# Formal Learning Theory Kernel: Blueprint v1

A Lean 4 formalization of the fundamental theorem of statistical learning

with paradigm separations and a measurability refinement

Dhruv Gupta
Indian Institute of Science

April 2026

# Contents

# Chapter 1

# Introduction

This is the mathematical blueprint for a Lean 4 formalization of the fundamental theorem of statistical learning. It sits between two other artifacts: the kernel source code at https://github.com/Zetetic-Dhruv/formal-learning-theory-kernel, which contains 354 machine-checked theorems in 21,728 lines with zero `sorry`, and a companion textbook on formal learning theory from which most of the exposition in Chapters 3 through 6 is imported directly. The blueprint's role is to annotate informal mathematical statements with hyperlinks into the formal declarations that prove them, so that a reader can click from a theorem in prose to the corresponding `theorem` or `def` in the Lean kernel and back.

Every statement carrying a `\lean{}` annotation in this document resolves to a proved declaration in the kernel. The formalization targets Lean 4 `v4.29.0-rc6` against mathlib4 pinned at commit `fde0cc5`. The per-module API reference is at https://zetetic-dhruv.github.io/formal-learning-theory-kernel/.

## Scope

Blueprint v1 covers the core story of the kernel:

- **PAC characterization** (Chapter 3): the five-way equivalence for PAC learning, with the `NullMeasurableSet` refinement of the standard Borel hypothesis used in the literature.
- **Online characterization** (Chapter 4): the Littlestone characterization theorem with the Standard Optimal Algorithm as the constructive witness.
- **Gold-style learning** (Chapter 5): Gold's theorem with the diagonalization proof and the mind-change characterization with ordinal bounds.
- **Three-paradigm separation** (Chapter 6): the 13-edge separation lattice between PAC, online, and Gold-style learning.
- **Measurability layer** (Chapter 7): the Borel-parameterized setting, the analytic measurability bridge via Choquet capacitability, and the Borel-analytic separation theorem. This chapter is net-new: it does not exist in the companion textbook because the Borel-analytic separation was discovered during the formalization effort.

Deferred to v2: compression via approximate minimax, the measurable batch learner monad, PAC-Bayes bounds, extended criteria (robust PAC, RKHS), and the Baxter multi-task base case. Compression and the MBL monad rely on infrastructure whose mathematical exposition is not yet stable enough to import from the textbook, and their blueprint treatment will ship once that stabilizes.

## Relationship to the companion textbook

Chapters 2 through 6 of this blueprint are imported verbatim from the companion textbook with `\lean{}` annotations added at each theorem or definition whose informal statement corresponds to a formal declaration. The prose is otherwise unchanged. Figures are reused from the textbook.

Chapter 7 is the one chapter whose content does not appear in the textbook, because the Borel-analytic separation and the `WellBehavedVCMeasTarget` refinement of the fundamental theorem were discovered during the formalization work and are outside the textbook's scope. The prose in that chapter is therefore written from scratch in the voice of the kernel rather than imported.

## How to read this document

A reader interested in the mathematical content can read the chapters linearly and ignore the hyperlink annotations. A reader interested in the formalization can click any `\lean{}`-annotated statement to open the corresponding Lean declaration page in the doc-gen4 API reference. A reader interested in the dependency structure of the theorems can consult the automatically generated dependency graph at `dep_graph_document.html`, which renders the `\uses{}` edges declared throughout the document.

# Chapter 2

# Primitives

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

## 2.1 Atomic vocabulary

**Definition 2.1** (Domain, Label, Concept)**.** Lean: `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 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 3).

**Definition 2.2** (Hypothesis, Target Concept, Proper vs. Improper)**.** 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 $\mathcal{H} \supseteq \mathcal{C}$.

## 2.2 The concept class

**Definition 2.3** (Concept Class)**.** Lean: `ConceptClass`

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.

The concept class is the node every complexity measure in the book connects to: VC dimension, Littlestone dimension, Rademacher complexity, covering number, growth function, mind-change dimension. It is the object whose structure determines what is learnable.

## 2.3 Batch learner

**Definition 2.4** (Batch Learner)**.** Lean: `BatchLearner`

A *batch learner* over $(X, Y)$ is a bundled structure carrying a hypothesis set $\mathcal{H} \subseteq Y^X$, a learning function $\mathcal{A}$ that takes a finite training sample $S : \operatorname{Fin} m \to X \times Y$ and returns a hypothesis in $\mathcal{H}$, and a proof that the output lies in $\mathcal{H}$.

The kernel formalizes the batch learner as a structure rather than as a bare function so that the constraint "output lies in the hypothesis set" travels with the algorithm. Chapter 4 will introduce a sequential learner with its own bundled type; the contrast between batch and sequential learning is the subject of Chapters 3 and 4.

## 2.4 Sample, loss, and error

**Definition 2.5** (I.i.d. sample)**.** For a distribution $D$ on $X$ and a target concept $c$, an *i.i.d. sample* of size $m$ is a tuple $S = ((x_1, c(x_1)), \dots, (x_m, c(x_m)))$ with each $x_i \sim D$ drawn independently.

**Definition 2.6** (Zero-one loss)**.** For $h, c : X \to \{0, 1\}$ and $x \in X$, the *zero-one loss* is $\ell(h, c, x) = \mathbf{1}[h(x) \neq c(x)]$.

**Definition 2.7** (Empirical and true error)**.** For a hypothesis $h$, a target $c$, a distribution $D$, and a sample $S$ of size $m$:

$$\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}[h(x_i) \neq c(x_i)] \qquad \text{(empirical error)},$$

$$R_D(h) = \mathbb{P}_{x \sim D}[h(x) \neq c(x)] \qquad \text{(true error)}.$$

# Chapter 3

# 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 3.1): the definition and its moving parts. Brief, the definition is not the centerpiece.
2. **The VC characterization proof** (Section 3.2): 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 3.3): all nine equivalent conditions, the equivalence web, and the directions we proved versus those we cited.
4. **The agnostic setting** (Section 3.4): 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 3.5): the matching lower bound $\Omega(d/\varepsilon)$ and the full NFL proof.
6. **Computational interlude** (Section 3.6): the information–computation gap.

## 3.1 The PAC Framework

Two assumptions frame the discussion.

**Definition 3.1** (Realizable setting)**.** `Lean: BatchLearner`

Learning is *realizable* if the target concept $c^*$ lies in the hypothesis class: $c^* \in \mathcal{H}$.

**Definition 3.2** (Agnostic setting)**.** `Lean: BatchLearner`

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 3.3** (PAC Learning [Val84])**.** Lean: `PACLearnable`

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 3.4** (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 $m = \frac{4}{\varepsilon} \ln \frac{4}{\delta} = O\big(\frac{d + \log(1/\delta)}{\varepsilon}\big)$ 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 $\delta$, but $\delta$ can be driven arbitrarily small by drawing more samples.

*Remark* 3.5 (The role of $\delta$). The $\delta$ 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 $\delta$ enters.

**Definition 3.6** (Empirical Risk Minimization)**.** Lean: `BatchLearner`

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?*

## 3.2 The VC Characterization

The main result of this section is:

**Theorem 3.7** (VC Characterization of PAC Learnability [BEHW89, VC71])**.** *Lean: `fundamental_theorem`*

*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*

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

*Remark* 3.8 (Refinement formalized in this kernel). The statement of Theorem 3.7 as written here is silent on the measurability of the hypothesis class. The symmetrization argument in Stage 2 below picks up a two-sided ghost-gap bad event on $X^m \times X^m$ whose measurability is a separate requirement on $\mathcal{H}$. The literature proof of Krapp and Wirth [KW24] states this requirement as a *Borel* hypothesis on the bad event. This kernel formalizes the fundamental theorem under a strictly weaker hypothesis: the bad event need only be *null-measurable* with respect to the completion of the product measure. The weakening is not cosmetic. Chapter 7 constructs a concept class whose ghost-gap bad event is analytic but not Borel, and thereby witnesses the strict gap between the two hypotheses. The Lean identifier

Lean: `WellBehavedVCMeasTarget`

names the refined hypothesis used in the kernel statement, and

Lean: `KrappWirthWellBehaved`

names the Krapp and Wirth version; the implication

Lean: `KrappWirthWellBehaved.toWellBehavedVC`

sits on one side of the strict gap, and Corollary 7.18 witnesses the other.

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 ($\varepsilon$-net/symmetrization argument).
**Stage 3:** Uniform convergence $\Longrightarrow$ ERM is a PAC learner.
**Stage 4 (Converse):** Infinite VC dimension $\Longrightarrow$ not PAC learnable.

### 3.2.1 Stage 1: Sauer–Shelah (Statement)

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

**Definition 3.9** (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 3.10** (Sauer–Shelah [Sau72, She72])**.** *Lean: sauer$_s$helah*

*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 the companion textbook's chapter on Combinatorial Dimensions.[1] 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.

### 3.2.2 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 3.11** (Uniform Convergence)**.** Lean: `HasUniformConvergence`

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 3.12** (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* $\varepsilon^2$ *denominator improves to* $\varepsilon$*, 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.

[1] See https://github.com/Zetetic-Dhruv/formal-learning-theory-book, Chapter 10.

**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 $T$:

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

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

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

Combining with the Sauer–Shelah bound:

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

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

*Remark* 3.13 (The Hanneke refinement). The optimal sample complexity in the realizable case is $\Theta\left(\frac{d+\log(1/\delta)}{\varepsilon}\right)$, 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.

### 3.2.3 Stage 3: Uniform Convergence Implies ERM Succeeds

**Proposition 3.14** (ERM is a PAC learner under uniform convergence)**.** *If $\mathcal{H}$ has the uniform convergence property, then* $\text{ERM}_{\mathcal{H}}$ *is a PAC learner for $\mathcal{H}$.*

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

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

Let $h_S = \text{ERM}_{\mathcal{H}}(S)$. 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:

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

### 3.2.4 Stage 4: Infinite VC Dimension Implies Failure

**Theorem 3.15** (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 $C = \{x_1, \ldots, x_{2m}\} \subseteq X$ 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 $\geq m$ 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* 3.16 (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.

## 3.3 The Fundamental Theorem of Statistical Learning

The VC characterization (Theorem 3.7) 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 3.17** (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 6.)

*Remark* 3.18 (On the numbering). Condition (F9) is an implication, not an equivalence: finite Littlestone dimension implies finite VC dimension (see the textbook's Chapter 10, Combinatorial Dimensions), but the converse fails, thresholds on $\mathbb{R}$ have VCdim $= 1$ but Ldim $= \infty$ (**??**). We include it to mark the boundary of the equivalence web: this is where the PAC–online separation begins.

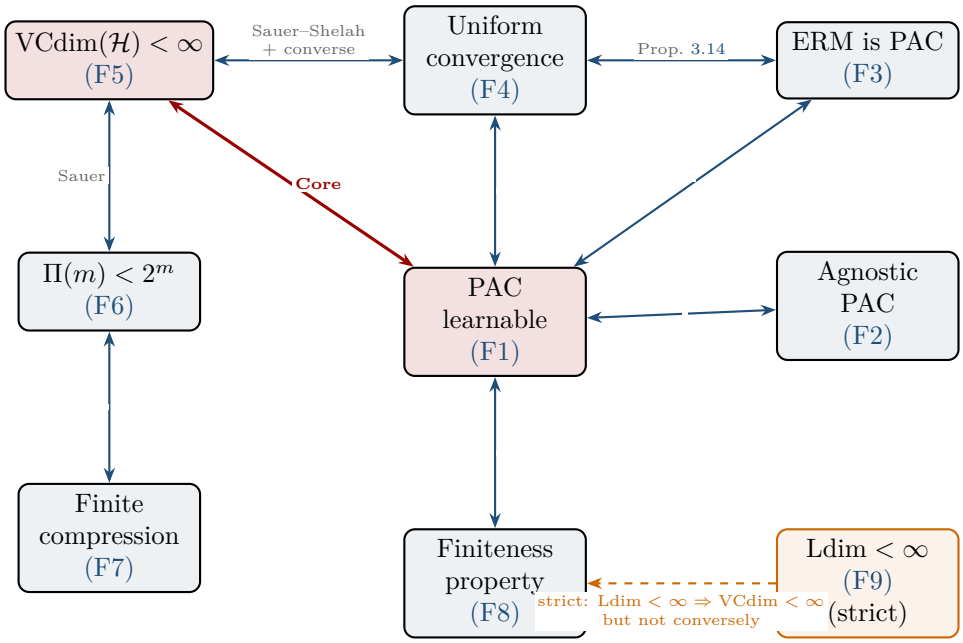


Figure 3.1: The equivalence web of the Fundamental Theorem. All solid bidirectional arrows are full equivalences. The dashed arrow from Ldim $< \infty$ is strict: finite Littlestone dimension implies PAC learnability, but not conversely. The thick red diagonal marks the VC characterization, the core equivalence proved in Section 3.2.

### 3.3.1 The Proof Architecture

We have already proved (in Section 3.2):

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

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

The remaining directions:

- (F5) $\Leftrightarrow$ (F6): By the Sauer–Shelah lemma (Lemma 3.10), VCdim$(\mathcal{H}) = d$ implies $\Pi_{\mathcal{H}}(m) \leq (em/d)^d < 2^m$ for $m$ large enough. Conversely, if 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) ⇔ (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 3.14 shows ERM succeeds in the agnostic setting as well (with the $\varepsilon^2$ sample complexity discussed in Section 3.4).

- (F5) ⇔ (F7): This is the deepest equivalence. We sketch the argument below.

- (F9) ⇒ (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 (textbook Chapter 10, Combinatorial Dimensions).

### 3.3.2 The Compression Direction (Sketch)

**Definition 3.19** (Compression Scheme)**.** A *compression scheme* of size $k$ for $\mathcal{H}$ is a pair $(\kappa, \rho)$ where:

- $\kappa$ (the compressor) maps any sample $S$ consistent with some $h \in \mathcal{H}$ to a subsequence $\kappa(S) \subseteq S$ of size $\leq k$;
- $\rho$ (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) ⇒ (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) ⇒ (F7) is the deep one. A weaker exponential bound was proved by Moran and Yehudayoff [MY16]; the conjectured linear bound remains open.

**Historical Note**

**The sample compression conjecture.** Littlestone and Warmuth conjectured in 1986 that every class of VC dimension $d$ has a compression scheme of size at most $d$ (the linear bound). Moran and Yehudayoff (2016) proved the weaker result that every class of VC dimension $d$ has a compression scheme of size at most $2^{O(d)}$ (an exponential bound). The original conjecture of a linear bound $O(d)$ remains one of the field's central open problems. This kernel formalizes the Moran-Yehudayoff exponential bound via an approximate minimax (MWU) construction, not the conjectured linear bound.

*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 $2^{O(d)}$. □

## 3.4 The Agnostic Setting and the $\varepsilon^2$ Price

**Definition 3.20** (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 $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$:

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

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

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

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

*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 $\varepsilon$-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 $\varepsilon$: $R_D(h_1) = \eta$ and $R_D(h_2) = \eta + \varepsilon$. Distinguishing these requires detecting a difference $\varepsilon$ against a background noise level $\eta$.

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* 3.22 (The revelation). The $\varepsilon$ vs. $\varepsilon^2$ 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 $\varepsilon$ (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.

## 3.5 Lower Bounds and the No-Free-Lunch Theorem

### 3.5.1 The PAC Lower Bound

The upper bound of Theorem 3.7 gives $m = O(d/\varepsilon)$ in the realizable case. The following shows this is tight up to constants.

**Theorem 3.23** (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 $m = \Omega(d/\varepsilon)$. □

### 3.5.2 The No-Free-Lunch Theorem (Full Proof)

The companion textbook's Chapter 1 (The Objects of Learning)[2] gave the statement and a brief argument. Here we give the full proof with the averaging argument over all target functions.

**Theorem 3.24** (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* 3.25 (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."

[2] https://github.com/Zetetic-Dhruv/formal-learning-theory-book, Chapter 1.

## 3.6 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 3.26** (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 the textbook's Chapter 16 (Computational Hardness).[a] 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 2.2) is computationally sharp even when it is information-theoretically irrelevant.

[a] https://github.com/Zetetic-Dhruv/formal-learning-theory-book, Chapter 16.

## 3.7 What This Chapter Established

The central achievement is the Fundamental Theorem (Theorem 3.17), 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 3.2 |
| Uniform convergence | Vapnik–Chervonenkis (1971) | Theorem 3.12 |
| ERM is a PAC learner | Blumer et al. (1989) | Proposition 3.14 |
| PAC learnable | Valiant (1984) | Theorem 3.7 |
| Agnostic PAC learnable | Haussler (1992) | Section 3.4 |
| $\Pi(m) < 2^m$ for some $m$ | Sauer–Shelah (1972) | Section 3.3.1 |
| Finite compression scheme | Moran–Yehudayoff (2016) | Section 3.3.2 |

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 $\varepsilon^2$ 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 the textbook's Chapter 16 (Computational Hardness).

**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 $\varepsilon$-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 $\varepsilon$ 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 3.21.

# Chapter 4

# 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.

## 4.1 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 4.1** (Online Learning Protocol)**.** `Lean: OnlineLearnable`

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 3:

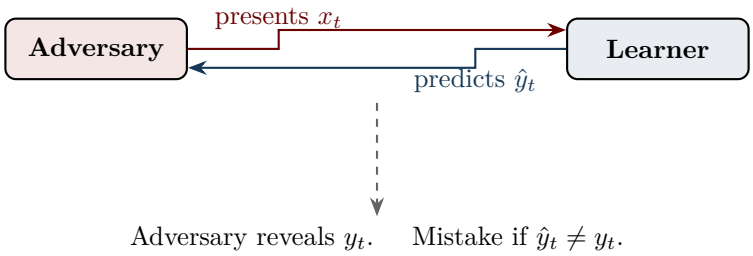


Figure 4.1: The online learning game. The adversary and learner interact sequentially; the adversary is constrained only by realizability. There is no distribution, no random sampling, and no distinction between training and test phases.

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 4.2** (Mistake Bound)**.** `Lean: OnlineLearnable`

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.

## 4.2 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 4.3** (Mistake Tree)**.** `Lean: LTree.isShattered`

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 4.4** (Littlestone Dimension [Lit88])**.** `Lean: LittlestoneDim`

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

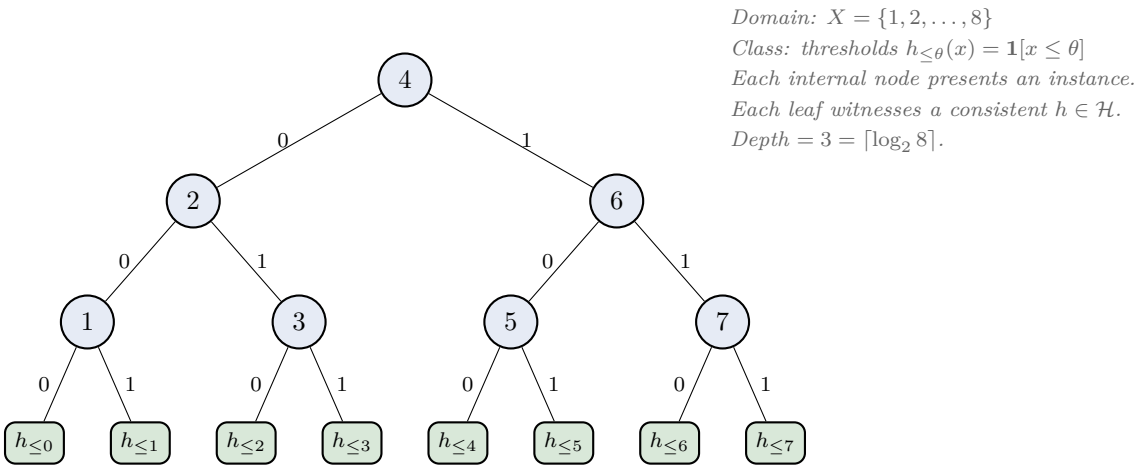


Figure 4.2: A complete mistake tree of depth 3 for the threshold class on $\{1, \ldots, 8\}$. Each internal node is labeled with an instance; the left and right children correspond to labels 0 and 1. Every root-to-leaf path is consistent with the threshold hypothesis shown at the leaf. The adversary can traverse this tree from root to any leaf, forcing 3 mistakes.

**Example 4.5** (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 4.2 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* 4.6 (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 VCdim $= 1$ but Ldim $= \infty$.

## 4.3 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 4.7** (Standard Optimal Algorithm (SOA)). Lean: `SOA_predict_eq`

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 4.8** (Upper Bound: SOA achieves $\mathrm{Ldim}(\mathcal{H})$ mistakes [Lit88]). *Lean: `backward_direction`*

*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:

$$\text{If the SOA makes a mistake at round } t, \text{ then } d_{t+1} \leq d_t - 1. \tag{4.1}$$

*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* 4.9 (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* 4.10 (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.

## 4.4 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 4.11** (Lower Bound: Any learner makes at least $\mathrm{Ldim}(\mathcal{H})$ mistakes [Lit88])**.** *Lean: `forward_direction`*

*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 **??** 4.8**??** 4.11, we obtain the fundamental characterization.

**Theorem 4.12** (Littlestone's Characterization [Lit88])**.** *Lean: `littlestone_characterization`*

*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.

## 4.5 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 4.13** (Regret)**.** Given a sequence $(x_1, y_1), \dots, (x_T, y_T)$ and the learner's predictions $\hat{y}_1, \dots, \hat{y}_T$, 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* 4.14 (Mistake bound vs. regret bound). In the realizable case, the best hypothesis $h^*$ makes zero mistakes, so $\mathrm{Regret}_T$ 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 4.15** (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* 4.16 (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.

## 4.6 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 4.17** (VCdim ≤ Ldim)**.** *Lean:* `BranchWiseLittlestoneDim_geVCDim`

*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 $\mathrm{Ldim}(\mathcal{H}) \geq d$. □

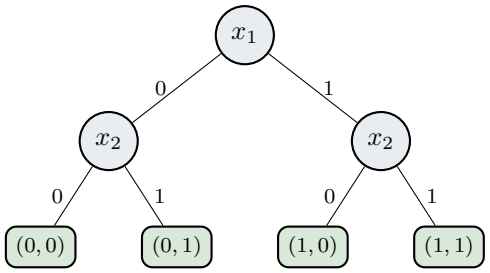


Figure 4.3: A shattered set of size 2 yields a (non-adaptive) mistake tree of depth 2. The instances at each depth are identical, so the tree does not exploit adaptivity. This construction proves $\mathrm{VCdim}(\mathcal{H}) \leq \mathrm{Ldim}(\mathcal{H})$.

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

**Proposition 4.18** (The PAC–Online separation: thresholds on $\mathbb{R}$)**.** *Lean:* `pac_not_implies_online`

*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 < ∞ implies VCdim < ∞ (by VCdim ≤ Ldim), so online learnability implies PAC learnability.

**Separation Result**

**PAC ⇏ 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 $\varepsilon$. The gap between PAC and online learnability is not a technicality: it reflects a fundamental difference between random and adversarial data.

*Remark* 4.19 (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 6.

## 4.7 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 5

# 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 $\varepsilon$, no $\delta$. 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: **FIN** $<$ **Ex** $<$ **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.

## 5.1 Gold's Question

Fix a countable domain. In Gold's original setting, this is the set of all strings over a finite alphabet $\Sigma$, 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 5.1** (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.

## 5.2 Ex-Learning and Gold's Impossibility Theorem

**Definition 5.2** (Explanatory Learning (**Ex**))**.** Lean: `EXLearnable`

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 5.3.2.)

**Example 5.3** (**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 5.4** (Gold's Impossibility Theorem [Gol67])**.** *Lean: `gold_theorem`*

*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 $\varepsilon$-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 $L_\infty$), 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 $L_\infty$ is eventually presented, so $t$ is a text for $L_\infty$. 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 $L_\infty$ 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 $L_\infty$.

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 $L_\infty$. This forces $M$ to fail for $L_\infty$ 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 $L_\infty$. 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* 5.5 (Diagonalization vs. concentration). Compare this proof to the lower bound for PAC learning (Chapter 3). 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 5.4 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 $\Sigma^0_2$-definable families, establishing a precise link to the arithmetical hierarchy.

## 5.3 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.

### 5.3.1 Finite Identification

**Definition 5.6** (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 5.3) but not **FIN**-identifiable, since the learner cannot know when the last element has been seen.

### 5.3.2 Behaviorally Correct Learning

**Definition 5.7** (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 5.8** (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. □

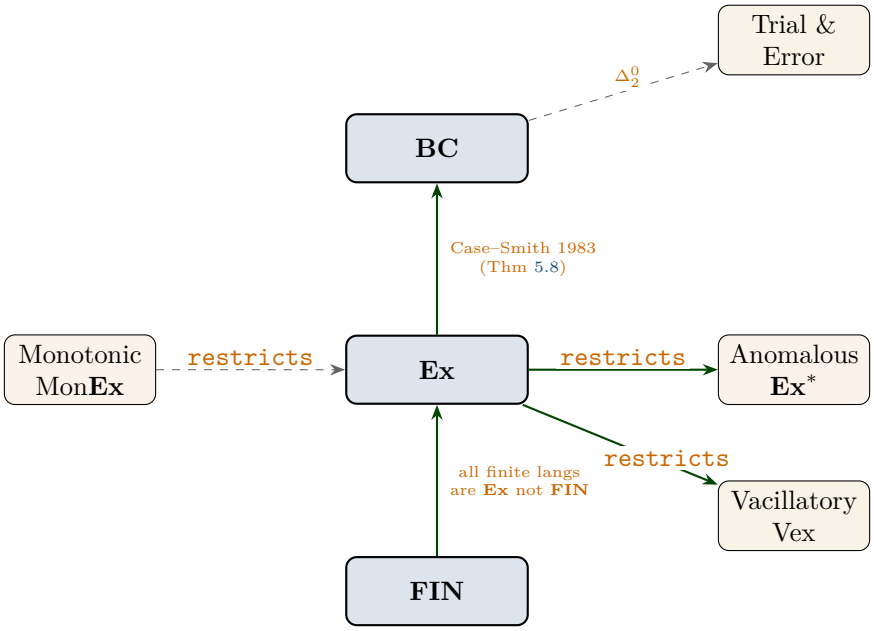


Figure 5.1: The identification hierarchy. Solid arrows indicate strict inclusion of identifiable classes (more hypotheses identified at the target). **FIN** $\subsetneq$ **Ex** $\subsetneq$ **BC**, with witnesses on each edge. Anomalous learning (**Ex**$^*$) extends **Ex** by removing the zero-error 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.

## 5.4 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.

### 5.4.1 Anomalous Learning

**Definition 5.9** (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].

### 5.4.2 Monotonic Learning

**Definition 5.10** (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.

### 5.4.3 Vacillatory Learning

**Definition 5.11** (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**.

### 5.4.4 Trial and Error

**Definition 5.12** (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 $\Sigma^0_2$ and $\Pi^0_2$. This is the precise recursion-theoretic characterization, due to Putnam [Gol65] and Gold: the limit-computable functions are exactly the $\Delta^0_2$ functions. This connection places Gold-style learning firmly within the landscape of classical recursion theory.

## 5.5 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 5.13** (Mind Change)**.** Lean: `EXLearnable`

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 5.3 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 5.14** (Freivalds–Smith [FS93]). *Lean:* `mind_change_characterization`

*The mind-change complexity of* **Ex***-identification is naturally measured by* countable ordinals. *Specifically:*

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

*(ii) There exist classes identifiable with mind-change bound $\omega$ (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 $\alpha$ but not with any bound $\beta < \alpha$.*

The idea behind ordinal mind-change bounds is as follows. A learner with mind-change bound $\omega$ begins with an ordinal counter set to $\omega$. 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 $\omega$ 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 $\omega$, 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; $\omega^2$ allows nested levels of resetting; and so on up through the constructive ordinals.

*Remark* 5.15 (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 the companion textbook's Chapter 13 (Mind-Change Ordinals),[1] 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.

## 5.6 Three Paradigms, Incomparable

We now have three learning paradigms: PAC (Chapter 3), online (Chapter 4), 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.

[1] https://github.com/Zetetic-Dhruv/formal-learning-theory-book, Chapter 13.

**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 5.3). 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.

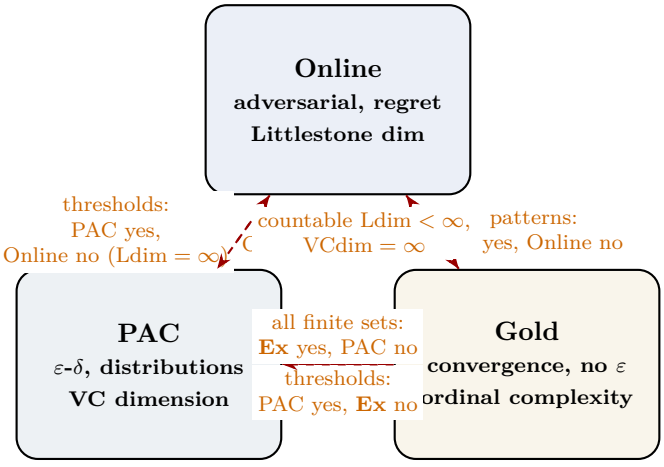


Figure 5.2: The three-paradigm separation. Each pair of paradigms is separated in both directions by explicit witnesses. No paradigm subsumes another. The dashed arrows represent `does_not_imply` edges in the graph, and each arrow is labelled with its witness class.

*Remark* 5.16 (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.

## 5.7 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 5.4): 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 5.3): the chain $\mathbf{FIN} \subsetneq \mathbf{Ex} \subsetneq \mathbf{BC}$ is strict, with the Case–Smith separation (Theorem 5.8) 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 5.4): anomalous, monotonic, and vacillatory learning branch off from **Ex**, each modifying a different aspect of the convergence requirement. Trial-and-error learning connects to $\Delta_2^0$ in the arithmetical hierarchy.
4. **Mind-change complexity is ordinal-valued** (Section 5.5): the Freivalds–Smith characterization shows that the right complexity measure for **Ex**-identification uses transfinite ordinals, not integers. The full treatment is in the textbook's Chapter 13 (Mind-Change Ordinals).
5. **The three paradigms are pairwise incomparable** (Section 5.6): 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 3 and the combinatorial character of Chapter 4. Each paradigm brings its own mathematical world. The separations of Section 5.6 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 "$\varphi_e$ 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 $L_\infty$ but the learner's inability to distinguish "finite set, done growing" from "co-finite set, still growing."

# Chapter 6

# 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.

## 6.1 The Separation Lattice

Figure 6.1 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.

## 6.2 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 $h_\theta(x) = \mathbf{1}[x \geq \theta]$.

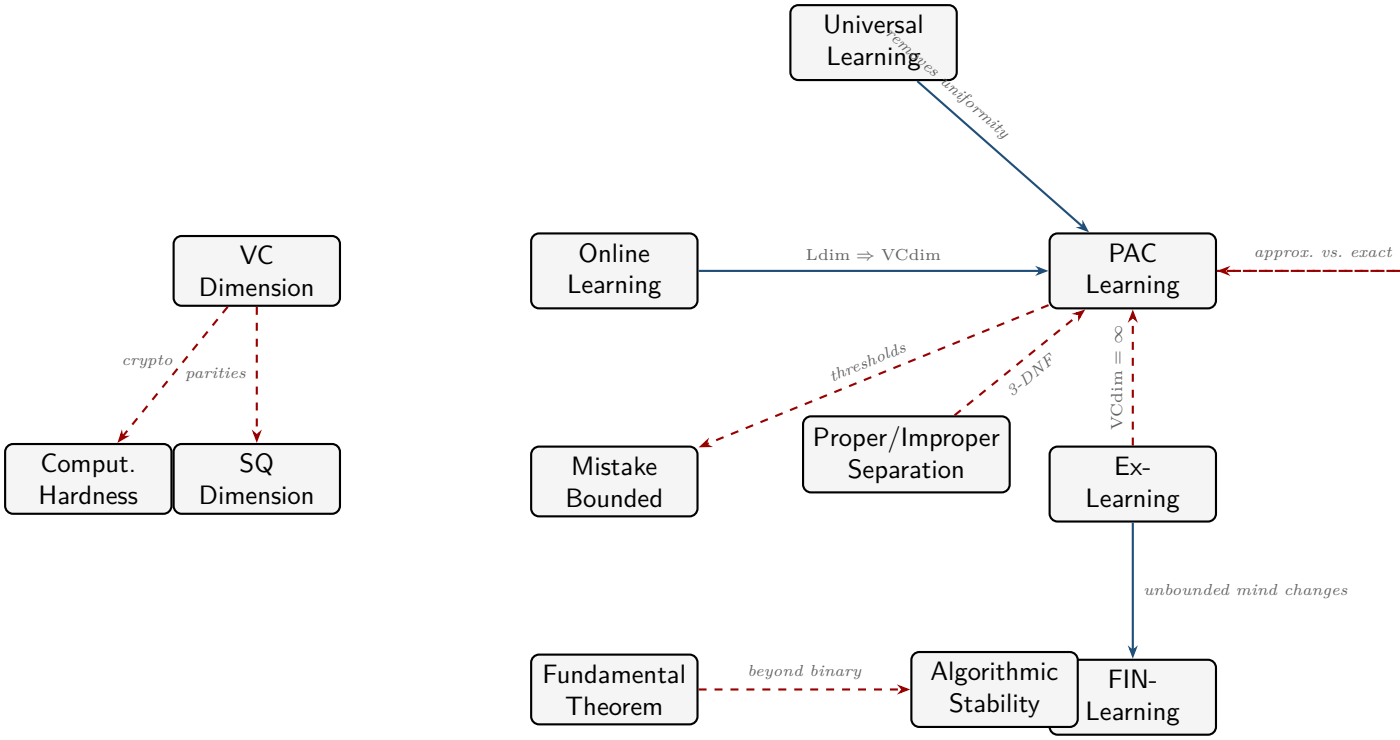


Figure 6.1: The separation lattice. Dashed red: `does_not_imply` (9 edges). Solid blue: `strictly_stronger` (4 edges). Each label names the witness.

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]

*Remark* 6.1. Lean: $\mathsf{pac}_{n}ot_{i}mplies_{o}nline$

Formalized in the kernel; see the source link.

**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]

*Remark* 6.2. Lean: $\mathsf{ex}_n ot_i mplies_p ac$

Formalized in the kernel; see the source link.

**Separation Result**

**PAC learning $\not\Rightarrow$ exact learning.**

*Witness.* PAC learning requires only $\varepsilon$-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 $\varepsilon$ 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 $\chi_S$, for $S \subseteq [n]$, maps $x \mapsto \bigoplus_{i \in S} x_i$. The parity class has 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 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 3) 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]

## 6.3 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 unbounded 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 6.2: 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]

## 6.4 What the Negative Layer Reveals

The separation lattice of Figure 6.1 carries a meta-theorem about the structure of formal learning theory, which we can now state explicitly.

**Theorem 6.3** (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 7

# The Measurability Layer

Up to Chapter 3 the fundamental theorem of statistical learning has been stated as if measurability were a side condition. It is not. The symmetrization argument in the forward direction picks up a set on $X^m \times X^m$ whose regularity determines whether Hoeffding's inequality can be applied pointwise and lifted to a supremum over the growth-function-many effective labelings. The literature, following Krapp and Wirth [KW24], states this regularity as a Borel hypothesis on the ghost-gap bad event. Every bad event in practice is then checked against the Borel $\sigma$-algebra on the product space, and everything goes through.

This chapter establishes that the Borel hypothesis is stronger than the proof needs. The actual sufficient condition is weaker: the bad event must be *null-measurable* with respect to the completion of the product measure. Section 7.1 identifies the point in the symmetrization chain where the weakening applies and names the kernel's refined hypothesis `WellBehavedVCMeasTarget`. Section 7.2 shows that every Borel-parameterized concept class automatically satisfies the refinement via a classical theorem of descriptive set theory (Choquet capacitability, Kechris GTM 156 [Kec95, Theorem 30.13]). Finally, Section 7.3 exhibits a concrete concept class whose ghost-gap bad event is analytic and null-measurable but *not* Borel, and which therefore witnesses that the refinement is strict.

None of this material appears in the companion textbook: the discoveries are tied to the formalization work, and their home is this chapter. The formalized chain is

$$\begin{aligned}
&\texttt{Lean: singletonClassOn} \\
&\longrightarrow \\
&\texttt{Lean: singleton}_b adEvent_e q_p reimage_p lanar \\
&\longrightarrow \\
&\texttt{Lean: planarWitnessEvent}_a nalytic \\
&\longrightarrow \\
&\texttt{Lean: planarWitnessEvent}_n ot_m easurable \\
&\longrightarrow \\
&\texttt{Lean: singleton}_b adEvent_n ot_m easurable
\end{aligned}$$

,

five theorems in `FLT_Proofs.Theorem.BorelAnalyticSeparation`, proved sorry-free against mathlib4 `fde0cc5`.

## 7.1 Borel parameterization and the symmetrization route

The symmetrization argument at the heart of the uniform convergence proof (Stage 2 of Section 3.2) rewrites the one-sided bad event

$$\mathrm{Bad}_1(C, m, \varepsilon) \;=\; \Big\{ S \in X^m \;:\; \sup_{h \in C} \big|\, R_D(h) - \hat{R}_S(h) \,\big| > \varepsilon \Big\}$$

in terms of a ghost sample $S' \sim D^m$ and the two-sided comparison event

$$\mathrm{Bad}_2(C, m, \varepsilon) \;=\; \Big\{ (S, S') \in X^m \times X^m \;:\; \sup_{h \in C} \big|\, \hat{R}_{S'}(h) - \hat{R}_S(h) \,\big| > \varepsilon \Big\}.$$

The bound $\mathbb{P}_S[\mathrm{Bad}_1] \le 2\,\mathbb{P}_{S,S'}[\mathrm{Bad}_2]$ transfers control of the deviation between empirical and true error over to the two-sample event. The union-bound-over-labelings step that follows requires the inner event at each fixed labeling to carry a well-defined probability: for each split of the combined $2m$ points into $S$ and $S'$, Hoeffding's inequality is applied to the $2m$ centered Bernoulli contributions, and the resulting tail bound is summed over the $\Pi_C(2m)$ distinct effective labelings.

What does this argument actually require of $\mathrm{Bad}_2$? Three things:

1. **A probability for** $\mathrm{Bad}_2$**.** The outer symmetrization bound compares $\mathbb{P}_{S,S'}[\mathrm{Bad}_2]$ against a sum of inner tail bounds. For this comparison to be meaningful, $\mathrm{Bad}_2$ must be assigned a probability with respect to the product measure $D^{\otimes 2m}$.
2. **Measurability of each slice at a fixed labeling.** The inner Hoeffding bound is applied to the event that a specific (fixed) labeling produces a large gap between the $S$-average and the $S'$-average. This inner event lives in the product Borel structure and is Borel automatically.
3. **Closure of the probability under countable operations.** The supremum over $h \in C$ is rewritten as a finite union over the growth-function-many effective labelings on the combined sample, so the only closure needed is under finite union inside the product $\sigma$-algebra extended with the $D^{\otimes 2m}$-null sets.

Requirement (1) is exactly null-measurability with respect to the completion of $D^{\otimes 2m}$. Requirements (2) and (3) are automatic for any $\mathrm{Bad}_2$ built from a class with a Borel parameter space. Nowhere does the proof need $\mathrm{Bad}_2$ itself to be Borel with respect to the product Borel structure.

*Remark* 7.1 (The Krapp and Wirth hypothesis, diagnosed). The natural place to plant a measurability hypothesis in the proof is after (1): require $\mathrm{Bad}_2$ to sit in whatever ambient $\sigma$-algebra the argument is using. Krapp and Wirth [KW24] plant it inside the product Borel $\sigma$-algebra. That is a strictly stronger requirement than (1) alone. The gap between "Borel on the product space" and "null-measurable for every product measure built from $D$" is invisible in every Borel-parameterized example, where both happen to hold, but it is a real gap in general.

The kernel names the refined hypothesis.

**Definition 7.2** (WellBehavedVCMeasTarget)**.** Lean: `WellBehavedVCMeasTarget`

A concept class $C$ over a measurable space $(X, \Sigma)$ satisfies *WellBehavedVCMeasTarget* if, for every sample size $m$, every gap threshold $\varepsilon$, and every probability distribution $D$ on $X$, the two-sided ghost-gap bad event $\mathrm{Bad}_2(C, m, \varepsilon)$ is a null-measurable subset of $(X^m \times X^m,\, \Sigma^{\otimes 2m})$ with respect to the product measure $D^{\otimes 2m}$. Here null-measurability means that the event differs from a set in $\Sigma^{\otimes 2m}$ by a $D^{\otimes 2m}$-null set, equivalently, it sits in the completion of $\Sigma^{\otimes 2m}$ under $D^{\otimes 2m}$.

**Definition 7.3** (KrappWirthWellBehaved)**.** Lean: `KrappWirthWellBehaved`

A concept class $C$ satisfies the *KrappWirthWellBehaved* hypothesis of [KW24] if, for every $m$ and $\varepsilon$, the two-sided ghost-gap bad event $\mathrm{Bad}_2(C, m, \varepsilon)$ is a Borel subset of $X^m \times X^m$ in the product Borel structure.

**Proposition 7.4** (KrappWirth is stronger than WellBehavedVCMeasTarget)**.** *Lean: `KrappWirthWellBehaved.toWell`*

*Every Borel set is null-measurable with respect to any completion of a Borel probability measure. Consequently any concept class carrying `KrappWirthWellBehaved` also carries `WellBehavedVCMeasTarget`.*

*Proof.* $\mathcal{B} \subseteq \mathcal{B}^*_\mu$ for every Borel probability measure $\mu$, because the completion only enlarges the $\sigma$-algebra by $\mu$-null sets. Apply this to the product structure $\Sigma^{\otimes 2m}$ with the product measure $D^{\otimes 2m}$ and the statement follows immediately. □

What Proposition 7.4 leaves open is the reverse. The reverse direction is the subject of Section 7.3. We prepare the ground by introducing the regularity the kernel uses in practice to produce instances of `WellBehavedVCMeasTarget`.

**Definition 7.5** (Borel-parameterized concept class)**.** A concept class $C \subseteq \{0,1\}^X$ over a measurable space $(X, \Sigma)$ is *Borel-parameterized* by a standard Borel space $(\Theta, \mathcal{B}_\Theta)$ if there exists a jointly measurable evaluation map $\mathrm{eval} : \Theta \times X \to \{0,1\}$ such that $C = \{\, \mathrm{eval}(\theta, \cdot) \,:\, \theta \in \Theta \,\}$.

Borel parameterization is the standard regularity assumption in descriptive-set-theory approaches to learning theory. Every concept class considered in practice admits a Borel parameterization: finite classes, neural networks with measurable weights, decision trees of bounded depth, thresholds on $\mathbb{R}$, halfspaces, axis-aligned rectangles, and so on. When the parameter space $\Theta$ is discrete or finite, joint measurability is automatic. When $\Theta = \mathbb{R}^k$ for some $k$, joint measurability amounts to continuity of eval in $\theta$ for each fixed $x$ together with measurability of eval in $x$ for each fixed $\theta$. The kernel's `WellBehavedVCMeasTarget` setting captures exactly this regularity, and the next section proves that the refinement is automatic for every such class.

**Definition 7.6** (Parameterized ghost-gap bad event)**.** Lean: `paramBadEvent`

For a Borel-parameterized class with evaluation map eval, sample size $m$, and gap threshold $\varepsilon$, the *parameterized ghost-gap bad event* is

$$\mathrm{Bad}_{\mathrm{eval}}(m, \varepsilon) \;=\; \bigcup_{\theta \in \Theta} \Big\{\, (S, S') \in X^m \times X^m \;:\; |\hat{R}_{S'}(\mathrm{eval}(\theta, \cdot)) - \hat{R}_S(\mathrm{eval}(\theta, \cdot))| > \varepsilon \,\Big\}.$$

Equivalently, $\mathrm{Bad}_{\mathrm{eval}}(m, \varepsilon)$ is the projection onto the last two factors of a jointly Borel subset of $\Theta \times X^m \times X^m$.

The "projection of a jointly Borel event" reading is the key. A projection of a Borel set onto a product factor is not Borel in general. It is, however, always analytic. This observation is what connects the kernel's refinement to classical descriptive set theory, and it is the topic of Section 7.2.

## 7.2 The analytic measurability bridge

Analytic sets are the sets obtainable as continuous images of Polish spaces, equivalently, the projections of Borel sets in a product of Polish spaces, equivalently, the result of the Souslin operation applied to a Borel scheme. Every Borel set is analytic, the converse fails in the strongest possible way (there exist analytic sets of every cardinality up to the continuum that are not Borel), and the class of analytic sets is strictly larger than the Borel class while staying manageable under projection. This is classical, and Kechris [Kec95, Chapter 14] is the standard modern reference.

**Lemma 7.7** (Projections of Borel sets are analytic)**.** *Let $Y$ and $Z$ be standard Borel spaces and let $B \subseteq Y \times Z$ be Borel. The projection $\pi_Z(B) = \{\, z : \exists y,\ (y,z) \in B \,\}$ is an analytic subset of $Z$. This is the defining property of analytic sets in the Kechris formulation.*

**Corollary 7.8** (The parameterized bad event is analytic)**.** *For every Borel-parameterized concept class $C$ with parameter space $\Theta$, every $m$, and every $\varepsilon > 0$, the parameterized ghost-gap bad event $\mathrm{Bad}_{\mathrm{eval}}(m,\varepsilon)$ is an analytic subset of $X^m \times X^m$ whenever $X$ is standard Borel.*

*Proof.* The condition $|\hat{R}_{S'}(\mathrm{eval}(\theta,\cdot)) - \hat{R}_S(\mathrm{eval}(\theta,\cdot))| > \varepsilon$ is jointly Borel in $(\theta, S, S') \in \Theta \times X^m \times X^m$ because eval is jointly measurable and the empirical risks are finite sums of indicator evaluations. The bad event $\mathrm{Bad}_{\mathrm{eval}}(m,\varepsilon)$ is the projection of this Borel set onto the last two factors, hence analytic by Lemma 7.7. □

Corollary 7.8 reduces the measurability question for the bad event to a single question: *are analytic sets null-measurable with respect to every Borel probability measure?* That question was answered in 1955 by Gustave Choquet. The answer is yes, and the proof route is the capacitability theorem.

**Theorem 7.9** (Choquet capacitability, [Kec95, Theorem 30.13])**.** *Lean:* `MeasureTheory.AnalyticSet.compactCapeq`

*Let $X$ be a Polish space and let $\mu$ be a Borel probability measure on $X$. For every analytic set $A \subseteq X$,*

$$\mu^*(A) \;=\; \sup\big\{\, \mu(K) \,:\, K \subseteq A,\ K \text{ compact} \,\big\},$$

*where $\mu^*$ denotes the outer measure induced by $\mu$. In particular, $A$ is $\mu$-capacitable: its outer measure equals its inner measure, and $A$ lies in the completion of the Borel $\sigma$-algebra with respect to $\mu$.*

*Sketch.* The argument proceeds by showing that $\mu^*$ extended to analytic sets is a Choquet capacity: it is countably subadditive, monotone, and continuous from below on increasing unions, and continuous from above on decreasing intersections of compact sets. Any Choquet capacity is capacitable on analytic sets by an alternating game between approximating unions and intersections of compact sets. The compact approximations then converge in outer measure to $\mu^*(A)$ from below, and the Borel completion absorbs the gap. The full formalization runs 416 lines in `FLT_Proofs.PureMath.ChoquetCapacity` and is independent of learning theory; it is written in a form suitable for contribution to Mathlib as a standalone result. The blueprint treats it as a black box from here on. □

**Corollary 7.10** (Analytic sets are null-measurable)**.** *Lean:* `analyticSet`$_n$`ullMeasurableSet`

*Every analytic set in a Polish space is null-measurable with respect to every Borel probability measure.*

*Proof.* By Theorem 7.9, the outer and inner measures of an analytic set agree, which is equivalent to the set lying in the $\mu$-completion of the Borel $\sigma$-algebra. □

The bridge closes with a one-line composition: Corollary 7.8 says the bad event is analytic, Corollary 7.10 says analytic sets are null-measurable, and null-measurability of the bad event is exactly `WellBehavedVCMeasTarget`.

**Proposition 7.11** (Borel-parameterized implies WellBehavedVCMeasTarget)**.** *Lean:* `analyticSet`$_n$`ullMeasurableSe`

*Every Borel-parameterized concept class over a standard Borel space $X$ satisfies* `WellBehavedVCMeasTarget`.

The fundamental theorem of statistical learning (Theorem 3.7) therefore applies, in the kernel's formulation, to every Borel-parameterized concept class, including the classes for which Krapp and Wirth originally stated it under the stronger Borel hypothesis. The refinement is automatic for practical examples. The question that remains is whether it is *strictly* weaker, and it is to that question we now turn.

## 7.3 The Borel-analytic separation theorem

Proposition 7.11 says that every reasonable concept class satisfies the kernel's measurability hypothesis via a classical descriptive-set-theory route. What the bridge does not rule out is that the underlying refinement might be cosmetic: perhaps every class that admits any kind of measurability certificate also happens to admit a Borel one, and the `WellBehavedVCMeasTarget` $\subsetneq$ `KrappWirthWellBehaved` inclusion is vacuous in content. This section rules that out with a witness.

**Historical Note**

**Krapp and Wirth 2024.** Shortly before this formalization effort began, Lothar Sebastian Krapp and Laura Wirth posted *Measurability in the Fundamental Theorem of Statistical Learning* (arXiv:2410.10243, October 2024). The paper performs exactly the audit this chapter is built around: it scrutinizes the standard proofs of the fundamental theorem of statistical learning in the agnostic setting, identifies the measurability assumptions that are tacitly imposed, and extracts a self-contained proof that makes those assumptions precise. The paper's measurability hypothesis on the ghost-gap bad event is that it be *Borel* in the product Borel structure. The refinement in this chapter observes that the Borel hypothesis is strictly stronger than the proof needs, and this section constructs a concept class that realizes the strict gap. The Krapp-Wirth paper is correct: everything stated in it under the Borel hypothesis also holds under the weaker null-measurability hypothesis used here. What the present chapter adds is that the Borel hypothesis is a genuine commitment, not a free choice, and the commitment can be relaxed.

The witness is a family of very thin concept classes on the real line. For each set $A \subseteq \mathbb{R}$, let $\mathbf{1}_{\{a\}}$ denote the indicator function of the singleton $\{a\}$, regarded as an element of $\{0,1\}^{\mathbb{R}}$.

**Definition 7.12** (Singleton class over a set)**.** Lean: `singletonClassOn`

For $A \subseteq \mathbb{R}$, define the *singleton class over* $A$ as

$$C_A \;=\; \{\,\mathbf{0}\,\} \cup \{\,\mathbf{1}_{\{a\}} : a \in A\,\},$$

where $\mathbf{0}$ is the everywhere-false concept ($\mathbf{0}(x) = 0$ for every $x \in \mathbb{R}$).

A singleton class looks unremarkable from the outside. It has VC dimension at most 1, which makes it PAC learnable under any reasonable measurability hypothesis (one can learn the identity of the non-trivial hypothesis from a single labeled example). Its individual hypotheses are all individually measurable in the strongest possible sense: the zero concept is constant, and each $\mathbf{1}_{\{a\}}$ is the characteristic function of a Borel singleton.

**Lemma 7.13** (Every hypothesis in $C_A$ is measurable)**.** *Lean: `singletonClassOn_measurable`*

*For every $A \subseteq \mathbb{R}$, every hypothesis $h \in C_A$ is Borel measurable.*

The surprise is that individual measurability of every hypothesis does *not* lift to measurability of the ghost-gap bad event for the class as a whole. The bad event is an indexed union over $C_A$, and the index set $A$ can be arbitrarily irregular. The kernel formalizes the reduction from the bad event to a planar witness.

**Definition 7.14** (Planar witness event)**.** Lean: `planarWitnessEvent`

For $A \subseteq \mathbb{R}$, the *planar witness event* is the subset of $\mathbb{R}^2$ given by

$$W_A \;=\; \{\,(x,y) \in \mathbb{R}^2 \;:\; y \in A \text{ and } x \neq y\,\}.$$

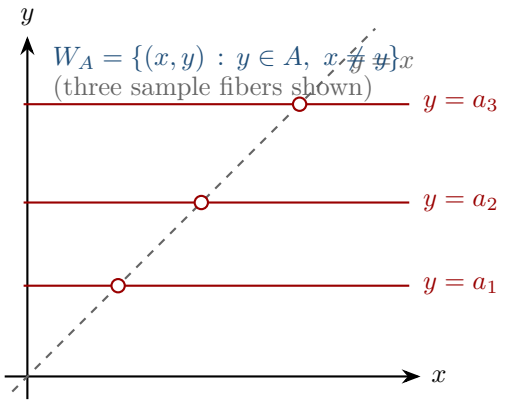


Figure 7.1: The planar witness $W_A$ depicted for three values $a_1, a_2, a_3 \in A$. Each fiber $\{y = a_i\}$ contributes the horizontal line at height $a_i$, punctured at the diagonal point $(a_i, a_i)$. The full set $W_A$ is the union of these punctured horizontal lines over all $y \in A$. When $A$ is analytic, $W_A$ is analytic. When $A$ is not Borel, $W_A$ is not Borel either.

**Lemma 7.15** (Bad event reduces to planar-witness preimage)**.** *Lean:* $singleton_bad Event_e q_p reimage_p lanar$

*For every $A \subseteq \mathbb{R}$, the one-step ghost-gap bad event $\mathrm{Bad}(C_A, 1, 1/2)$ on $(\mathrm{Fin}\,1 \to \mathbb{R}) \times (\mathrm{Fin}\,1 \to \mathbb{R})$ equals the preimage of the planar witness $W_A$ under the sample-pair projection $(s, s') \mapsto (s(0), s'(0))$.*

The reduction replaces a class-indexed existential quantifier with a fixed planar set, trading one measurability problem for another. The remaining problem is to understand the measurability of the planar set.

**Theorem 7.16** ($W_A$ is analytic for analytic $A$)**.** *Lean:* $planarWitnessEvent_a nalytic$

*If $A \subseteq \mathbb{R}$ is analytic then $W_A \subseteq \mathbb{R}^2$ is analytic.*

*Proof.* The set $W_A$ is the intersection of the Borel set $\{(x, y) : x \neq y\}$ with the cylinder $\mathbb{R} \times A$. Analytic sets are closed under intersection with Borel sets, so $W_A$ is analytic. □

**Theorem 7.17** ($W_A$ is not Borel for some analytic non-Borel $A$)**.** *Lean:* $planarWitnessEvent_n ot_m easurable$

*There exists an analytic set $A \subseteq \mathbb{R}$ such that $W_A$ is analytic but* not *Borel. The classical construction takes any complete analytic subset of $\mathbb{R}$, for example the set of codes of ill-founded trees on $\mathbb{N}$ under a standard encoding; this set is analytic and not Borel. The projection $A \mapsto W_A$ preserves the non-Borel property in the following sense: if $W_A$ were Borel, then the fiber of $W_A$ over any $x_0 \notin A$ would be Borel, but that fiber is exactly $A$, a contradiction.*

With $C_A$ defined, the planar reduction established, and the planar witness classified by descriptive set theory, the main separation theorem assembles in a single line of logic.

**Separation Result**

**Borel-analytic separation (Corollary 7.18).**

*Statement.* There exists a concept class $C \subseteq \{0,1\}^{\mathbb{R}}$ such that $C$ carries `WellBehavedVCMeasTarget` but $C$ does *not* carry `KrappWirthWellBehaved`.

*Witness.* Let $A \subseteq \mathbb{R}$ be any analytic non-Borel set, for example the set of codes of ill-founded trees as in Theorem 7.17. Take $C = C_A$, the singleton class over $A$.

*Why the witness works.* Every hypothesis in $C_A$ is individually Borel measurable (Lemma 7.13), so the class has the individual regularity one might expect. The ghost-gap bad event coincides, by Lemma 7.15, with the preimage of $W_A$ under the sample-pair projection. The planar witness $W_A$ is analytic by Theorem 7.16 and is not Borel by Theorem 7.17. The sample-pair projection is a homeomorphism onto its image, so the bad event is itself analytic and not Borel. Analytic sets are null-measurable by Corol-

lary 7.10, giving `WellBehavedVCMeasTarget`. Borel sets are not: the bad event fails `KrappWirthWellBehaved` by the non-Borel witness.

*What the witness exploits.* Individual measurability of the hypotheses in a class does not lift to measurability of the class-indexed bad event, because the bad event is an uncountable union whose indexing set can be as wild as the analytic hierarchy permits. The Borel $\sigma$-algebra is stable under countable unions but not under analytic-indexed unions; the completion under any Borel measure is exactly the extra layer needed to accommodate such unions. The singleton class is the minimal construction that exposes this gap: it has VC dimension 1, a single non-trivial hypothesis per index, and a planar bad event whose irregularity is inherited entirely from $A$.

[KW24], [Kec95]

**Corollary 7.18** (Borel-analytic separation)**.** *Lean:* $singleton_bad Event_n ot_m easurable$

*There exists a concept class $C_A \subseteq \{0,1\}^{\mathbb{R}}$ such that $C_A$ satisfies `WellBehavedVCMeasTarget` but fails `KrappWirthWellBehaved`. Consequently the fundamental theorem of statistical learning (Theorem 3.7) holds for $C_A$ under the kernel's NullMeasurableSet refinement and does not hold for $C_A$ under the Krapp-Wirth Borel hypothesis.*

*Remark* 7.19 (What the refinement changes and what it does not). The refinement changes the shape of the measurability hypothesis used in the fundamental theorem from a set-theoretic constraint on the class (Borel bad event) to a measure-theoretic constraint (null-measurable bad event). For every class that admits a Borel parameterization, both hypotheses hold and the two versions of the theorem are interchangeable. For classes outside Borel parameterization, only the null-measurability version applies. The Krapp-Wirth paper and this kernel therefore prove the same theorem on the same universe of practically interesting classes; the difference is at the edge of the universe, where concept classes over analytic non-Borel sets live. That edge is sparse in practice but non-empty, and Corollary 7.18 exhibits it in full.

The chain of five theorems formalized in `FLT_Proofs.Theorem.BorelAnalyticSeparation` thus completes the chapter. Every step is sorry-free, cross-linked with `FLT_Proofs.Complexity.BorelAnalytic` on the analytic measurability side, and cited from `FLT_Proofs.Complexity.Measurability` where the `WellBehavedVCMeasTarget` typeclass lives.

# Chapter 8

# Closing Notes and Forward Pointers

This document covers the core story of the kernel plus the Borel-analytic separation that was discovered during formalization. Three strands of the kernel are deferred to a successor blueprint.

- **Compression.** The kernel formalizes a compression scheme for concept classes via an approximate minimax argument using multiplicative weights, giving a constructive route to compression schemes where the classical Moran–Yehudayoff argument [MY16] uses the exact minimax theorem. This material belongs to a dedicated v2 chapter that will import the textbook's Chapter 11 on compression and add the kernel-specific MWU formalization on top.
- **The measurable batch learner monad.** The kernel introduces `MeasurableBatchLearner`, a regularity axis isolating joint learner measurability, and proves that it forms a monad closed under Boolean combination, majority vote, piecewise interpolation, and countable selection. The first proof that non-neural learners (version spaces) satisfy `MeasurableBatchLearner` uses a rectangle decomposition that bypasses the Kuratowski–Ryll-Nardzewski measurable selection theorem for countable families. This material belongs to a dedicated v2 chapter on learner regularity.
- **PAC-Bayes, extended criteria, and the Baxter multi-task base case.** These are additional characterization results the kernel formalizes but which sit outside the core story of v1. They will be covered as separate strands in v2 (PAC-Bayes as a frequentist–Bayesian bridge following McAllester [McA99], the extended criteria as a refinement of the measurability hierarchy, and the Baxter base case [Bax00] as a multi-task analogue of the fundamental theorem).

The kernel also contributes a proof engineering layer, the typed proof operad `TPG_FLT` together with the measurable inner event metaprogram, that sits orthogonal to the mathematical content. That layer is documented in a companion artifact at `https://zetetic-dhruv.github.io/formal-learning-theory-kernel/methodology/` (in preparation) which reuses the same leanblueprint infrastructure and is cross-linked with this blueprint on the case study of the measurable inner event pattern used by the symmetrization step of Chapter 7.

For the full mathematical exposition of every strand deferred above, together with the six paradigms not covered by the kernel (Ch 8 exact learning, Ch 9 universal learning, Ch 12 generalization bounds, Ch 13 mind-change ordinals, Ch 15 analogies, Ch 17 extensions, Ch 18 frontiers), see the companion textbook whose chapters this blueprint reuses.

# Appendix A

# Key Lean 4 Declarations

This appendix displays the formal Lean 4 code for the definitions and theorems claimed in the preceding chapters. Every declaration compiles against mathlib4 `fde0cc5` with zero `sorry`.

## A.1 Core types

Listing A.1: Concept and ConceptClass (Basic.lean)

```
def Concept (X : Type u) (Y : Type v) := X -> Y

abbrev ConceptClass (X : Type u) (Y : Type v) := Set (Concept X Y)
```

Listing A.2: BatchLearner (Learner/Core.lean)

```
structure BatchLearner (X : Type u) (Y : Type v) where
  hypotheses : HypothesisSpace X Y
  learn : {m : N} -> (Fin m -> X * Y) -> Concept X Y
  output_in_H : forall {m : N} (S : Fin m -> X * Y), learn S ∈ hypotheses
```

## A.2 Combinatorial dimensions

Listing A.3: Shattering and VC dimension (Complexity/VCDimension.lean)

```
def Shatters (X : Type u) (C : ConceptClass X Bool) (S : Finset X) : Prop :=
  forall f : S -> Bool, exists c ∈ C, forall x : S, c (x : X) = f x

noncomputable def VCDim (X : Type u) (C : ConceptClass X Bool) : WithTop N
    :=
  iSup (S : Finset X) (_ : Shatters X C S), (S.card : WithTop N)
```

Listing A.4: Littlestone dimension (Complexity/GameInfra.lean)

```
noncomputable def LittlestoneDim (X : Type) (C : ConceptClass X Bool) :
    WithBot (WithTop N) :=
  iSup (n : N) (_ : exists T : LTree X n, T.isShattered C),
    ((n : WithTop N) : WithBot (WithTop N))
```

## A.3 Learnability predicates

Listing A.5: PAC learnability (Criterion/PAC.lean)

```
def PACLearnable (X : Type u) [MeasurableSpace X]
    (C : ConceptClass X Bool) : Prop :=
  exists (L : BatchLearner X Bool) (mf : R -> R -> N),
    forall (e d : R), 0 < e -> 0 < d ->
      forall (D : Measure X), IsProbabilityMeasure D ->
        forall (c : Concept X Bool), c ∈ C ->
          let m := mf e d
          Measure.pi (fun _ : Fin m => D)
            { xs : Fin m -> X |
              D { x | L.learn (fun i => (xs i, c (xs i))) x ≠ c x }
                ≤ ENNReal.ofReal e }
            ≥ ENNReal.ofReal (1 - d)
```

Listing A.6: Online learnability (Criterion/Online.lean)

```
def OnlineLearnable (X : Type u) (Y : Type v) [DecidableEq Y]
    (C : ConceptClass X Y) : Prop :=
  exists (M : N), MistakeBounded X Y C M
```

Listing A.7: EX-learnability, Gold-style (Criterion/Gold.lean)

```
def EXLearnable (X : Type u) (C : ConceptClass X Bool) : Prop :=
  exists (L : GoldLearner X Bool),
    forall (c : Concept X Bool), c ∈ C ->
      forall (T : TextPresentation X c),
        exists (t0 : N), forall (t : N), t ≥ t0 ->
          L.conjecture (dataUpTo T.toDataStream t) = c
```

## A.4 The fundamental theorem (5-way equivalence)

Listing A.8: Fundamental theorem of statistical learning (Theorem/PAC.lean:293)

```
theorem fundamental_theorem (X : Type u) [MeasurableSpace X]
    [MeasurableSingletonClass X]
    (C : ConceptClass X Bool)
    [MeasurableConceptClass X C] :
    (PACLearnable X C ↔ VCDim X C < ⊤) ∧
    ((VCDim X C < ⊤) ↔
      exists (k : N) (cs : CompressionSchemeWithInfo0 X Bool C), cs.size = k
        ) ∧
    ((VCDim X C < ⊤) ↔
      forall e > 0, exists m0, forall (D : Measure X),
        IsProbabilityMeasure D ->
        forall m ≥ m0, RademacherComplexity X C D m < e) ∧
    (PACLearnable X C ->
      exists (L : BatchLearner X Bool) (mf : R -> R -> N),
        -- sample complexity sandwich + lower bound
        ...) ∧
    ((VCDim X C < ⊤) ↔
      exists (d : N), forall (m : N), d ≤ m ->
        GrowthFunction X C m ≤
          Finset.sum (Finset.range (d + 1)) (Nat.choose m)) :=
  -- Proof: assembles from component theorems
  ⟨vc_characterization X C,
   fundamental_vc_compression X C,
   (vc_characterization X C).symm.trans (fundamental_rademacher X C),
   pac_sample_complexity_sandwich X C,
   ⟨vcdim_finite_imp_growth_bounded X C,
```

```
    growth_bounded_imp_vcdim_finite X C⟩⟩
```

## A.5 Characterization theorems

Listing A.9: Littlestone characterization (Theorem/Online.lean:419)

```
theorem littlestone_characterization (X : Type)
    (C : ConceptClass X Bool) :
    OnlineLearnable X Bool C ↔ LittlestoneDim X C < ⊤ :=
  ⟨forward_direction X C, backward_direction X C⟩
```

Listing A.10: Gold's theorem (Theorem/Gold.lean:19, first 30 lines of 200)

```
theorem gold_theorem (X : Type u) [Countable X] [DecidableEq X]
    (C : ConceptClass X Bool)
    (h_finite : forall (S : Finset X), (fun x => decide (x ∈ S)) ∈ C)
    (h_infinite : exists c ∈ C, Set.Infinite { x | c x = true }) :
    ¬ EXLearnable X C := by
  intro ⟨L, hL⟩
  obtain ⟨c_inf, hc_inf_mem, hc_inf_inf⟩ := h_infinite
  let phi := hc_inf_inf.natEmbedding
  have hpos_ne : Nonempty { x : X | c_inf x = true } :=
    hc_inf_inf.nonempty.to_subtype
  obtain ⟨enum, henum⟩ :=
    exists_surjective_nat ({ x : X | c_inf x = true })
  let S : N -> Finset X := fun n =>
    ((Finset.range (n + 1)).image (fun i => (phi i).val)) ∪
    ((Finset.range (n + 1)).image (fun i => (enum i).val))
  -- ... (152 more lines: constructs locking sequence via
  --       interleaving enumeration of c_inf's support with
  --       finite-set indicators, derives contradiction)
```

## A.6 Paradigm separations

Listing A.11: PAC does not imply Online (Theorem/Separation.lean:1175)

```
theorem pac_not_implies_online :
    exists (X : Type) (_ : MeasurableSpace X) (C : ConceptClass X Bool),
      PACLearnable X C ∧ ¬ OnlineLearnable X Bool C := by
  refine ⟨N, ⊤, thresholdClass, ?_, ?_⟩
  . -- PAC learnable: VCDim = 1 < ⊤, via UC route
    exact vcdim_finite_imp_pac_via_uc' N thresholdClass
      vcdim_threshold_finite ...
  . -- ¬ OnlineLearnable: LittlestoneDim = ⊤
    intro hol
    have hfin := forward_direction N thresholdClass hol
    rw [ldim_threshold_top] at hfin
    exact lt_irrefl _ hfin
```

Listing A.12: Online implies PAC (Theorem/Separation.lean:132)

```
theorem online_imp_pac (X : Type u) [MeasurableSpace X]
    (C : ConceptClass X Bool) (hol : OnlineLearnable X Bool C)
    [MeasurableConceptClass X C] :
    PACLearnable X C := by
  obtain ⟨M, hM⟩ := hol
```

```
  have hvcdim : VCDim X C < ⊤ := by
    calc VCDim X C ≤ M := vcdim_le_of_mistake_bounded hM
      _ < ⊤ := WithTop.coe_lt_top M
  exact vcdim_finite_imp_pac_via_uc' X C hvcdim ...
```

## A.7 Measurability layer (kernel's novel contribution)

Listing A.13: WellBehavedVCMeasTarget (Complexity/Measurability.lean:388)

```
def WellBehavedVCMeasTarget
    (X : Type u) [MeasurableSpace X]
    (C : ConceptClass X Bool) : Prop :=
  forall (D : Measure X) [IsProbabilityMeasure D]
    (c : Concept X Bool), Measurable c ->
    forall (m : N) (e : R),
      NullMeasurableSet
        {p : (Fin m -> X) * (Fin m -> X) | exists h ∈ C,
          EmpiricalError X Bool h
            (fun i => (p.2 i, c (p.2 i))) (zeroOneLoss Bool) -
          EmpiricalError X Bool h
            (fun i => (p.1 i, c (p.1 i))) (zeroOneLoss Bool) ≥ e / 2}
        ((Measure.pi (fun _ : Fin m => D)).prod
         (Measure.pi (fun _ : Fin m => D)))
```

Listing A.14: KrappWirthWellBehaved (Complexity/Measurability.lean:241)

```
class KrappWirthWellBehaved (X : Type u) [MeasurableSpace X]
    (C : ConceptClass X Bool) : Prop extends MeasurableHypotheses X C where
  V_measurable : KrappWirthV X C
  U_measurable : KrappWirthU X C
```

Listing A.15: Borel-analytic separation witness (Theorem/BorelAnalyticSeparation.lean:207)

```
theorem singleton_badEvent_not_measurable
    (A : Set R) (hA_non : ¬ MeasurableSet A) :
    ¬ MeasurableSet (singletonBadEvent A) := by
  intro hbad
  rw [singleton_badEvent_eq_preimage_planar A] at hbad
  have hmeas : Measurable samplePair1ToPlane :=
    Measurable.prod
      ((measurable_pi_apply 0).comp measurable_fst)
      ((measurable_pi_apply 0).comp measurable_snd)
  have hsurj : Function.Surjective samplePair1ToPlane := by
    intro ⟨x, y⟩
    exact ⟨(fun _ => x, fun _ => y), by simp [samplePair1ToPlane]⟩
  have hplanar :=
    (hmeas.measurableSet_preimage_iff_of_surjective hsurj).mp hbad
  exact planarWitnessEvent_not_measurable A hA_non hplanar
```

Listing A.16: Analytic sets are null-measurable (PureMath/AnalyticMeasurability.lean:91)

```
theorem analyticSet_nullMeasurableSet
    {α : Type*}
    [TopologicalSpace α] [MeasurableSpace α]
    [BorelSpace α] [PolishSpace α]
    {μ : Measure α} [IsFiniteMeasure μ]
    {s : Set α} (hs : AnalyticSet s) :
    NullMeasurableSet s μ := by
  classical
```

```
  obtain ⟨t, hst, ht_meas, ht_eq⟩ :=
    exists_measurable_superset μ s
  have hzero : μ (t \ s) = 0 := by
    by_contra hne
    have hfin_diff : μ (t \ s) ≠ ⊤ := measure_ne_top μ _
    have hpos : 0 < μ.real (t \ s) :=
      ENNReal.toReal_pos hne hfin_diff
    obtain ⟨K, hKc, hKs, hKapprox⟩ :=
      hs.exists_isCompact_measureReal_gt
        (μ := μ) (μ.real (t \ s) / 2) (by positivity)
    have hKt : K ⊆ t := fun x hx => hst (hKs hx)
    have hKmeas : MeasurableSet K := hKc.isClosed.measurableSet
    have hdiff_eq : μ.real (t \ K) = μ.real t - μ.real K :=
      measureReal_diff hKt hKmeas
    have ht_real : μ.real t = μ.real s := by
      simp only [Measure.real]; rw [ht_eq]
    have hsub : t \ s ⊆ t \ K :=
      Set.diff_subset_diff_right hKs
    have hle : μ.real (t \ s) ≤ μ.real (t \ K) :=
      measureReal_mono hsub
    linarith
  have h_ae : s =ᵃᵉ[μ] t := by
    rw [Filter.eventuallyEq_comm, ae_eq_set]
    exact ⟨hzero, by simp [Set.diff_eq_empty.mpr hst]⟩
  exact ht_meas.nullMeasurableSet.congr h_ae.symm
```

# Bibliography


[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.

[BDPSS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *Proceedings of the 22nd Conference on Learning Theory (COLT)*, 2009.

[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.

[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.

[CS83] John Case and Carl H. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25(2):193–220, 1983.

[FS93] Rūsiņš Freivalds and Carl H. Smith. On the role of procrastination in machine learning. *Information and Computation*, 107(2):237–271, 1993.

[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.

[Kec95] Alexander S. Kechris. *Classical Descriptive Set Theory*, volume 156 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. The standard modern reference for analytic sets, Polish spaces, Borel hierarchy, and Choquet capacitability. Theorem 30.13 (capacitability of analytic sets) is the bridge used in Chapter 7 to prove that every Borel-parameterized concept class satisfies the kernel's `WellBehavedVCMeasTarget` hypothesis.

[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.

[KW24] Lothar Sebastian Krapp and Laura Wirth. Measurability in the fundamental theorem of statistical learning (with an appendix by laura wirth). Preprint, arXiv:2410.10243, October 2024. Scrutinizes the measurability hypotheses tacitly imposed in classical proofs of the Fundamental Theorem of Statistical Learning in the agnostic setting, extracts a sound statement together with a detailed proof showcasing the minimal measurability requirements. This kernel's NullMeasurableSet refinement (Chapter 7) weakens the Borel hypothesis Krapp and Wirth work under, and Corollary 7.18 exhibits a concrete concept class witnessing the strict gap.

[Lit88] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[McA99] David McAllester. PAC-Bayesian model averaging. In *Proceedings of the 12th Annual Conference on Computational Learning Theory (COLT)*, pages 164–170, 1999.

[MY16] Shay Moran and Amir Yehudayoff. Sample compression schemes for VC classes. *Journal of the ACM*, 63(3):1–10, 2016.

[PT20] Dömötör Pálvölgyi and Gábor Tardos. Unlabeled compression schemes exceeding the VC dimension. *Discrete Applied Mathematics*, 276:102–107, 2020.

[PV88] Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35(4):965–984, 1988.

[Sau72] Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13:145–147, 1972.

[She72] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.

[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.

[Val84] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[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.