler, and Warmuth [BEHW89] completed the characterization of PAC learnability through VC dimension. The conjecture was motivated by the observation that specific learnable classes—halfspaces, rectangles, decision lists—all admit small compression schemes, and the belief that this was not coincidental but structural. Four decades later, the general case remains open.

18.1.2 Where the Gap Is

The gap is between $O(d)$ and $2^{O(d)}$. No class is known to require compression size $\omega(d)$, but no proof technique is known to achieve $o(2^d)$ in general.

The difficulty is structural. The Moran–Yehudayoff proof proceeds through a topological argument (the staircase lemma) that inherently produces exponential-size compressions. Improving the bound requires a fundamentally different approach—one that exploits the combinatorial structure of VC classes more directly.

Several partial strategies have been explored:

- **Unlabeled compression.** Allowing the compression to store unlabeled points (without their labels) does not help: the reconstruction map must still determine labels, and known reductions show that unlabeled compression of size k implies labeled compression of size $O(k)$.

- **Algebraic approaches.** For classes definable by polynomial threshold functions, algebraic arguments yield compression of size $O(d)$. But this exploits algebraic structure absent from general VC classes.

- **Recursive approaches.** Decomposing a class into subclasses of smaller VC dimension and compressing each recursively is a natural strategy, but known decompositions (e.g., via Helly-type theorems) produce too many subclasses, losing the linear bound.

18.1.3 What Would Close the Gap

A resolution likely requires one of the following:

1. A new structural theorem about VC classes that controls the geometry of their restrictions to finite sets more tightly than Sauer–Shelah.

2. A connection to a different branch of combinatorics (e.g., matroid theory or extremal set theory) that provides the right decomposition lemma.

3. A counterexample: a family of classes with VC dimension d and no compression scheme of size $o(d^{1+c})$ for some $c > 0$. Even a superlinear lower bound would be a breakthrough.

Remark 18.2. The compression conjecture illustrates a recurring theme in this chapter: the gap between *existential* and *constructive* knowledge. We know that every finite-VC-dimension class is PAC learnable (Chapter 5). We know that compression of size $2^{O(d)}$ exists (Chapter 11). What we do not know is whether these two facts are reflections of the same underlying structure at the right quantitative scale.

18.2 Deep Learning and Generalization

Open Problem 18.3 (Generalization in Overparameterized Networks). Let $\mathcal{H}$ be the class of functions computed by a neural network with $p \gg n$ parameters, trained by gradient descent on n samples. Why does the trained network generalize—i.e., why is $R(\hat{h}) \approx \hat{R}(\hat{h})$—despite $\mathrm{VCdim}(\mathcal{H}) \geq p$?

This is perhaps the most practically urgent open problem in learning theory. Classical generalization bounds predict that a model with more parameters than training examples should overfit catastrophically. Deep networks routinely violate this prediction.

18.2.1 What is Known

Several partial explanations exist, none fully satisfactory.

- **Margin-based bounds.** Bartlett, Foster, and Telgarsky [BFT17] proved that for a depth-L network with weight matrices $W_1, \dots, W_L$, the generalization error is controlled by

$$\frac{1}{n}\left(\frac{\prod_{i=1}^{L} \|W_i\|_\sigma}{\gamma}\right)^2 \cdot \sum_{i=1}^{L} \frac{\|W_i\|_F^2}{\|W_i\|_\sigma^2}$$

  where γ is the margin, $\|\cdot\|_\sigma$ is the spectral norm, and $\|\cdot\|_F$ is the Frobenius norm. This bound is independent of the number of parameters p and depends instead on the *spectral complexity* of the learned weights.

- **PAC-Bayes bounds.** Treating the trained weights as a posterior distribution (e.g., by adding Gaussian noise), PAC-Bayes bounds (Chapter 17) can yield non-vacuous generalization certificates for specific trained networks. Dziugaite and Roy [DR17] demonstrated this empirically.

- **Algorithmic stability.** SGD with bounded iterations satisfies uniform stability with $\beta = O(1/n)$ under convexity assumptions. For non-convex losses, Hardt, Recht, and Singer [HRS16] showed that early stopping provides stability guarantees, but the bounds depend on the number of SGD steps and become vacuous for the training durations used in practice.

- **Compression-based arguments.** If the effective hypothesis after training can be compressed to $k \ll p$ bits, then compression bounds (Chapter 11) apply. Lottery ticket observations—that trained networks contain sparse subnetworks matching full performance—provide indirect evidence for compression, but no rigorous connection to sample complexity has been established.

Computational Illustration

The double descent curve. Consider interpolation threshold: the point where the number of parameters p equals the number of training examples n. Classical theory predicts that test error is U-shaped as a function of model complexity, with minimum at $p \approx n$. Empirically, the test error *decreases again* for $p \gg n$, forming a double descent curve. This was systematically documented by Belkin et al. [BHMM19].
The double descent phenomenon is observed across architectures (linear models, random features, deep networks) and datasets. No existing generalization bound fully explains it: all known bounds are monotone in model complexity and cannot predict the second descent. The phenomenon suggests that the relevant notion of complexity for overparameterized models is not a simple function of parameter count.

18.2.2 Where the Gap Is

The gap has three aspects.

1. **Quantitative vacuity.** Most existing bounds, when evaluated on practical networks, predict generalization error close to 1 (i.e., they are vacuous). The margin bounds of Bartlett–Foster–Telgarsky are the tightest known, but still far from observed test error on benchmarks such as ImageNet.

2. **Architecture dependence.** Existing bounds treat networks as generic function classes parameterized by weight norms. They do not explain why certain architectures (e.g., ResNets, Transformers) generalize better than others with the same parameter count and weight norms.

3. **Optimization dependence.** The generalization behavior of deep networks depends heavily on the optimization algorithm (SGD vs. Adam), learning rate schedule, batch size, and initialization. No existing framework captures how the optimization trajectory selects hypotheses with low population risk from the vast set of interpolating solutions.

18.2.3 What Would Close the Gap

The field appears to need a complexity measure that simultaneously accounts for:

- the architecture (not just the function class),

- the optimization trajectory (not just the endpoint),

- the data distribution (not just worst-case analysis).

Possible directions include:

- **Implicit regularization theory:** proving that gradient descent on specific architectures converges to solutions with low complexity in a suitable measure (e.g., low rank, low spectral norm product, or low description length).

- **Distribution-dependent bounds:** replacing uniform convergence (which bounds $\sup_D |R - \hat{R}|$) with bounds that depend on the specific data distribution, potentially through conditional mutual information (Steinke–Zakynthinou, Chapter 12).

- **Feature learning theory:** understanding how networks transform input representations during training, and why the learned features support generalization.

18.3 Universal Learning Beyond Countable Classes

Open Problem 18.4 (Universal Learning for Uncountable Classes). The trichotomy theorem of Bousquet et al. [BHM+21] characterizes universal learning rates for classes $\mathcal{H}$ with $|\mathcal{H}| \geq 3$. Does an analogous characterization hold for uncountable classes of functions $f\colon X \to [0,1]$ under general loss functions?

18.3.1 What is Known

The trichotomy theorem (Chapter 9) establishes that for hypothesis classes of size at least 3, the optimal universal learning rate is exactly one of three types:
1. Exponential (e^{-n}): if and only if the Littlestone dimension is finite.
2. Linear ($1/n$): if and only if the Littlestone dimension is infinite but no infinite VCL tree exists.
3. Arbitrarily slow: if and only if an infinite VCL tree exists.

This characterization applies to binary classification with the 0-1 loss.

For regression and general losses, partial results exist:

- Hanneke, Moran, and others have extended the trichotomy to multiclass settings with finite label spaces, showing that the DS dimension (Chapter 17) plays a role analogous to the Littlestone dimension.

- For real-valued regression under squared loss, the fat-shattering dimension at all scales characterizes PAC learnability, but the universal learning rate structure is not yet characterized.

18.3.2 Where the Gap Is

The gap has two components:

1. **The loss function.** The trichotomy proof relies on the discreteness of 0-1 loss. For continuous losses (squared loss, absolute loss, logistic loss), the proof technique does not directly apply. The VCL tree, which is the combinatorial object governing the third regime, has no established analogue for continuous losses.

2. **The function class.** For uncountable classes (e.g., all Lipschitz functions, all bounded-variation functions), measurability issues arise: the supremum of uncountably many random variables may not be measurable without additional regularity assumptions. The universal learning framework requires that the learning rate hold for *all* realizable distributions, and handling uncountable classes demands careful measure-theoretic foundations.

18.3.3 What Would Close the Gap

Two ingredients appear necessary:

- A combinatorial object for continuous losses that plays the role of the Littlestone tree and VCL tree in the binary case—a "scale-sensitive game tree" that captures the sequential complexity of the class at all accuracy levels.
- A measurability framework (e.g., based on metric entropy or covering numbers) that allows the universal learning definition to extend cleanly to uncountable classes without requiring separability or other ad hoc assumptions.

18.4 Computational–Statistical Tradeoffs

Open Problem 18.5 (Computational–Statistical Gap). For which concept classes $\mathcal{C}$ does a gap exist between the information-theoretic sample complexity of PAC learning $\mathcal{C}$ and the sample complexity achievable by any polynomial-time algorithm? Can such gaps be characterized combinatorially?

The results of Chapter 16 show that computational hardness can prevent efficient PAC learning even when the VC dimension is finite. But those results are conditional: they assume the hardness of specific cryptographic problems (factoring, LWE, discrete cube root). The deeper question is whether the computational–statistical gap is a structural feature of learning or an artifact of our inability to prove $P \neq NP$.

18.4.1 What is Known

- **Cryptographic hardness.** Kearns and Valiant [KV94] showed that under the Discrete Cube Root Assumption, polynomial-size circuits are not PAC learnable in polynomial time, despite having polynomial VC dimension. The gap is between $O(d/\varepsilon)$ samples (information-theoretically sufficient) and "no polynomial-time algorithm suffices" (computationally).
- **Statistical query lower bounds.** The SQ dimension provides *unconditional* computational lower bounds within the SQ model (Chapter 16). Parities over $\{0,1\}^n$ have VCdim $= n$ but SQdim $= 2^n$: any SQ algorithm needs exponentially many queries. Yet Gaussian elimination learns parities in polynomial time—it is simply not an SQ algorithm. This shows that the SQ model captures one type of computational constraint but not all.
- **Planted clique and related problems.** A growing body of work studies problems where the information-theoretic threshold differs from the apparent computational threshold: planted clique, sparse PCA, community detection. In each case, $\Theta(\sqrt{n})$ appears to be a computational barrier, while information-theoretically $O(\log n)$ samples suffice. But none of these gaps are proven unconditionally.
- **The barrier of Applebaum–Barak–Xiao.** One cannot prove "$P \neq NP$ implies hard PAC learning" via standard Karp reductions. Cryptographic or average-case assumptions are inherently necessary for hardness results in PAC learning. This is a meta-theoretic constraint on the kind of hardness proofs that are possible.

Historical Note

The computational–statistical tradeoff emerged as a distinct research direction in the 2010s, though its roots go back to Kearns and Valiant's 1994 cryptographic hardness results. The key conceptual shift was the recognition that sample complexity and computational complexity are not independent axes of difficulty, but interact in subtle ways.

A problem may have low sample complexity (few examples suffice) but high computational complexity (no efficient algorithm can use those examples), or vice versa. The SQ model, introduced by Kearns [Kea98], provided the first framework where computational lower bounds could be proved unconditionally, but only within a restricted model of computation.

18.4.2 Where the Gap Is

The fundamental gap is proof-theoretic: we lack techniques to prove unconditional computational lower bounds for PAC learning.

1. **No unconditional separations.** Every known example of a computational–statistical gap relies on a complexity-theoretic assumption. Proving an unconditional gap would require proving $P \neq NP$ or something similarly difficult.

2. **No combinatorial characterization.** For information-theoretic learnability, we have a complete combinatorial characterization: finite VC dimension. For computationally efficient learnability, no analogous characterization exists. We do not know what combinatorial property of a concept class determines whether it is efficiently learnable.

3. **Model dependence.** Lower bounds proved in the SQ model, the low-degree polynomial model, or the sum-of-squares hierarchy apply only to those specific computational models. An algorithm outside the model (like Gaussian elimination for parities) may circumvent the lower bound. No model of computation is known to be simultaneously natural, powerful enough to capture practical algorithms, and amenable to unconditional lower bounds.

18.4.3 What Would Close the Gap

- **A natural proof barrier for learning.** Razborov and Rudich's natural proofs barrier explains why circuit lower bounds are hard. An analogous barrier for learning—explaining why computational lower bounds for PAC learning are hard—would clarify the structural limits of current proof techniques.

- **A unifying computational model.** A model of bounded computation that captures SQ algorithms, low-degree methods, sum-of-squares, and gradient descent, while admitting provable lower bounds, would provide a natural home for computational–statistical tradeoff results.

- **Conditional characterizations.** Even under assumptions like $P \neq NP$ or the hardness of LWE, a characterization of efficiently learnable classes (analogous to the fundamental theorem for information-theoretic learnability) would be a major advance.

18.5 Kolmogorov Complexity and the Ideal Learner

Open Problem 18.6 (Computable Approximations to Solomonoff Induction). Solomonoff's universal prior $M(x) = \sum_{p:\, U(p)=x} 2^{-|p|}$ defines an ideal but uncomputable learner. To what extent can computable approximations to M achieve the learning-theoretic properties of the ideal learner?

This section concerns the deepest stratum of the theory: the connection between algorithmic information theory and learning.

18.5.1 What is Known

- **Solomonoff's completeness theorem.** For any computable measure μ on infinite binary sequences, the universal semimeasure M satisfies

$$\sum_{n=1}^{\infty} \mathbb{E}_{x_{1:n}\sim\mu}\left[D_{\mathrm{KL}}(\mu(\cdot \mid x_{1:n}) \,\|\, M(\cdot \mid x_{1:n}))\right] < \infty.$$

  That is, the cumulative prediction error of M relative to any computable environment is bounded. This makes M the "ideal learner": it converges to the true distribution at a rate dominated by the Kolmogorov complexity of the environment.

- **MDL and MML as approximations.** The minimum description length principle (Rissanen [Ris84]) and minimum message length (Wallace and Boulton [WB68]) are computable approximations to the Kolmogorov-optimal encoding. Both embody a version of Occam's razor: prefer the hypothesis with the shortest total description (model + data given model).

- **Occam bounds via Kolmogorov complexity.** Li, Tromp, and Vitányi [LV08] showed that replacing description length with Kolmogorov complexity in Occam-style PAC bounds yields tighter (but uncomputable) generalization guarantees. The bound states that the generalization error of a hypothesis h is controlled by $K(h)/n$, where $K(h)$ is the Kolmogorov complexity of h and n is the sample size.

- **Algorithmic probability and universal learning.** The connection between Solomonoff's prior and the universal learning framework of Chapter 9 is suggestive but not formal. Universal learning allows distribution-dependent rates; Solomonoff's prior achieves distribution-dependent convergence. But the universal learning framework is information-theoretic (it asks only about sample complexity), while Solomonoff's prior is algorithmic (it asks about description complexity). The precise relationship between these two notions of universality is open.

Historical Note

Solomonoff's 1964 paper [Sol64] predates Gold's 1967 formalization of identification in the limit and Valiant's 1984 introduction of PAC learning. It is arguably the first formal proposal for a universal learning algorithm, though it was formulated in the language of algorithmic information theory rather than learning theory. The intellectual genealogy runs: Solomonoff (1964) → Kolmogorov complexity (1965) → MDL (1978) → Occam bounds in PAC (1987) → compression schemes (1986). The compression conjecture of Section 18.1 can be viewed as asking whether the Occam principle—short description implies good generalization—holds at the tightest possible quantitative level.

18.5.2 Where the Gap Is

1. **Uncomputability.** Kolmogorov complexity and Solomonoff's prior are uncomputable. Every computable approximation introduces a gap: MDL uses a fixed model class rather than all programs; bounded Solomonoff approximations (running programs for at most t steps) introduce a time–complexity tradeoff that is poorly understood.

2. **No convergence rate for approximations.** For the ideal learner M, the convergence rate is known: the expected KL divergence is $O(K(\mu)/n)$ where $K(\mu)$ is the complexity of the true environment. For computable approximations, no analogous rate is established that holds uniformly over all computable environments.

3. **The gap between prediction and classification.** Solomonoff's framework concerns *sequence prediction*: predicting the next bit. PAC learning concerns *concept identification*: finding a hypothesis close to the target. The formal relationship between these two tasks is well understood in specific settings (e.g., the realizable case with i.i.d. data), but the general connection—especially in the agnostic or adversarial setting—remains underdeveloped.

18.5.3 What Would Close the Gap

- **Time-bounded Solomonoff induction.** A theory of $M^t(x) = \sum_{p:\, U(p)=x,\, \mathrm{time}(p)\leq t} 2^{-|p|}$ with provable convergence rates as a function of both n (sample size) and t (computation time) would bridge the gap between the ideal and the computable. Such a theory would need to handle the fact that M^t is not a semimeasure for finite t, and that the dominant programs may have very long running times.
- **Kolmogorov complexity as a complexity measure for PAC learning.** Formalizing $K(\mathcal{H})$ (the description complexity of a hypothesis class) as a complexity measure in the sense of Chapter 10—with upper and lower bound relationships to VC dimension, Rademacher complexity, and compression size—would integrate algorithmic information theory into the concept graph of this book.
- **A computable universal learner.** An algorithm that, for every computable concept class $\mathcal{C}$ and every realizable distribution D, achieves PAC learning with sample complexity depending on $K(\mathcal{C})$, would be a computable analogue of the Solomonoff ideal. Whether such an algorithm exists is itself an open question, closely related to the universal learning framework of Chapter 9.

18.6 Further Frontiers

The five problems above are those most tightly connected to the conceptual framework of this book. We briefly note several other directions where the theory is actively expanding. Each lies at the boundary of the framework developed in Chapters 1–17 and involves mathematical structures that do not reduce to the types established there.

Private learning. Differential privacy constrains the learner: the output hypothesis must not reveal too much about any individual training example. Kasiviswanathan et al. showed that the sample complexity of private PAC learning is $\Theta(d/\varepsilon + \log|\mathcal{X}|/(\varepsilon\alpha))$ for pure differential privacy, where α is the privacy parameter. For approximate differential privacy, the Littlestone dimension characterizes private learnability (Alon et al., Bun et al.), connecting online learning theory to privacy in an unexpected way. The tight sample complexity of private agnostic learning remains open.

Active learning. An active learner chooses which examples to query, rather than receiving them passively. The label complexity—the number of label queries needed for PAC learning—can be exponentially smaller than the sample complexity of passive learning. Hanneke showed that the disagreement coefficient controls the label complexity for general hypothesis classes. Open problems include tight characterizations of active learning under agnostic noise and the computational complexity of active learning algorithms.

Transfer, meta-learning, and continual learning. These paradigms concern learning from multiple related tasks. Transfer learning asks: when does experience on a source task improve learning on a target task? Meta-learning asks: can the learner learn to learn faster across a distribution of tasks? Continual learning asks: can the learner acquire new tasks without forgetting old ones? Formal frameworks exist (e.g., Baxter's model of learning to learn, the

PAC-Bayes approach to meta-learning), but no characterization theorems analogous to the VC characterization have been established. The fundamental question is whether the combinatorial objects governing multi-task learning can be expressed in terms of the single-task dimensions of this book, or whether genuinely new complexity measures are needed.

Online multiclass learnability. The Littlestone dimension characterizes online binary learnability. For multiclass online learning, the situation is more complex: the Littlestone dimension of the binary reductions does not always capture the optimal mistake bound. A combinatorial characterization of online multiclass learnability, and its relationship to the DS dimension, is an active area of research.

Quantum sample complexity. Quantum PAC learning, where the learner receives quantum superpositions of labeled examples, can have polynomially different sample complexity from the classical setting. The no-signaling constraint on learnability is a recent development. A full characterization of when quantum examples help—in terms of classical complexity measures—is open.

Computational Illustration

The shape of the frontier. The open problems of this chapter are not isolated. They form a connected subgraph in the concept graph, linked by shared dependencies. The compression conjecture involves the same VC dimension that governs PAC learning, which connects to the computational hardness that creates computational–statistical gaps, which connects to the SQ dimension, which connects to the Kolmogorov complexity that underlies the ideal learner. The deep learning generalization problem is linked to compression (through lottery tickets), to margin theory (through spectral bounds), and to algorithmic stability (through SGD analysis). Universal learning connects to the Littlestone dimension and therefore to private learning.
This interconnection is not coincidental. The open problems persist *because* they are entangled: a resolution of the compression conjecture would likely yield insight into generalization bounds; a combinatorial characterization of efficient learnability would likely require new structural theorems about VC classes; a computable approximation to Solomonoff induction would likely connect to both compression and universal learning. The frontier of formal learning theory is not a collection of independent problems but a single, tightly coupled surface.

This chapter has surveyed what we do not know. The theory developed in Chapters 1–17 provides a rigorous, interconnected framework for what we do know: characterization theorems, tight bounds, separation results, and structural analogies. The open problems of this chapter mark the places where that framework reaches its current limits. They are not embarrassments but invitations.

Appendix A

Complete Edge Inventory

This appendix lists all 260 edges in the companion knowledge graph, organized by the 13 relation types. Each table shows representative edges; the full inventory is machine-readable in `flt_concept_graph.json`.

Column conventions. Source and Target give graph node identifiers. *Citation* gives the abbreviated bibliography key. Relation-specific columns (*Witness*, *Obstruction type*, *Generalization type*) appear where the schema requires them.

A.1 `defined_using` (99 edges)

Table A.1: `defined_using` edges (representative sample; 99 total).

Source	Target	Note
concept	domain	
concept	label	
concept_class	concept	
hypothesis_space	concept	
proper_flag	hypothesis_space	
··· *94 additional edges in* `flt_concept_graph.json`		

A.2 `instance_of` (25 edges)

Table A.2: `instance_of` edges (representative sample; 25 total).

Source	Target	Note
text_presentation	data_stream	
informant_presentation	data_stream	
noisy_input	data_stream	
time_index	domain	
iterative_learner	learner	
··· *20 additional edges in* `flt_concept_graph.json`		

A.3 `characterizes` (24 edges)

Table A.3: `characterizes` edges (representative sample; 24 total).

Source	Target	Citation
vc_characterization	pac_learning	[BEHW89]
vc_characterization	vc_dimension	[BEHW89]
star_number	vc_dimension	[Han24]
trial_and_error	ex_learning	[Gol65]
ds_dimension	pac_learning	[BCD+22]
· · · 19 additional edges in `flt_concept_graph.json`		

A.4 `analogy` (32 edges)

Table A.4: `analogy` edges (representative sample; 32 total).

Source	Target	Obstruction type	Citation
sample_complexity	vc_dimension	one_way_theorem_only	
ordinal_vc_dim	mind_change_ordinal	proof_method_mismatch	
mistake_bound	littlestone_dimension	missing_equiv_witness	
label_complexity	sample_complexity	missing_equiv_witness	
concept_drift	online_learning	data_model_mismatch	
· · · 27 additional edges in `flt_concept_graph.json`			

A.5 `measures` (22 edges)

Table A.5: `measures` edges (representative sample; 22 total).

Source	Target	Note
vc_dimension	concept_class	
shatters	concept_class	
littlestone_dimension	concept_class	
star_number	version_space	
eluder_dimension	version_space	
· · · 17 additional edges in `flt_concept_graph.json`		

A.6 `used_in_proof` (14 edges)

Table A.6: `used_in_proof` edges (representative sample; 14 total).

Source	Target	Citation
gold_theorem	text_presentation	[Gol67]
gold_theorem	concept_class	[Gol67]
nfl_theorem	learner	[WM97]
nfl_theorem	inductive_bias	[WM97]

continued on next page

Source	Target	Citation
pac_lower_bound	vc_dimension	[BEHW89]
··· 9 additional edges in `flt_concept_graph.json`		

A.7 `does_not_imply` (9 edges)

Table A.7: `does_not_imply` edges (all 9).

Source	Target	Witness	Citation
pac_learning	mistake_bounded	Thresholds on $\mathbb{R}$: VCdim $= 1$, Ldim $= \infty$	[Lit88]
ex_learning	pac_learning	All finite subsets of $\mathbb{N}$: identifiable from text, VCdim $= \infty$	[Gol67]
vc_dimension	computational_hardness	Poly-size circuits under DCRA: finite VCdim but not efficiently PAC-learnable	[KV94]
vc_dimension	sq_dimension	Parities: VCdim $= n$, SQdim $= 2^n$	[BFJ+94]
proper_improper_separation	pac_learning	3-term DNF: NP-hard proper, poly-time improper	[PV88]
4 additional edges in `flt_concept_graph.json`			

A.8 `upper_bounds` (8 edges)

Table A.8: `upper_bounds` edges (all 8).

Source	Target	Citation
littlestone_dimension	vc_dimension	[Lit88]
rademacher_complexity	generalization_error	[BM02]
pac_bayes_bound	generalization_error	[McA99]
information_theoretic_bound	generalization_error	[XR17]
vc_dimension	rademacher_complexity	[Sau72]
margin_theory	generalization_error	[BFT17]
2 additional edges in `flt_concept_graph.json`		

A.9 `restricts` (8 edges)

Table A.9: `restricts` edges (representative sample; 8 total).

Source	Target	Citation
bc_learning	ex_learning	[CS83]
agnostic_pac_learning	pac_learning	[Hau92]
universal_learning	pac_learning	[BHM+21]
meta_pac_bound	pac_learning	[Bax00]

Source	Target	Citation
anomalous_learning	ex_learning	
3 additional edges in `flt_concept_graph.json`		

A.10 `extends_grammar` (8 edges)

Table A.10: `extends_grammar` edges (representative sample; 8 total).

Source	Target	Generalization type	Citation
ordinal_vc_dim	vc_dimension	new_primitives_required	
pseudodimension	vc_dimension	new_primitives_required	
natarajan_dimension	vc_dimension	new_primitives_required	
mind_change_ordinal	mind_change_count	new_primitives_required	
pushdown_automaton	dfa	new_primitives_required	
3 additional edges in `flt_concept_graph.json`			

A.11 `lower_bounds` (5 edges)

Table A.11: `lower_bounds` edges (all 5).

Source	Target	Citation
gold_theorem	ex_learning	[Gol67]
pac_lower_bound	pac_learning	[BEHW89]
pac_lower_bound	sample_complexity	[BEHW89]
proper_improper_separation	pac_learning	[PV88]
computational_hardness	pac_learning	[KV94]

A.12 `strictly_stronger` (4 edges)

Table A.12: `strictly_stronger` edges (all 4).

Source	Target	Witness	Citation
universal_learning	pac_learning	Classes with infinite VCdim but no infinite Littlestone tree	[BHM+21]
ex_learning	finite_learning	Classes requiring unbounded mind changes (e.g., pattern languages)	[Gol67]
online_learning	pac_learning	Finite Ldim $\Rightarrow$ finite VCdim; thresholds on finite domains: Ldim $<$ VCdim	[Lit88]
ds_dimension	natarajan_dimension	Hyperbolic pseudo-manifold construction	[BCD+22]

A.13 `requires_assumption` (2 edges)

Table A.13: `requires_assumption` edges (all 2).

Source	Target	Citation
computational_hardness	inductive_bias	[ABX08]
posterior_consistency	bayesian_inference	[Sch65]

Full inventory. The complete 260-edge listing is available in `flt_concept_graph.json` (v1.0). Programmatic access is recommended for any systematic traversal; see Appendix C for the validation suite.

Appendix B

Graph Traversal Demonstrations

These 15 tasks demonstrate the graph's utility for cross-paradigm reasoning. Each traversal states a query, lists the sequence of edges traversed, and assembles an answer from the path. Edge notation: source $\xrightarrow{\texttt{relation}}$ target.

Graph Traversal

T01: Prerequisite retrieval for VC dimension.

Query: What concepts must be defined before VCdim can be stated?
Path:

1. vc_dimension $\xrightarrow{\texttt{defined\_using}}$ shatters
2. shatters $\xrightarrow{\texttt{defined\_using}}$ concept_class (via shatters prerequisites)
3. concept_class $\xrightarrow{\texttt{defined\_using}}$ concept
4. concept $\xrightarrow{\texttt{defined\_using}}$ domain
5. concept $\xrightarrow{\texttt{defined\_using}}$ label

Answer: The prerequisite chain is: domain, label → concept → concept_class → shatters → vc_dimension. In the graph, the single direct `defined_using` edge from vc_dimension targets shatters.

Graph Traversal

T02: What characterizes PAC learnability?

Query: What are all known characterizations of PAC learnability?
Path: Collect all nodes X where X $\xrightarrow{\texttt{characterizes}}$ pac_learning.

1. vc_characterization $\xrightarrow{\texttt{characterizes}}$ pac_learning
2. vc_dimension $\xrightarrow{\texttt{characterizes}}$ pac_learning (via vc_characterization)
3. ds_dimension $\xrightarrow{\texttt{characterizes}}$ pac_learning
4. fundamental_theorem $\xrightarrow{\texttt{characterizes}}$ pac_learning
5. occam_algorithm $\xrightarrow{\texttt{characterizes}}$ pac_learning

Answer: Five nodes characterize PAC learnability: vc_dimension, vc_characterization, ds_dimension, fundamental_theorem, and occam_algorithm.

Graph Traversal

T03: Is a PAC-learnable class online-learnable?

Query: Is a PAC-learnable class necessarily online-learnable with finite mistakes?
Path:

1. pac_learning $\xrightarrow{\texttt{does\_not\_imply}}$ mistake_bounded

Answer: No. The edge carries witness: thresholds on $\mathbb{R}$ have VCdim $= 1$ but Ldim $= \infty$ [Lit88].

Graph Traversal

T04: Upper bounds on generalization error.

Query: What generalization bounds are available and what do they each depend on?
Path: Find all X with X $\xrightarrow{\texttt{upper\_bounds}}$ generalization_error, then follow `defined_using` from each.

1. rademacher_complexity $\xrightarrow{\texttt{upper\_bounds}}$ generalization_error [BM02]
2. pac_bayes_bound $\xrightarrow{\texttt{upper\_bounds}}$ generalization_error [McA99]
3. information_theoretic_bound $\xrightarrow{\texttt{upper\_bounds}}$ generalization_error [XR17]
4. margin_theory $\xrightarrow{\texttt{upper\_bounds}}$ generalization_error [BFT17]

Answer: Four bound families provide upper bounds on generalization error: Rademacher complexity, PAC-Bayes, information-theoretic, and margin-based bounds. Each has distinct `defined_using` dependencies (e.g., Rademacher complexity depends on the growth function and hypothesis space).

Graph Traversal

T05: Cross-paradigm Bayesian–multiclass reasoning.

Query: Can a Bayesian learner be PAC-Bayes analyzed for a multiclass problem where $d_{\mathrm{N}} < \infty$ but the class is not learnable?
Path:

1. bayesian_learner $\xrightarrow{\texttt{instance\_of}}$ learner
2. pac_bayes_bound $\xrightarrow{\texttt{upper\_bounds}}$ generalization_error
3. natarajan_dimension $\xrightarrow{\texttt{does\_not\_imply}}$ pac_learning
4. ds_dimension $\xrightarrow{\texttt{characterizes}}$ pac_learning

Answer: PAC-Bayes bounds apply to any posterior Q (edge 2), so a Bayesian learner (edge 1) can be analyzed. However, finite Natarajan dimension does not imply PAC learnability (edge 3); the DS dimension is the correct multiclass characterization (edge 4). If DSdim $= \infty$, the bounds may be vacuous despite $d_{\mathrm{N}} < \infty$.

Graph Traversal

T06: BC $\supsetneq$ Ex $\supsetneq$ FIN hierarchy.

Query: What is the strict power hierarchy among success criteria in Gold's model?
Path:

1. bc_learning $\xrightarrow{\texttt{restricts}}$ ex_learning [CS83]
2. ex_learning $\xrightarrow{\texttt{strictly\_stronger}}$ finite_learning [Gol67]

Answer: **BC $\supsetneq$ Ex $\supsetneq$ FIN**. BC generalizes Ex by removing the syntactic convergence

requirement (edge 1). Ex is strictly stronger than FIN because some classes require unbounded mind changes (edge 2).

Graph Traversal

T07: Why concept drift ≠ online learning (obstruction).

Query: Why is the analogy between concept drift and online learning not a formal theorem?
Path:

1. concept_drift $\xrightarrow{\texttt{analogy}}$ online_learning, obstruction type: `data_model_mismatch`

Answer: Data model mismatch: drift uses i.i.d. draws per step from a slowly changing distribution; online learning uses adversarially chosen instances from a fixed class.

Graph Traversal

T08: Which generalizations of VC dimension required new primitives?

Query: Which generalizations of VCdim required inventing new formal primitives?
Path: Find all X with X $\xrightarrow{\texttt{extends\_grammar}}$ vc_dimension.

1. pseudodimension $\xrightarrow{\texttt{extends\_grammar}}$ vc_dimension
2. natarajan_dimension $\xrightarrow{\texttt{extends\_grammar}}$ vc_dimension
3. ordinal_vc_dim $\xrightarrow{\texttt{extends\_grammar}}$ vc_dimension

Answer: Three nodes extend VC grammar: pseudodimension (witness thresholds for real-valued functions), Natarajan dimension (two-coloring for multiclass), and ordinal VC dimension (transfinite generalization).

Graph Traversal

T09: What prevents efficient PAC learning even when VCdim $< \infty$?

Query: What prevents efficient PAC learning even when the VC dimension is finite?
Path:

1. computational_hardness $\xrightarrow{\texttt{lower\_bounds}}$ pac_learning [KV94]
2. computational_hardness $\xrightarrow{\texttt{used\_in\_proof}}$ vc_dimension
3. computational_hardness $\xrightarrow{\texttt{requires\_assumption}}$ inductive_bias [ABX08]
4. vc_dimension $\xrightarrow{\texttt{does\_not\_imply}}$ computational_hardness [KV94]

Answer: Computational hardness results (edge 1) show that under cryptographic assumptions (edge 3), polynomial-size circuits have finite VCdim but are not efficiently PAC-learnable. Finite VCdim alone does not imply tractability (edge 4).

Graph Traversal

T10: Curriculum ordering for PAC learning.

Query: Generate a reading order for learning PAC theory from scratch.
Path: Topological sort of the `defined_using` subgraph restricted to PAC-related nodes.
Answer: One valid ordering:

1. domain, label
2. concept (depends on domain, label)

3. shatters, iid_sample
4. concept_class, hypothesis_space
5. pac_learning, generalization_error
6. sample_complexity
7. vc_dimension
8. vc_characterization
9. fundamental_theorem

Multiple valid topological orderings exist; the key constraint is that each node appears after all its `defined_using` targets.

Graph Traversal

T11: Is reinforcement learning covered in this graph?

Query: Is reinforcement learning covered in this graph?
Path: Check for scope_boundary_rl node.
Answer: Explicitly excluded. The node scope_boundary_rl states that RL extends bandits to sequential decision-making with a different problem structure (MDP, policy, value function). No edges connect to kernel nodes.

Graph Traversal

T12: Count separation results.

Query: List all known separations between learning paradigms with their witnesses.
Path: Collect all `does_not_imply` and `strictly_stronger` edges.
Answer: 9 `does_not_imply` edges + 4 `strictly_stronger` edges = 13 total separation results. Each carries a `witness` field and a `citation`. See Tables A.7 and A.12 in Appendix A.

Graph Traversal

T13: How do VCdim, Ldim, SQdim, and star number relate?

Query: How do VCdim, Littlestone dim, SQ dim, and star number relate to each other?
Path:

1. star_number $\xrightarrow{\texttt{characterizes}}$ vc_dimension [Han24]
2. littlestone_dimension $\xrightarrow{\texttt{upper\_bounds}}$ vc_dimension [Lit88]
3. sq_dimension $\xrightarrow{\texttt{analogy}}$ vc_dimension (exponential gap possible: parities)
4. sq_dimension $\xrightarrow{\texttt{does\_not\_imply}}$ vc_dimension (parities: VCdim $= n$, SQdim $= 2^n$)

Answer: Star number is equivalent to VCdim (edge 1). Finite Ldim implies finite VCdim but the gap can be infinite (edge 2). SQ dimension is structurally analogous but formally independent, with exponential separations (edges 3–4).

Graph Traversal

T14: Real-valued extension of VC dimension.

Query: What changes when moving from binary classification to real-valued prediction?
Path:

1. pseudodimension $\xrightarrow{\texttt{extends\_grammar}}$ vc_dimension

2. pseudodimension $\xrightarrow{\texttt{restricts}}$ fat_shattering_dimension
3. fat_shattering_dimension $\xrightarrow{\texttt{characterizes}}$ real_valued_pac (via appropriate node)

Answer: Pseudodimension extends VC grammar by introducing witness thresholds for real-valued functions (edge 1). It removes the scale-sensitivity from fat-shattering dimension (edge 2, constraint removal). Fat-shattering adds a margin parameter γ and characterizes real-valued PAC learnability. Three nodes in total extend the VC grammar: pseudodimension, natarajan_dimension, ordinal_vc_dim.

Graph Traversal

T15: Does the fundamental theorem extend to multiclass?

Query: Does the fundamental theorem of statistical learning extend to multiclass?
Path:

1. Check: fundamental_theorem has no direct edges to multiclass-specific nodes.
2. natarajan_dimension $\xrightarrow{\texttt{does\_not\_imply}}$ pac_learning — finite d_{N} insufficient when $|Y| = \infty$.
3. ds_dimension $\xrightarrow{\texttt{characterizes}}$ pac_learning [BCD+22]
4. natarajan_dimension $\xrightarrow{\texttt{extends\_grammar}}$ vc_dimension
5. ds_dimension $\xrightarrow{\texttt{strictly\_stronger}}$ natarajan_dimension [BCD+22]

Answer: The fundamental theorem (VCdim $\leftrightarrow$ PAC) is specific to binary classification (no multiclass edges from edge 1). For multiclass: Natarajan dimension gives bounds with a $\log k$ gap but does not characterize learnability when $|Y| = \infty$ (edge 2). The DS dimension is the correct multiclass characterization (edge 3), and it is strictly stronger than Natarajan dimension (edge 5).

Appendix C

Graph Validation

This appendix documents the validation infrastructure that enforces structural integrity of `flt_concept_graph.json`. Two scripts, both located in `scripts/`, perform complementary checks: `validate_graph.py` enforces the embedded JSON schema and all relational constraints, while `validate_bibliography_links.py` cross-references every citation field against the BibTeX file. Together they ensure that the graph remains self-consistent, fully cited, and release-ready.

C.1 Validation Scripts

The concept graph embeds its own JSON schema in the `meta.json_schema` object. Both validators read this schema at runtime, so adding a new relation type, required field, or status value automatically propagates to the checks without modifying validator code.

Table C.1: Companion files relevant to validation.

File	Purpose
`flt_concept_graph.json`	The knowledge graph (142 nodes, 260 edges, 13 relation types).
`flt_bibliography.bib`	Complete BibTeX bibliography (∼120 entries).
`scripts/validate_graph.py`	Schema and constraint validator (13 checks).
`scripts/validate_bibliography_links.py`	Bibliography cross-reference validator (3 checks).
`scripts/run_reference_tasks.py`	Executes the 15 benchmark tasks against the graph.
`scripts/evaluate_reference_tasks.py`	Scores task outputs against gold answers.

Invocation. Both validators accept `--graph` and `--bib` flags with sensible defaults (the repository root).

```
python3 scripts/validate_graph.py \
    --graph flt_concept_graph.json \
    --bib   flt_bibliography.bib

python3 scripts/validate_bibliography_links.py \
    --graph flt_concept_graph.json \
    --bib   flt_bibliography.bib
```

Both scripts exit with code 0 on success and code 1 on failure, making them suitable for continuous-integration pipelines. A trailing-comma tolerance layer (regex-based) strips syntac-

tic noise before JSON parsing, so hand-edited graph files with accidental trailing commas do not cause spurious parse failures.

C.2 The Thirteen Validation Checks

`validate_graph.py` executes the following checks in order. A single failure in any check causes the overall result to be `FAIL`.

Table C.2: Validation checks performed by `validate_graph.py`.

#	Check name	Description
1	JSON parses correctly	The graph file is valid JSON after trailing-comma stripping. Executed before the `Validator` object is constructed; a parse failure terminates immediately.
2	Unique node IDs	Every `id` field across the 142-element node array is unique. Collected IDs are cached for use by subsequent checks.
3	Edge endpoints exist	Every edge `source` and `target` value appears in the set of declared node IDs from Check 2.
4	Relation values in enum	Every edge `relation` value is one of the 13 strings declared in `meta.json_schema.relation_enum`.
5	Required node fields	Every node contains all 8 required fields: `id`, `name`, `category`, `layer`, `status`, `claim_type`, `description`, `formal_definition`.
6	Required edge fields	Every edge contains all 3 universal required fields: `source`, `target`, `relation`.
7	Edge constraints	Relation-specific required fields are present and non-empty. See §C.3 for the full constraint table.
8	No duplicate edges	No two edges share the same (`source`, `relation`, `target`) triple.
9	Bib keys resolve	All values in node `bib_keys` arrays, `provenance.introduced_by`, `provenance.proved_by`, and edge `citation` fields that match the BibTeX key pattern `[A-Za-z]+[0-9]{4}[a-z]?` resolve against entries in the bibliography file.
10	Node count matches meta	The actual length of the `nodes` array equals `meta.node_count` (currently 142).
11	Edge count matches meta	The actual length of the `edges` array equals `meta.edge_count` (currently 260).
12	Status values in enum	Every node `status` value is one of: `defined`, `proved`, `deferred`, `scope_note`.

continued on next page

#	Check name	Description
13	No `generalizes_unclassified`	No edges carry the legacy relation `generalizes_unclassified`. All such edges must be reclassified as either `restricts` (constraint removal) or `extends_grammar` (new primitives) before release.

C.3 Edge Constraints

Not all relation types carry the same metadata. The schema declares relation-specific *required fields* beyond the universal triple (`source`, `target`, `relation`). Check 7 enforces these constraints.

Table C.3: Relation-specific required fields enforced by Check 7.

Relation type	Required fields	Rationale
`strictly_stronger`	`witness`, `citation`	Strict hierarchy requires a separating construction and a proof reference.
`does_not_imply`	`witness`, `citation`	Negative result requires an explicit counterexample and a proof reference.
`characterizes`	`citation`	Equivalence theorem requires a traceable reference.
`upper_bounds`	`citation`	Bound statement requires a proof reference.
`lower_bounds`	`citation`	Bound statement requires a proof reference.
`requires_assumption`	`citation`	Conditional result requires a reference.
`used_in_proof`	`citation`	Proof dependency requires a reference to the theorem that uses the concept.
`defined_using`	*none beyond triple*	Definitional edges are self-documenting from node definitions.
`instance_of`	*none beyond triple*	Taxonomic specializations are self-documenting.
`measures`	*none beyond triple*	Measurement relationships are self-documenting.
`analogy`	*none beyond triple*	Carries optional `obstruction_type` and `obstruction` fields.
`restricts`	*none beyond triple*	May carry optional `note`.
`extends_grammar`	*none beyond triple*	May carry optional `generalization_type`.

Design principle. The seven relations in the upper block encode mathematical claims—equivalences, bounds, separations, conditional validity, proof dependencies—and therefore re-

quire traceable citations. The two separation types (`strictly_stronger` and `does_not_imply`) additionally require a `witness` field: a concrete concept class, construction, or example that demonstrates the separation. The six relations in the lower block encode structural or definitional relationships that do not assert a mathematical result and therefore need no external citation.

C.4 The Thirteen Relation Types

Each relation type has a forward reading, an inverse reading, and specific semantics. Table C.4 documents all 13, ordered by frequency in the graph.

Table C.4: Relation types with semantics and edge counts.

Relation	Edges	Semantics (forward → inverse)
`defined_using`	99	X's formal definition directly invokes Y. Forms the definitional dependency DAG. Inverse: `definition_of`.
`analogy`	32	Structural parallel that is formally independent. Symmetric: A `analogy` B iff B `analogy` A. Each edge carries an optional `obstruction_type` field (one of: `data_model_mismatch`, `missing_equiv_witness`, `one_way_theorem_only`, `proof_method_mismatch`) classifying why the analogy does not upgrade to a theorem.
`instance_of`	25	X is a specific instance or specialization of Y. Inverse: `has_instance`.
`characterizes`	24	$X \leftrightarrow Y$: equivalence under the stated conditions (the fundamental "iff" edges). Inverse: `characterized_by`. Requires `citation`.
`measures`	22	Complexity measure X quantifies a property of object Y. Inverse: `measured_by`.
`used_in_proof`	14	Theorem X invokes concept Y in its statement or proof. Distinct from `defined_using`: the latter is a definitional prerequisite, while `used_in_proof` records proof-level dependencies. Inverse: `proves_about`. Requires `citation`.
`does_not_imply`	9	X does *not* imply Y, with a concrete separating witness. These are the negative-layer edges that encode the field's "what does not hold" structure. Inverse: `not_implied_by`. Requires `witness` and `citation`.

continued on next page

Relation	Edges	Semantics (forward → inverse)
`restricts`	8	X generalizes Y by removing a constraint: $Y = X+$ additional restriction. Conservative generalization—no new formal primitives are needed. Inverse: `restricted_from`.
`extends_grammar`	8	X generalizes Y but required inventing new concepts or primitives not present in Y. Generative grammar expansion. Each edge carries an optional `generalization_type` field (typically `new_primitives_required`). Inverse: `grammar_extended_by`.
`upper_bounds`	8	X provides an upper bound on Y. Inverse: `bounded_above_by`. Requires `citation`.
`lower_bounds`	5	X provides a lower bound on Y. Inverse: `bounded_below_by`. Requires `citation`.
`strictly_stronger`	4	$X \supsetneq Y$: X implies Y but not conversely, with a known separating witness. These edges encode the strict hierarchies among learning criteria. Inverse: `strictly_weaker`. Requires `witness` and `citation`.
`requires_assumption`	2	X holds only if Y is assumed (e.g., cryptographic assumptions for hardness results). Inverse: `assumed_by`. Requires `citation`.

C.5 Bibliography Link Validation

`validate_bibliography_links.py` performs three additional checks beyond those in the graph validator:

1. **Missing keys** (FAIL). BibTeX keys referenced in node `bib_keys`, `provenance.introduced_by`, `provenance.proved_by`, or edge `citation` fields that do not appear in `flt_bibliography.bib`. Each missing key is listed with all locations where it is referenced.
2. **Malformed bib keys** (FAIL). Entries in the `.bib` file whose keys do not match the canonical pattern `[A-Za-z]+[0-9]{4}[a-z]?` (e.g., `BlumerEhrenfeuchtHausslerWarmuth1989`).
3. **Orphan keys** (WARN). Entries in the `.bib` file that are never referenced by any node or edge. These do not cause a failure but indicate potential dead references.

The script reports the total number of BibTeX entries and unique keys referenced in the graph, then lists any missing, malformed, or orphaned keys with their locations.

C.6 Sample Validator Output

When all checks pass, `validate_graph.py` produces:

```
================================================================
  FLT Concept Graph Validation
================================================================

  [PASS]  JSON parses correctly

  [PASS]  Unique node IDs

  [PASS]  Edge endpoints exist

  [PASS]  Relation values in enum

  [PASS]  Required node fields

  [PASS]  Required edge fields

  [PASS]  Edge constraints

  [PASS]  No duplicate edges

  [PASS]  Bib keys resolve

  [PASS]  Node count matches meta

  [PASS]  Edge count matches meta

  [PASS]  Status values in enum

  [PASS]  No generalizes_unclassified edges

================================================================
  13/13 checks passed, 0 failed.
  RESULT: PASS
================================================================
```

When a check fails, the validator prints up to 20 detail lines identifying specific violations. For example, a dangling edge endpoint produces:

```
  [FAIL]  Edge endpoints exist
          Edge 47: source 'nonexistent_node' is not a node ID
```

A missing witness on a separation edge produces:

```
  [FAIL]  Edge constraints
          Edge pac_learning-[does_not_imply]->mistake_bounded:
          constraint requires 'witness' but it is missing or empty
```

C.7 Running the Benchmark Suite

The benchmark consists of 15 tasks (see Appendix B). Execution and evaluation are separate steps.

Step 1: Execute tasks.

```
python3 scripts/run_reference_tasks.py \
    flt_concept_graph.json flt_tasks.json \
    --output results.json
```

This traverses the graph for each task and writes structured outputs to `results.json`.

Step 2: Evaluate against gold answers.

```
python3 scripts/evaluate_reference_tasks.py \
    results.json flt_task_answers.json
```

Scoring modes include `set_match` (exact set equality), `chain_match` (ordered sequence), and `structured_explanation` (required elements present in structured output). The expected output on a correct implementation is 15/15 tasks passed.

Appendix D

Notation Index

This appendix collects all notation used in the text, organized by category. The *Introduced* column gives the chapter of first use. Symbols defined by LaTeX macros in `main.tex` are marked with the macro name in the *Notes* column.

D.1 Sets and Spaces

Symbol	Description	Introduced
X	Domain (input space). Typically $\{0,1\}^n$, $\mathbb{R}^d$, or a countable set.	Ch. 1
Y	Label space (output space). $\{0,1\}$ for binary, finite set for multiclass, $\mathbb{R}$ for regression.	Ch. 1
Σ	Alphabet (finite, nonempty set of symbols).	Ch. 3
Σ^*	Free monoid: all finite strings over Σ.	Ch. 3
$\mathcal{C}$	Concept class: a collection of concepts $c : X \to Y$.	Ch. 1
$\mathcal{H}$	Hypothesis space: the set from which the learner selects hypotheses.	Ch. 1
$\mathcal{F}$	Real-valued function class $\mathcal{F} \subseteq Y^X$ (typically $Y = \mathbb{R}$).	Ch. 10
$S = \{(x_i, y_i)\}_{i=1}^m$	Training sample of m labeled examples.	Ch. 2
D	Distribution on $X \times Y$ (or on X in the realizable setting).	Ch. 2
$\mathbb{N}$	Natural numbers $\{0, 1, 2, \ldots\}$.	Ch. 1
$\mathbb{Z}$	Integers.	Ch. 1
$\mathbb{R}$	Real numbers.	Ch. 1
$\{0,1\}^n$	Binary strings of length n.	Ch. 1
$\mathcal{B}_\varepsilon(x)$	Open ball of radius ε centered at x.	Ch. 10

D.2 Complexity Measures

Symbol	Description	Introduced
$\mathrm{VCdim}(\mathcal{H})$	Vapnik–Chervonenkis dimension: largest set shattered by $\mathcal{H}$. Macro: `\VC`.	Ch. 5
$\mathrm{Ldim}(\mathcal{H})$	Littlestone dimension: depth of deepest mistake tree for $\mathcal{H}$. Macro: `\Ldim`.	Ch. 6
$\mathrm{DSdim}(\mathcal{H})$	DS dimension (Daniely–Shalev-Shwartz): multiclass PAC characterization. Macro: `\DSdim`.	Ch. 17
$d_{\mathrm{N}}(\mathcal{H})$	Natarajan dimension: multiclass generalization of VC dimension via two-colorings. Macro: `\Ndim`.	Ch. 17
$\mathrm{Pdim}(\mathcal{F})$	Pseudodimension: VC dimension of the subgraph class $\{(x,t) : f(x) \geq t\}$. Macro: `\Pdim`.	Ch. 10
$\mathrm{fat}_\gamma(\mathcal{F})$	Fat-shattering dimension at scale $\gamma > 0$. Macro: `\fatshat`.	Ch. 10
$\mathrm{SQdim}(\mathcal{C})$	Statistical query dimension. Macro: `\SQdim`.	Ch. 10
$\widehat{\mathcal{R}}_n(\mathcal{H})$	Empirical Rademacher complexity: $\mathbb{E}_\sigma\big[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \sigma_i h(x_i)\big]$. Macro: `\Rad`.	Ch. 12
$\Pi_{\mathcal{H}}(n)$	Growth function (Sauer–Shelah): $\max_{\lvert S\rvert=n} \lvert\{h\rvert_S : h \in \mathcal{H}\}\rvert$. Macro: `\growth`.	Ch. 10
$\mathrm{KL}(P\Vert Q)$	Kullback–Leibler divergence between distributions P and Q. Macro: `\KL`.	Ch. 12
$\mathcal{N}(\varepsilon, \mathcal{F}, d)$	Covering number: minimum ε-net size under metric d.	Ch. 10
$K(x)$	Kolmogorov complexity of string x.	Ch. 13
$\mathrm{dl}(h)$	Description length of hypothesis h.	Ch. 11

D.3 Learning Parameters

Symbol	Description	Introduced
ε	Accuracy parameter: $R(h) \leq \varepsilon$ or $R(h) \leq R(h^*) + \varepsilon$.	Ch. 5
δ	Confidence parameter: bounds hold with probability $\geq 1 - \delta$.	Ch. 5
m	Sample size (number of training examples).	Ch. 2
$m(\varepsilon, \delta)$	Sample complexity: minimum m sufficient for (ε, δ)-learning.	Ch. 5
T	Number of rounds in online learning or time horizon.	Ch. 6
γ	Margin parameter (for fat-shattering dimension and margin bounds).	Ch. 10
k	Number of labels $\lvert Y\rvert$ in multiclass settings.	Ch. 17
n	Input dimensionality $\lvert x\rvert$ or sample subset size, depending on context.	Ch. 1

D.4 Risk and Loss

Symbol	Description	Introduced
$\ell(h(x), y)$	Loss function (0-1 loss unless otherwise stated).	Ch. 5
$R(h)$	True (population) risk: $\mathbb{E}_{(x,y)\sim D}[\ell(h(x), y)]$. Macro: `\risk`.	Ch. 5
$\hat{R}(h)$ or $\hat{R}_S(h)$	Empirical risk: $\frac{1}{m}\sum_{i=1}^{m} \ell(h(x_i), y_i)$. Macro: `\emprisk`.	Ch. 5
$R(h) - \hat{R}(h)$	Generalization gap.	Ch. 12
R_T	Cumulative regret over T rounds.	Ch. 6
M_T	Mistake count over T rounds.	Ch. 6

D.5 Distributions and Probability

Symbol	Description	Introduced
D	Data-generating distribution on $X \times Y$.	Ch. 2
D_X	Marginal distribution on X.	Ch. 5
$\mathbb{P}(\cdot)$ or $\mathbb{P}(\cdot)$	Probability measure. Macro: `\Pr`.	Ch. 2
$\mathbb{E}[\cdot]$	Expectation. Macro: `\E`.	Ch. 2
$\mathbf{1}_A$ or $\mathbf{1}$[event]	Indicator function / indicator random variable. Macro: `\ind`.	Ch. 2
σ_i	Rademacher random variable: $\mathbb{P}(\sigma_i = +1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}$.	Ch. 12
Q	Posterior distribution (PAC-Bayes) or generic distribution over $\mathcal{H}$.	Ch. 12
P	Prior distribution over $\mathcal{H}$ (PAC-Bayes).	Ch. 12
$S \sim D^m$	Sample drawn i.i.d. from D.	Ch. 2

D.6 Ordinals and Mind-Changes

Symbol	Description	Introduced
ω	First infinite ordinal; cardinality of $\mathbb{N}$.	Ch. 7
ω^k	Ordinal exponentiation (iterated ω).	Ch. 13
ω^ω	First epsilon-unreachable ordinal power (limit of $\omega, \omega^2, \omega^3, \ldots$).	Ch. 13
α, β	Arbitrary ordinals.	Ch. 13
$\mathrm{MC}(L, \mathcal{C})$	Mind-change count: maximum number of hypothesis changes by learner L on class $\mathcal{C}$.	Ch. 7
MC_α	Ordinal mind-change bound at level α.	Ch. 13
ω-VCdim	Ordinal VC dimension (transfinite extension).	Ch. 13
Δ_2^0	Arithmetic hierarchy class: limits of computable functions.	Ch. 7

D.7 Information-Theoretic Quantities

Symbol	Description	Introduced
$H(X)$	Shannon entropy of random variable X.	Ch. 12
$H(X \mid Y)$	Conditional Shannon entropy.	Ch. 12
$I(X;Y)$	Mutual information between X and Y: $H(X) - H(X \mid Y)$.	Ch. 12
$\mathrm{KL}(P\|Q)$	Kullback–Leibler divergence: $\mathbb{E}_P[\log(P/Q)]$. Macro: `\KL`.	Ch. 12
$I(S; A(S))$	Input–output mutual information of algorithm A on sample S (Xu–Raginsky framework).	Ch. 12

D.8 Learning Criteria and Algorithms

Symbol	Description	Introduced
Ex	Explanatory learning (Gold-style identification in the limit). Macro: `\Ex`.	Ch. 7
BC	Behaviorally correct learning. Macro: `\BC`.	Ch. 7
FIN	Finite identification (the learner eventually stops and is correct). Macro: `\FIN`.	Ch. 7
ERM	Empirical risk minimization. Macro: `\ERM`.	Ch. 5
SOA	Standard optimal algorithm (online prediction). Macro: `\SOA`.	Ch. 6
L^*	Angluin's L^* algorithm for learning regular languages from queries.	Ch. 8
CEGIS	Counterexample-guided inductive synthesis.	Ch. 8

D.9 Functions and Maps

Symbol	Description	Introduced
$c : X \to Y$	Target concept (the function the learner aims to identify).	Ch. 1
$h : X \to Y$	Hypothesis produced by the learner.	Ch. 1
$h^* \in \mathcal{H}$	Best hypothesis in $\mathcal{H}$: $h^* = \arg\min_{h \in \mathcal{H}} R(h)$.	Ch. 5
$A : (X \times Y)^m \to \mathcal{H}$	Learning algorithm (maps a sample to a hypothesis).	Ch. 4
φ_e	Partial computable function with Gödel number e.	Ch. 3
W_e	Domain of φ_e (recursively enumerable set with index e).	Ch. 3
$h\|_S$	Restriction of h to the set S: the behavior pattern of h on points in S.	Ch. 5

D.10 Graph-Theoretic Notation

The following notation is specific to the knowledge graph structure described in this book.

Symbol / Macro	Description	Introduced
`\nde{id}` → id	Graph node identifier (sans-serif).	Preface
`\rel{R}` → R	Relation type label (monospaced).	Preface
A $\xrightarrow{\texttt{R}}$ B	Typed directed edge from A to B via R. Macro: `\edge{A}{R}{B}`.	Preface
Layer $\ell \in \{0, \ldots, 7\}$	Stratification of nodes: 0 = formal objects, 1 = base types, 2 = data presentations, 3 = learner types, 4 = success criteria, 5 = complexity measures, 6 = characterizations/impossibilities, 7 = processes/scope boundaries.	Ch. 1
Kernel node	A node with status `defined` or `proved` (133 of 142).	Preface
Deferred node	A node with status `deferred` (6 of 142).	Preface
Scope boundary	A node with status `scope_note` marking an explicit exclusion (3 of 142).	Preface

Complete symbol list. This appendix covers the most frequently used symbols. Additional notation local to a single proof or example is defined at the point of use. All LaTeX macros are defined in the preamble of `main.tex`; see §D above for the graph-specific macros `\nde`, `\rel`, and `\edge`.

Bibliography

[ABDCBH97] Noga Alon, Shai Ben-David, Nicolò Cesa-Bianchi, and David Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *JACM*, 44(4):615–631, 1997.

[ABX08] Benny Applebaum, Boaz Barak, and David Xiao. On basing lower-bounds for learning on worst-case assumptions. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 211–220, 2008.

[AJS99] Andris Ambainis, Sanjay Jain, and Arun Sharma. Ordinal mind change complexity of language identification. *Theoretical Computer Science*, 220:323–343, 1999.

[AL88] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[ALMM19] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite Littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 852–860, 2019.

[Ang87] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

[Ang88] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[Bar74] Jānis Barzdiņš. Inductive inference of automata, functions and programs. In *Proceedings of the International Congress of Mathematicians (ICM 1974)*, pages 455–460, 1974.

[Bax00] Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

[BCD+22] Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2022.

[BDCBHL95] Shai Ben-David, Nicolò Cesa-Bianchi, David Haussler, and Philip M. Long. Characterizations of learnability for classes of {0,...,n}-valued functions. *JCSS*, 50:74–86, 1995.

[BDPSS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *Proceedings of the 22nd Conference on Learning Theory (COLT)*, 2009.

[BE02] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[BEHW87] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, 1987.

[BEHW89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[BFJ+94] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC)*, pages 253–262, 1994.

[BFT17] Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *NeurIPS*, 2017.

[BHM+21] Olivier Bousquet, Steve Hanneke, Shay Moran, Ramon van Handel, and Amir Yehudayoff. A theory of universal learning. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 532–541, 2021.

[BHMM19] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[BM02] Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[BP92] Raymond Board and Leonard Pitt. On the necessity of Occam algorithms. *Theoretical Computer Science*, 100(1):157–184, 1992.

[Cat07] Olivier Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *IMS Lecture Notes – Monograph Series*. Institute of Mathematical Statistics, 2007.

[CS83] John Case and Carl H. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25(2):193–220, 1983.

[DF86] Persi Diaconis and David Freedman. On the consistency of Bayes estimates. *Annals of Statistics*, 14(1):1–26, 1986.

[DR17] Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

[DSS14] Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In *Proceedings of the 27th Conference on Learning Theory (COLT)*, pages 287–316, 2014.

[Dud67] Richard M. Dudley. The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. *Journal of Functional Analysis*, 1:290–330, 1967.

[DV21] Amit Daniely and Gal Vardi. From local pseudorandom generators to hardness of learning. In *COLT*, 2021.

[EHKV89] Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989.

[FJO95] Mark Fulk, Sanjay Jain, and Daniel Osherson. Open problems in inductive inference. *Bulletin of the EATCS*, 55:89–95, 1995.

[FS93] Rūsiņš Freivalds and Carl H. Smith. On the role of procrastination in machine learning. *Information and Computation*, 107(2):237–271, 1993.

[FW95] Sally Floyd and Manfred Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

[Gol65] E. Mark Gold. Limiting recursion. *Journal of Symbolic Logic*, 30(1):28–48, 1965.

[Gol67] E. Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[Han16] Steve Hanneke. The optimal sample complexity of PAC learning. *Journal of Machine Learning Research*, 17(38):1–15, 2016.

[Han24] Steve Hanneke. The star number and eluder dimension. In *Proceedings of the 37th Annual Conference on Learning Theory (COLT)*, 2024.

[Hau88] David Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 36:177–221, 1988.

[Hau92] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[HD20] Fredrik Hellström and Giuseppe Durisi. Generalization bounds via information density and conditional information density. *IEEE Journal on Selected Areas in Information Theory*, 1(3):824–839, 2020.

[Hel18] David Helmbold. On compression schemes for intersection-closed concept classes. In *Proceedings of the 31st Conference on Learning Theory (COLT)*, 2018.

[HLW94] David Haussler, Nick Littlestone, and Manfred K. Warmuth. Predicting $\{0,1\}$-functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.

[HRS16] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1225–1234, 2016.

[HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[JORS99] Sanjay Jain, Daniel Osherson, James S. Royer, and Arun Sharma. *Systems That Learn: An Introduction to Learning Theory*. MIT Press, Cambridge, MA, 2nd edition, 1999.

[Kea98] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[KS01] Susanne Kaufmann and Frank Stephan. Robust learning with infinite additional information. *Theoretical Computer Science*, 259(1–2):427–449, 2001.

[KV94] Michael J. Kearns and Leslie G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

[KW07] Dima Kuzmin and Manfred K. Warmuth. Unlabeled compression schemes for maximum classes. *Journal of Machine Learning Research*, 8:2047–2081, 2007.

[Lit88] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[LV08] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 3rd edition, 2008.

[LW86] Nick Littlestone and Manfred K. Warmuth. Relating data compression and learnability. Technical report, University of California, Santa Cruz, 1986.

[McA99] David McAllester. PAC-Bayesian model averaging. In *Proceedings of the 12th Annual Conference on Computational Learning Theory (COLT)*, pages 164–170, 1999.

[MSS03] Eric Martin, Arun Sharma, and Frank Stephan. On ordinal VC-dimension and some notions of complexity. In *Algorithmic Learning Theory (ALT)*, volume 2842 of *LNCS*, pages 54–68. Springer, 2003. Full version: TCS 364(1):62–76, 2006.

[MST22] Shay Moran, Amir Shpilka, and Iska Tsubari. Compressing concept classes with unlabeled samples. In *Proceedings of the 35th Conference on Learning Theory (COLT)*, 2022. See Moran and Yehudayoff (2016) for the labeled case.

[MY16] Shay Moran and Amir Yehudayoff. Sample compression schemes for VC classes. *Journal of the ACM*, 63(3):1–10, 2016.

[Nat89] Balas K. Natarajan. On learning sets and functions. *Machine Learning*, 4:67–97, 1989.

[Pol84] David Pollard. *Convergence of Stochastic Processes*. Springer, 1984.

[PT20] Dömötör Pálvölgyi and Gábor Tardos. Unlabeled compression schemes exceeding the VC dimension. *Discrete Applied Mathematics*, 276:102–107, 2020.

[Put65] Hilary Putnam. Trial and error predicates and the solution to a problem of Mostowski. *Journal of Symbolic Logic*, 30(1):49–57, 1965.

[PV88] Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35(4):965–984, 1988.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th ACM Symposium on Theory of Computing (STOC)*, pages 84–93, 2005.

[Ris78] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[Ris84] Jorma Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, 30(4):629–636, 1984.

[RK07] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo Method*. Wiley, 2nd edition, 2007.

[RV13] Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems 26 (NeurIPS)*, 2013.

[Sau72] Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13:145–147, 1972.

[Sch65] Lorraine Schwartz. On Bayes procedures. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 4:10–26, 1965.

[She72] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.

[Sho59] Joseph R. Shoenfield. On degrees of unsolvability. *Annals of Mathematics*, 69(3):644–653, 1959.

[Sip13] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 3rd edition, 2013.

[Smi82] Carl H. Smith. The power of pluralism for automatic program synthesis. *Journal of the ACM*, 29(4):1144–1165, 1982.

[Sol64] Ray J. Solomonoff. A formal theory of inductive inference, parts I and II. *Information and Control*, 7:1–22, 224–254, 1964.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[SSSSS10] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670, 2010.

[SSV04] Arun Sharma, Frank Stephan, and Yuri Ventsov. Generalized notions of mind change complexity. *Information and Computation*, 189:235–262, 2004.

[SZ15] Hans Ulrich Simon and Sandra Zilles. Open problem: Recursive teaching dimension versus vc dimension. In *Proceedings of the 28th Conference on Learning Theory (COLT)*, 2015.

[Val84] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vap98] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[VC71] 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.

[Vel89] M. Velauthapillai. Inductive inference with a bounded number of mind changes. In *Proceedings of the 2nd Annual Workshop on Computational Learning Theory (COLT)*, pages 200–213, 1989.

[WB68] Chris S. Wallace and David M. Boulton. An information measure for classification. *The Computer Journal*, 11(2):185–194, 1968.

[WM97] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[Wol96] David H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8:1341–1390, 1996.

[XR17] Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.