CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

| | |
|---|---|
| Title | Lean formalization of generalization error bound by Rademacher complexity and Dudley's entropy integral |
| Authors | Sonoda, Sho;Kasaura, Kazumi;Mizuno, Yuma;Tsukamoto, Kei;Onda, Naoto |
| Publication date | 2026-07-16 |
| Original Citation | Sonoda, S, Kasaura, K, Mizuno, Y, Tsukamoto, K & Onda, N 2026, Lean formalization of generalization error bound by Rademacher complexity and Dudley's entropy integral. in E Komendantskaya, E Komendantskaya & T Nipkow (eds), 17th International Conference on Interactive Theorem Proving, ITP 2026., 8, Leibniz International Proceedings in Informatics, LIPIcs, vol. 382, Schloss Dagstuhl- Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, pp. 1-17, 17th International Conference on Interactive Theorem Proving, ITP 2026, Lisbon, Portugal, 26/07/26. https://doi.org/10.4230/LIPIcs.ITP.2026.8 |
| Type of publication | Conference item |
| Link to publisher's version | https://www.scopus.com/pages/publications/105045843790 - 10.4230/LIPIcs.ITP.2026.8 |
| Rights | © 2026, Sho Sonoda, Kazumi Kasaura, Yuma Mizuno, Kei Tsukamoto, and Naoto Onda. Licensed under Creative Commons License CC-BY 4.0 - https://creativecommons.org/licenses/by/4.0/ |
| Download date | 2026-11-08 10:48:41 |
| Item downloaded from | https://hdl.handle.net/10468/19181 |

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

# Lean Formalization of Generalization Error Bound by Rademacher Complexity and Dudley's Entropy Integral

**Sho Sonoda**[1]
RIKEN AIP, Tokyo, Japan
CyberAgent Inc., Tokyo, Japan

**Kazumi Kasaura**
OMRON SINIC X Corporation, Tokyo, Japan

**Yuma Mizuno**
University College Cork, Ireland

**Kei Tsukamoto**
The University of Tokyo, Japan

**Naoto Onda**
OMRON SINIC X Corporation, Tokyo, Japan

## Abstract

Understanding and certifying the generalization performance of machine learning algorithms – i.e. obtaining theoretical estimates of the test error from the training error – is a central theme of statistical learning theory. Among the many complexity measures used to derive such guarantees, *Rademacher complexity* yields sharp, data-dependent bounds that apply well beyond classical VC-dimension theory. In this study, we formalize the generalization error bound by *Rademacher complexity* in Lean 4, building on measure-theoretic probability theory available in the Mathlib library. Our development provides a mechanically-checked pipeline from the definitions of empirical and expected Rademacher complexity, through a formal symmetrization argument and a bounded-differences analysis, to high-probability uniform deviation bounds via a formally proved McDiarmid inequality. A key technical contribution is a reusable mechanism for lifting results from *countable* hypothesis classes (where measurability of suprema is straightforward in Mathlib) to *separable* topological index sets via a reduction to a countable dense subset. As worked applications of the abstract theorem, we mechanize standard empirical Rademacher bounds for linear predictors under $\ell_2$ and $\ell_1$ regularizations, and we also formalize a Dudley-type entropy integral bound based on covering numbers and a chaining construction.

**2012 ACM Subject Classification** Theory of computation → Sample complexity and generalization bounds; Theory of computation → Logic and verification

**Keywords and phrases** Lean, generalization error bound, Rademacher complexity, McDiarmid's inequality, Hoeffding's lemma, symmetrization arguments, chaining, Dudley's entropy integral

**Digital Object Identifier** 10.4230/LIPIcs.ITP.2026.8

**Related Version** *Preprint Version*: `https://arxiv.org/abs/2503.19605`

**Supplementary Material** *Software (Source Code)*: `https://github.com/auto-res/lean-rademacher/tree/f34ab4f0a9029682f1af3179a3fe9b5e114e511f` [15], archived at `swh:1:dir:06614ca069f4562b630fca2903e3b286a426e317`

**Funding** This work was supported by JSPS KAKENHI 24K21316, and JST Moonshot R&D Program JPMJMS2236, PRESTO JPMJPR2125, BOOST JPMJBY24E2, and CREST JPMJCR25I5.

**Acknowledgements** We thank the reviewers for useful feedback on our manuscript.

**Declaration about GenAI/LLM use** Generative AI was used to assist with routine Lean 4 formalization tasks, including type-system adjustments, casts, and algebraic rewrites.

[1] corresponding author

© Sho Sonoda, Kazumi Kasaura, Yuma Mizuno, Kei Tsukamoto, and Naoto Onda;
licensed under Creative Commons License CC-BY 4.0

# 1 Introduction

Generalization analysis is the task of estimating, from finite data, how accurately a learned model will perform on *unseen* test data. Formally, let $\mathcal{X}$ be a data domain and let $\mu$ be an unknown data-generating distribution on $\mathcal{X}$. A learning algorithm observes an i.i.d. sample $S = (x_1, \ldots, x_n)$ drawn from $\mu$ and selects a hypothesis $h$ from some hypothesis class $\mathcal{H}$ (or, more specifically, a function from a parameterized family). If $\ell(h, x)$ denotes a loss value on data $x$, the *population risk* (test error) and the *empirical risk* (training error) are defined as

$$L(h) := \mathbb{E}_{x\sim\mu}[\ell(h, x)], \qquad \widehat{L}_S(h) := \frac{1}{n}\sum_{k=1}^{n} \ell(h, x_k).$$

Here a "data point" may already contain both an input and a label. For example, in supervised regression one may take $\mathcal{X} = \mathcal{U} \times \mathbb{R}$, write a sample point as $x = (u, y)$, and let a predictor $h : \mathcal{U} \to \mathbb{R}$ have squared-error loss

$$\ell(h, (u, y)) := (h(u) - y)^2.$$

Then

$$L(h) = \mathbb{E}_{(u,y)\sim\mu}\big[(h(u) - y)^2\big], \qquad \widehat{L}_S(h) = \frac{1}{n}\sum_{k=1}^{n}(h(u_k) - y_k)^2$$

for a sample $S = ((u_1, y_1), \ldots, (u_n, y_n))$. A typical goal in statistical learning theory is to bound the *generalization gap* $|\widehat{L}_S(h) - L(h)|$, uniformly over all hypotheses in the class (or at least over the hypotheses produced by a learning algorithm), with high probability over the sample draw.

In this study, we formalize the generalization error bound by *Rademacher complexity* [5] in Lean 4 [8] based on probability theory formalized in Lean 4's mathematical library, Mathlib [19]. We refer to Mohri et al. (2018) [14] as a standard textbook on statistical machine learning theory, and Wainwright (2019) [13] for more mathematical details on the Rademacher complexity and concentration inequalities used in generalization error analysis. Concretely, rather than fixing a particular learning algorithm, we formalize a uniform law of large numbers (ULLN) style deviation functional, which we call the *uniform deviation*: for a family of measurable real-valued functions $F = \{f_i : \mathcal{X} \to \mathbb{R}\}_{i\in\iota}$ we consider

$$\mathrm{UD}_n(F; S) := \sup_{i\in\iota}\left|\frac{1}{n}\sum_{k=1}^{n} f_i(x_k) - \mathbb{E}_{x\sim\mu}[f_i(x)]\right|.$$

This quantity subsumes classical generalization-gap bounds by taking $f_i$ to be loss functions $\ell(h_i, \cdot)$, and it also aligns well with the measure-theoretic infrastructure in Mathlib. In the Lean code, the corresponding objects are defined as `uniformDeviation` and bounded using `empiricalRademacherComplexity` and `rademacherComplexity` (Section 4 of the main body).

### Why Rademacher complexity?

Classical sample-complexity theory for the *PAC learning* framework [22] often proceeds via the *Vapnik–Chervonenkis (VC) dimension* [23]. While powerful, VC-dimension-based bounds are typically *data-independent* and tailored to worst-case uniform guarantees for 0–1 classification. In modern learning settings, however, one often seeks bounds that (i) apply to real-valued losses and function classes beyond binary classifiers, and (ii) adapt to the

*observed sample* through a data-dependent complexity. Rademacher complexity provides precisely such a tool: it measures the ability of a function class to correlate with random signs, and yields sharp uniform deviation bounds for a wide variety of learning problems, including analyses that underpin *kernel methods* and *deep learning* [14, 18, 3]. This flexibility motivates a foundational formalization effort: once the abstract Rademacher pipeline is mechanized, many concrete generalization analyses can be obtained as *specializations* by instantiating the function class and bounding its complexity.

### A textbook-level bound and its formal counterpart

At a high level, the classical Rademacher approach yields the following pattern. Assume a uniform bound $|f_i(x)| \leq b$. Then one first proves an *expected* bound

$$\mathbb{E}[\mathrm{UD}_n(F;S)] \leq 2\,\mathfrak{R}_n(F),$$

where $\mathfrak{R}_n(F)$ is the (distribution-dependent) Rademacher complexity of $F$. Then one upgrades this to a *high-probability* statement using a concentration inequality (typically McDiarmid's inequality), obtaining for all $\delta \in (0,1)$

$$\mathbb{P}\left(\mathrm{UD}_n(F;S) \leq 2\,\mathfrak{R}_n(F) + b\sqrt{\frac{2\log(1/\delta)}{n}}\right) \geq 1-\delta.$$

Our Lean development follows and formalizes this standard "symmetrization + bounded differences" route, and exposes the corresponding countable-index and separable-index tail theorems in `FoML/Main.lean`.

### Why Lean and Mathlib?

A recurring theme in modern learning theory is that generalization bounds are most naturally expressed and proved in a measure-theoretic language: product measures for i.i.d. sampling, measurability of suprema, integrability of envelopes, and concentration inequalities stated for measurable functions on probability spaces. Over the past several years, measure theory and probability theory in Lean's ecosystem have developed rapidly, and Mathlib has become a particularly convenient target for such work due to its extensive analysis libraries and the large community of mathematically oriented contributors. These factors make Lean 4 + Mathlib a natural setting for formalizing generalization analysis at a level close to contemporary textbooks (as opposed to highly simplified, finite or discrete proxies).

### Novelty and contributions

Prior work has formalized generalization bounds, but only in relatively simplified settings. For example, Bagnall and Stewart (2019) [4] formalize a generalization bound for neural networks, but restrict parameters to a finite set, which is insufficient for formalizing today's diverse generalization analyses. Our contribution is, to our knowledge, the *first* systematic formalization of generalization analysis via Rademacher complexity. The classical inequalities we use are mostly known on paper; the new mathematical clarification produced by the formalization is the precise package of sufficient conditions needed to make uncountable suprema measurable in the Rademacher pipeline. In particular, the Lean proof isolates a reusable countable-dense reduction for separable index spaces, together with continuity-under-integration hypotheses such as first countability and uniform domination, conditions that are routinely implicit in empirical-process arguments but are not presented as a single textbook lemma. More specifically, the repository provides:

- a reusable Lean formalization of empirical and expected Rademacher complexity and the associated uniform deviation functional;
- a mechanically checked symmetrization argument connecting uniform deviations to Rademacher averages;
- a formally verified bounded-differences (Lipschitz) analysis of the uniform deviation functional and a full McDiarmid inequality proof, yielding high-probability generalization bounds;
- a principled extension from countable index sets to separable topological parameter spaces via a countable dense reduction, resolving measurability issues of uncountable suprema;
- reusable formal infrastructure for maximal inequalities, Massart's lemma, PMF-based Rademacher variables, empirical pseudo-metrics, covering numbers, covering chains, and Dudley's entropy integral;
- worked applications: formal generalization-relevant Rademacher bounds for $\ell_2$- and $\ell_1$-regularized linear predictors, and a formal Dudley entropy integral bound via covering numbers and chaining.

### Paper organization

Section 2 summarizes the repository and its measure-theoretic scope. Sections 3–4 define the core objects and present the main Rademacher-based expectation and tail bounds. Section 5 develops the $\ell_2$ and $\ell_1$ linear predictor examples. Section 6 presents the Dudley entropy integral formalization. Section 7 discusses key design choices and engineering challenges that arise when translating textbook arguments into Mathlib's formal framework, and Section 8 situates the contribution in the broader literature on formalized learning theory and concentration inequalities.

## 2 Repository Overview and Scope

The repository's core contribution is a mechanically-checked pipeline from (i) definitions of empirical and expected Rademacher complexity, through (ii) a formal symmetrization argument, to (iii) high-probability uniform deviation (ULLN-style) bounds via a formally proved McDiarmid inequality, with an additional reduction step that extends the results from countable hypothesis classes to separable topological index sets. It also includes worked examples showing how $\ell_2$- and $\ell_1$-type norm constraints yield standard $O(1/\sqrt{n})$ and $O(\sqrt{\log(d)/n})$ empirical Rademacher-complexity bounds for linear prediction, and a formal Dudley entropy-integral upper bound for empirical Rademacher complexity (via covering numbers and a chaining construction).

From a theorem proving perspective, the notable aspects are: the explicit handling of measurability/integrability hypotheses (often forcing `Countable` index sets, or separability+continuity to reduce to a countable dense subclass), the use of product measures $\mu^n$ rather than i.i.d. sequences as primitive, and the systematic bridging between informal learning-theory arguments and Mathlib's measure-theoretic infrastructure. Conceptually, the main theorems align with classical results in Rademacher complexity and generalization theory as in Bartlett–Mendelson (risk bounds via Rademacher averages), McDiarmid's bounded-differences method, and Dudley's metric-entropy/chaining bounds. The formalization contribution is to make the side conditions, especially measurability of suprema and integrability under product measures, explicit enough to be reused by later learning-theory developments.

### Scope and assumptions

Our Lean development is structured around a general measure-theoretic formulation of empirical processes: a probability space $(\Omega, m\Omega, \mu)$, a data space $\mathcal{X}$ (sometimes `Z` in code), and a random variable $X : \Omega \to \mathcal{X}$ (encoding the data distribution). Fix an index type $\iota$ (parameterizing hypotheses), and a family of real-valued functions

$$\mathcal{F} \;=\; \{f_i : \mathcal{X} \to \mathbb{R}\}_{i \in \iota}.$$

In the code, such a class is represented by a single curried term `f : ι → 𝒳 → ℝ`.

Samples of size $n$ are modeled as functions $\omega : \mathrm{Fin}\ n \to \Omega$, pushed forward through $X$ to obtain $S := X \circ \omega : \mathrm{Fin}\ n \to \mathcal{X}$. Expectations are integrals in Mathlib notation, e.g. $\mu$`[g]` for $\int g \, d\mu$.

We assume the reader is comfortable with basic probability/analysis (product measures, integrals, concentration inequalities, norms) but not necessarily with ML-specific vocabulary (hypothesis classes, generalization error, empirical risk). We will use natural logarithms log (Lean's `Real.log`) and the convention that $n \in \mathbb{N}$, $n > 0$ when dividing by $n$ or using $\sqrt{n}$. We interpret `toReal` of an `ENNReal` probability as the ordinary probability in $[0, 1]$ (a standard Mathlib bridge).

## 3 Core Definitions and Notational Conventions

The central definitions in the file `FoML/Defs.lean` include the empirical and (expected) Rademacher complexities:

### Rademacher signs as a finite type

The repository models a sign vector as a function `Fin n → {-1,1}` in $\mathbb{Z}$, with `Fintype` structure giving a finite uniform averaging over all $2^n$ sign assignments.

```
-- FoML/Defs.lean
def Signs (n : ℕ) : Type := Fin n → ({-1, 1} : Finset ℤ)
```

Mathematically, one can view `Signs n` as $\{\pm 1\}^n$ equipped with the uniform distribution; Lean often uses explicit finite sums multiplied by $(\#\mathrm{Signs}\ n)^{-1}$ instead of an expectation over i.i.d. $\sigma_k$.

### Empirical Rademacher complexity

For a sample $S : \mathrm{Fin}\ n \to \mathcal{X}$ and a class $f : \iota \to \mathcal{X} \to \mathbb{R}$, the code defines:

$$\widehat{\mathfrak{R}}_n(f; S) := \frac{1}{|\{\pm 1\}^n|} \sum_{\sigma \in \{\pm 1\}^n} \sup_{i \in \iota} \left| \frac{1}{n} \sum_{k=1}^{n} \sigma_k \, f_i(S_k) \right|.$$

This matches the common ML definition up to the choice of explicit averaging versus probabilistic expectation.

```
-- FoML/Defs.lean
def empiricalRademacherComplexity (n : ℕ) (f : ι → 𝒳 → ℝ) (S : Fin n → 𝒳) : ℝ :=
  (Fintype.card (Signs n) : ℝ)⁻¹ *
    Σ σ : Signs n, ⨆ i, |(n : ℝ)⁻¹ * Σ k : Fin n, (σ k : ℝ) * f i (S k)|
```

### Expected Rademacher complexity

For a random variable $X : \Omega \to \mathcal{X}$ and probability measure $\mu$ on $\Omega$, the development forms the product measure $\mu^n$ on $\Omega^n$ and defines:

$$\mathfrak{R}_n(f; \mu, X) := \mathbb{E}_{\omega \sim \mu^n}\left[\widehat{\mathfrak{R}}_n(f; X \circ \omega)\right].$$

```
-- FoML/Defs.lean
def rademacherComplexity (n : ℕ) (f : ι → 𝒳 → ℝ) (μ : Measure Ω) (X : Ω → 𝒳) :
    ℝ :=
    μ^n[fun ω : Fin n → Ω ↦ empiricalRademacherComplexity n f (X ∘ ω)]
```

### Uniform deviation

The central "generalization" quantity is the supremum of the absolute difference between the empirical mean and the population mean:

$$\mathrm{UD}_n(f; \mu, X; S) := \sup_{i \in \iota} \left| \frac{1}{n} \sum_{k=1}^{n} f_i(S_k) \; - \; \mathbb{E}_\mu[f_i(X)] \right|.$$

```
-- FoML/Defs.lean
def uniformDeviation (n : ℕ) (f : ι → 𝒳 → ℝ) (μ : Measure Ω) (X : Ω → 𝒳) (S :
    Fin n → 𝒳) : ℝ :=
    ⨆ i, |(n : ℝ)⁻¹ * Σ k : Fin n, f i (S k) - μ[fun ω' ↦ f i (X ω')]|
```

### One-sided Rademacher complexity

The repo also defines `empiricalRademacherComplexity_without_abs`, a one-sided empirical Rademacher complexity obtained by dropping the absolute value inside the supremum. This variant appears because Massart- and Dudley-style chaining arguments naturally bound a one-sided supremum of signed sums. The absolute-value version remains the main quantity for uniform deviations; the development keeps both definitions explicit and proves comparison lemmas so that the exact assumptions needed to move between one-sided and absolute-value statements are visible.

## 4 Generalization via Rademacher Complexity and Concentration

### Symmetrization as a first-class formal object

The file `FoML/Symmetrization.lean` develops an integral identity that replaces a supremum over differences of two independent samples by an average over sign-flipped sums. This is the backbone of standard Rademacher complexity arguments, and it is exposed as a theorem used downstream: `abs_symmetrization_equation`. The formal version is phrased over the product measure on $(\Omega \times \Omega)^n$ and uses finitary sign averaging.

### Expected supremum bound

In `FoML/Rademacher.lean`, the key theorem `expectation_le_rademacher` states (informally):

$$\mathbb{E}_{\omega \sim \mu^n}\left[\sup_{i \in \iota} \left| \sum_{k=1}^{n} f_i(X(\omega_k)) - n\,\mathbb{E}_\mu[f_i(X)] \right|\right] \;\le\; 2n \cdot \mathfrak{R}_n(f; \mu, X),$$

under `Nonempty` $\iota$, `Countable` $\iota$, measurability hypotheses, and a uniform bound $|f_i(z)| \le b$. The proof explicitly composes: (i) rewriting the mean as a coordinatewise mean on the product space, (ii) applying the symmetrization identity, and (iii) splitting the supremum via triangle inequality (a step corresponding to the classical "ghost sample + symmetrization $\Rightarrow 2\mathfrak{R}$" argument). For context, this is the standard route by which Rademacher complexity controls expected uniform deviations and risk bounds.

```
-- FoML/Rademacher.lean (excerpt)
theorem expectation_le_rademacher {X : Ω → Z} [Nonempty ι] [Countable ι]
    (hf : ∀ i, Measurable (f i ∘ X))
    {b : ℝ} (hb : b ≥ 0) (hf' : ∀ (i : ι) (z : Z), |f i z| ≤ b) :
    μ^n[fun ω : Fin n → Ω ↦ ⨆ i, |Σ k : Fin n, f i (X (ω k)) - n · μ[fun ω' ↦ f i
      (X ω')]|]
    ≤ (2 * n) · rademacherComplexity n f μ X := by ...
```

### Normalized form

The file `FoML/Main.lean` packages a normalized statement for `uniformDeviation`, dividing the previous bound by $n$:

$$\mathbb{E}_{\omega\sim\mu^n}\left[\mathrm{UD}_n(f;\mu,X;X\circ\omega)\right] \;\le\; 2\mathfrak{R}_n(f;\mu,X).$$

This is the canonical "expected generalization gap bound" in Rademacher complexity theory.

```
-- FoML/Main.lean (excerpt)
theorem uniform_deviation_expectation_le_two_smul_rademacher_complexity
    [Nonempty ι] [Countable ι] [IsProbabilityMeasure μ]
    (hn : 0 < n) (X : Ω → 𝒳)
    (hf : ∀ i, Measurable (f i ∘ X))
    {b : ℝ} (hb : 0 ≤ b) (hf' : ∀ i x, |f i x| ≤ b) :
    μ^n[fun ω : Fin n → Ω ↦ uniformDeviation n f μ X (X ∘ ω)]
    ≤ 2 · rademacherComplexity n f μ X := by ...
```

### Bounded differences for samples

To obtain high-probability bounds, the repo proves that changing one sample point changes `uniformDeviation` by at most $2b/n$ (the standard bounded differences constant) when all functions are bounded by $b$:

$$\left|\mathrm{UD}_n(S) - \mathrm{UD}_n(S^{(i\leftarrow x')})\right| \le \frac{2b}{n}.$$

```
-- FoML/BoundedDifference.lean (excerpt)
theorem uniformDeviation_bounded_difference
    (hn : 0 < n) (X : Ω → 𝒳) (hf : ∀ i, Measurable (f i ∘ X))
    {b : ℝ} (hb : 0 ≤ b) (hf': ∀ i z, |f i z| ≤ b)
    (i : Fin n) (S : Fin n → 𝒳) (x' : 𝒳) :
    |uniformDeviation n f μ X S - uniformDeviation n f μ X (Function.update S i x')|
    ≤ (n : ℝ)⁻¹ * 2 * b := by ...
```

### McDiarmid tail bound and the countable theorem

Using a formal McDiarmid inequality (file `FoML/McDiarmid.lean`), `FoML/Main.lean` proves a tail bound around the mean and then combines it with the expectation bound to obtain the standard Rademacher-complexity tail estimate:

$$\mathbb{P}(\mathrm{UD}_n \;\ge\; 2\mathfrak{R}_n + \varepsilon) \;\le\; \exp\!\left(-\frac{\varepsilon^2\, n}{2b^2}\right),$$

where $|f_i(x)| \le b$ uniformly. This is precisely the usual "symmetrization + McDiarmid" pathway found in standard textbooks and original papers on learning bounds.

```
-- FoML/Main.lean (excerpt)
theorem uniform_deviation_tail_bound_countable_of_pos
    [MeasurableSpace 𝒳] [Nonempty 𝒳] [Nonempty ι] [Countable ι] [
    IsProbabilityMeasure μ]
    (f : ι → 𝒳 → ℝ) (hf : ∀ i, Measurable (f i))
    (X : Ω → 𝒳) (hX : Measurable X)
    {b : ℝ} (hb : 0 < b) (hf' : ∀ i x, |f i x| ≤ b)
    {ε : ℝ} (hε : 0 ≤ ε) :
    (μ^n (fun ω ↦ 2 · rademacherComplexity n f μ X + ε ≤ uniformDeviation n f μ X (X
     ∘ ω))).toReal
    ≤ (- ε ^ 2 * n / (2 * b ^ 2)).exp := by ...
```

### Extending from countable to separable index sets

A recurring obstacle in mechanizing learning theory is measurability of suprema: `mathlib` provides `Measurable.iSup` for countable index types, but for uncountable parameter spaces measurability is not automatic. The development addresses this by separating *proof obligations* from *modeling convenience*. Theorem `main_countable` is proved for `[Countable ι]`; then `main_separable` lifts the result to the common case where $\iota$ is a separable topological space and the map $i \mapsto f_i(x)$ is continuous for each $x$.

The lifting works by reducing $\sup_{i \in \iota}$ to a supremum over a countable dense sequence. Formally, `FoML/SeparableSpaceSup.lean` proves a lemma (for real-valued continuous functions)

$$\sup_{x \in X} g(x) \;=\; \sup_{n \in \mathbb{N}} g(\mathrm{denseSeq}(n)),$$

stated as `separableSpaceSup_eq_real`. This lemma is then used to show definitional equivalence of the learning-theoretic quantities when replacing $f : \iota \to (\mathcal{X} \to \mathbb{R})$ by its restriction along `denseSeq ι : Nat → ι`. In particular, `FoML/Main.lean` proves equalities

$$\widehat{\mathfrak{R}}_n(\mathcal{F}\ x) = \widehat{\mathfrak{R}}_n(\mathcal{F} \circ \mathrm{denseSeq}\ x), \qquad \mathfrak{R}_n(\mathcal{F}) = \mathfrak{R}_n(\mathcal{F} \circ \mathrm{denseSeq}), \qquad \Delta_n = \Delta_n^{\mathrm{denseSeq}},$$

These are packaged as equality lemmas for empirical complexity, expected complexity, and uniform deviation. The last equality is technically the most subtle: it requires continuity of the integral term $i \mapsto \int f_i(X(\omega)), d\mu(\omega)$, which is obtained via a dominated-continuity lemma under the uniform bound $|f_i(x)| \le b$.

The final theorem `main_separable` applies `main_countable` to the countable family `f ∘ denseSeq ι` and rewrites the result back using the aforementioned equalities. Mathematically, it yields the same tail bound as `main_countable` but under topological assumptions (`[SeparableSpace ι]` plus continuity) rather than the purely set-theoretic `[Countable ι]` assumption.

```
-- FoML/Main.lean (excerpt)
theorem uniform_deviation_tail_bound_separable_of_pos
    [MeasurableSpace 𝒳] [Nonempty 𝒳] [Nonempty ι]
    [TopologicalSpace ι] [SeparableSpace ι] [FirstCountableTopology ι]
    [IsProbabilityMeasure μ]
    (f : ι → 𝒳 → ℝ) (hf : ∀ i, Measurable (f i))
    (X : Ω → 𝒳) (hX : Measurable X)
    {b : ℝ} (hb : 0 < b) (hf' : ∀ i x, |f i x| ≤ b)
    (hf'' : ∀ x : 𝒳, Continuous fun i ↦ f i x)
    {ε : ℝ} (hε : 0 ≤ ε) :
    (μ^n (fun ω ↦ 2 · rademacherComplexity n f μ X + ε ≤ uniformDeviation n f μ X (X
     ∘ ω))).toReal
    ≤ (- ε ^ 2 * n / (2 * b ^ 2)).exp := by ...
```

## 5 Linear Predictors with $\ell_2$ and $\ell_1$ Regularization

To demonstrate the power of the abstract infrastructure, we specialize the Rademacher complexity to linear hypothesis classes with norm constraints. These are the simplest continuous model classes used in learning theory: a parameter vector $w$ defines a prediction rule $x \mapsto \langle w, x \rangle$, and different norm constraints on $w$ capture different geometric assumptions on the model class.

### $\ell_2$-bounded predictors

Let $d \in \mathbb{N}$ and consider the Euclidean space $\mathbb{R}^d$ (modeled as `EuclideanSpace ℝ (Fin d)`). For weights $w$ in an $\ell_2$ ball of radius $W$ and inputs $x$ in an $\ell_2$ ball of radius $X$, the class is

$$f_w(x) = \langle w, x \rangle.$$

The repo formalizes the standard empirical bound:

$$\widehat{\mathfrak{R}}_n(\{f_w : \|w\|_2 \le W\}; S) \;\le\; \frac{XW}{\sqrt{n}}.$$

The proof is recognizably the classical argument: rewrite the supremum via Cauchy–Schwarz, then compute the second moment of the Rademacher-weighted sum using sign orthogonality (all cross terms vanish). The orthogonality lemma is proved by an explicit sign-flip involution (`rademacher_flip`) in `FoML/RademacherVariableProperty.lean`.

```
-- FoML/Main.lean (excerpt)
theorem linear_predictor_l2_bound
    [Nonempty ι] (d : ℕ) (W X : ℝ) (hx : 0 ≤ X) (hw : 0 ≤ W)
    (Y' : Fin n → Metric.closedBall (0 : EuclideanSpace ℝ (Fin d)) X)
    (w' : ι → Metric.closedBall (0 : EuclideanSpace ℝ (Fin d)) W) :
    empiricalRademacherComplexity
      n (fun (i : ι) a ↦ ⟪((Subtype.val ∘ w') i), a⟫) (Subtype.val ∘ Y')
    ≤ X * W / √(n : ℝ) := by ...
```

### $\ell_1$-bounded predictors

For $\ell_1$-bounded weights $\|w\|_1 \le W$ and $\ell_\infty$-bounded inputs $\|x\|_\infty \le X_\infty$, standard learning theory gives

$$\widehat{\mathfrak{R}}_n(\{x \mapsto \langle w, x \rangle : \|w\|_1 \le W\}; S) \;\lesssim\; \frac{X_\infty W}{\sqrt{n}} \sqrt{2\log(2d)}.$$

The repo proves essentially this statement (with explicit constants), using a Massart-type finite-class lemma for a $2d$-sized auxiliary class corresponding to signed coordinate projections; and a duality inequality reflecting $\sup_{\|w\|_1 \le W} \langle w, z \rangle = W\|z\|_\infty$ (implemented with careful absolute values and finiteness). Classical references typically attribute the finite-class step to Massart and standard concentration methods.

```
-- FoML/Main.lean (excerpt)
theorem linear_predictor_l1_bound
    [Nonempty ι] (d : ℕ) (Xinf W : ℝ)
    (hX : 0 ≤ Xinf) (hW : 0 ≤ W) (d_pos : 0 < d) (n_pos : 0 < n)
    (Y' : Fin n → LinftyBall (d := d) Xinf)
    (w' : ι → L1Ball (d := d) W) :
    empiricalRademacherComplexity n
      (fun i a => (Σ j : Fin d, (w' i).1 j * a j))
      (Subtype.val ∘ Y')
    ≤ (Xinf * W / Real.sqrt (n : ℝ)) * Real.sqrt (2 * Real.log (2 * d)) := by ...
```

## 6 Dudley Entropy Integral Bound

Beyond linear predictors, we formalized the Dudley entropy integral bound, which relates the Rademacher complexity of an arbitrary function class to its covering numbers under the empirical $L^2$ metric. Here "entropy" means the logarithm of a covering number: it measures how many functions are needed to approximate the whole class at scale $\varepsilon$ on the observed sample. This required a substantial additional infrastructure layer, because Dudley's theorem is naturally expressed in terms of metric entropy with respect to a sample-dependent pseudometric.

### Empirical pseudo-metric and covering numbers

For a fixed sample $S : \mathrm{Fin}\ n \to \mathcal{X}$ and functions $f, g : \mathcal{X} \to \mathbb{R}$, the repo defines an empirical $\ell_2$-type seminorm

$$\|f - g\|_S := \sqrt{\frac{1}{n}\sum_{k=1}^{n}(f(S_k) - g(S_k))^2},$$

implemented via `empiricalNorm` and `empiricalDist` in `FoML/PseudoMetric.lean`. This induces a pseudo-metric structure on the set of functions restricted to the sample.

Covering numbers are then defined for totally bounded sets in a pseudo-metric space:

$$N(\varepsilon) := \min\left\{|T| : T \subseteq A,\ A \subseteq \bigcup_{t\in T} B(t,\varepsilon)\right\},$$

implemented as `coveringNumber ha ε` in `FoML/CoveringNumber.lean`. The Lean term uses `Nat.find`, Lean's operator for extracting the least natural number satisfying a proven existential statement; here the existential statement is the finite-cover lemma for totally bounded sets.

### The Dudley-type theorem as formalized

The central statement in `FoML/Main.lean` is a wrapper around `dudley_entropy_integral'` from `FoML/DudleyEntropy.lean`. It bounds the *without-absolute-value* empirical Rademacher complexity in terms of an entropy integral:

$$\widehat{\mathfrak{R}}_n^{(\text{no abs})}(F; S) \ \le\ 4\varepsilon \ +\ \frac{12}{\sqrt{n}}\int_{\varepsilon}^{c/2}\sqrt{\log N(u)}\,du,$$

where $c$ is an upper bound on all empirical norms $\|F(f)\|_S$ and $\varepsilon < c/2$, and where $N(u)$ is the covering number of the class realized as a subset of `EmpiricalFunctionSpace`.

```
-- FoML/Main.lean (excerpt)
theorem dudley_entropy_integral_bound
  {𝒳 : Type v} {n : ℕ} {ι : Type u} [Nonempty ι] {F : ι → 𝒳 → ℝ} {S : Fin n → 𝒳}
     {c ε : ℝ}
  (ε_pos : 0 < ε) (h' : TotallyBounded (Set.univ : Set (EmpiricalFunctionSpace F S)))
  (m_pos : 0 < n) (cs : ∀ f : ι, empiricalNorm S (F f) ≤ c)
  (ε_le_c_div_2 : ε < c/2) :
    empiricalRademacherComplexity_without_abs n F S ≤
    (4 * ε + (12 / Real.sqrt n) *
    (∫ (x : ℝ) in ε..(c/2), √(Real.log (coveringNumber h' x)))) := by ...
```

### Relation to classical Dudley bounds

Dudley's 1967 work [9] introduced entropy integrals to control sample regularity and suprema of Gaussian processes. Modern empirical-process theory adapts similar "chaining" ideas to Rademacher/Gaussian complexities, yielding inequalities of the form

$$\mathbb{E}_\sigma \left[ \sup_{f \in \mathcal{F}} \sum_i \sigma_i f(x_i) \right] \lesssim \int \sqrt{\log N(\varepsilon)}\, d\varepsilon.$$

The Lean development follows this blueprint explicitly: it constructs dyadic radii $\varepsilon_j = c/2^j$, chooses finite $\varepsilon_j$-covers (via `coveringFinset`), defines a chaining approximation `chainApprox`, proves a telescoping decomposition, bounds the "tail" by $\varepsilon$ (Part A), bounds increments by a Massart finite-class lemma (Part B), and converts the resulting finite sum into an interval integral using a monotonicity-based Riemann-sum comparison.

## 7 Design Choices

We review the particularly challenging components of the development and the methodological choices that enabled progress.

### Independent variables

In the Rademacher-complexity argument, the training dataset is assumed to be i.i.d. This is essential, for example, in the symmetrization argument. In a textbook-style formulation, i.i.d. variables $X_1, \ldots, X_n$ are formulated as a map $\mathrm{Fin}\, n \to \Omega \to \mathcal{X}$. However, we noticed that it is more convenient to formalize them as compositions of a single variable $X : \Omega \to \mathcal{X}$ with the coordinate projections $\Omega^n \to \Omega$, taking $\Omega^n$ to be the base probability space. In this view, independence need not be explicitly assumed as it follows directly from the construction, which simplifies descriptions of theorems.

On the other hand, in McDiarmid's inequality, the random variables $X_1, \ldots, X_n$ are assumed to be independent but need not be identically distributed. So we formalize them as distinct functions $X : \mathrm{Fin}\, n \to \Omega \to \mathbb{R}$ and explicitly assume independence as an explicit condition. Here the base probability space is $\Omega$.

The fact that the former construction $\Omega^n \to \Omega \to \mathcal{X}$ satisfies the independence condition required in the latter is a theorem that should be proved. We could not find this result in Mathlib, so we supplied our own proof. We also attempted another formalization that defines Rademacher complexity via probability mass functions, but deriving its properties along this path proved difficult, and we abandoned that approach.

### Topological details of the index set

The Rademacher complexity is an expectation of the *supremum* over a hypothesis class $\mathcal{F}$, and in modern machine learning settings, $\mathcal{F}$ is often *uncountable*. When $\mathcal{F}$ is uncountable, however, measurability is not generally preserved under pointwise sup. We therefore first prove results in the *countable* case and then extend to the *separable* case. Even when uncountable, a separable family allows the sup of continuous functions to be computed over a countable dense subset, reducing to the countable case. The formalization clarified the exact hypotheses needed for this reduction in the learning-theoretic setting: continuity must also be preserved after taking the population integral, which we prove using uniform domination and a first-countability assumption. This package of sufficient conditions is routine in applications, but we were unable to locate it as a ready-made statement in standard learning-theory or empirical-process texts.

### Conditional expectation

Textbook proofs of McDiarmid's inequality typically proceed via conditional expectations. However, because conditional expectation is defined abstractly, it is difficult to carry out concrete computations directly from that definition. (At the time we were developing the formalization, Mathlib had not yet included the Doob–Dynkin lemma; now there is `Mathlib.MeasureTheory.Function.FactorsThrough`.) We therefore avoided conditional expectations and instead defined the relevant quantities directly by integration, using independence. This changes the proof order: in the original argument the constructed sequence $Y$ is a martingale by construction, whereas in our approach (without conditional expectations) we establish the martingale property from independence.

### Integral and supremum

A large fraction of the lines in many lemmas is devoted to handling integrability conditions (and the treatment of sup), which are often considered routine. Each time we performed an algebraic manipulation under the integral sign, we had to supply a fresh proof of integrability.

### Dudley entropy integral and `without_abs`

The Dudley entropy integral bound is stated for `empiricalRademacherComplexity_without_abs` because the chaining proof controls a one-sided supremum of signed increments. In standard learning-theory presentations, this distinction is often hidden by working with centered or symmetric classes, or by applying the argument to both a class and its negation. Our Lean development keeps the one-sided and absolute-value versions separate, and records comparison lemmas in `FoML/RademacherVariableProperty.lean` so that later users can see exactly which symmetry or boundedness assumptions are being used.

### Generative AI assistance

Our project spans more than two years of manual formalization. During the early stages, available large language models tended to produce Lean 3 code and were not practically useful. Since mid-2025 the models improved markedly and began producing syntactically correct Lean 4 code; we used generative AI to assist with routine yet tedious tasks such as rewriting expressions to fit Lean's type system, managing casts between numeric types, and expanding algebraic identities. For example, writing the empirical Rademacher complexity as a measure-theoretic expectation required meticulous adjustments of types. While generative AI streamlined these technical steps, it could not reliably construct full proofs; we therefore designed the high-level strategy and wrote most proofs manually.

## 8 Literature Overview

Tables 1 and 2 summarize close literature and related formalizations used as context for this study.

### 8.1 PAC Learning and VC Dimension (Lean, Rocq)

A hypothesis class with finite VC dimension (or simply a finite class) admits Probably Approximately Correct (PAC) generalization bounds. Early formalizations focused on specific cases as follows:

**Table 1** Formalization of machine-learning theory and related probabilistic analyses.

| | | |
|---|---|---|
| Bentkamp et al. (2016,2019) [6, 7] | Isabelle/HOL | Expressive power superiority of deep over shallow |
| Tassarotti and Harper (2018) [16] | Coq | Verified tail bounds for randomized programs via Karp-style recurrences |
| Bagnall and Stewart (2019) [4] | Coq | Generalization error bounds for finite hypothesis class |
| Tassarotti et al. (2021) [17] | Lean | PAC learnability of decision stumps |
| Vajjha et al. (2021) [20] | Coq | Convergence of reinforcement learning algorithms |
| Vajjha et al. (2022) [21] | Coq | Stochastic approximation theorem |
| Hirata (2025) [11] | Isabelle/HOL | No free lunch theorem |
| (**ours**) | Lean | Generalization error bound by Rademacher complexity and Dudley's entropy integral |
| Zhang et al. (2026) [24] | Lean | SLT with Gaussian Lipschitz concentration and Dudley's entropy integral |

**Table 2** Formalization of Concentration Inequalities.

| | |
|---|---|
| Markov/Chebyshev | Lean (Mathlib), Rocq (MathComp-Analysis [1], IBM/FormalML [20]), Isabelle/HOL (HOL-Probability) |
| Weak/strong laws of large numbers | Isabelle/HOL (AFP) [10] |
| Azuma–Hoeffding | Lean (Mathlib) |
| McDiarmid | (**ours**), Isabelle/HOL (AFP) [12] |
| Gaussian Lipschitz | Lean [24] |

## Finite Hypothesis Classes

Bagnall & Stewart (2019) [4] proved a general PAC bound in *Rocq* for any finite hypothesis class using Hoeffding's inequality. Essentially, if $\mathcal{F}$ is finite, with probability $1-\delta$ the true error is within $\varepsilon$ of the training error for $n \gtrsim \frac{1}{\varepsilon^2}(\ln|\mathcal{F}| + \ln\frac{1}{\delta})$. Their Rocq development (part of the *MLCERT* system) used a union bound and Chernoff/Hoeffding bounds to link training and test errors. This formal result was applied to certify small neural network models' performance. However, it was limited to a finite set $\mathcal{F}$, excluding most modern model classes because $\mathcal{F}$ with finite-dimensional parameters is an infinite set (so $|\mathcal{F}| = \infty$).

## Decision Stump Class ($VC = 1$)

Tassarotti et al. (2021) [17] gave a full *Lean 3* proof that the concept class of *decision stumps* (threshold classifiers in $\mathbb{R}$) is PAC-learnable. This is a classic textbook example with VC dimension 1. The formal proof uncovered subtle measure-theoretic issues that are glossed over in informal proofs. For instance, textbooks often assume measurability of argmax operations on sample data without proof – the formalization had to rigorously prove measurability and proper probability space definitions for the learning algorithm. The authors structured the proof to separate combinatorial reasoning about the algorithm's behavior from the analytic

reasoning about probabilities. They employed the Giry monad (in Lean's category theory library) to handle distributions, and ultimately derived the standard PAC guarantee for decision stumps.

### Verified Tail Bounds for Randomized Programs

Tassarotti and Harper (2018) [16] mechanized Karp's theorem and extensions for randomized divide-and-conquer recurrences in Coq. Their case studies include sequential and parallel QuickSort, random binary-search-tree height, and randomized leader election, with concrete numerical tail bounds. Although this work is about randomized algorithms rather than statistical learning theory, it is related methodologically: it shows how nontrivial tail-bound arguments can be packaged into reusable theorem-proving infrastructure, albeit in a mainly discrete-probability setting rather than the measure-theoretic empirical-process setting used here.

## 8.2 Formalizing Machine Learning Theory

Statistical learning theory extends beyond classical generalization bounds.

### Expressiveness of Deep Neural Networks

Bentkamp et al. (2016, 2019) [6, 7] formalized in *Isabelle/HOL* a theorem that deep networks can represent certain functions exponentially more efficiently than shallow ones. Their formalization simplified and generalized the original proof within Isabelle's logic. To support the proof, Bentkamp developed libraries for linear algebra (matrix ranks), multivariate polynomials, and even Lebesgue measure integration. The result is not about generalization error, but rather about the capacity/representation power of deep vs. shallow networks – nevertheless, it showcases the application of proof assistants to core theoretical ML questions. It also enriched Isabelle's libraries with notions like tensor products and rank, which are useful in learning theory.

### Convergence of Reinforcement Learning

Vajjha et al. (2021) [20] verified the convergence of value iteration and policy iteration for Markov Decision Processes in *Rocq*. In a follow-up, they formalized a stochastic approximation theorem [21] useful for analyzing RL algorithms. These efforts, under the IBM FormalML project, build a bridge between learning theory and formal verification by proving probabilistic convergence properties in Rocq. They required heavy use of Rocq's analysis libraries and bespoke techniques (e.g. coinduction for probabilistic processes).

### Statistical Learning Theory Based on Empirical Processes

In early 2026, Zhang et al. [24] open-sourced an AI-assisted formalization of statistical learning theory in Lean 4, focusing on empirical process theory. They describe their generated development as being built from scratch. At the level of initial seeds, a small amount of existing Lean code from developments such as `lean-rademacher`[2] and `CLT`[3] was also present. Given the scale of the generated library and its different goals, this is nevertheless best

[2] `https://github.com/auto-res/lean-rademacher.git`
[3] `https://github.com/RemyDegenne/CLT.git`

viewed as an independent, large-scale development. Their system automatically generated a large number of definitions and proofs including Gaussian Lipschitz concentration and Dudley entropy integral for sub-Gaussian processes.

## 8.3 Concentration Inequalities and Background Formalizations

Many learning-theoretic proofs rely on *concentration of measure* results and related probabilistic inequalities. Over the 2010s–2020s, these foundational results have been increasingly formalized, often as prerequisites for the theorems above:

### Isabelle/HOL

Its probability theory library (*HOL-Probability*) already included basic inequalities like *Markov's inequality*, *Chebyshev's inequality*, and exponential tail (Chernoff/Hoeffding) bounds by the late 2010s. In 2023, Karayel and Tan [12] contributed an AFP entry *"Concentration Inequalities"* which adds more advanced results. This includes *Bennett's and Bernstein's inequalities* (for sub-exponential random variables), *Efron–Stein's inequality* (variance bound via variance decomposition), *McDiarmid's inequality* (bounded differences), and the *Paley–Zygmund inequality*. Isabelle also has background limit theorems relevant to statistical reasoning: Eberl's AFP entry on the laws of large numbers [10] proves the strong law for i.i.d. random variables and derives the weak law from it, using ergodic-theoretic infrastructure. Thus Isabelle/HOL provides a rich library of probability results. For the present Lean development, however, the goal was not merely to reproduce McDiarmid's inequality: the Rademacher-complexity pipeline also requires product-measure i.i.d. samples, measurability of uniform deviations, the countable-to-separable supremum reduction, and links to Mathlib's integration and topology APIs. The Isabelle results therefore served primarily as external sanity checks for the shape of statements, while the missing Lean-side infrastructure still had to be developed.

### Lean

Lean's Mathlib library [19] gained basic measure theory around 2019, including Lebesgue integration and statistical independence. By mid-2020s, Mathlib had formal proofs of *Markov and Chebyshev inequalities*. This study also introduced *Hoeffding's lemma* (a result that a bounded zero-mean variable is sub-Gaussian) and then proved *Hoeffding's inequality* as a corollary of Chernoff bound techniques. It also implemented *McDiarmid's inequality* from scratch in Lean 4. In 2025, Degenne implemented *Azuma–Hoeffding* on Mathlib independently of our work. In 2026, Zhang et al. [24] implemented *Dudley's entropy integral* (with other concentration inequalities such as *Gaussian Lipschitz*) independently of our work.

### Rocq

Historically, Rocq's standard library did not include measure-theoretic concentration results. But developments like *MathComp-Analysis* [1] have added some pieces. Using these, the IBM FormalML team [20, 21] have formalized basic inequalities such as Markov and Chebyshev in Rocq. Hoeffding's and McDiarmid's inequalities were not present in Rocq as of early 2020s except in special-case proofs (e.g. MLCert implicitly used a form of Hoeffding's bound for finite samples). Tassarotti and Harper's Coq development [16] provides another related point in the design space: it verifies tail bounds for randomized programs using discrete probability and reusable recurrence theorems. In 2025, Affeldt et al. [2] formalized concentration inequalities in the Rocq proof assistant, developing infrastructure for measure-theoretic reasoning and exponential bounds. Their work includes Hoeffding's and Bernstein's inequalities.

## 9 Conclusion

We presented a Lean 4 formalization of generalization analysis via Rademacher complexity, built on Mathlib's measure-theoretic probability infrastructure. At the core is a mechanically checked pipeline that mirrors standard textbook proofs: (i) formal definitions of empirical and expected Rademacher complexity and of the uniform deviation functional, (ii) a symmetrization theorem that turns uniform deviations into sign-averaged suprema, and (iii) a bounded-differences argument combined with a fully formal McDiarmid inequality to obtain high-probability generalization bounds. A major methodological contribution is the systematic treatment of measurability obstacles: results are first established for countable hypothesis classes (where measurability of suprema is readily available), and then lifted to separable topological parameter spaces through a countable dense reduction, together with continuity-under-integration lemmas. This does not change the classical learning-theory bounds, but it clarifies a reusable set of sufficient conditions for the uncountable suprema that are routinely used in empirical-process arguments.

Beyond the abstract theorem, we demonstrated that this infrastructure supports nontrivial specializations. We formalized classical empirical Rademacher bounds for linear predictors under $\ell_2$ and $\ell_1$ norm constraints, and we mechanized a Dudley entropy integral bound using covering numbers and a chaining construction. These examples illustrate that once the general Rademacher pipeline is established in a proof assistant, many concrete bounds become "executable corollaries" obtained by instantiating the abstract objects and discharging model-specific analytic estimates.

Looking forward, we expect several extensions to be both practically useful and methodologically interesting: incorporating standard contraction inequalities for Lipschitz losses, connecting Rademacher complexity to Gaussian complexity and sub-Gaussian process tools, expanding libraries for covering-number estimates for common model classes, and improving automation for routine measurability and integrability obligations. More broadly, we hope this development contributes to a foundation in which modern learning-theoretic proofs can be stated and verified at a faithful, measure-theoretic level, enabling trustworthy reuse in formal verification and certified ML pipelines.

## References


1 Reynald Affeldt, Alessandro Bruni, Yves Bertot, Cyril Cohen, Marie Kerjean, Assia Mahboubi, Damien Rouhling, Pierre Roux, Kazuhiko Sakaguchi, Zachary Stone, Pierre-Yves Strub, and Laurent Théry. Mathcomp-analysis: Analysis library compatible with mathematical components, 2025. URL: `https://github.com/math-comp/analysis`.

2 Reynald Affeldt, Alessandro Bruni, Cyril Cohen, Pierre Roux, and Takafumi Saikawa. Formalizing concentration inequalities in Rocq: Infrastructure and automation. In Yannick Forster and Chantal Keller, editors, *16th International Conference on Interactive Theorem Proving (ITP 2025)*, volume 352, pages 21:1–21:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. `doi:10.4230/LIPIcs.ITP.2025.21`.

3 Francis Bach. *Learning Theory from First Principles*. MIT Press, 2024. URL: `https://mitpress.mit.edu/9780262049443/learning-theory-from-first-principles/`.

4 Alexander Bagnall and Gordon Stewart. Certifying the True Error: Machine Learning in Coq with Verified Generalization Guarantees. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):2662–2669, 2019. `doi:10.1609/AAAI.V33I01.33012662`.

5 Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 3:463–482, 2002. URL: `http://www.jmlr.org/papers/volume3/bartlett02a/bartlett02a.pdf`.

6 Alexander Bentkamp. Expressiveness of deep learning. *Archive of Formal Proofs*, 2016. URL: `https://isa-afp.org/entries/Deep_Learning.html`.

7 Alexander Bentkamp, Jasmin Christian Blanchette, and Dietrich Klakow. A formal proof of the expressiveness of deep learning. *Journal of Automated Reasoning*, 63(2):347–368, 2019. doi:10.1007/s10817-018-9481-5.

8 Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In *International Conference on Automated Deduction, CADE 28*, pages 625–635. Springer International Publishing, 2021. doi:10.1007/978-3-030-79876-5_37.

9 R M Dudley. The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1:290–330, 1967. doi:10.1016/0022-1236(67)90017-1.

10 Manuel Eberl. The laws of large numbers. *Archive of Formal Proofs*, February 2021. URL: https://isa-afp.org/entries/Laws_of_Large_Numbers.html.

11 Michikazu Hirata. No-free-lunch theorem for machine learning. *Archive of Formal Proofs*, 2025. URL: https://isa-afp.org/entries/No_Free_Lunch_ML.html.

12 Emin Karayel and Yong Kiam Tan. Concentration Inequalities. *Archive of Formal Proofs*, 2023. URL: https://www.isa-afp.org/entries/Concentration_Inequalities.html.

13 Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019. doi:10.1017/9781108627771.

14 Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning series. MIT Press, second edition, 2018. URL: https://mitpress.mit.edu/9780262039406/foundations-of-machine-learning/.

15 Sho Sonoda, Kazumi Kasaura, Yuma Mizuno, Kei Tsukamoto, and Naoto Onda. auto-res/lean-rademacher. Software, swhId: swh:1:dir:06614ca069f4562b630fca2903e3b286a426e317 (visited on 2026-07-13). URL: https://github.com/auto-res/lean-rademacher/tree/f34ab4f0a9029682f1af3179a3fe9b5e114e511f, doi:10.4230/artifacts.27120.

16 Joseph Tassarotti and Robert Harper. Verified tail bounds for randomized programs. In Jeremy Avigad and Assia Mahboubi, editors, *Interactive Theorem Proving (ITP2018)*, pages 560–578. Springer International Publishing, 2018. doi:10.1007/978-3-319-94821-8_33.

17 Joseph Tassarotti, Koundinya Vajjha, Anindya Banerjee, and Jean-Baptiste Tristan. A Formal Proof of PAC Learnability for Decision Stumps. In *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2021*, pages 5–17. Association for Computing Machinery, 2021. doi:10.1145/3437992.3439917.

18 Matus Telgarsky. Deep learning theory lecture notes, 2021. URL: https://mjt.cs.illinois.edu/dlt/.

19 The mathlib community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020*, pages 367–381, 2020. URL: http://doi.org/10.1145/3372885.3373824.

20 Koundinya Vajjha, Avraham Shinnar, Barry Trager, Vasily Pestun, and Nathan Fulton. CertRL: formalizing convergence proofs for value and policy iteration in Coq. In *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2021*, pages 18–31. Association for Computing Machinery, 2021. doi:10.1145/3437992.3439927.

21 Koundinya Vajjha, Barry Trager, Avraham Shinnar, and Vasily Pestun. Formalization of a Stochastic Approximation Theorem. In *13th International Conference on Interactive Theorem Proving, ITP 2022*, volume 237 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITP.2022.31.

22 L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. doi:10.1145/1968.1972.

23 Vladimir N Vapnik and A. 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. URL: http://doi.org/10.1137/1116025.

24 Yuanhe Zhang, Jason D Lee, and Fanghui Liu. Statistical learning theory in lean 4: Empirical processes from scratch. *arXiv preprint: 2602.02285*, 2026. doi:10.48550/arXiv.2602.02285.