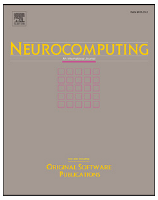
Neurocomputing 674 (2026) 132826

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



# Reconciling grokking with statistical learning theory through the lens of norm- and stability-based generalization bounds

Luca Oneto[a,*], Sandro Ridella[a], Simone Minisi[a], Andrea Coraddu[b], Davide Anguita[a]

[a] University of Genoa, Genova, Italy
[b] Delft University of Technology, Delft, Netherlands

## HIGHLIGHTS

- Emerging phenomena in ML challenge current theoretical frameworks.
- Grokking: sudden learning surge after stagnation or decline.
- Insights via norm- and stability-based generalization bounds.
- Theories reconciled with practice through concrete examples.

## ARTICLE INFO

*Keywords:*
Machine learning
Deep learning
Grokking
Statistical learning theory
Norm-based generalization bounds
Stability-based generalization bounds
Empirical evidence

## ABSTRACT

In recent years, Artificial Intelligence, particularly Machine Learning, has achieved remarkable success in solving complex problems. However, this progress has also revealed the emergence of unexpected, poorly understood, and elusive phenomena that characterize the behavior of machine intelligence and learning processes. These phenomena often challenge researchers to interpret them within the boundaries of existing Machine Learning theoretical frameworks, thereby motivating the development of new and more comprehensive theoretical foundations. One such phenomenon, known as *grokking*, refers to the sudden and substantial improvement in a model's performance following a prolonged period of stagnant or even regressive learning. In this paper, we argue that it is possible to provide insights into grokking by leveraging the existing theoretical foundations of Machine Learning, in particular concepts from Statistical Learning Theory, such as norm-based and stability-based generalization bounds. We further show how these theories can help reconcile the phenomenon of grokking with established principles of learning and generalization. Furthermore, we demonstrate the practical applicability of these insights through concrete examples.

## 1. Introduction

In recent years, Artificial Intelligence (AI), particularly Machine Learning (ML), has profoundly transformed society, industry, and science. AI systems have demonstrated remarkable capabilities across a wide spectrum of domains: from solving well-defined challenges such as mastering board games [1] and predicting protein structures [2], to enabling broad, open-ended tasks like multimodal conversational agents [3]. More recently, the emergence of agentic AI systems [4], capable of autonomously pursuing complex goals, has highlighted the accelerating potential of this technology. The capabilities of this new generation of intelligent machines increasingly appear boundless.

Nevertheless, all these systems rely on a common underlying principle: transforming a problem into a series of prediction tasks. In certain domains, this formulation becomes very explicit, for instance, in protein folding, the task consists of predicting a protein's three-dimensional structure from its amino acid sequence [2]. In other domains, the predictive framing is more subtle. For example, in text generation, the model predicts the next word (or, more precisely, the next token) [5], while in image generation, such as with diffusion models, the task involves predicting how to iteratively denoise an image to recover a coherent visual representation [6].

Despite significant advancements in our ability to learn predictive models from data, we are increasingly witnessing the emergence of

[*] Corresponding author.
*Email addresses:* luca.oneto@unige.it (L. Oneto), sandro.ridella@unige.it (S. Ridella), simone.minisi@unige.it (S. Minisi), a.coraddu@tudelft.nl (A. Coraddu), davide.anguita@unige.it (D. Anguita).

https://doi.org/10.1016/j.neucom.2026.132826
Received 11 September 2025; Received in revised form 18 December 2025; Accepted 20 January 2026
Available online 28 January 2026

unexpected, poorly understood, and elusive phenomena that characterize the behavior of machine intelligence and learning processes. These include *catastrophic forgetting* [7–11], where models lose previously acquired knowledge when exposed to new data during fine-tuning; *mode collapse* [12–15], where generative models fail to capture the full diversity of the underlying data distribution; *shortcut learning* [16–18], where models rely on superficial correlations rather than learning deeper insights; *spurious correlations* [19–21], where models learn associations that do not generalize due to non-causal statistical dependencies in the data; *double descent* [22–28], where performance initially improves during learning, then temporarily deteriorates before improving again; *benign overfitting* [29–35], where models overfit the training data without harming or even improving generalization; *over-parameterization* [24, 36–38], where model performance continues to improve despite excessive complexity; *physical implausibility* [39–41], where models generate accurate predictions without capturing the underlying physics; *bias amplification* [42–44], where models reinforce historical societal biases present in the data; *adversarial vulnerability* [42,45,46], where models are highly susceptible to targeted manipulations of input data; *lack of explainability* [42,47–50], where model decisions and internals remain opaque and difficult to interpret; and *privacy violations* [42,51,52], where models may inadvertently leak sensitive information from the training data. These phenomena represent some of the most pressing and complex challenges facing modern ML systems. Moreover, there is a significant interplay between these issues [13,21,24,42,53–56].

In this work, we explore the phenomenon of *grokking*, where during learning, following a prolonged phase of stagnation or even regression, a model suddenly experiences a rapid and substantial improvement in task performance [57]. This abrupt shift resembles the moment when a person achieves a breakthrough in understanding after struggling with a concept. What makes grokking particularly fascinating is its divergence from the traditional expectations of statistical learning, which generally follows a pattern of steady, incremental progress [58–63]. Instead, grokking suggests a dynamic in which models may initially show little or no improvement, or even a decline in performance, before unexpectedly surging in capability.

Originally observed in supervised ML on algorithmic datasets [57, 64], this phenomenon has since been identified in real-world data [65–67] and other learning contexts [68,69]. This unpredictability has posed significant challenges for researchers attempting to explain grokking within existing ML theoretical frameworks, spurring the search for more sophisticated interpretations and explanations [70–76].

These phenomena often challenge researchers to gain insights into them within the boundaries of existing Statistical Learning Theories (SLTs), thereby motivating the development of new and more comprehensive theoretical foundations in Machine Learning. SLT enables the study of the generalization abilities of predictive models. Current SLT frameworks can be broadly categorized into six main families [60,63]. When the space of functions can be explicitly or implicitly defined [77], complexity-based theories are applicable. These include the Vapnik-Chervonenkis theory [61] and Rademacher complexity theory [78]. From these, norm-based generalization bounds are derived [79–81], which account for fully connected networks [78,79,82–90], convolutional networks [80,91–94], and attention-based networks [81,95–98]. When the learning algorithm tends to significantly compress the original dataset, compression-based theories become more suitable. Examples include the compression bound [99] and the minimum description length principle [100]. If the function space cannot be explicitly defined or the algorithm does not compress the data, algorithmic stability theory becomes relevant. This encompasses notions such as uniform stability and hypothesis stability [101,102]. When dealing with randomized models, PAC-Bayes theory offers a powerful analytical tool [103]. In scenarios where the algorithms themselves are randomized, differential privacy theory can be employed to derive generalization guarantees [104]. Finally, when learning algorithms are adaptive, only information-theoretic approaches are suitable for analysis [63].

In this paper, we argue that grokking can be interpreted within the existing theoretical foundations of ML by leveraging concepts from SLT, specifically norm-based and stability-based generalization bounds. We provide insights into how these frameworks can help reconcile grokking with established principles of learning and generalization. Furthermore, we demonstrate the practical relevance of these insights through concrete examples.

The remainder of the paper is organized as follows. Section 2 introduces the preliminary concepts necessary to understand our work. Section 3 presents the proposed theoretical analysis. Section 4 provides the empirical evidence supporting our theoretical findings. Finally, Section 5 concludes the paper.

## 2. Preliminaries

Let us consider the supervised learning setting [58,59]. Given a random observation $X \in \mathcal{X}$, the objective is to estimate the corresponding output $Y \in \mathcal{Y}$. Observations are drawn according to an unknown distribution $\mu$ over the space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. To approximate the conditional distribution $\mathbb{P}\{Y \mid X\}$, we aim to select a function $f : \mathcal{X} \to \mathcal{Y}$ from a (possibly unknown) set of candidate functions $\mathcal{F}$. This is accomplished through a learning algorithm $\mathscr{A}_{\mathcal{H}} : \mathcal{Z}^n \to \mathcal{F}$, parameterized by a set of hyperparameters $\mathcal{H}$. Given a labeled dataset of $n$ samples, $\mathcal{D}_n = \left\{(X_1, Y_1), \dots, (X_n, Y_n)\right\} = \left\{Z_1, \dots, Z_n\right\} \in \mathcal{Z}^n$, drawn i.i.d. from $\mu$, the algorithm outputs a hypothesis $\hat{f} = \mathscr{A}_{\mathcal{H}}(\mathcal{D}_n) \in \mathcal{F}$. The generalization error of $\hat{f}$, which quantifies its performance in approximating $\mathbb{P}\{Y \mid X\}$, is defined as

$$\mathsf{L}(\hat{f}) = \mathbb{E}_Z\{\ell(\hat{f}, Z)\}, \tag{1}$$

where $\ell : \mathcal{F} \times \mathcal{Z} \to [0, 1]$ is a loss function that evaluates the accuracy of the prediction on individual data points. Since the true generalization error $\mathsf{L}(\hat{f})$ is unknown, it is commonly approximated by the empirical error, defined as

$$\hat{\mathsf{L}}(\hat{f}) = \hat{\mathsf{L}}_{\mathrm{emp}}(\hat{f}) = \frac{1}{n} \sum_{Z \in \mathcal{D}_n} \ell(\hat{f}, Z). \tag{2}$$

Ideally, a learning algorithm should return the so-called oracle predictor, defined as

$$f^* = \arg\min_{\mathcal{M}} \mathsf{L}(f), \tag{3}$$

which minimizes the generalization error over the space of all possible models $\mathcal{M}$, i.e., the set of all measurable functions. However, since the true risk $\mathsf{L}(f)$ is unknown, we rely on the algorithm $\mathscr{A}_{\mathcal{H}}$, which effectively performs the following procedure:

$$\hat{f} = \mathscr{A}_{\mathcal{H}}(\mathcal{D}_n) = \arg\widetilde{\min}_{f \in \mathcal{F}} \hat{\mathsf{L}}(f). \tag{4}$$

In other words, we restrict our attention to a subset $\mathcal{F} \subset \mathcal{M}$, use $\hat{\mathsf{L}}(f)$ as an empirical estimator of the true risk $\mathsf{L}(f)$, and aim to find the function in $\mathcal{F}$ that minimizes $\hat{\mathsf{L}}(f)$ through a practical (possibly approximate) procedure, denoted by $\widetilde{\min}$. An alternative to the empirical error as a performance estimator is the leave-one-out error

$$\mathsf{L}_{\mathrm{loo}}(\hat{f}) = \frac{1}{n} \sum_{Z \in \mathcal{D}_n} \ell(\mathscr{A}_{\mathcal{H}}(\mathcal{D}_n \setminus Z), Z), \tag{5}$$

which computes the average prediction error on individual samples from $\mathcal{D}_n$, each of which is excluded once during training. The hypothesis class $\mathcal{F}$ is determined by the choice of algorithm $\mathscr{A}_{\mathcal{H}}$ and its associated hyperparameters. This includes the functional form of $f$ (e.g., linear, tree-based, ensemble-based, convolutional, or attention-based), as well as implicit or explicit regularization mechanisms (e.g., weight norms, dropout, early stopping, or over-parameterization), and the corresponding regularization strength [24,58,59]. The notation $\widetilde{\min}$ highlights that, given $\mathcal{F}$ and $\hat{\mathsf{L}}(f)$, exactly minimizing the empirical risk over $\mathcal{F}$ may be computationally infeasible. Therefore, in practice, we resort to approximate optimization procedures (e.g., gradient-based methods or greedy algorithms) to solve this minimization problem [58,59].

Typically, the model is expressed in the form

$$f(X) = t_\tau(r_\rho(X)), \tag{6}$$

where $r : \mathcal{X} \to \mathbb{R}^d$ is a representation function, $t : \mathbb{R}^d \to \mathcal{Y}$ is a task-specific function, and $\tau$ and $\rho$ denote their respective parameters. This decomposition allows us to rewrite Problem (4) as

$$\hat{\tau}, \hat{\rho} = \arg \widetilde{\min}_{\tau,\rho\in\mathcal{F}} \tilde{\mathsf{C}}(\tau, \rho), \tag{7}$$

where $\mathcal{F}$ denotes the (possibly implicit) parameter search space [58,59], $\tilde{\mathsf{C}}(\tau, \rho)$ is a practical objective function that incorporates the empirical error, or a convex or differentiable surrogate thereof, potentially augmented with regularization terms, and we optimize this using a practical algorithm denoted by $\widetilde{\min}$ [58,59]. For further details, please refer to Section 2.1.

Note that, in the absence of side information, the *no-free-lunch* theorem [105] states that it is impossible to identify the best learning algorithm $\mathscr{A}_{\mathcal{H}}$ a priori. As a result, the only practical strategy is to define a set of candidate algorithms and search for the most suitable one. This search can be carried out using simple grid search or more sophisticated techniques [106–111], guided by theoretical guarantees or empirical performance metrics [60,109]. While the gold standard in this setting is to employ resampling methods [60], these approaches are not without limitations, such as overfitting and overvalidation [112,113]. Moreover, they offer little insight into the underlying learning process. For this reason, theoretical tools are necessary to analyze and understand the learning behavior of a given algorithm [60,63]. One common approach is to derive bounds on the generalization ability of the learned model [60,63]. In fact, in the setting we just described, it can be shown that

$$\mathbb{P}\{\mathsf{L}(\hat{f}) \leq \hat{\mathsf{L}}_*(\hat{f}) + \mathsf{M}(\mathscr{A}_{\mathcal{H}}) + \Delta(n, \delta)\} \geq 1 - \delta, \tag{8}$$

indicating that the generalization error of $\hat{f}$ is bounded by an empirical estimate, denoted $\mathsf{L}_*$ with $* \in \{\hat{\mathsf{L}}_{\text{emp}}, \hat{\mathsf{L}}_{\text{loo}}\}$, along with two additional terms. The term $\mathsf{M}(\mathscr{A}_{\mathcal{H}})$ reflects the complexity or risk introduced by the choice of algorithm and its hyperparameters. This quantity tends to increase when the algorithm overfits the data, prioritizing memorization over generalization. The second term,[1] denoted $\Delta(n, \delta)$, captures the confidence level of the bound and depends on the sample size $n$ and a user-defined confidence parameter $\delta$. It increases as $n$ decreases or as stronger guarantees (smaller $\delta$) are desired. Statistical Learning Theory (SLT) proposes six major families of techniques to derive bounds of the form of Eq. (8) [60,63]:

- *Complexity*-based approaches (e.g., Vapnik-Chervonenkis theory [61], Rademacher complexity [78], and norm-based generalization bounds [79–81]) are applicable when $\mathcal{F}$ is explicitly or implicitly defined [77].
- *Compression*-based approaches (e.g., compression bounds [99] and the Minimum Description Length principle [100]) are useful when the algorithm $\mathscr{A}_{\mathcal{H}}$ effectively compresses the training dataset $\mathcal{D}_n$.
- *Stability*-based approaches (e.g., uniform stability and hypothesis stability [101,102]) are relevant when $\mathcal{F}$ is not explicitly defined or the algorithm does not perform data compression.
- *PAC-Bayes* theory is suited for randomized predictors $f$ [103].
- *Differential privacy*-based approaches provide generalization guarantees for randomized algorithms that preserve privacy [104].
- *Information-theoretic* approaches are applicable when $\mathscr{A}_{\mathcal{H}}$ is adaptive [63].

In this work, we focus on norm-based and stability-based approaches, as they are broadly applicable and rely on relatively mild assumptions. In particular, they are well-suited for analyzing the most commonly used algorithms, which, according to the discussion in Section 2.1, naturally satisfy their foundational hypotheses. For further details, see Section 2.2.

### 2.1. Learning algorithms

Let us consider the general case where $\mathcal{Y} \subseteq \mathbb{R}^p$ or $\mathcal{Y} \subseteq \{0, 1\}^p$. In this section, we recall a simple yet general framework that captures both shallow and deep ML algorithms.

For shallow ML models [58], the model can be expressed as

$$f(X) = t_\tau(r_\rho(X)) = W r(X), \tag{9}$$

where $\tau = W \in \mathbb{R}^{p\times d}$, so that $t_\tau$ denotes a linear transformation, and $r(X)$ is a fixed feature representation. This representation may be obtained either explicitly (e.g., through manual feature engineering [114]) or implicitly (e.g., via kernel methods [115]). Importantly, $r(X)$ does not depend on learnable parameters $\rho$. The learning objective is then defined as

$$\tilde{\mathsf{C}}(\tau) = \tilde{\mathsf{L}}(W) + \tilde{\mathsf{R}}(W), \tag{10}$$

where $\tilde{\mathsf{L}}(W)$ is a convex (or at least differentiable) surrogate of the empirical loss $\hat{\mathsf{L}}(f)$, and $\tilde{\mathsf{R}}(W)$ is a regularization term, typically involving a norm of $W$ [58]. In some cases, $\tilde{\mathsf{R}}(W)$ also accounts for additional quantities that allow addressing more complex challenges, such as trustworthiness [116]. As for the $\widetilde{\min}$ operator, when $\tilde{\mathsf{C}}(\tau)$ is convex, the main concern is computational efficiency or convergence speed. When $\tilde{\mathsf{C}}(\tau)$ is only differentiable, gradient-based methods are employed [58]. This simple formulation encompasses a wide range of classical ML models [58], including Ridge Regression, Support Vector Machines, and Random Forests.

In the case of deep ML models [59], the model can be expressed as a composition of functions called layers $L$, which we formalize as

$$f = t_\tau \circ r_\rho, \quad t_\tau = \bigcirc_{i=1}^{L^t} t^{(i)}_{\tau^{(i)}}, \quad r_\rho = \bigcirc_{i=1}^{L^r} r^{(i)}_{\rho^{(i)}}, \tag{11}$$

where $L = L^t + L^r$, $\tau = \{\tau^{(1)}, \dots, \tau^{(L^t)}\}$, and $\rho = \{\rho^{(1)}, \dots, \rho^{(L^r)}\}$. The first $L^t$ layers are the task-specific layers, while the last $L^r$ layers are the representation layers. There are multiple types of layers. Some are learnable (parametrized), such as:

- *Fully Connected*: This layer operates on vectors. Given an input vector $X$, a learnable matrix $W$, and a learnable bias vector $B$, the output is computed as:

  $$WX + B. \tag{12}$$

  This extends shallow ML by learning complex mappings through compositions of linear transformations.
- *Convolutional*: This layer operates on structured data (e.g., images, sequences, or graphs). Given an input tensor $X$ and a learnable kernel $K$, the output is computed via a sliding window operation:

  $$\sum_{j\in\mathcal{J}} K_j \cdot X_{i+j}, \tag{13}$$

  where $\mathcal{J}$ denotes the receptive field of the convolution. This allows the model to learn local patterns and translation-invariant features.
- *Attention*: Attention layer computes a weighted sum of values based on the similarity between queries and keys. Given query $Q$, key $K$, and value $V$ matrices, the attention mechanism is defined as:

  $$\texttt{Attention}(Q, K, V) = \texttt{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{14}$$

[1] We will not delve into this term, as it is independent of $\mathscr{A}_{\mathcal{H}}$.

where $d_k$ is the dimensionality of the keys. Multi-head attention extends the attention mechanism by computing multiple attention outputs in parallel (called heads), each with different learned projections. Formally:

$$\texttt{MultiHead}(Q, K, V) = \texttt{Concat}(\texttt{head}_1, \dots, \texttt{head}_h) W^O, \tag{15}$$

where each head is computed as

$$\texttt{head}_i = \texttt{Attention}(Q W_i^Q, K W_i^K, V W_i^V), \tag{16}$$

and $W^O$, $W_i^Q$, $W_i^K$, $W_i^V$ are learnable projection matrices.

Other layers are non-learnable, such as activation functions (e.g., linear, ReLU, and softmax), residual connections, max pooling, batch normalization, and dropout [59]. Nevertheless, these non-learnable layers implicitly influence the behavior of the learnable ones, thereby contributing indirectly to the overall function class $\mathcal{F}$ [59]. The learning objective is then defined as

$$\tilde{\mathsf{C}}(\breve{\tau}, \breve{\rho}) = \tilde{\mathsf{L}}(\breve{\tau}, \breve{\rho}) + \tilde{\mathsf{R}}(\breve{\tau}, \breve{\rho}), \tag{17}$$

where $\breve{\tau}$ and $\breve{\rho}$ denote either a subset or the entirety of the parameters $\tau$ and $\rho$, depending on whether the network is being fully trained, undergoing transfer learning, or fine-tuned [59]. $\tilde{\mathsf{L}}(\breve{\tau}, \breve{\rho})$ is a differentiable surrogate of the empirical loss $\hat{\mathsf{L}}(f)$ [59]. Finally, $\tilde{\mathsf{R}}(\breve{\tau}, \breve{\rho})$, unlike in shallow ML, is not an explicit regularization term (which is typically embedded in the model structure, e.g., via dropout, or in the optimization algorithm $\widetilde{\min}$). Instead, it often addresses more complex challenges, such as trustworthiness [59,116]. As for the $\widetilde{\min}$ operator, gradient-based methods such as Stochastic Gradient Descent or ADAM are employed [59].

### 2.2. Generalization bounds

The generalization bound in Eq. (8) [60,63] provides a probabilistic guarantee that the true risk $\mathsf{L}(\hat{f})$ of a learned model $\hat{f}$ is not much larger than its empirical estimator $\hat{\mathsf{L}}_*(\hat{f})$, plus a complexity or risk term $\mathsf{M}(\mathscr{A}_\mathcal{H})$ and a confidence term $\Delta(n, \delta)$ that depends on the sample size $n$ and confidence $\delta$.

The term $\mathsf{M}(\mathscr{A}_\mathcal{H})$ captures the properties of the learning algorithm $\mathscr{A}_\mathcal{H}$ operating over the possibly unknown class of functions $\mathcal{F}$, and controlling this term is essential for achieving good generalization. Two classical approaches to upper bounding $\mathsf{M}(\mathscr{A}_\mathcal{H})$ in the family of complexity-based methods are *Rademacher Complexity* (RC) [78] and *Covering Numbers* (CN) [117]. The RC $\hat{\mathsf{R}}_n(\mathcal{F})$ is defined as

$$\hat{\mathsf{RC}}(\mathcal{F}) = \mathbb{E}_{\sigma_1, \dots, \sigma_n} \left[ \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell(f, Z_i) \right], \tag{18}$$

where $\{\sigma_1, \dots, \sigma_n\}$ are i.i.d. Rademacher variables taking values in $\{-1, +1\}$ with uniform probability. The RC measures the ability of the hypothesis class to fit random noise and thus serves as a data-dependent complexity measure. Then, for a universal constant $c$, we have

$$\mathsf{M}(\mathscr{A}_\mathcal{H}) \leq c \hat{\mathsf{RC}}(\mathcal{F}). \tag{19}$$

Note that more advanced approaches, such as Local RC [118], exist, but these are beyond the scope of this paper. Alternatively, Covering Numbers (CN) provide a combinatorial method to quantify the richness of $\mathcal{F}$. Given a metric, the CN with respect to this metric $\mathsf{CN}(\epsilon, \mathcal{F})$ is the minimal number of balls of radius $\epsilon$ needed to cover $\mathcal{F}$. Localized versions also exist [119]. Dudley's entropy integral and chaining techniques [120] allow bounding RC in terms of these CN. For a universal constant $c$, we have

$$\hat{\mathsf{RC}}(\mathcal{F}) \leq c \inf_{\alpha \geq 0} \left( \alpha + \int_{\alpha}^{\infty} \sqrt{\frac{\log(\mathsf{CN}(\epsilon, \mathcal{F}))}{n}} d\epsilon \right). \tag{20}$$

This connection enables bounding $\mathsf{M}(\mathscr{A}_\mathcal{H})$ using empirical entropy conditions of the hypothesis class. However, both RC and CN require explicit knowledge of $\mathcal{F}$ in order to be computed.

Rather than directly characterizing the function space, an upper bound on the RC or CN can be provided by leveraging the norm of the model's weights [79–81]. This method is highly versatile, allowing for the derivation of bounds not only for shallow models but also for deep architectures, including fully connected, convolutional-based, and attention-based neural networks. By considering weight norms, we can derive generalization bounds for a broad range of ML models, irrespective of their depth or architecture, as long as the weight norms can be effectively controlled. Formally, it can be shown that

$$\mathsf{M}(\mathscr{A}_\mathcal{H}) \leq \mathsf{N}(\hat{\tau}, \hat{\rho}), \tag{21}$$

where $\mathsf{N}$ is a function of the norm of the parameters $\tau$ and $\rho$. Note that $\mathsf{N}$ also depends on the number of samples $n$, but for the purposes of this paper, this dependency is not fundamental, as in the case of *grokking*, where the dataset is kept fixed. The literature contains numerous works that explore this approach for fully connected networks [78,79,82–90], convolutional-based networks [80,91–94], and attention-based networks [81,95–98].

Algorithmic Stability has many variants [60,101,101,102,121–126], e.g., uniform stability, Hypothesis Stability (HS), cross-validation or leave-one-out stability, error stability, and pointwise HS. Nevertheless, HS has emerged as the most powerful and insightful for understanding intricate behaviors in learning models [24,102,127,128]. Although HS generally induces looser bounds compared to uniform stability [101,121,122], it can be estimated directly from the data and strongly depends on the properties of the underlying learning algorithm [24,101,122,124,126]. HS theory can be used to derive a bound of the form in Eq. (8), implying that for a universal constant $c$:

$$\mathsf{M}(\mathscr{A}_\mathcal{H}) \leq c\beta, \tag{22}$$

where $\beta$ can be defined in two distinct ways [101,129,130]:

$$\mathbb{E}_{\mathcal{D}_n, Z'} \left| \ell \left( \mathscr{A}_\mathcal{H}(\mathcal{D}_n), Z_i \right) - \ell \left( \mathscr{A}_\mathcal{H}(\mathcal{D}_n \setminus Z_i \cup Z'), Z_i \right) \right| \leq \beta, \tag{23}$$

$$\mathbb{E}_{\mathcal{D}_n, Z'} \left| \ell \left( \mathscr{A}_\mathcal{H}(\mathcal{D}_n), Z' \right) - \ell \left( \mathscr{A}_\mathcal{H}(\mathcal{D}_n \setminus Z_i), Z' \right) \right| \leq \beta, \tag{24}$$

where $Z'$ is an example drawn from $\mu$. If $\ell$ is Lipschitz continuous with respect to a distance metric $\mathtt{d} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, one can write [127]:

$$\beta \propto \mathbb{E}_{\mathcal{D}_n, Z'} \mathtt{d} \left( \mathscr{A}_\mathcal{H}(\mathcal{D}_n)(X_i), \mathscr{A}_\mathcal{H}(\mathcal{D}_n \setminus Z_i \cup Z')(X_i) \right), \tag{25}$$

$$\beta \propto \mathbb{E}_{\mathcal{D}_n, Z'} \mathtt{d} \left( \mathscr{A}_\mathcal{H}(\mathcal{D}_n)(X'), \mathscr{A}_\mathcal{H}(\mathcal{D}_n \setminus Z_i)(X') \right). \tag{26}$$

These forms of HS can be estimated in a fully unsupervised manner once the models are trained. This feature is especially helpful when assessing the stability not only of the final end-to-end model but also of various intermediate representation layers in deep architectures. In practice, one can estimate these quantities via resampling techniques (e.g., Bootstrap [131] or Bag of Little Bootstrap [132] to reduce computational overhead) or via theoretical approaches [24,122], obtaining an estimate $\hat{\beta}$. Moreover, in specific settings, it is possible to derive bounds (rather than direct estimates) for HS [24,126].

## 3. Theoretical analysis

In this work, we investigate the phenomenon of *grokking*. When models are trained with gradient descent based methods on Problem (7), whether in the shallow case (Eq. 10) or the deep case (Eq. 17), they may display a prolonged phase of stagnation or even a decline in performance, followed by a sudden and substantial improvement in task accuracy as the number of training iterations increases [57]. What makes grokking particularly intriguing is its deviation from the classical expectations of statistical learning, which typically assume a pattern of steady, incremental improvement [58–63]. Instead, grokking reveals a dynamic where models may initially show little to no progress or even deteriorating performance before unexpectedly achieving high generalization.

In this section, we argue that grokking can be interpreted within the established theoretical framework of ML by drawing on concepts from SLT, specifically norm-based (Section 3.1) and stability-based (Section 3.2) generalization bounds. In particular, we offer insights into how these theoretical tools can help reconcile grokking with classical learning principles. Our core idea is straightforward: rather than solely focusing on training and validation/test errors, one should analyze the generalization bound of Eq. (8), and in particular the quantity $\mathtt{M}(\mathscr{A}_{\mathcal{H}})$, which may provide valuable information about the internal dynamics of the learning process. However, estimating $\mathtt{M}(\mathscr{A}_{\mathcal{H}})$ for real-world, complex architectures remains a significant challenge. Therefore, in Sections 3.1 and 3.2, we discuss how recent advances in norm- and stability-based analysis can be used to approximate $\mathtt{M}(\mathscr{A}_{\mathcal{H}})$ effectively.

### *3.1. Norm-based analysis*

In this section, we present the state-of-the-art norm-based generalization bounds for various classes of shallow and deep ML models. It is important to note that these bounds have evolved over time: early bounds were often loose and uninformative, while more recent ones provide meaningful and tight analyses [78–98].

For shallow ML models, let us consider the simplified case where $p = 1$, which can be extended to more complex scenarios. In this case, the learned model, found by solving Problem (7), takes the form

$$\hat{f}(X) = \hat{\boldsymbol{w}}^T r(X), \tag{27}$$

where $\hat{\boldsymbol{w}} \in \mathbb{R}^p$. Under mild assumptions [133], such as the smoothness of the loss function, it can be shown [78] that

$$\mathtt{N}(\hat{\boldsymbol{w}}) \propto \|\hat{\boldsymbol{w}}\|_2, \tag{28}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. Note that the dependency from other quantities depending on the data or the Lipschitz constant of the loss function, is not taken into account, as in *grokking* these are assumed to be fixed. Note also that the distribution of the data is implicitly captured by $\hat{f}$, since the solution $\hat{\boldsymbol{w}}$ to Problem (7) is obtained via minimization over the dataset. However, a limitation of the bound in Eq. (28) is that it only accounts for the functional form of the model, without considering the specific regularization employed by the learning algorithm (see Eq. (10)). For instance, in most common shallow ML algorithms [123,133], regularization takes the form

$$\tilde{\mathtt{R}}(\boldsymbol{w}) = \|M(\hat{\boldsymbol{w}} - \boldsymbol{c})\|, \tag{29}$$

for some matrix $M \in \mathbb{R}^{p\times p}$, vector $\boldsymbol{c} \in \mathbb{R}^p$, and a norm $\|\cdot\|$. In this setting, it can be proven [133] that

$$\mathtt{N}(\hat{\boldsymbol{w}}) \propto \|M(\hat{\boldsymbol{w}} - \boldsymbol{c})\|. \tag{30}$$

This refined bound takes into account both the structure of the model and the associated regularization, and represents one of the tightest analyses available in this setting.

For deep ML models, things can become significantly more complex. Depending on the chosen architecture, we need to rely on more recent and sophisticated results to obtain meaningful and tight generalization bounds. Here, we consider three powerful and widely used architectures:

- *Fully Connected Neural Network (FNN)*. We consider a composition of $L$ fully connected layers, each with weights $\hat{W}_i$ for $i \in \{1, \dots, L\}$ (bias terms are excluded), and Lipschitz activation functions [59]. In this setting, one of the most effective norm-based generalization bounds is due to [85], which shows that

$$\mathtt{N}(\hat{W}_1, \dots, \hat{W}_L) \propto \prod_{i=1}^{L} \|\hat{W}_i\|_2 \sqrt{\sum_{i=1}^{L} \frac{\|\hat{W}_i\|_F^2}{\|\hat{W}_i\|_2^2}}. \tag{31}$$

  Here, $\|\cdot\|_2$ and $\|\cdot\|_F$ denote the spectral norm and Frobenius norm, respectively. Note also that the dependency on other parameters, such as the Lipschitz constant of the activation function or the dimensionality of the layers, is not taken into account, as in *grokking* these are assumed to be fixed. Note also that, when performing fine tuning or transfer learning, meaning we learn $\hat{W}_i$ with $i \in \{1, \dots, L\}$ starting from $\hat{W}_i^0$ with $i \in \{1, \dots, L\}$ we can further improve [85] the result of Eq. (31)

$$\mathtt{N}(\hat{W}_1, \dots, \hat{W}_L) \propto \prod_{i=1}^{L} \|\hat{W}_i\|_2 \sqrt{\sum_{i=1}^{L} \frac{\|\hat{W}_i - \hat{W}_i^0\|_F^2}{\|\hat{W}_i\|_2^2}}. \tag{32}$$

  Several other bounds exist [78,79,82–90], but this result remains among the most meaningful and tightest bounds currently available [80,81,85]. Nevertheless, in this setting, one can state that the bounds of Eqs. (31) and (32) roughly scale as

$$\mathtt{N}(\hat{W}_1, \dots, \hat{W}_L) \propto \prod_{i=1}^{L} \|\hat{W}_i\|_F. \tag{33}$$

  Note also that this bound accounts for the addition of non-learnable layers such as residual connections and batch normalization.
- *Convolutional Neural Network (CNN)*. We consider a classical architecture for CNNs, composed of learnable layers—namely, convolutional and fully connected layers—interleaved with non-learnable ones. For the convolutional layers, we consider the standard case where, in layer $l$, there are $k^l$ kernels denoted by $\hat{w}_l^{k\in\{1,\dots,k^l\}}$, while for the fully connected layers, the weight matrices are denoted by $\hat{W}_l$ (bias terms are excluded). We assume all activation functions are Lipschitz continuous [59]. Convolutional layers (e.g., in image processing tasks) can be equivalently represented as fully connected layers via associated Toeplitz matrices $\hat{W}_{T,l}$. Consequently, one might apply the result of Eq. (33) to derive a norm-based generalization bound. Nevertheless, as analyzed in [80], this approach is suboptimal and typically leads to loose bounds. To address this, it was shown that, for convolutional layers, the term $\|\hat{W}_{T,l}\|_F$ in Eq. (33) can be replaced by $\max_{k\in\{1,\dots,k^l\}} \|\hat{w}_l^k\|_F$, resulting in significantly sharper and more meaningful bounds. In conclusion, consider a CNN with $L$ layers, where the set of convolutional layers is denoted by $\mathcal{L}^C \subseteq \{1, \dots, L\}$, and the set of fully connected layers is given by $\mathcal{L}^F = \{1, \dots, L\} \setminus \mathcal{L}^C$. Then, the norm-based complexity term can be approximated as

$$\mathtt{N}(\hat{W}_{T,l\in\mathcal{L}^C}, \hat{W}_{l\in\mathcal{L}^F}) \propto \prod_{l\in\mathcal{L}^C} \max_{k\in\{1,\dots,k^l\}} \|\hat{w}_l^k\|_F \prod_{l\in\mathcal{L}^F} \|\hat{W}_l\|_F. \tag{34}$$

  Note that, as with the FNN, there are additional dependencies that we did not take into account, since in *grokking* these are assumed to be fixed. Several other norm-based generalization bounds have been proposed [80,91–94], yet this refinement remains among the most interpretable and tightest currently available [80,81,85].
- *Transformer Neural Network (TNN)*. In a standard TNN, inputs are first passed through a non-learnable embedding layer for both elements and positions. These embeddings are then processed by a sequence of $L^T$ transformer decoder blocks. Each block, indexed by $b \in \{1, \dots, L^T\} = \mathcal{L}^T$, consists of three main components, each followed by residual connections and layer normalization:

  1. Multi-Head Attention, with parameters $\hat{W}_{i,b}^Q$, $\hat{W}_{i,b}^K$, and $\hat{W}_{i,b}^V$, where $i \in \{1, \dots, h^b\} = \mathcal{L}^{H_b}$ and $h^b$ is the number of attention heads in block $b$. The output projection matrix is denoted by $\hat{W}_b^O$;
  2. Two-Layer FNN with a ReLU or GELU activation in between, and parameters $\hat{W}_{1,b}^F$ and $\hat{W}_{2,b}^F$.

  Finally, a fully connected output layer with parameters $\hat{W}^F$ is applied to generate the final output. In this setting, we can leverage the results from [85] and [81], which provide the tightest generalization bounds currently available in the literature, to obtain the following norm-based bound:

$$\mathbb{N}(\hat{W}^F, \hat{W}^Q_{i,b}, \hat{W}^K_{i,b}, \hat{W}^V_{i,b}, \hat{W}^O_b, \hat{W}^F_{1,b}, \hat{W}^F_{2,b}, i \in \mathcal{L}^{H_b}, b \in \mathcal{L}^T)$$

$$\propto \|\hat{W}^F\|_F \prod_{b \in \mathcal{L}^T} \max_{i \in \mathcal{L}^H} \left[ \max \left\{ \begin{array}{c} \|\hat{W}^Q_{i,b}\|_F, \\ \|\hat{W}^K_{i,b}\|_F, \\ \|\hat{W}^V_{i,b}\|_F, \\ \|\hat{W}^O_b\|_F \end{array} \right\} \right] \|\hat{W}^F_{1,b}\|_F \|\hat{W}^F_{2,b}\|_F. \tag{35}$$

Note that, as with FNN and CNN, there are additional dependencies that we did not take into account, since in *grokking* these are assumed to be fixed. While several other norm-based bounds exist [81,95–98], this result remains among the most meaningful and tightest available to date [80,81,85].

The various norm-based generalization bounds reported in this section correspond to the tightest results available in the literature—namely, those that, all else being equal, yield the smallest values for the different architectures. Their differences arise solely from their dependence on the specific architecture under consideration; for any given architecture, they represent the best bounds currently known. They are not the theoretically minimal possible bounds, but rather the smallest ones established so far, to the best of the authors' knowledge.

To the best of our knowledge, no prior work has theoretically related these bounds to the grokking phenomenon. This is likely difficult, or even impossible, because grokking is not consistently observed: it depends on both the dataset and the model. In contrast, norm-based bounds incorporate data dependence mostly through the empirical error, while the complexity term (the norm-based component) is data-independent. Although the weights implicitly encode some data dependence—being obtained through minimization on the training set—the complexity term itself does not explicitly account for it.

In our study, we instead show empirically that computing these theoretical quantities across training epochs provides insights into grokking (see Section 4). Even in cases where the test error shows no improvement, the norm-based complexity terms evolve during training and steadily decrease, indicating that the model continues to learn—even when this is not reflected in standard error metrics.

We also note that a theoretical investigation of the dynamics of norm-based bounds across epochs is, to the best of the authors' knowledge, not available in the literature, as such an analysis lies beyond the scope of norm-based bounds, which assume a fixed model. For this reason, in Section 4, we report the empirical trends.

### 3.2. Stability analysis

In this section, we present the state-of-the-art hypothesis stability HS-based generalization bounds for both shallow and deep machine learning models . These bounds have undergone significant development over time [24,60,101,102,121–126]. Initially, they were difficult to apply in practice due to the challenge of estimating HS directly from data. More recently, however, it has been demonstrated that HS can either be estimated directly from data [24,122] or tightly bounded [24,126]. HS is well suited for analyzing the properties of a learning algorithm, as it is strongly influenced by both the data-generating distribution and the fine-grained characteristics of the machine learning model. Unlike other notions of stability, such as uniform stability, HS captures the average behavior of the model over data and training, rather than its worst-case behavior [24,60,101,102,121–126,133].

For both shallow and deep machine learning models, the HS coefficient $\beta$ can be directly estimated from data, provided that sufficient computational power is available [24,122]. Indeed, computing $\beta$ requires multiple training iterations, which may be computationally prohibitive if the dataset is large or the model is complex. In practice, $\beta$ can be estimated using resampling techniques—such as the Bootstrap [131] or the Bag of Little Bootstraps [132]—to reduce computational overhead, or through theoretical approaches [24,122], yielding an estimate $\hat{\beta}$. Note that theoretical approaches rely on certain assumptions, such as the monotonic learning property [122]; however, this assumption can often be easily satisfied in practice [134]. In practice, for shallow machine learning models, directly estimating $\beta$ is typically both applicable and effective [24,133].

For deep ML models, directly estimating $\beta$ from data is generally impractical, as the computational requirements quickly become prohibitive. For this reason, one can instead resort to computing an upper bound on $\beta$ that is data-dependent [24,126,133]. When the model $f$ can be expressed as in Eq. (6), where $r_\rho(X) \in \mathbb{R}^d$, the following proportionality holds:

$$\hat{\beta} \propto \mathsf{Cond}\left( \begin{bmatrix} r^T(X_1) \\ \vdots \\ r^T(X_n) \end{bmatrix} \begin{bmatrix} r^T(X_1) \\ \vdots \\ r^T(X_n) \end{bmatrix}^T \right), \tag{36}$$

where $\mathsf{Cond}$ denotes the condition number, i.e., the ratio of the largest to the smallest singular value.

Analogously to what has been done in the previous section for norm-based bounds, in this section we report the tightest hypothesis-stability-based generalization bounds available in the literature, restricting ourselves to those that are computable and both data- and algorithm-dependent, so as to extract the highest amount of information from these theoretical quantities.

Unfortunately, computing hypothesis stability is a computationally demanding process that can be performed only on limited datasets and small models. For large datasets and/or large models, the only viable solution is to bound it using highly informative upper bounds, as previously discussed.

Note also that, in some cases, these bounds can be specialized only when considering specific algorithms, but such specializations usually require additional upper-bounding steps that increase the gap between the quantity we would like to measure and the one we are actually able to compute.

Grokking is not directly linked to stability unless one examines how stability evolves across training epochs. Observing a decreasing trend indicates that the algorithm continues to learn, even when this is not reflected in traditional error metrics. This is precisely what we will show in Section 4. A theoretical analysis of these trends does not currently exist in the literature, as it lies beyond the present capabilities of hypothesis-stability bounds.

## 4. Empirical evidence

This section presents empirical evidence supporting the claim that the theoretical analysis in Section 3 sometimes provides valuable insights into the phenomenon of *grokking*, but at other times fails, leading to a discussion of the reasons behind these failures. In particular, Section 4.1 illustrates results for shallow learning on a toy example, while Sections 4.2-4.4 present results for deep models, namely FNN, CNN, and TNN, on more realistic tasks.

### 4.1. Grokking in shallow models

In this section, we present an example of *grokking* for shallow (linear) models using a toy dataset.[2]

The TOY dataset illustrates the role of implicit bias as a possible explanation for the phenomenon of grokking. Machine learning algorithms are often designed with an inherent preference for certain classes of solutions, typically favoring simpler or more regular ones. In some cases, however, this bias is insufficient, and memorization becomes necessary to achieve optimal performance. To demonstrate this, we consider

[2] https://xanderdavies.com/writing/toy_grok/toy_grok.html.

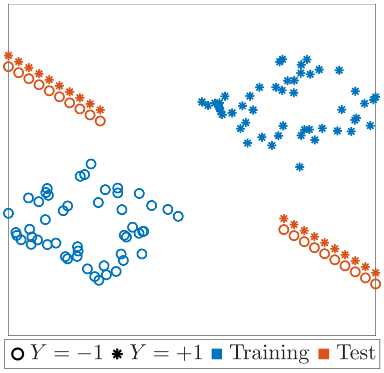


**Fig. 1.** Grokking in Shallow Model - Toy dataset.

a binary classification task on a dataset of two-dimensional points. A linearly separable training set is first constructed, and a hard-margin Support Vector Machine is applied to determine the maximum-margin solution. This solution is then used to generate a test set of points positioned near the decision boundary. Subsequently, central points in the test set are removed, creating a situation in which test accuracy cannot improve unless the learned solution aligns closely with the maximum-margin classifier. This dataset is depicted in Fig. 1.

For this dataset, the label space is given by $\mathcal{Y} = \{\pm 1\}$ and the input space by $\mathcal{X} = \mathbb{R}^2$. We train a linear model of the form $f(X) = \boldsymbol{w}^\top X$, where the weight vector $\boldsymbol{w} \in \mathbb{R}^2$ is optimized via empirical risk minimization using gradient descent (learning rate = 0.02) with the loss function $\ell(f, Z) = \mathrm{e}^{-Y f(X)}$.

In Fig. 2, we report the training and test error percentages, the stability (see Section 3.2), and the norm of the weights (see Section 3.1 Eq. (28)) as functions of the number of optimization steps. In this case, since the model is simple and computationally efficient, stability (Eq. 24) is directly estimated from the data using both the misclassification loss and the training loss [24,122]. As a consequence, stability depends on the chosen perspective (e.g., misclassification or training loss), whereas the norm of the weights does not.

As illustrated in Fig. 2 and anticipated in Section 3, stability does not increase with the number of optimization steps. On the contrary, it decreases, which indicates potential improvements in test error—improvements that are indeed observed. Notably, even when the training curve appears to plateau, stability continues to decrease significantly, suggesting enhanced generalization performance, as reflected in the reduction of test error.

In contrast, Fig. 2 shows that the norm of the weights is not informative in this example: it increases with the number of optimization steps, incorrectly suggesting a decrease in test error that does not occur. This behavior is expected, since the best solution in this setting corresponds to the hard-margin classifier, which is characterized by a large weight magnitude. This highlights one of the main limitations of norm-based bounds: they provide only upper bounds on the true complexity of the model, and such bounds are not always tight [24,133].

### 4.2. Grokking in deep ML: FNN

In this section, we present an example of *grokking* in an FNN on the MNIST dataset, taken from [66]. We rely on the implementation of [135], which also provides the corresponding code,[3] precisely replicating the baseline version of Fig. 9 in [135].

In particular, Fig. 3 shows the counterpart of Fig. 2 in Section 4.1: training and test loss, error percentage, stability (see Section 3.2 Eq. (36)), and the norm of the weights (see Section 3.1 Eq. (33)). The FNN considered in this section is too complex and computationally demanding to estimate stability directly from the data. Therefore, we use the bound based on the condition number (see Section 3.2). Consequently, both stability and the norm of the weights remain unchanged whether we consider loss or error percentage.

From Fig. 3, it can be observed that both stability and the norm of the weights provide insights into the behavior of the FFN: the larger the number of optimization steps, the smaller their values, indicating a possible trend of improvement in the generalization ability of the model. This confirms the intuition behind the idea of using stability- and norm-based bounds to gain insights into grokking. The model, while still learning and not yet exhibiting any improvement in performance, is in fact searching in the background for solutions that minimize the complexity of the representations, thereby reducing both stability and weight norm. For FNNs, previous works [66,135] have already shown that the norm of the weights serves as an indicator of the background progress of the network toward grokking, since older norm-based bounds were relatively tight. For CNNs and TNNs, instead, past bounds were too loose, and it was necessary to wait for the newly developed bounds described in Section 3.1 in order to provide meaningful insights. The stability results are consistent with previous studies, further confirming the ability of stability-based analyses to yield insights into the complex behavior of learning algorithms [24,133].

### 4.3. Grokking in deep ML: CNN

In this section, we present an example of *grokking* in a Graph CNN on the QM9 dataset, taken from [66], which also provides the corresponding code.[4] In particular, we replicate Fig. 5 of [66] with $\alpha = 3$.

In particular, Fig. 4 shows the counterpart of Fig. 3 in Section 4.2. As the QM9 dataset is a regression problem rather than a classification one, we report only the training and test loss (measured using the mean squared error), the stability bound based on the condition number (see Section 3.2 Eq. (36)), and the norm of the weights (see Section 3.1 Eq. (34)).

Fig. 4 shows a behavior consistent with that observed for FNNs in Section 4.2. In this case, however, the insights provided by norm-based bounds are made possible by recent advances in the theory of norm-based bounds for sparse models (see Section 3.1).

### 4.4. Grokking in deep ML: TNN

In this section, we present an example of *grokking* in a TNN on the MOD97 dataset, based on the seminal work of [57]. We use the implementation provided in a public repository,[5] faithfully reproducing Figs. 1 and 4 of [57].

In particular, Fig. 5 shows the counterpart of Fig. 3 in Section 4.2: training and test loss, error percentage, the stability bound based on the condition number (see Section 3.2 Eq. (36)), and the norm of the weights (see Section 3.1 Eq. (35)).

Fig. 5 exhibits a behavior consistent with that observed for FNNs (Section 4.2) and CNNs (Section 4.3). In this case as well, the applicability of norm-based bounds is due to recent theoretical advances specifically addressing transformer-based architectures (see Section 3.1).

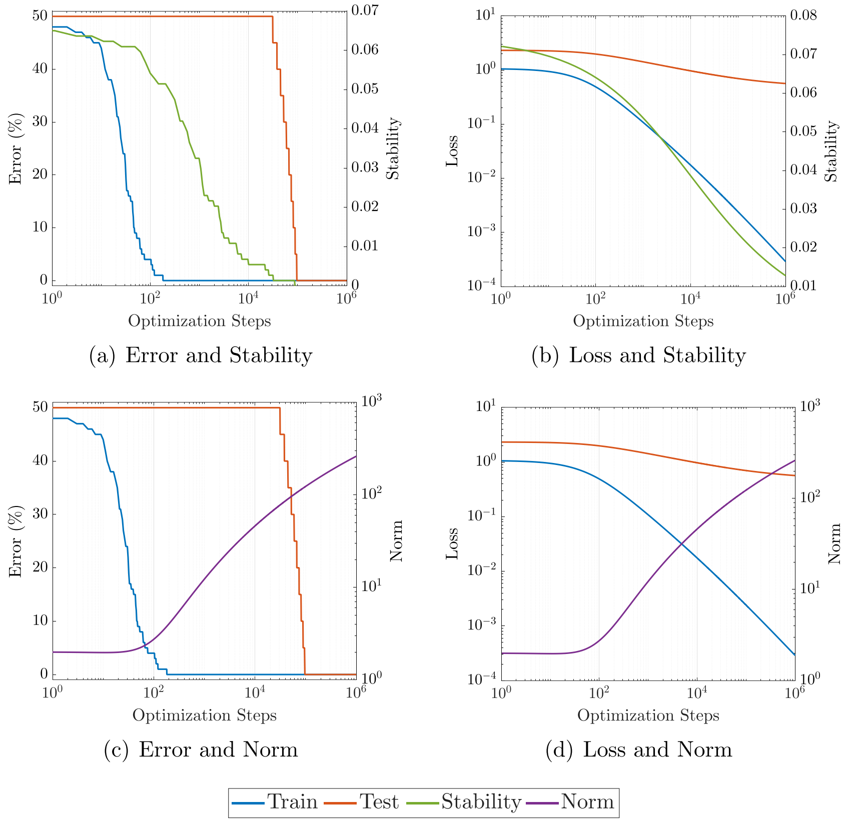
### 4.5. Discussion

The four empirical studies in Sections 4.1-4.4 jointly probe the extent to which the theoretical framework of Section 3 sheds light on the grokking phenomenon across different architectures, tasks, and types of generalization bounds. Table 1 summarizes the main experimental dimensions and their connection to the theory.

[3] https://github.com/ironjr/grokfast.

[4] https://github.com/KindXiaoming/Omnigrok.

[5] https://github.com/teddykoker/grokking.



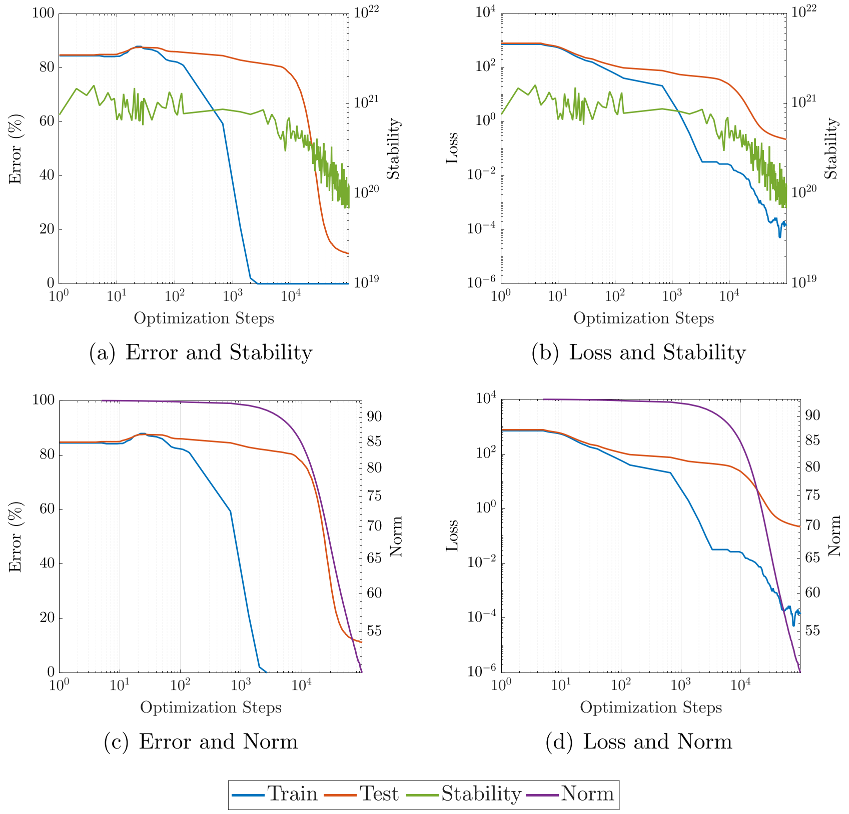
(a) Error and Stability

(b) Loss and Stability

(c) Error and Norm

(d) Loss and Norm

**Fig. 2.** Grokking in Shallow Model.



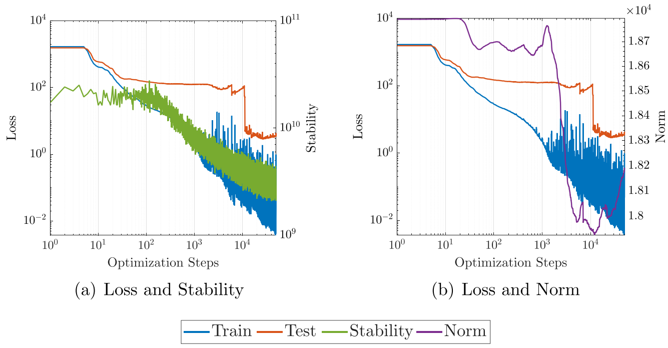
(a) Error and Stability

(b) Loss and Stability

(c) Error and Norm

(d) Loss and Norm

**Fig. 3.** Grokking in Deep ML: FNN.



(a) Loss and Stability

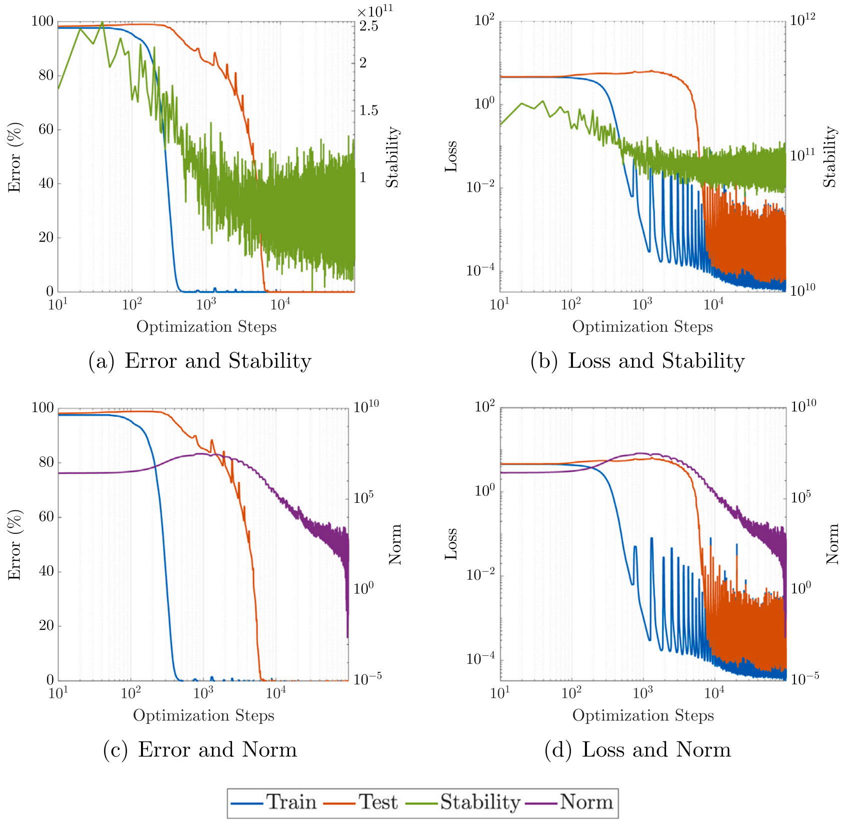
(b) Loss and Norm

**Fig. 4.** Grokking in Deep ML: CNN.

The shallow linear experiment in Section 4.1 provides a setting where the theoretical notions of hypothesis stability and norm based complexities can both be computed in a relatively direct way from the learned classifier. Here, we estimate hypothesis stability empirically via leave-one-out perturbations of the training set Eq. (24), and we measure the Euclidean norm entering the shallow bound of Eq. (28). The results clearly show that hypothesis stability decreases over time in parallel with the improvement in test error, while the weight norm increases and thus fails to track generalization. This is a concrete instance where the hypothesis stability based view of Section 3.2 offers a faithful description of grokking, whereas the norm-based bounds of Section 3.1 are too loose to be predictive in practice. In contrast, the deep settings (FNN, CNN, and TNN) in Sections 4.2–4.4 are characterized by highly overparameterized models for which direct hypothesis stability estimation is computationally infeasible and naive norms are uninformative. In these cases we rely instead on (i) the condition-number-based hypothesis stability bound of Eq. (36) and (ii) the architecture-specific norm bounds of Eqs. (33)–(35). Empirically, both quantities decrease during the grokking phase, aligning well with the theoretical prediction that improved hypothesis stability and reduced norm imply tighter generalization bounds.

The TOY, MNIST, and MOD97 experiments (Sections 4.1, 4.2, and 4.4) are binary or multi-class classification problems, whereas the QM9 experiment in Section 4.3 is a regression task. In classification, we can track both misclassification error and training loss, and—whenever feasible—define hypothesis stability with respect to either loss function. This is explicitly exploited in the shallow learning setting, where hypothesis stability is estimated with respect to both the misclassification loss and the exponential training loss, illustrating that hypothesis stability is inherently loss-dependent. In the deep learning settings, however, hypothesis stability is assessed only through the condition-number-based upper bound of Eq. (36), which abstracts away this dependence and yields a single complexity proxy that can be used across both classification and regression. The QM9 regression experiment shows that the same hypothesis stability and norm-based tools can still capture the “background progress” toward grokking when one only observes a continuous loss (mean squared error). This supports the theoretical claim that the framework of Section 3 applies beyond classification, as long as the loss satisfies the regularity assumptions required in the derivation of the bounds.

A central theme of Section 3.1 is that the tightness of norm-based bounds strongly depends on the architecture. The shallow toy example demonstrates a failure mode of classical norm bounds: the best solution corresponds to a large-margin classifier with large norm, so



(a) Error and Stability

(b) Loss and Stability

(c) Error and Norm

(d) Loss and Norm

**Fig. 5.** Grokking in Deep ML: TNN.

**Table 1**
Summary of the four empirical settings of Sections 4.1–4.4 and their relation to the theoretical framework.

| Sec. | Model | Dataset & Task | Hypothesis Stability | Norm | Insights |
|---|---|---|---|---|---|
| 4.1 | Linear (Shallow) | TOY Classification | Eq. (24) | Eq. (28) | Just Hypothesis Stability is informative |
| 4.2 | FNN | MNIST Classification | Eq. (36) | Eq. (33) | Both Hypothesis Stability and Norm are informative |
| 4.3 | (Graph) CNN | QM9 Regression | Eq. (36) | Eq. (34) | Both Hypothesis Stability and Norm are informative |
| 4.4 | TNN | MOD97 Classification | Eq. (36) | Eq. (35) | Both Hypothesis Stability and Norm are informative |

that the upper bound in Eq. (28) fails to reflect the actual evolution of test error. In the FNN, CNN, and TNN experiments, by contrast, we employ recently developed norm bounds that are tailored to fully-connected, sparse convolutional, and transformer architectures, respectively (Eqs. (33)–(35)). The empirical results show that these norms systematically decrease during the grokking regime, in line with the theoretical prediction that the learned representations are becoming simpler in a way that is captured by the corresponding bounds. This contrast between shallow and deep settings highlights that norm-based analyses can be informative for grokking only when the bounds are sufficiently adapted to the architecture under consideration.

Taken together, the four experiments illustrate that hypothesis stability- and norm-based bounds offer complementary perspectives on grokking. In all settings, a decrease in the hypothesis stability proxy (empirical hypothesis stability for the shallow model, condition-number-based bound for the deep models) accompanies the eventual improvement in test performance, thereby supporting the theoretical view of Section 3.2 that increased hypothesis stability is a key driver of generalization. Norm-based bounds, on the other hand, are informative in the FNN, CNN, and TNN settings—where architecture-specific bounds are available—but not in the shallow toy setting - where the bound is too coarse. This systematic comparison clarifies when the theoretical tools of Section 3 successfully anticipate grokking (deep architectures with tailored bounds, both hypothesis stability and norm) and when they fall short (shallow toy example, norm-based analysis), thereby delineating the practical scope and limitations of the proposed framework.

## 5. Conclusions

The phenomenon of *grokking*, a sudden leap in model performance after an extended period of stagnation, challenges conventional expectations about how learning and generalization progress in machine learning systems. While it may initially appear to be an anomaly, our work demonstrates that grokking can, in fact, be reconciled with existing principles from statistical learning theory.

By examining grokking through the lens of norm-based and stability-based generalization bounds, we have shown that the theoretical tools already developed within statistical learning theory provide valuable insights into this behavior. In particular, we have argued that changes in implicit regularization dynamics, stability properties of the training algorithm, and the evolving geometry of the learned representations can collectively explain the transition from poor to strong generalization performance.

Our theoretical analysis was supported by empirical evidence illustrating how models may undergo structural shifts in norm and stability metrics prior to grokking, even when surface-level performance metrics remain flat or regress. These observations not only support our theoretical framing but also suggest that the onset of grokking may, in principle, be anticipated or influenced.

Ultimately, this work contributes to the broader effort of understanding the complex and sometimes counterintuitive behaviors of modern machine learning systems. Rather than requiring a departure from existing theory, we argue that grokking exemplifies the richness and flexibility of statistical learning theory when applied thoughtfully. Future work should aim to further unify these theoretical perspectives and extend them to other emerging phenomena, fostering a deeper and more predictive understanding of learning in highly expressive models.

## CRediT authorship contribution statement

**Luca Oneto:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Sandro Ridella:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Simone Minisi:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Data curation. **Andrea Coraddu:** Writing – review & editing, Validation, Supervision, Investigation. **Davide Anguita:** Writing – review & editing, Investigation, Funding acquisition, Formal analysis, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work is partially supported by (i) project ELSA - European Lighthouse on Secure and Safe AI funded by the European Union's Horizon Europe under the grant agreement No. 101070617,(ii) project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU,(iii) project FAIR (PE00000013) under the NRRP MUR program funded by the EU - NGEU, and (iv) project FISA-2023–00128 funded by the MUR program “Fondo Italiano per le Scienze Applicate”.

## Data availability

Data will be made available on request.

## References

[1] D. Silver, A. Huang, C.J. Maddison, A. Guez, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.
[2] M. Varadi, D. Bertoni, P. Magana, U. Paramval, et al., Alphafold protein structure database in 2024: providing structure coverage for over 214 million protein sequences, Nucl. Acids Res. 52 (D1) (2024) D368–D375.
[3] A. Hurst, A. Lerer, A.P. Goucher, A. Perelman, et al., Gpt-4o system card, arXiv preprint arXiv:2410.21276, 2024.
[4] D.B. Acharya, K. Kuppan, B. Divya, Agentic AI: autonomous intelligence for complex goals-a comprehensive survey, IEEE Access (2025).
[5] OpenAI, Gpt-4 technical report, arXiv preprint arXiv:2303.08774, 2023.
[6] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Neural Inf. Process. Syst. (2020).
[7] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, et al., Overcoming catastrophic forgetting in neural networks, Proc. Natl. Acad. Sci. 114 (13) (2017) 3521–3526.
[8] V.V. Ramasesh, A. Lewkowycz, E. Dyer, Effect of scale on catastrophic forgetting in neural networks, in: International Conference on Learning Representations, 2021.
[9] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, Y. Zhang, An empirical study of catastrophic forgetting in large language models during continual fine-tuning, arXiv preprint arXiv:2308.08747, 2023.
[10] G. Shi, J. Chen, W. Zhang, L.M. Zhan, X.M. Wu, Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima, in: Neural Information Processing Systems, 2021.

[11] E.L. Aleixo, J.G. Colonna, M. Cristo, E. Fernandes, Catastrophic forgetting in deep learning: A comprehensive taxonomy, arXiv preprint arXiv:2312.10549, 2023.

[12] Y. Kossale, M. Airaj, A. Darouichi, Mode collapse in generative adversarial networks: an overview, in: International Conference on Optimization and Applications, 2022.

[13] H. Thanh-Tung, T. Tran, Catastrophic forgetting and mode collapse in gans, in: International Joint Conference on Neural Networks, 2020.

[14] K. Zhang, On mode collapse in generative adversarial networks, in: International Conference on Artificial Neural Networks, 2021, pp. 563–574.

[15] F.L. Barsha, W. Eberle, An in-depth review and analysis of mode collapse in generative adversarial networks, Mach. Learn. 114 (6) (2025) 141.

[16] R. Geirhos, J.H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, F.A. Wichmann, Shortcut learning in deep neural networks, Nat. Mach. Intell. 2 (11) (2020) 665–673.

[17] Y. Yuan, L. Zhao, K. Zhang, G. Zheng, Q. Liu, Do llms overcome shortcut learning? An evaluation of shortcut challenges in large language models, in: Conference on Empirical Methods in Natural Language Processing, 2024.

[18] K.L. Hermann, H. Mobahi, T. Fel, M.C. Mozer, On the foundations of shortcut learning, arXiv preprint arXiv:2310.16228, 2023.

[19] M. Srivastava, T. Hashimoto, P. Liang, Robustness to spurious correlations via human annotations, in: International Conference on Machine Learning, 2020.

[20] W. Ye, G. Zheng, X. Cao, Y. Ma, A. Zhang, Spurious correlations in machine learning: A survey, arXiv preprint arXiv:2402.12715, 2024.

[21] S. Sagawa, A. Raghunathan, P.W. Koh, P. Liang, An investigation of why overparameterization exacerbates spurious correlations, in: International Conference on Machine Learning, 2020, pp. 8346–8356.

[22] V. Cherkassky, E.H. Lee, To understand double descent, we need to understand VC theory, Neural Netw. 169 (2024) 242–256.

[23] M. Loog, T. Viering, A. Mey, J.H. Krijthe, D.M.J. Tax, A brief prehistory of double descent, Proc. Natl. Acad. Sci. 117 (20) (2020) 10625–10626.

[24] L. Oneto, S. Ridella, D. Anguita, Do we really need a new theory to understand over-parameterization? Neurocomputing 543 (2023) 126227.

[25] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, I. Sutskever, Deep double descent: where bigger models and more data hurt, J. Stat. Mech.: Theory Exp. 2021 (12) (2021) 124003.

[26] S. D'Ascoli, M. Refinetti, G. Biroli, F. Krzakala, Double trouble in double descent: bias and variance (s) in the lazy regime, in: International Conference on Machine Learning, 2020.

[27] S. Mei, A. Montanari, The generalization error of random features regression: precise asymptotics and the double descent curve, Commun. Pure Appl. Math. 75 (4) (2022) 667–766.

[28] M. Belkin, D. Hsu, S. Ma, S. Mandal, Reconciling modern machine-learning practice and the classical bias-variance trade-off, Proc. Natl. Acad. Sci. 116 (32) (2019) 15849–15854.

[29] O. Shamir, The implicit bias of benign overfitting, in: Conference on Learning Theory, 2022.

[30] P.L. Bartlett, P.M. Long, G. Lugosi, A. Tsigler, Benign overfitting in linear regression, Proc. Natl. Acad. Sci. 117 (48) (2020) 30063–30070.

[31] Y. Cao, Z. Chen, M. Belkin, Q. Gu, Benign overfitting in two-layer convolutional neural networks, in: Neural Information Processing Systems, 2022.

[32] A. Sanyal, P.K. Dokania, V. Kanade, P. Torr, How benign is benign overfitting? in: International Conference on Learning Representations, 2020.

[33] A. Tsigler, P.L. Bartlett, Benign overfitting in ridge regression, J. Mach. Learn. Res. 24 (123) (2023) 1–76.

[34] Z. Li, Z.H. Zhou, A. Gretton, Towards an understanding of benign overfitting in neural networks, arXiv preprint arXiv:2106.03212, 2021.

[35] Y. Kou, Z. Chen, Y. Chen, Q. Gu, Benign overfitting in two-layer RELU convolutional neural networks, in: International Conference on Machine Learning, 2023.

[36] Z. Allen-Zhu, Y. Li, Z. Song, A convergence theory for deep learning via over-parameterization, in: International Conference on Machine Learning, 2019.

[37] Z.H. Zhou, Why over-parameterization of deep neural networks does not overfit? Sci. China Inf. Sci. 64 (1) (2021) 116101.

[38] Y. Dar, V. Muthukumar, R.G. Baraniuk, A farewell to the bias-variance tradeoff? an overview of the theory of overparameterized machine learning, arXiv preprint arXiv:2109.02355, 2021.

[39] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422–440.

[40] S. Zampini, G. Parodi, L. Oneto, A. Coraddu, D. Anguita, A review on full-, zero-, and partial-knowledge based predictive models for industrial applications, Inf. Fusion (2025) 102996.

[41] Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, H. Su, J. Zhu, Physics-informed machine learning: A survey on problems, methods and applications, arXiv preprint arXiv:2211.08064, 2022.

[42] L. Oneto, N. Navarin, B. Biggio, F. Errica, et al., Towards learning trustworthily, automatically, and with guarantees on graphs: an overview, Neurocomputing 493 (2022) 217–243.

[43] D. Pessach, E. Shmueli, A review on fairness in machine learning, ACM Comput. Surv. 55 (3) (2022) 1–44.

[44] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM Comput. Surv. 54 (6) (2021) 1–35.

[45] B. Biggio, F. Roli, Wild patterns: ten years after the rise of adversarial machine learning, Pattern Recognition 84 (2018) 317–331.

[46] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, X. Yi, A survey of safety and trustworthiness of deep neural networks: verification, testing, adversarial attack and defence, and interpretability, Comput. Sci. Rev. 37 (2020) 100270.

[47] N. Burkart, M.F. Huber, A survey on the explainability of supervised machine learning, J. Artif. Intell. Res. 70 (2021) 245–317.

[48] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, et al., Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.

[49] P.P. Angelov, E.A. Soares, R. Jiang, N.I. Arnold, P.M. Atkinson, Explainable artificial intelligence: an analytical review, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 11 (5) (2021) e1424.

[50] C. Molnar, Interpretable Machine Learning, Lulu. Com, 2020.

[51] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, Z. Lin, When machine learning meets privacy: a survey and outlook, ACM Comput. Surv. 54 (2) (2021) 1–36.

[52] R. Xu, N. Baracaldo, J. Joshi, Privacy-preserving machine learning: Methods, challenges and directions, arXiv preprint arXiv:2108.04417, 2021.

[53] J. Chen, Y. Cao, Q. Gu, Benign overfitting in adversarially robust linear classification, in: Uncertainty in Artificial Intelligence, 2023.

[54] J. Zheng, M. Makar, Causally motivated multi-shortcut identification and removal, in: Neural Information Processing Systems, 2022.

[55] J. Zheng, X. Cai, S. Qiu, Q. Ma, Spurious forgetting in continual learning of language models, arXiv preprint arXiv:2501.13453, 2025.

[56] X. Davies, L. Langosco, D. Krueger, Unifying grokking and double descent, arXiv preprint arXiv:2303.06173, 2023.

[57] A. Power, Y. Burda, H. Edwards, I. Babuschkin, V. Misra, Grokking: Generalization beyond overfitting on small algorithmic datasets, arXiv preprint arXiv:2201.02177, 2022.

[58] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: from Theory to Algorithms, Cambridge University Press, 2014.

[59] C.M. Bishop, H. Bishop, Deep Learning: Foundations and Concepts, Springer Nature, 2023.

[60] L. Oneto, Model Selection and Error Estimation in a Nutshell, Springer, 2020.

[61] V.N. Vapnik, Statistical Learning Theory, Wiley New York, 1998.

[62] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, Springer, 2013.

[63] D. Russo, J. Zou, How much does your data exploration overfit? Controlling bias via information usage, IEEE Trans. Inf. Theory 66 (1) (2019) 302–323.

[64] A. Gromov, Grokking modular arithmetic, arXiv preprint arXiv:2301.02679, 2023.

[65] B. Žunkovič, E. Ilievski, Grokking phase transitions in learning local rules with gradient descent, J. Mach. Learn. Res. 25 (199) (2024) 1–52.

[66] Z. Liu, E.J. Michaud, M. Tegmark, Omnigrok: grokking beyond algorithmic data, in: International Conference on Learning Representations, 2022.

[67] S. Murty, P. Sharma, J. Andreas, C.D. Manning, Grokking of hierarchical structure in vanilla transformers, arXiv preprint arXiv:2305.18741, 2023.

[68] S. Samothrakis, A. Matran-Fernandez, U. Abdullahi, M. Fairbank, M. Fasli, Grokking-like effects in counterfactual inference, in: International Joint Conference on Neural Networks, 2022.

[69] Z. Xu, Y. Wang, S. Frei, G. Vardi, W. Hu, Benign overfitting and grokking in relu networks for xor cluster data, arXiv preprint arXiv:2310.02541, 2023.

[70] Z. Liu, Z. Zhong, M. Tegmark, Grokking as compression: A nonlinear complexity perspective, arXiv preprint arXiv:2310.05918, 2023.

[71] Z. Tan, W. Huang, Understanding grokking through a robustness viewpoint, arXiv preprint arXiv:2311.06597, 2023.

[72] Z. Liu, O. Kitouni, N.S. Nolte, E. Michaud, M. Tegmark, M. Williams, Towards understanding grokking: an effective theory of representation learning, Neural Inf. Process. Syst. (2022).

[73] N. Nanda, L. Chan, T. Lieberum, J. Smith, J. Steinhardt, Progress measures for grokking via mechanistic interpretability, arXiv preprint arXiv:2301.05217, 2023.

[74] K. Lyu, J. Jin, Z. Li, S.S. Du, J.D. Lee, W. Hu, Dichotomy of early and late phase implicit biases can provably induce grokking, in: International Conference on Learning Representations, 2023.

[75] V. Thilak, E. Littwin, S. Zhai, O. Saremi, R. Paiss, J. Susskind, The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon, arXiv preprint arXiv:2206.04817, 2022.

[76] T. Kumar, B. Bordelon, S.J. Gershman, C. Pehlevan, Grokking as the transition from lazy to rich training dynamics, arXiv preprint arXiv:2310.06110, 2023.

[77] P. Klesk, M. Korzen, Sets of approximating functions with finite vapnik-chervonenkis dimension for nearest-neighbors algorithms, Pattern Recognit. Lett. 32 (14) (2011) 1882–1893.

[78] P.L. Bartlett, D. Mendelson, Rademacher and Gaussian complexities: risk bounds and structural results, J. Mach. Learn. Res. 3 (2002) 463–482.

[79] P.L. Bartlett, For valid generalization the size of the weights is more important than the size of the network, in: Neural Information Processing Systems, 1996.

[80] T. Galanti, M. Xu, L. Galanti, T. Poggio, Norm-based generalization bounds for sparse neural networks, in: Neural Information Processing Systems, 2023.

[81] J. Trauger, A. Tewari, Sequence length independent norm-based generalization bounds for transformers, in: International Conference on Artificial Intelligence and Statistics, 2024.

[82] B. Neyshabur, R. Tomioka, N. Srebro, Norm-based capacity control in neural networks, in: Conference on Learning Theory, 2015.

[83] N. Golowich, A. Rakhlin, O. Shamir, Size-independent sample complexity of neural networks, in: Conference on Learning Theory, 2018.

[84] P.L. Bartlett, N. Harvey, C. Liaw, A. Mehrabian, Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks, J. Mach. Learn. Res. 20 (63) (2019) 1–17.

[85] P.L. Bartlett, D.J. Foster, M.J. Telgarsky, Spectrally-normalized margin bounds for neural networks, in: Neural Information Processing Systems, 2017.

[86] B. Neyshabur, S. Bhojanapalli, N. Srebro, A pac-bayesian approach to spectrally-normalized margin bounds for neural networks, arXiv preprint arXiv:1707.09564, 2017.

[87] Y. Cao, Q. Gu, Generalization bounds of stochastic gradient descent for wide and deep neural networks, in: Neural Information Processing Systems, 2019.

[88] A. Daniely, E. Granot, Generalization bounds for neural networks via approximate description length, in: Neural Information Processing Systems, 2019.

[89] C. Wei, T. Ma, Data-dependent sample complexity of deep neural networks via lipschitz augmentation, in: Neural Information Processing Systems, 2019.

[90] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, S. Bengio, Fantastic generalization measures and where to find them, in: International Conference on Learning Representations, 2020.

[91] X. Li, J. Lu, Z. Wang, J. Haupt, T. Zhao, On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond, arXiv preprint arXiv:1806.05159, 2018.

[92] M.L. Philip, H. Sedghi, Generalization bounds for deep convolutional neural networks, in: International Conference on Learning Representations, 2020.

[93] S.S. Du, Y. Wang, X. Zhai, S. Balakrishnan, R.R. Salakhutdinov, A. Singh, How many samples are needed to estimate a convolutional neural network? in: Neural Information Processing Systems, 2018.

[94] A. Ledent, W. Mustafa, Y. Lei, M. Kloft, Norm-based generalisation bounds for deep multi-class convolutional neural networks, in: AAAI Conference on Artificial Intelligence, 2021.

[95] B.L. Edelman, S. Goel, S. Kakade, C. Zhang, Inductive biases and variable creation in self-attention mechanisms, in: International Conference on Machine Learning, 2022.

[96] Y. Zhang, B. Liu, Q. Cai, L. Wang, Z. Wang, An analysis of attention via the lens of exchangeability and latent variable models, arXiv preprint arXiv:2212.14852, 2024.

[97] C. Wei, Y. Chen, T. Ma, Statistically meaningful approximation: a case study on approximating turing machines with transformers, in: Neural Information Processing Systems, 2022.

[98] H. Fu, T. Guo, Y. Bai, S. Mei, What can a single attention layer learn? A study through the random features lens, in: Neural Information Processing Systems, 2023.

[99] J. Langford, R. Schapire, Tutorial on practical prediction theory for classification, J. Mach. Learn. Res. 6 (3) (2005).

[100] P.D. Grünwald, The Minimum Description Length Principle, MIT Press, 2007.

[101] O. Bousquet, A. Elisseeff, Stability and generalization, J. Mach. Learn. Res. 2 (2002) 499–526.

[102] T. Poggio, R. Rifkin, S. Mukherjee, P. Niyogi, General conditions for predictivity in learning theory, Nature 428 (6981) (2004) 419–422.

[103] P. Alquier, User-friendly introduction to pac-bayes bounds, arXiv preprint arXiv:2110.11216, 2021.

[104] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. 9 (3–4) (2014) 211–407.

[105] D.H. Wolpert, The supervised learning no-free-lunch theorems, in: Soft Computing and Industry, 2002.

[106] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2) (2012).

[107] T. Elsken, J.H. Metzen, F. Hutter, Neural architecture search: a survey, J. Mach. Learn. Res. 20 (55) (2019) 1–21.

[108] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: a next-generation hyperparameter optimization framework, in: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.

[109] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: theory and practice, Neurocomputing 415 (2020) 295–316.

[110] M. Feurer, F. Hutter, Hyperparameter optimization, in: Automated Machine Learning: Methods, Systems, Challenges, 2019.

[111] X. He, K. Zhao, X. Chu, Automl: a survey of the state-of-the-art, Knowl.-Based Syst. 212 (2021) 106622.

[112] T. Dietterich, Overfitting and undercomputing in machine learning, ACM Comput. Surv. 27 (3) (1995) 326–327.

[113] F. Mori, A.E. Cinà, F. Roli, D. Anguita, L. Oneto, Toward measuring and understanding the overvalidation phenomena, in: International Conference on Machine Learning and Applications, 2024.

[114] P. Duboue, The Art of Feature Engineering: Essentials for Machine Learning, Cambridge University Press, 2020.

[115] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.

[116] L. Oneto, S. Ridella, D. Anguita, Towards algorithms and models that we can trust: a theoretical perspective, Neurocomputing 592 (2024) 127798.

[117] D.X. Zhou, The covering number in learning theory, J. Complex. 18 (3) (2002) 739–767.

[118] P.L. Bartlett, O. Bousquet, S. Mendelson, Local rademacher complexities, Ann. Stat. 33 (4) (2005) 1497–1537.

[119] L. Oneto, D. Anguita, S. Ridella, A local vapnik-chervonenkis complexity, Neural Netw. 82 (2016) 62–75.

[120] R.M. Dudley, The sizes of compact subsets of hilbert space and continuity of Gaussian processes, J. Funct. Anal. 1 (3) (1967) 290–330.

[121] A. Elisseeff, T. Evgeniou, M. Pontil, L.P. Kaelbing, Stability of randomized learning algorithms, J. Mach. Learn. Res. 6 (1) (2005) 55–79.

[122] L. Oneto, A. Ghio, S. Ridella, D. Anguita, Fully empirical and data-dependent stability-based bounds, IEEE Trans. Cybern. 45 (9) (2014) 1913–1926.

[123] S. Shalev-Shwartz, O. Shamir, N. Srebro, K. Sridharan, Learnability, stability and uniform convergence, J. Mach. Learn. Res. 11 (2010) 2635–2670.

[124] S. Mukherjee, P. Niyogi, T. Poggio, R. Rifkin, Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization, Adv. Comput. Math. 25 (2006) 161–193.

[125] A. Maurer, A second-order look at stability and generalization, in: Conference on Learning Theory, 2017.

[126] A. Rangamani, L. Rosasco, T. Poggio, For interpolating kernel machines, minimizing the norm of the erm solution minimizes stability, arXiv preprint arXiv:2006.15522, 2020.

[127] G. Donghi, L. Pasa, L. Oneto, C. Gallicchio, A. Micheli, D. Anguita, A. Sperduti, N. Navarin, Investigating over-parameterized randomized graph networks, Neurocomputing 606 (2024) 128281.

[128] L. Oneto, S. Ridella, A. Coraddu, D. Anguita, Reconciling grokking with statistical learning theory, in: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2025.

[129] L. Devroye, T. Wagner, Distribution-free inequalities for the deleted and holdout error estimates, IEEE Trans. Inf. Theory 25 (2) (1979) 202–207.

[130] M. Kearns, D. Ron, Algorithmic stability and sanity-check bounds for leave-one-out cross-validation, in: International Conference on Computational Learning Theory, 1997.

[131] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, CRC Press, 1994.

[132] A. Kleiner, A. Talwalkar, P. Sarkar, M.I. Jordan, A scalable bootstrap for massive data, J. R. Stat. Soc. Ser. B Stat. Methodol. 76 (4) (2014) 795–816.

[133] L. Oneto, S. Ridella, D. Anguita, Informed machine learning: excess risk and generalization, Neurocomputing (2025) 130521.

[134] O.J. Bousquet, A. Daniely, H. Kaplan, Y. Mansour, S. Moran, U. Stemmer, Monotone learning, in: Conference on Learning Theory, 2022.

[135] J. Lee, B.G. Kang, K. Kim, K.M. Lee, Grokfast: Accelerated grokking by amplifying slow gradients, arXiv preprint arXiv:2405.20233, 2024.

## Author biography

**Luca Oneto** born in 1986 in Rapallo, Italy, completed his BSc and MSc in Electronic Engineering at the University of Genoa in 2008 and 2010, respectively. In 2014, he earned his PhD in Computer Engineering from the same institution. From 2014 to 2016, he worked as a Postdoc in Computer Engineering at the University of Genoa, where he then served as an Assistant Professor from 2016 to 2019. Luca co-founded the company ZenaByte s.r.l. in 2018. In 2019, he became an Associate Professor in Computer Science at the University of Pisa, and from 2019 to 2024, he held the position of Associate Professor in Computer Engineering at the University of Genoa. Currently, he is a Full Professor in Computer Engineering at the University of Genoa. He has been coordinator and local responsible in numerous industrial, H2020, and Horizon Europe projects. He has received prestigious recognitions, including the Amazon AWS Machine Learning Award and the Somalvico Award for the best young AI researcher in Italy. His primary research interests lie in Statistical Learning Theory and Trustworthy AI. Additionally, he focuses on data science, utilizing and improving cutting-edge machine learning and AI algorithms to tackle real-world problems.

**Sandro Ridella** received the “Laurea” degree in electronic engineering from the University of Genoa, Genoa, Italy, in 1966. Currently, he is a Full Professor at the Department of Biophysical and Electronic Engineering (DIBE, now DITEN Dept.), University of Genoa, where he teaches circuits and algorithms for signal processing. In the last five years, his scientific activity has been mainly focused on the field of neural networks.

**Simone Minisi** was born in Genova, Italy in 1991, holds a Master’s degree earned in 2018 in Computer Science from the University of Genoa. He is an IT professional and Research Fellow at the Department of Computer Science, Bioengineering, Robotics, and Systems Engineering (DIBRIS) at the University of Genoa. He specializes in Foundational and Trustworthy AI as part of the sAIfer Lab. His research focuses on developing methodologies to enhance transparency, robustness, and data privacy in AI systems, and he actively contributes to projects advancing these critical areas of artificial intelligence.

**Andrea Coraddu** was born in Pietrasanta, Italy, in 1979. He received the Laurea degree in naval architecture and marine engineering from the University of Genoa, Italy, in 2006, and the Ph.D. degree from the School of Fluid and Solid Mechanics, University of Genoa, in 2012. His Ph.D. dissertation was titled “Modelling and Control of Naval Electric Propulsion Plants.” Currently, he is an Associate Professor of intelligent and sustainable energy systems with the Maritime and Transport Technology Department, Delft University of Technology, Delft, The Netherlands. His relevant professional and academic experiences include working as an Associate Professor with the Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde, a Research Associate with the School of Marine Science and Technology, Newcastle University, and a Research Engineer as part of the DAMEN Research and Development Department, Singapore. He is also a Postdoctoral Research Fellow with the University of Genoa. He has been involved in a number of successful grant 2 applications from research councils, industry, and international governmental agencies focusing on the design, integration, and control of complex marine energy and power management systems enabling the development of next-generation complex and multi-function vessels that can meet the social challenges regarding the environmental impact of human-related activities.

**Davide Anguita** received the “Laurea” degree in Electronic Engineering and a Ph.D. degree in Computer Science and Electronic Engineering from the University of Genoa, Genoa, Italy, in 1989 and 1993, respectively. After working as a Research Associate at the International Computer Science Institute, Berkeley, CA, on special-purpose processors for neurocomputing, he returned to the University of Genoa. He is currently Associate Professor of Computer Engineering with the Department of Informatics, BioEngineering, Robotics, and Systems Engineering (DIBRIS). His current research focuses on the theory and application of kernel methods and artificial neural networks.