European Journal of Applied Mathematics (2025), 1–31
doi:10.1017/S0956792525100193

CAMBRIDGE UNIVERSITY PRESS

PAPER

The mathematics of adversarial attacks in AI – why deep learning is unstable despite the existence of stable neural networks

Alexander Bastounis[1], Anders Hansen[2] and Verner Vlačić[3]

[1]Department of Mathematics, King's College London, London, UK
[2]DAMTP, University of Cambridge, Cambridge, UK
[3]D-ITET, ETH Zürich, Zürich, Switzerland
Corresponding author: Anders Hansen; Email: ach70@cam.ac.uk

Received: 06 June 2024; **Revised:** 05 September 2025; **Accepted:** 08 September 2025

Keywords: instability in deep learning; methodological barriers; existence of algorithms (deterministic and randomised); approximation theory; robust optimisation; numerical analysis; solvability complexity index hierarchy

2020 Mathematics Subject Classification: 68T07, 65K10, 41A30, 90C17 (Primary)

Abstract

The unprecedented success of deep learning (DL) makes it unchallenged when it comes to classification problems. However, it is well established that the current DL methodology produces universally unstable neural networks (NNs). The instability problem has caused a substantial research effort – with a vast literature on so-called adversarial attacks – yet there has been no solution to the problem. Our paper addresses why there has been no solution to the problem, as we prove the following: any training procedure based on training rectified linear unit (ReLU) neural networks for classification problems with a fixed architecture will yield neural networks that are either inaccurate or unstable (if accurate) – despite the provable existence of both accurate and stable neural networks for the same classification problems. The key is that the stable and accurate neural networks must have variable dimensions depending on the input, in particular, variable dimensions is a necessary condition for stability. Our result points towards the paradox that accurate and stable neural networks exist; however, modern algorithms do not compute them. This yields the question: if the existence of neural networks with desirable properties can be proven, can one also find algorithms that compute them? There are cases in mathematics where provable existence implies computability, but will this be the case for neural networks? The contrary is true, as we demonstrate how neural networks can provably exist as approximate minimisers to standard optimisation problems with standard cost functions; however, no randomised algorithm can compute them with probability better than $1/2$.

1. Introduction

Neural networks (NNs) [29, 48, 67] and deep learning (DL) [52] have seen incredible success, in particular in classification problems [58]. However, neural networks become universally unstable (non-robust) when trained to solve such problems in virtually any application [2–4, 10, 20–22, 36, 47, 49, 74], making the non-robustness issue one of the fundamental problems in artificial intelligence (AI). The vast literature on this issue – often referring to the instability phenomenon as vulnerability to adversarial attacks – has not been able to solve the problem. Thus, we are left with the key question:

Why does deep learning yield universally unstable methods for classification?

In this paper, we provide mathematical answers to this question in connection with Smale's 18th problem on the limits of AI.

The above problem has become particularly relevant as the instability phenomenon yields non-human-like behaviour of AI with misclassifications by DL methods being caused by small perturbations

that are so tiny that human sensor systems such as eyes and ears cannot detect the tiny change. The non-robustness issue has thus caused serious concerns among scientists [2, 36, 47], in particular in applications where trustworthiness of AI is a key feature. Moreover, the instability phenomenon has become a grave matter for policy-makers for regulating AI in safety critical areas where trustworthiness is a must, as suggested by the European Commission's outline for a legal framework for AI:

> *'In the light of the recent advances in artificial intelligence (AI), the serious negative consequences of its use for EU citizens and organisations have led to multiple initiatives from the European Commission to set up the principles of a trustworthy and secure AI. Among the identified requirements, the concepts of robustness and explainability of AI systems have emerged as key elements for a future regulation of this technology'.*
>
> – Europ. Comm. JCR Tech. Rep. (January 2020) [42].
>
> *'On AI, trust is a must, not a nice to have. [. . .] The new AI regulation will make sure that Europeans can trust what AI has to offer. [. . .] High-risk AI systems will be subject to strict obligations before they can be put on the market: [requiring] High level of robustness, security and accuracy'.*
>
> – Europ. Comm. outline for legal AI (April 2021) [27].

The concern is also shared on the American continent, especially regarding security and military applications. Indeed, the US Department of Defence has spent millions of dollars through DARPA on project calls to cure the instability problem. The strong regulatory emphasis on trustworthiness, stability (robustness), security and accuracy leads to potential serious consequences given that modern AI techniques are universally non-robust. Current state-of-the-art AI techniques may be illegal in certain key sectors given their fundamental lack of robustness. The lack of a cure for the instability phenomenon in modern AI suggests a methodological barrier applicable to current AI techniques and hence should be viewed in connection with Smale's 18th problem on the limits of AI.

1.1. Main theorems – methodological barriers, Smale's 18th problem and the limits of AI

Smale's 18th problem, from the list of mathematical problems for the 21st century [71], echoes Turing's paper from 1950 [76] on the question of existence of AI. Turing asks if a computer can think and suggests the imitation game (Turing test) as a test for his question about AI. Smale takes the question even further and asks in his 18th problem: *what are the limits of AI?* The question is followed by a discussion on the problem that ends as follows. '*Learning is a part of human intelligent activity. The corresponding mathematics is suggested by the theory of repeated games, neural nets and genetic algorithms'*.

Our contribution to the program on Smale's 18th problem are the following limitations and methodological barriers on modern AI highlighted in (I) and (II). These results provide mathematical answers to the question on why there has been no solution to the instability problem.

(I) **Theorem 2.2**: There are basic methodological barriers in state-of-the-art DL based on ReLU NNs. Indeed, any training procedure based on training ReLU NNs for many simple classification problems with a fixed architecture will yield neural networks that are either inaccurate or unstable (if accurate) – despite the provable existence of both accurate and stable neural networks for the same classification problems. Moreover, variable dimensions on NNs is necessary for stability for ReLU NNs.

Theorem 2.2 points towards the paradox that accurate and stable neural networks exist; however, modern algorithms do not compute them. This yields the question:

> *If the existence of neural networks can be proven, can one also find algorithms that compute them? In particular, there are cases in mathematics where provable existence implies computability, but will this be the case for neural networks?*

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

We address this question even for provable existence of NNs in standard training scenarios.

(II) **Theorem 3.5**: There are NNs that provably exist as approximate minimisers to standard optimisation problems with standard cost functions; however, no randomised algorithm can compute them with probability better than $1/2$.

A detailed account of the results and the consequences can be found in Sections 2 and 3.

1.2. Phase transitions and generalised hardness of approximation (GHA)

Theorem 3.5 can be understood within the framework of *generalised hardness of approximation* (GHA) [2, 6, 7, 9, 25, 33, 37, 45, 81], which describes a specific phase transition phenomenon. In many cases, it is straightforward to compute an ϵ-approximation to a solution of a computational problem for $\epsilon > \epsilon_1 > 0$. However, when $\epsilon < \epsilon_1$ (the approximation threshold), a phase transition occurs, wherein it is suddenly difficult, or even infeasible, to obtain an ϵ-approximation. This difficulty could manifest as non-computability or intractability (e.g., non-polynomial time complexity). GHA extends the concept of hardness of approximation [5] from discrete computations to more general computational problems.

In particular, Theorem 3.5 establishes lower bounds on the approximation threshold $\epsilon_1 > 0$ for computing NNs in classification tasks. This theorem builds upon the initial work on GHA introduced in [9] for convex optimisation (see also Problem 5 (J. Lagarias) in [33]) and further developed in [25, 37] for NNs in AI and inverse problems. The theory of GHA is part of the larger framework of the Solvability Complexity Index (SCI) hierarchy [11–13, 23–26, 43, 44].

2. Main results I – trained NNs become unstable despite the existence of stable and accurate NNs

In this section, we will explain our contributions to understanding the instability phenomenon. We consider the simplest DL problem of approximating a given classification function:

$$f{:}[0,1]^d \to \{0,1\}, \tag{2.1}$$

by constructing a neural network from training data. Let $\mathcal{NN}_{\mathbf{N},L}$ with $\mathbf{N} := (N_L = 1, N_{L-1}, \ldots, N_1, N_0 = d)$ denote the set of all L-layer neural networks (with $L \geq 2$) under the ReLU non-linearity with N_ℓ neurons in the ℓ-th layer (see Section 5.1 for definitions and explanations of these concepts). We assume that the cost function $\mathcal{R}$ is an element of

$$\mathcal{CF}_r = \{\mathcal{R} : \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}_+ \cup \{\infty\} \mid \mathcal{R}(v,w) = 0 \text{ iff } v = w\}. \tag{2.2}$$

Remark 2.1 (Choice of cost functions). Note that the choice of class of cost functions defined in (2.2) will be used in Theorem 2.2 is to demonstrate how one can achieve great generalisability properties of the trained network. It is worth mentioning however that we show that expanding this class to include, for example, regularised cost functions will not cure the instability phenomenon (see Section 2.1 (II) for more detail).

As we will discuss the stability of neural networks, we introduce the idea of *well-separated and stable sets* to exclude pathological examples whereby the training and validation sets have elements that are arbitrarily close to each other in a way that could make the classification function jump subject to a small perturbation. Specifically, given a classification function $f{:}[0,1]^d \to \{0,1\}$, we define the family of well-separated and stable sets $\mathcal{S}_\delta^f$ with separation at least 2δ according to

$$\mathcal{S}_\delta^f = \Big\{ \{x^1, \ldots, x^m\} \subset [0,1]^d \mid m \in \mathbb{N}, \\ \min_{x^i \neq x^j} \|x^i - x^j\|_\infty \geq 2\delta, f(x^j + y) = f(x^j) \text{ for } \|y\|_\infty < \delta \text{ satisfying } x^j + y \in [0,1]^d \Big\}.$$

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

We also set $r \vee s$ to be the maximum of r and s and $r \wedge s$ to be the minimum of r and s. Finally, we use the notation $\mathcal{B}_\epsilon^\infty$ to refer to the open ball of radius ϵ in the ℓ^∞ norm. With this notation established, we are now ready to state our first main result.

Theorem 2.2 (Instability of trained NNs despite existence of a stable NN). *There is an uncountable collection $\mathcal{C}_1$ of classification functions f as in (2.1) – with fixed $d \geq 2$ – and a constant $C > 0$ such that the following holds. For every $f \in \mathcal{C}_1$, any norm $\|\cdot\|$ and every $\epsilon > 0$, there is an uncountable family $\mathcal{C}_2$ of probability distributions on $[0,1]^d$ so that for any $\mathcal{D} \in \mathcal{C}_2$, any neural network dimensions $\mathbf{N} = (N_L = 1, N_{L-1}, \ldots, N_1, N_0 = d)$ with $L \geq 2$, any $\mathrm{p} \in (0,1)$, any positive integers q, r, s with*

$$r + s \geq C \max\left\{ \mathrm{p}^{-3},\ q^{3/2}\left[(N_1+1)\cdots(N_{L-1}+1)\right]^{3/2} \right\}, \tag{2.3}$$

any training data $\mathcal{T} = \{x^1, \ldots, x^r\}$ and validation data $\mathcal{V} = \{y^1, \ldots, y^s\}$, where the x^j and y^j are drawn independently at random from $\mathcal{D}$, the following happens with probability exceeding $1 - \mathrm{p}$.

(i) *(Success – great generalisability). We have $\mathcal{T}, \mathcal{V} \in \mathcal{S}^f_{\varepsilon((r\vee s)/\mathrm{p})}$, where $\varepsilon(n) = (Cn)^{-4}$, and, for every $\mathcal{R} \in \mathcal{CF}_r$, there exists a ϕ such that*

$$\phi \in \operatorname*{arg\,min}_{\varphi \in \mathcal{NN}_{\mathbf{N},L}} \mathcal{R}\left(\{\varphi(x^j)\}_{j=1}^r, \{f(x^j)\}_{j=1}^r\right) \tag{2.4}$$

and

$$\phi(x) = f(x) \quad \forall x \in \mathcal{T} \cup \mathcal{V}. \tag{2.5}$$

(ii) *(Any successful NN in $\mathcal{NN}_{\mathbf{N},L}$ – regardless of architecture – becomes universally unstable). Yet, for any $\hat{\phi} \in \mathcal{NN}_{\mathbf{N},L}$ (and thus, in particular, for $\hat{\phi} = \phi$) and any monotonic $g\colon \mathbb{R} \to \mathbb{R}$, there is a subset $\tilde{\mathcal{T}} \subset \mathcal{T} \cup \mathcal{V}$ of the combined training and validation set of size $|\tilde{\mathcal{T}}| \geq q$, such that there exist uncountably many universal adversarial perturbations $\eta \in \mathbb{R}^d$ so that for each $x \in \tilde{\mathcal{T}}$ we have*

$$|g \circ \hat{\phi}(x+\eta) - f(x+\eta)| \geq 1/2, \quad \|\eta\| < \epsilon, \quad |supp(\eta)| \leq 2. \tag{2.6}$$

(iii) *(Other stable and accurate NNs exist). However, there exists a stable and accurate neural network ψ that satisfies $\psi(x) = f(x)$ for all $x \in \mathcal{B}_\epsilon^\infty(\mathcal{T} \cup \mathcal{V})$, when $\epsilon \leq \varepsilon((r \vee s)/\mathrm{p})$.*

We remark in passing that the training and validation data $\mathcal{T}$ and $\mathcal{V}$ in Theorem 2.2 are technically not sets, but randomised multisets, as some of the samples x^j or y^j may be repeated.

Remark 2.3 (The role of g in (ii) in Theorem 2.2). The purpose of the monotone function $g\colon \mathbb{R} \to \mathbb{R}$ in (ii) in Theorem 2.2 is to make the theorem as general as possible. In particular, a popular way of creating an approximation to f is to have a network combined with a thresholding function g. This would potentially increase the approximation power compared to only having a neural network; however, Theorem 2.2 shows that adding such a function does not cure the instability problem.

2.1. Interpreting Theorem 2.2

In this section, we discuss in detail the implications of Theorem 2.2 with regard to Smale's 18th problem. First, note that Theorem 2.2 demonstrates a methodological barrier applicable to current DL approaches. This does not imply that the instability problem in classification cannot be resolved, but it does imply that in order to overcome these instability issues one will have to change the methodology. Second, Theorem 2.2 provides guidance on which methodologies will not solve the instability issues. In order to make the exposition easy to read, we will now summarise in non-technical terms what Theorem 2.2 says.

(I) *Performance comes at a cost – Accurate DL methods inevitably become unstable*. Theorem 2.2 shows that there are basic classification functions and distributions where standard DL methodology yields trained NNs with great success in terms of generalisability and performance – note

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

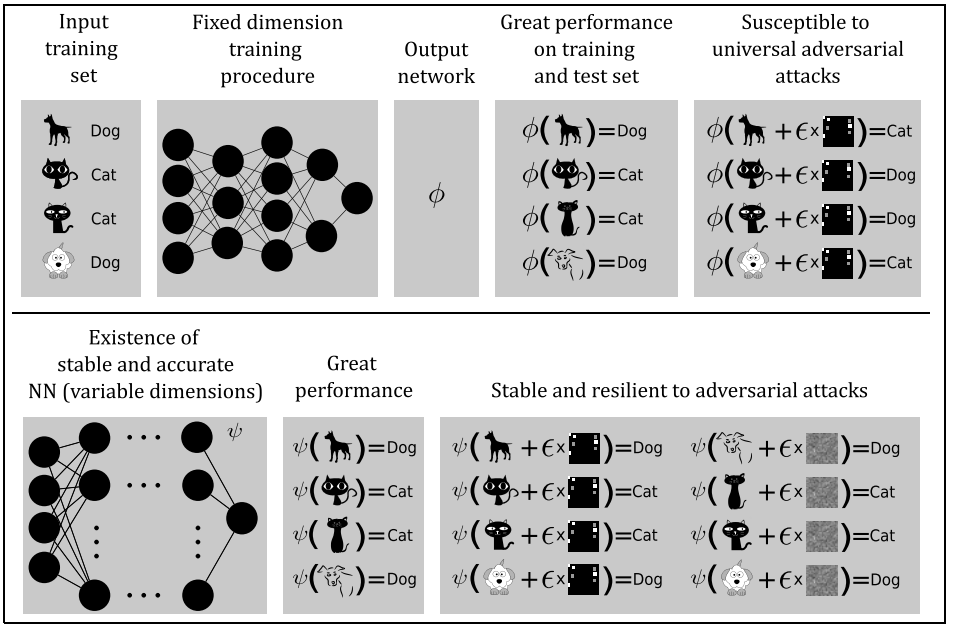

Figure 1 *(**Training with fixed architecture yields instability – variable dimensions on NNs is necessary for stability for ReLu NNs**). A visual interpretation of Theorem 2.2. A fixed dimension training procedure can lead to excellent performance and yet be highly susceptible to adversarial attacks, even if there exists a NN which has both great performance and excellent stability properties. However, such a stable and accurate ReLu network must have variable dimensions depending on the input.*

that the size of the validation set $\mathcal{V}$ in Theorem 2.2 can become arbitrary large. However, (2.3) demonstrates how greater success (better generalisability) implies more instabilities. Indeed, the NNs – regardless of architecture and training procedure – become either successful and unstable, or unsuccessful.

(II) *There is no remedy within the standard DL framework*. Note that (ii) in Theorem 2.2 demonstrates that there is no remedy within the standard DL framework to cure the instability issue described in Theorem 2.2. The reason why is that standard DL methods will fix the architecture (i.e., the class $\mathcal{N}$ of NNs that one minimises over) of the neural networks. Indeed, the misclassification in (2.6) happens for any neural network $\hat{\phi} \in \mathcal{NN}_{\mathbf{N},L}$. This means that, for example, zero loss training [66], or any attempt using adversarial training [39, 56] – that is, computing

$$\min_{\phi \in \mathcal{N}} E_{x \sim \mathcal{D}} \max_{z \in \mathcal{U}} \mathcal{L}(\phi(x+z), f(x)),$$

where $\mathcal{N} \subset \mathcal{NN}_{\mathbf{N},L}$ is any collection of NNs described by a specific architecture, $\mathcal{U} \subset \mathbb{R}^d$ and $\mathcal{L}$ is any real-valued cost function – will not solve the problem. In fact, (ii) in Theorem 2.2 immediately implies that adversarial training will reduce the performance if it increases the stability.

(III) *There are accurate and stable NNs, but DL methods do not find them*. Note that (iii) in Theorem 2.2 demonstrates that there are stable and accurate NNs for many classification problems where DL methods produce unstable NNs. Thus, the problem is not that stable and accurate NNs do not exist; instead, the problem is that DL methods do not find them. The reason is that the dimensions and architectures of the stable and accurate networks will change depending on the size and properties of the data set.

(IV) *Why instability? – Unstable correlating features are picked up by the trained NN*. In addition to the statement of Theorem 2.2, the proof techniques illustrate the root causes of the problem. The

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

reason why one achieves the great success described by (i) in Theorem 2.2 is that the successful NN picks up a feature in the training set that correlates well with the classification function but is itself unstable. This phenomenon is empirically investigated in [50].

(V) *No training model where the dimensions of the NNs are fixed can cure instability*. Note that (ii) in Theorem 2.2 describes the reason for the failure of the DL methodology to produce a stable and accurate NN. Indeed, as pointed out above, the dimensions of the stable and accurate NN will necessary change with the amount of data in the training and validation set.

(VI) *Adding more training data cannot cure the general instability problem*. Note that (2.3) in Theorem 2.2 shows that adding more training data will not help. In fact, it can make the problem worse. Indeed, (2.3) allows s – the number of elements in the test set – to be set so that $s = 1$, and r – the number of training data – can be arbitrary large. Hence, if r becomes large and s is small, then the trained NN – if successful – will (because of (ii)) start getting instabilities on the training data. In particular, the network has seen the data, but will misclassify elements arbitrary close to seen data.

(VII) *Comparison with the No-Free-Lunch Theorem*. The celebrated No-Free-Lunch Theorem has many forms. However, the classical impossibility result we refer to (Theorem 5.1 in [70]) states that for any learning algorithm for classification problems there exists a classification function f and a distribution $\mathcal{D}$ that makes the algorithm fail. Our Theorem 2.2 is very different in the way that it is about instability. Moreover, it is an impossibility result specific for DL. Thus, the statements are much stronger. Indeed – in contrast to the single classification function f and distribution $\mathcal{D}$ making a fixed algorithm fail in the No-Free-Lunch Theorem – Theorem 2.2 shows the existence of uncountably many classification functions f and distributions $\mathcal{D}$ such that for any fixed architecture DL will either yield unstable and successful NNs or unsuccessful NNs. This happens despite the existence of stable and accurate NNs for exactly the same problem. Moreover, Theorem 2.2 shows how NNs can generalise well given relatively few training data compared to the test data, but at the cost of non-robustness (note that this is in contrast to the No-Free-Lunch theorem wherein few training samples leads to a lack of generalisation). See also [40] for other 'No-Free-Lunch' theorems.

3. Main results II – NNs may provably exist, but no algorithm can compute them

The much celebrated Universal Approximation Theorem is widely used as an explanation for the wide success of NNs in the sciences and in AI, as it guarantees the existence of neural networks that can approximate arbitrary continuous functions. In essence, any task that can be handled by a continuous function can also be handled by a neural network.

Theorem 3.1 (Universal Approximation Theorem [67]). *Suppose that $\sigma \in C(\mathbb{R})$, where $C(\mathbb{R})$ denotes the set of continuous functions on $\mathbb{R}$. Then the set of neural networks with non-linearity σ is dense in $C(\mathbb{R}^d)$ in the topology of uniform convergence on compact sets, if and only if σ is not a polynomial.*

Theorem 2.2 illustrates basic methodological barriers in DL and suggests the following fundamental question:

> *If we can prove that a stable and well generalisable neural network exists, why do algorithms fail to compute them?*

This question is not only relevant because of Theorem 2.2 but also the Universal Approximation Theorem that demonstrates – in theory – that there are very few limitations on what NNs can do. Yet, there is clearly a barrier that the desirable NNs that one can prove exist – as shown in Theorem 2.2 and in many cases follow from the Universal Approximation Theorem – are not captured by standard algorithms.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

3.1. The weakness of the Universal Approximation Theorem – When will existence imply computability?

The connection between the *provable existence* of a mathematical object and its *computability* (that there is an algorithm that can compute it) touches on the foundations of mathematics [72]. Indeed, there are cases in mathematics – with the ZFC[1] axioms – where the fact that one can prove mathematically a statement about the existence of the object will imply that one can find an algorithm that will compute the object when it exists. Consider the following example:

Example 3.2 (When provable existence implies computability). *Consider the following basic computational problem concerning Diophantine equations:*

> *Let Θ be a collection of polynomials in $\mathbb{Z}[x_1, x_2, \ldots, x_n]$ with integer coefficients, where $n \in \mathbb{N}$ can be arbitrary. Given a polynomial $p \in \Theta$, does there exist an integer vector $a \in \mathbb{Z}^n$ such that $p(a) = 0$, and if so, compute such an a.*

Note that in this case we have that 'being able to prove $\Rightarrow$ being able to compute' as the following implication holds for the ZFC model [68]:

$$\boxed{\begin{array}{c}\text{For any polynomial } p \in \Theta, \text{ one can prove – given the ZFC axioms – that there exists} \\ \text{an integer vector } a \in \mathbb{Z}^n \text{ such that } p(a) = 0, \text{ or a negation of this statement.}\end{array}}$$

$$\Downarrow \tag{3.1}$$

$$\boxed{\begin{array}{c}\text{There exists an algorithm } \Gamma \text{ taking any polynomial } p \in \Theta \text{ such that } \Gamma(p) = \text{'no' if} \\ \text{there is no } a \in \mathbb{Z}^n \text{ such that } p(a) = 0, \text{ otherwise } \Gamma(p) = a \text{ where } a \in \mathbb{Z}^n \text{ such that} \\ p(a) = 0.\end{array}}$$

The above implication is true, subject to ZFC being consistent and that theorems in ZFC about integers are true [68]. In particular, being able to prove existence or not of integer-valued zeroes of polynomials in Θ implies the existence of an algorithm that can compute integer-valued zeroes of polynomials in Θ and determine if no integer-valued zero exists.

There is a substantial weakness with the Universal Approximation Theorem and the vast literature on approximation properties of NNs, in that they provide little insight into how NNs should be computed, or indeed if they can be computed. As Example 3.2 suggests, there are cases where provable existence implies computability. Hence, we are left with the following basic problem:

> *If neural networks can be proven to exist, will there exist algorithms that can compute them? If this is not the case in general, what about neural networks that can be proven to be approximate minimisers of basic cost functions?*

As we see in the next sections, the answer to the above question is rather delicate.

Remark 3.3. Although the Universal Approximation Theorem does not directly apply to non-constant classification functions in the class (2.1), if we consider a classification function restricted to a finite set (e.g., training and validation sets), it will have a continuous extension and hence the Universal Approximation Theorem will apply. Furthermore, recent results discuss existence theorems in the general setting of (2.1) [53].

[1] Zermelo-Fraenkel axiomatic system with the axiom of choice, which is the standard axiomatic system for modern mathematics.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

3.2. Inexactness and floating point arithmetic

The standard model for computation in most modern software is floating point arithmetic. This means that even a rational number like $1/3$ will be approximated by a base-2 approximation. Moreover, the floating point operations yield errors, that – in certain cases – can be analysed through backward error analysis, which typically show how the computed solution in floating point arithmetic is equivalent to a correct computation with an approximated input. Hence, in order to provide a realistic analysis, we use the model of computation with inexact input as emphasised by S. Smale in his list of mathematical problems for the 21st century:

> *'But real number computations and algorithms which work only in exact arithmetic can offer only limited understanding. Models which process approximate inputs and which permit round-off computations are called for'.*
>
> – S. Smale (from the list of mathematical problems for the 21st century [71])

To model this situation, we shall assume that an algorithm designed to compute a neural network is allowed to see the training set to an arbitrary accuracy decided at runtime. More precisely, for a given training set $\mathcal{T} = \{x^1, x^2, \dots, x^r\}$, we assume that the algorithm (a Turing [75] or Blum-Shub-Smale (BSS) [18] machine) is equipped with an oracle $\mathscr{O}$ that can acquire the true input to any accuracy ϵ. Specifically, the algorithm cannot access the vectors $x^1, x^2, \dots, x^r$ but rather, for any $k \in \mathbb{N}$, it can call the oracle $\mathscr{O}$ to obtain $x^{1,k}, x^{2,k}, \dots, x^{r,k}$ such that

$$\|x^{j,k} - x^j\|_\infty \leq 2^{-k}, \qquad \text{for } i = 1, 2, \dots, r \text{ and } \forall k \in \mathbb{N}, \tag{3.2}$$

see Section 5.4.4 for details.

Another key assumption when discussing the success of the algorithm is that it must be 'oracle agnostic', that is, it must work with any choice of the oracle $\mathscr{O}$ satisfying (3.2). In the Turing model, the Turing machine accesses the oracle via an oracle tape and in the BSS model the BSS machine accesses the oracle through an oracle node. This extended computational model of having inexact input is standard and can be found in many areas of the mathematical literature – we mention only a small subset here: Bishop [17], Cucker & Smale [28], Fefferman & Klartag [34, 35], Ko [51] and Lovász [54].

3.3. Being able to prove existence may imply being able to compute – but not in DL

We now examine the difference between being able to prove the existence of a neural network and the ability to compute it, even in the case when the neural network is an approximate minimiser. Recall the typical training scenario of neural networks in (2.4) where one tries to find

$$\phi \in \underset{\varphi \in \mathcal{NN}_{\mathbf{N},L}}{\arg\min} \mathcal{R}\left(\{\varphi(x^j)\}_{j=1}^r, \{f(x^j)\}_{j=1}^r\right),$$

where f is the decision function, $\mathcal{R}$ is the cost function and $\mathcal{T} = \{x^j\}_{j=1}^r$ is the training set. However, one will typically not reach the actual minimiser, but rather an approximation. Hence, we define *the approximate argmin*.

Definition 3.4 (The approximate argmin). *Given an $\epsilon > 0$, an arbitrary set X, a totally ordered set Y and a function $g\colon X \to Y$, the approximate* $\operatorname{argmin}_\epsilon$ *over some subset $S \subset X$ is defined by*

$$\underset{x \in S}{\operatorname{argmin}_\epsilon}\, g(x) := \{x \in S \mid g(x) \leq g(y) + \epsilon \ \ \forall\, y \in X\} \tag{3.3}$$

To accompany the idea of the approximate argmin, we will also consider cost functions that are bounded with respect to the ℓ^∞ norm:

$$\mathcal{CF}_r^{\epsilon,\hat{\epsilon}} = \{\mathcal{R} \in \mathcal{CF}_r : \mathcal{R}(v, w) \leq \epsilon \implies \|v - w\|_\infty \leq \hat{\epsilon}\}. \tag{3.4}$$

The computational problem we now consider is to compute a neural network that is an approximate minimiser and evaluate it on the training set (this is the simplest task that we should be able to compute):

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

$$\phi(x^j), \qquad \phi \in \operatorname*{arg\,min}_{\varphi \in \mathcal{NN}_{\mathbf{N},L}} \epsilon \, \mathcal{R}\left(\{\varphi(x^j)\}_{j=1}^r, \{f(x^j)\}_{j=1}^r\right), \quad \epsilon > 0, \quad j = 1, \ldots, r. \tag{3.5}$$

Hence, an algorithm Γ trying to compute (3.5) takes the training set $\mathcal{T}$ as an input (or to be precise, it calls oracles providing approximations to the x^js to any precision, see seeSection 5.4.4 for details) and outputs a vector in $\mathbb{R}^r$. Hopefully, $\|\Gamma(\mathcal{T}) - \{\phi(x^j)\}_{j=1}^r\|$ is sufficiently small.

The next theorem shows that even if one can prove the existence of neural networks that are approximate minimisers to optimisation problems with standard cost functions, one may not be able to compute them.

Theorem 3.5 (NNs may provably exist, but no algorithm can compute them). *There is an uncountable collection $\mathcal{C}_1$ of classification functions f as in (2.1) – with fixed $d \geq 2$ – such that the following holds. For*

(1) any neural network dimensions $\mathbf{N} = (N_L = 1, N_{L-1}, \ldots, N_1, N_0 = d)$ with $L \geq 2$,

(2) any $r \geq 3(N_1 + 1) \cdots (N_{L-1} + 1)$,

(3) any $\epsilon > 0$, $\hat{\epsilon} \in (0, 1/2)$ and cost function $\mathcal{R} \in \mathcal{CF}_r^{\epsilon,\hat{\epsilon}}$,

(4) any randomised algorithm Γ,

(5) any $\mathrm{p} \in [0, 1/2)$,

there is an uncountable collection $\mathcal{C}_2$ of training sets $\mathcal{T} = \{x^1, x^2, \ldots, x^r\} \in \mathcal{S}^f_{\varepsilon'(r)}$ such that for each $\mathcal{T} \in \mathcal{C}_2$ there exists a neural network ϕ, where

$$\phi \in \operatorname*{argmin}_{\varphi \in \mathcal{NN}_{\mathbf{N},L}} \epsilon \, \mathcal{R}\left(\{\varphi(x^j)\}_{j=1}^r, \{f(x^j)\}_{j=1}^r\right),$$

however, the algorithm Γ applied to the input $\mathcal{T}$ will fail to compute any such ϕ in the following way:

$$\mathbb{P}\Big(\|\Gamma(\mathcal{T}) - \{\phi(x^j)\}_{j=1}^r\|_* \geq 1/4 - 3\hat{\epsilon}/4\Big) > p,$$

where $* = 1, 2$ or ∞.

3.4. A missing theory in AI – Which NNs can be computed?

If provable existence results about NNs were to imply that they could be computed by algorithms, the research effort to secure stable and accurate AI would – in most cases – be about finding the right algorithms that we know exist, due to the many neural network existence results [29, 67]. In particular, the key limitation for providing stable and accurate AI via DL – at least in theory – would be the capability of the research community. However, as Theorem 3.5 reveals, the simplest existence results of NNs as approximate minimisers do not imply that they can be computed. Therefore, the research effort moving forward must be about which NNs that can be computed by algorithms and how. Indeed, the limitations of DL as an instrument in AI will be determined by the limitations of existence of algorithms and their efficiency for computing NNs.

Remark 3.6 (Theorem 3.5 is independent of the exact computational model). Note that the result above is independent of whether the underlying computational device is a BSS machine or a Turing machine. To achieve this, we work with a definition of an algorithm termed a general algorithm. The corresponding definitions as well as a formal statement of Theorem 3.5 are detailed inSection 5.4 and Proposition 5.26, respectively.

Remark 3.7 (Irrelevance of local minima). Note that Theorem 3.5 has nothing to do with the potential issue of the optimisation problem having several local minima. Indeed, the general algorithms used in Theorem 3.5 are more powerful than any Turing machine or BSS machine as will be discussed further in Remark 5.13.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

Remark 3.8 (Hilbert's 10th problem). Finally, we mention in passing that Theorem 3.5 demonstrates – in contrast to Hilbert's 10th problem [57] – that non-computability results in DL do not prevent provable existence results. Indeed, because of the implication in (3.1) and the non-computability of Hilbert's 10th problem [57, 68] (when Θ is the collection of all polynomials with integer coefficients in Example 3.2), there are infinitely many Diophantine equations for which one cannot prove existence of an integer solution – or a negation of the statement.

4. Connection to previous work

The literature documenting the instability phenomenon in DL is so vast that we can only cite a tiny subset here [2–4, 20, 31, 32, 36, 39, 47, 49, 61, 62, 69, 74, 77], see the references in the survey paper [3] for a more comprehensive collection. Below we will highlight some of the most important connections to our work:

(i) ***Universality of instabilities in AI***. A key feature of Theorem 2.2 is that it demonstrates how the perturbations are universal, meaning that one adversarial perturbation works for all the cases where the instability occurs – as opposed to a specific input-dependent adversarial perturbation. The DeepFool program [32, 61, 62] – created by S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi and P. Frossard – was the first to establish empirically that adversarial perturbations can be made universal, and this phenomenon is also universal across different methods and architectures. For recent and related developments, see D. Higham and I. Tyukin et al. [77, 78] which describe instabilities generated by perturbations to the structure of a neural network, [8, 73] wherein instability to randomised perturbations are considered, as well as the results by L. Bungert and G. Trillos et al. [19], and S. Wang, N. Si, and J. Blanchet [79].

(ii) ***Approximation theory and numerical analysis***. There is a vast literature proving existence results of ReLU networks for DL, investigating their approximation power, where the recent work of R. DeVore, B. Hanin and G. Petrova [29] also provides a comprehensive account of the contemporary developments. The huge approximation literature on existence results and approximation properties of NNs prior to the year 2000 is well summarised by A. Pinkus in [67]. Our results suggest a program combining recent approximation theory [29, 41] results with foundations of mathematics and numerical analysis to characterise the NNs that can be computed by algorithms. This aligns with the work of B. Adcock and N. Dexter [1] that demonstrates the gap between what algorithms compute and the theoretical existence of NNs in function approximation with deep NNs. Note that results on existence of algorithms in learning – with performance and stability guarantees – do exist (see the work of P. Niyogi, S. Smale and S. Weinberger [63]), but so far not in DL.

(iii) ***Mathematical explanation of instability and impossibility results***. Our paper is very much related to the work of H. Owhadi, C. Scovel and T. Sullivan [64, 65] who 'observe that learning and robustness are antagonistic properties'. The recent work of I. Tyukin, D. Higham and A. Gorban [77] and A. Shafahi, R. Huang, C. Studer, S. Feizi and T. Goldstein [69] demonstrate how the instability phenomenon increases with dimension showing universal lower bound on stability as a function of the dimension of the domain of the classification function. Note, however, that the results in this paper are independent of dimensions. The work of A. Fawzi, H. Fawzi and O. Fawzi [31] is also related; however, their results are about adversarial perturbations for any function, which is somewhat different from the problem that DL is unstable to perturbations that humans do not perceive. In particular, our results focus on how DL becomes unstable despite the fact that there is another device (in our case another NN) or a human that can be both accurate and stable.

(iv) ***Proof techniques – The SCI hierarchy***. Initiated in [43], the mathematics behind the SCI hierarchy provides a variety of techniques to show lower bounds and impossibility results for algorithms – in a variety of mathematical fields – that provide the foundations for the proof techniques in this paper, see the works by V. Antun, J. Ben-Artzi, M. Colbrook, A. C. Hansen, M. Marletta, O. Nevanlinna,

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

F. Rösler and M. Seidel. [11–13, 25, 43]. This is strongly related to the work of S. Weinberger [80] on the existence of algorithms for computational problems in topology. The authors of this paper have also extended the SCI framework [9] in connection with the extended Smale's 9th problem.

(v) ***Robust optimisation***. Robust optimisation [14–16], pioneered by A. Ben-Tal, L. El Ghaoui and A. Nemirovski, is an essential part of optimisation theory addressing sensitivity to perturbations and inexact data in optimisation problems. There are crucial links to our results – indeed, a key issue is that the instability phenomenon in DL leads to non-robust optimisation problems. In fact, there is a fundamental relationship between Theorem 2.2, Theorem 3.5 and robust optimisation. Theorem 3.5 yields impossibility results in optimisation, where non-robustness is a key element. The big question is whether stable and accurate NNs – with variable dimensions – that exist as a result of Theorem 2.2 can be shown to be approximate minimisers of robust optimisation problems. This leads to the final question, would such problems be computable and have efficient algorithms? The results in this paper can be viewed as an instance of where robust optimisation meets the SCI hierarchy. This was also the case in the recent results on the extended Smale's 9th problem [9].

5. Proofs of the main results

5.1. Some well-known definitions and ideas from DL

In this section, we outline some basic well-known definitions and explain the notation that will be useful for this paper. Many of these definitions can be found in [38]. For a vector $x \in \mathbb{R}^{N_1}$, we denote x_i by the ith coordinate. Similarly, for a matrix $A \in \mathbb{R}^{N_1 \times N_2}$ for some dimensions $N_1 \in \mathbb{N}$ and $N_2 \in \mathbb{N}$, we denote $A_{i,j}$ by the entry of A contained on the ith row and the jth column.

Recall that for natural numbers n_1, n_2, an affine map $W{:}\mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ is a map such that there exists $A \in \mathbb{R}^{n_2 \times n_1}$ and $b \in \mathbb{R}^{n_2}$ so that for all $x \in \mathbb{R}^{n_1}$, $Wx = Ax + b$. Let L, d be natural numbers and let $\mathbf{N} := (N_L = 1, N_{L-1}, \ldots, N_1, N_0)$ be a vector in $\mathbb{N}^{L+1}$ with $N_0 = d$. An *neural network with dimensions* $(\mathbf{N}, L)$ is a map $\phi : \mathbb{R}^d \to \mathbb{R}$ such that

$$\phi = W^L \sigma W^{L-1} \sigma W^{L-2} \ldots \sigma W^1$$

where, for $l = 1, 2, \ldots, L$, the map W^l is an affine map from $\mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$, that is, $W^l x^l = A^l x^l + b^l$ where $b^l \in \mathbb{R}^{N_l}$ and $A^l \in \mathbb{R}^{N_l \times N_{l-1}}$. The map $\sigma : \mathbb{R} \to \mathbb{R}$ is interpreted as a coordinate-wise map and is called the *non-linearity* or *activation function*: typically, σ is chosen to be continuous and non-polynomial [67].

In this paper, we focus on the well-known *ReLU* non-linearity, which we denote by ρ. More specifically, for $x \in \mathbb{R}$, we define $\rho(x)$ by $\rho(x) = 0$ if $x < 0$ and $\rho(x) = x$ if $x \geq 0$. We denote all neural networks with dimensions $(\mathbf{N}, L)$ and the ReLU non-linearity by $\mathcal{NN}_{\mathbf{N},L}$. This will be the central object for our arguments.

Remark 5.1. Although we focus on the ReLU non-linearity, it is possible to use the techniques presented in this paper to prove similar results for other non-linearities like the leaky ReLU [55] ρ^{leaky} and the parameterised ReLU [46] ρ_α^{param} where

$$\rho^{\text{leaky}}(x) = \begin{cases} 0.01 \cdot x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}, \quad \rho_\alpha^{param}(x) = \begin{cases} \alpha x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

In this paper, the most common norms we use are the ℓ^p norms: for a vector $x \in \mathbb{R}^d$ for some natural number d and some $p \in [1, \infty)$, the ℓ^p norm of x (which we denote by $\|x\|_p$) is given by $\|x\|_p = (\sum_{i=1}^d |x_i|^p)^{1/p}$. We also define the ℓ^∞ norm of x (which we denote by $\|x\|_\infty$) by $\|x\|_\infty := \max_{i=1,2,\ldots,d} |x_i|$. It is easy to see the following inequality: $\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1$. We will denote the ball of radius ϵ about x in the infinity norm by $\mathcal{B}_\epsilon^\infty(x)$, that is, $\mathcal{B}_\epsilon^\infty(x) = \{y \in \mathbb{R}^d \mid \|y - x\|_\infty \leq \epsilon\}$. For a set S, we denote $\mathcal{B}_\epsilon^\infty(S)$ by $\cup_{x \in S} \mathcal{B}_\epsilon^\infty(x)$.

The cost function of a neural network is used in the training procedure: typically, one attempts to compute solutions to (2.4) where the function $\mathcal{R}$ is known as the cost function. In optimisation theory,

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

the cost function is sometimes known as the objective function and sometimes the loss function. Some standard choices for $\mathcal{R}$ include the following:

(1) *Cross-entropy cost function*, where $\mathcal{R}$ is defined by

$$\mathcal{R}(\{v^j\}_{j=1}^r, \{w^j\}_{j=1}^r) := -\frac{1}{r}\sum_{j=1}^{r}\left(w^j \log(v^j) + (1-w^j)\log(1-v^j)\right)$$

The cross-entropy function is only defined if $v^j \in [0,1]$: it is easy to extend this definition to $\mathcal{R}(\{v^j\}_{j=1}^r, \{w^j\}_{j=1}^r) := \infty$ when $v^j \notin [0,1]$ for some j.

(2) *Mean square error*, where $\mathcal{R}$ is defined by

$$\mathcal{R}(\{v^j\}_{j=1}^r, \{w^j\}_{j=1}^r) := \frac{1}{r}\|\{w^j\}_{j=1}^r - \{v^j\}_{j=1}^r\|_2^2$$

(3) *Root mean square error*, where $\mathcal{R}$ is defined by

$$\mathcal{R}(\{v^j\}_{j=1}^r, \{w^j\}_{j=1}^r) := \frac{1}{r}\|\{w^j\}_{j=1}^r - \{v^j\}_{j=1}^r\|_2$$

(4) *Mean absolute error*, where

$$\mathcal{R}(\{v^j\}_{j=1}^r, \{w^j\}_{j=1}^r) := \frac{1}{r}\|\{w^j\}_{j=1}^r - \{v^j\}_{j=1}^r\|_1$$

Note that each of these functions are in $\mathcal{CF}_r$ where $\mathcal{CF}_r$ is defined in (2.2).

5.2. Lemmas and definitions common to the proofs of both Theorems 2.2 and 3.5

For both theorems, the proof relies on the points $x^{k,\delta} \in \mathbb{R}^{N_0}$, defined for $k \in \mathbb{N}$, $\delta \geq 0$, $\kappa \in [1/4, 3/4]$ and $a \in [1/2, 1]$ as follows:

$$x^{k,\delta} = \begin{cases} (a(k+1-\kappa)^{-1}, 0, \ldots, 0), & \text{if } k \text{ is odd} \\ (a(k+1-\kappa)^{-1}, \delta, 0, 0, \ldots, 0), & \text{if } k \text{ is even} \end{cases}. \tag{5.1}$$

Both theorems also rely on some classification functions f_a for $a \in [1/2, 1]$, defined as follows: we set $f_a : \mathbb{R}^{N_0} \to \{0, 1\}$

$$f_a(x) = \begin{cases} 1 & \text{if } \lceil a/x_1 \rceil \text{ is an odd integer} \\ 0 & \text{otherwise (including } x = 0) \end{cases} \tag{5.2}$$

In particular, note that for any $\delta \geq 0$, $f_a(x^{k,\delta}) = 1$ if k is even and $f_a(x^{k,\delta}) = 0$ if k is odd. The following three lemmas will be useful in both proofs. The first of these lemmas shows that finite collections of $x^{k,\delta}$ are well separated. Precisely, we will prove the following:

Lemma 5.2. *Let $a \in [1/2, 1]$, $\kappa \in [1/4, 3/4]$ and $\delta \geq 0$, and consider the points $x^{k,\delta}$ as given in (5.1) and f_a given as in (5.2). Then, for every $K \in \mathbb{N}$, we have $\{x^{1,\delta}, \ldots, x^{K,\delta}\} \in \mathcal{S}^{f_a}_{\varepsilon'(K)}$, where $\varepsilon'(n) := [(4n+3)(4n+4)]^{-1}$.*

The purpose of the next lemma is to show that if $\delta > 0$, there is a neural network that matches f_a on the $x^{k,\delta}$:

Lemma 5.3. *Let d be a natural number with $d \geq 2$, let $a \in [1/2, 1]$, $\kappa \in [1/4, 3/4]$ and $\delta > 0$, and consider the points $x^{k,\delta}$ as given in (5.1) and f_a given as in (5.2). Fix neural networks dimensions $\mathbf{N} = (N_L = 1, N_{L-1}, \ldots, N_1, N_0 = d)$ with $L \geq 2$. Then there exists a neural network $\tilde{\varphi} \in \mathcal{NN}_{\mathbf{N},L}$ with $\tilde{\varphi}(x^{k,\delta}) = f_a(x^{k,\delta})$ for all $k \in \mathbb{N}$.*

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

Finally, the next lemma will be used to give examples of sets of vectors $\mathcal{W}$ and functions f for which neural networks with fixed dimensions cannot exactly match f on $\mathcal{W}$. More precisely, we shall show the following:

Lemma 5.4. *Let $d, t, m, L, N_1, N_2, \ldots, N_L$ each be natural numbers and let $\mathcal{W}$ be a set of vectors with $\mathcal{W} = \{w^1, w^2, \ldots, w^t\} \subset \mathbb{R}^d$. Suppose that each of the following apply*

(1) $t \geq 3m \cdot (N_1 + 1)(N_2 + 1) \cdots (N_L + 1)$.

(2) $w_1^1 > w_1^2 > w_1^3 > \cdots > w_1^t$ *and* $w_j^1 = w_j^2 = \cdots = w_j^t = 0$ *for* $j = 2, \ldots, d$.

(3) $f:\mathbb{R}^d \to \{0, 1\}$ *is such that* $f(w^i) \neq f(w^{i+1})$ *for* $i = 1, 2, \ldots, t-1$.

Then for any neural network $\varphi \in \mathcal{NN}_{\mathbf{N},L}$ and any monotonic function $g:\mathbb{R} \to \mathbb{R}$, there exists a set $\mathcal{U} \subset \mathcal{W}$ such that $|\mathcal{U}| \geq m$ and $|g(\varphi(w)) - f(w)| \geq 1/2$ for all $w \in \mathcal{U}$.

The remainder of this subsection will be concerned with proving Lemmas 5.2–5.4.

5.2.1. Proof of Lemma 5.2

Proof of Lemma 5.2. We must verify that $\min_{1 \leq i < j \leq K} \|x^{i,\delta} - x^{j,\delta}\|_\infty \geq 2\varepsilon'(K)$ and that for $k \leq K$ and vectors $y \in \mathbb{R}^{N_0}$ with $\|y\|_\infty < \varepsilon'(r)$ we have $f_a(x^{k,\delta} + y) = f_a(x^{k,\delta})$.

For the first part, note that for distinct i, j with $i, j \leq K$ we have

$$\|x^{i,\delta} - x^{j,\delta}\|_\infty \geq \left| \frac{a}{i+1-\kappa} - \frac{a}{j+1-\kappa} \right| = \frac{|a(j-i)|}{(i+1-\kappa)(j+1-\kappa)} \geq \frac{1}{2(K+1-\kappa)(K-\kappa)} \tag{5.3}$$

since $a|j-i| \geq a \geq 1/2$ and the condition that $i, j \leq K$ with at least one bounded by $K-1$ implies that $(i+1-\kappa)^{-1}(j+1-\kappa)^{-1} \geq (K+1-\kappa)^{-1}(K-\kappa)^{-1}$. Since $\kappa \geq 1/4$, we get $\|x^{i,\delta} - x^{j,\delta}\|_\infty \geq \left[2(K+1-1/4)(K-1/4)\right]^{-1} \geq 2\varepsilon'(K)$.

Next, we let $k \leq K$ and $y \in \mathbb{R}^{N_0}$ be such that $\|y\|_\infty \leq \varepsilon'(K)$. We will establish that $f_a(x^{k,\delta} + y) = f_a(x^{k,\delta})$. Since $k \leq K$ and $\kappa \in [1/4, 3/4]$, we have

$$\frac{a(1-\kappa)}{(k+1-\kappa)k} > \frac{1}{(4K+3)(2K+2)} \geq y_1 \geq \frac{-1}{(4K+3)(2K+2)} \geq \frac{-a\kappa}{(k+1-\kappa)(k+1)}.$$

We claim that this implies $a(x_1^{k,\delta} + y_1)^{-1} \in (k, k+1]$. For the upper bound, note that

$$\frac{y_1}{a} \geq \frac{-\kappa}{(k+1-\kappa)(k+1)} = \frac{1}{k+1} - \frac{1}{k+1-\kappa} = \frac{1}{k+1} - \frac{x_1^{k,\delta}}{a}.$$

Similarly, for the lower bound, we have

$$\frac{y_1}{a} < \frac{1-\kappa}{k(k+1-\kappa)} = k^{-1}\left(\frac{k+1-\kappa}{k+1-\kappa} - \frac{k}{k+1-\kappa}\right) = \frac{1}{k} - \frac{x_1^{k,\delta}}{a}.$$

Therefore, $\lceil a/(x_1^{k,\delta} + y_1) \rceil = k+1$. Thus, for all $\|y\|_\infty < \varepsilon'(K)$, we have $f_a(x^{k,\delta} + y) = f_a(x^{k,\delta}) = 1$ for even k and $f_a(x^{k,\delta} + y) = f_a(x^{k,\delta}) = 0$ for odd k, therefore establishing $\{x^{1,\delta}, \ldots, x^{K,\delta}\} \in \mathcal{S}_{\varepsilon'(K)}^{f_a}$. □

5.2.2. Proof of Lemma 5.3

Proof of Lemma 5.3. We set

$$\tilde{\varphi} = W^L \rho W^{L-1} \rho W^{L-2} \ldots \rho W^1$$

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

where $W^\ell x = A^\ell x + b^\ell$ and $A^\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$, $b^\ell \in \mathbb{R}^{N_\ell}$ are defined as follows: let $A^1_{1,1} = 0$, $A^1_{1,2} = \delta^{-1}$ and $A^1_{i,j} = 0$ otherwise, and, for $\ell > 1$, $A^\ell_{1,1} = 1$ and $A^\ell_{i,j} = 0$ otherwise, and $b^\ell = 0$ for every ℓ. Clearly

$$W^1 x^{k,\delta} = \begin{cases} e_1 \in \mathbb{R}^{N_1} & \text{if } k \text{ is even} \\ \mathbf{0} \in \mathbb{R}^{N_1} & \text{if } k \text{ is odd} \end{cases}$$

and it is therefore easy to see that $\tilde{\varphi}(x^{k,\delta}) = 1$ if k is even and $\tilde{\varphi}(x^{k,\delta}) = 0$ if k is odd. By the definition of $x^{k,\delta}$, we have $f_a(x^{k,\delta}) = 1$ if k is even and $f_a(x^{k,\delta}) = 0$ if k is odd, and therefore $\tilde{\varphi}(x^{k,\delta}) = f_a(x^{k,\delta})$ for all k. □

5.2.3. Proof of Lemma 5.4

To prove Lemma 5.4, we will state and prove the following:

Lemma 5.5. *Fix $m, n \in \mathbb{N}$, $A \in \mathbb{R}^{N \times N_0}$, $B \in \mathbb{R}^{m \times N}$ and $z \in \mathbb{R}^N$. Suppose that*

$$R = \{\alpha^q, \alpha^{q+1}, \alpha^{q+2}, \ldots, \alpha^{q+r-1}\} \subset \mathbb{R}^{N_0}$$

is a set such that $|R| \geq N+1$, the sequence $\{\alpha^k_1\}_{k=q}^{q+r-1}$ is strictly decreasing and $\alpha^k_j = 0$ for $j > 1$ and all k. Then there exist a matrix $C \in \mathbb{R}^{m \times N_0}$, a vector $v \in \mathbb{R}^m$ and a set $\mathcal{S} \subseteq R$ of the form $\mathcal{S} = \{\alpha^s, \alpha^{s+1}, \ldots, \alpha^{s+t-1}\}$ such that $|\mathcal{S}| \geq |R|/(N+1)$ and $B\rho(A\alpha + z) = C\alpha + v$, for all $\alpha \in \mathcal{S}$.

Proof of Lemma 5.5. Write $B = (b_{j,k})_{j=1,k=1}^{j=m,k=N}$, $A = (a_{j,k})_{j=1,k=1}^{j=N,k=N_0}$. We claim that the set $\mathcal{Q}$ defined by

$$\mathcal{Q} = \{\big(\operatorname{sgn}(a_{1,1}u_1 + w_1), \operatorname{sgn}(a_{2,1}u_1 + w_2), \ldots, \operatorname{sgn}(a_{N,1}u_1 + w_N)\big) \mid u \in R\}.$$

contains at most $N+1$ (unique) elements, that is, $|\mathcal{Q}| \leq N+1$, where we define $\operatorname{sgn}(x) = 1$ for $x \geq 0$ and $\operatorname{sgn}(x) = -1$ for $x < 0$. To see this, note that if we allow the value of β to vary over $\mathbb{R}$, then each of the lines $y = a_{1,1}\beta + z_1$, $y = a_{2,1}\beta + z_2$, …, $y = a_{N,1}\beta + z_N$ intersect the line $y = 0$ at most once. Between each of these intersections, the vector $(\operatorname{sgn}(a_{1,1}\beta + z_1), \operatorname{sgn}(a_{2,1}\beta + z_2), \ldots, \operatorname{sgn}(a_{N,1}\beta + z_N))$ is constant. As there are at most N such intersections, we note that if

$$\mathcal{Q}' := \{\big(\operatorname{sgn}(a_{1,1}\beta + w_1), \operatorname{sgn}(a_{2,1}\beta + w_2), \ldots, \operatorname{sgn}(a_{N,1}\beta + w_N)\big) \mid \beta \in \mathbb{R}\}.$$

then $|\mathcal{Q}'| \leq N+1$ follows because partitioning a line by at most N intersections gives at most $N+1$ regions between the intersections. As $\mathcal{Q} \subseteq \mathcal{Q}'$, the proof that $|\mathcal{Q}| \leq N+1$ is complete.

We can now define $\mathcal{S}$. By the pigeonhole principle and the fact that $|\mathcal{Q}| \leq N+1$, there exists a subset of R with cardinality at least $|R|/(N+1)$ such that the vector

$$\operatorname{sgn}(a_{\cdot,1}\alpha_1 + z) = \big(\operatorname{sgn}(a_{1,1}\alpha_1 + z_1), \operatorname{sgn}(a_{2,1}\alpha_1 + z_2), \ldots, \operatorname{sgn}(a_{N,1}\alpha_1 + z_N)\big)$$

is constant over α in this subset. Let $\mathcal{S}$ be a subset of R of maximal cardinality satisfying this constant sign condition. Then clearly $|\mathcal{S}| \geq |R|/(N+1)$. To see that $\mathcal{S} = \{\alpha^s, \alpha^{s+1}, \ldots, \alpha^{s+t-1}\}$, for some s and t, suppose by way of contradiction that no such s and t exist. Then there are j_1 and k_1 such that $j_1 + 1 < k_1$, $\alpha^{j_1}, \alpha^{k_1} \in \mathcal{S}$ and $\alpha^{j_1+1} \notin \mathcal{S}$. But then, as $\mathcal{S}$ is assumed to be of maximal cardinality, there must be an ℓ for which $\operatorname{sgn}(a_{\ell,1}\alpha_1^{j_1} + z_1) = \operatorname{sgn}(a_{\ell,1}\alpha_1^{k_1} + z_1) \neq \operatorname{sgn}(a_{\ell,1}\alpha_1^{j_1+1} + z_1)$. However, since $\{\alpha^j_1\}_{j=j_1}^{k_1}$ is a strictly decreasing sequence by assumption, we see that if $a_{\ell,1} \geq 0$ then $a_{\ell,1}\alpha_1^{j_1} + z_1 \geq a_{\ell,1}\alpha_1^{j_1+1} + z_1 \geq a_{\ell,1}\alpha_1^{k_1} + z_1$ and similarly if $a_{\ell,1} < 0$ then $a_{\ell,1}\alpha_1^{j_1} + z_1 < a_{\ell,1}\alpha_1^{j_1+1} + z_1 < a_{\ell,1}\alpha_1^{k_1} + z_1$ which is a contradiction. This establishes that $\mathcal{S} = \{\alpha^s, \alpha^{s+1}, \ldots, \alpha^{s+t-1}\}$, for some s and t.

We now show how to construct C and v. Recall that, for all $\alpha \in \mathcal{S}$, $\alpha_2 = \alpha_3 = \cdots = \alpha_{N_0} = 0$, and so the i-th row of $B\rho(A\alpha + z)$ is given by $\sum_{j=1}^N b_{i,j}\rho(a_{j,1}\alpha_1 + z_j)$. Since $\operatorname{sgn}(a_{j,1}\alpha_1 + z_j)$ is constant over $\alpha \in \mathcal{S}$, we must have that for each j either $\rho(a_{j,1}\alpha_1 + z_j) = 0$ or $\rho(a_{j,1}\alpha_1 + w_j) = a_{j,1}\alpha_1 + z_j$, for all $\alpha \in \mathcal{S}$. In the former case, we define $d_{i,j} = 0$ and $y_{i,j} = 0$ and in the latter case we define $d_{i,j} = b_{i,j}a_{j,1}$ and $y_{i,j} = b_{i,j}z_j$.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

Therefore, by construction, the i-th row of $B\rho(A\alpha + z)$ is given by $\sum_{j=1}^{N} \left(d_{i,j}\alpha_1 + y_{i,j}\right)$. Thus, defining the matrix $C = (c_{i,j})_{i=1,j=1}^{i=m,j=N_0}$ and the vector $v \in \mathbb{R}^m$ according to

$$c_{i,1} = \sum_{k=1}^{N} d_{i,k}, \quad c_{i,j} = 0, \text{ for } j > 1, \qquad \text{and} \qquad v_i = \sum_{k=1}^{N} y_{i,k}$$

immediately yields that the i-th row of $B\rho(A\alpha + z)$ satisfies $\sum_{k=1}^{N} \left(d_{i,k}\alpha_1 + y_{i,k}\right) = \sum_{k=1}^{N} c_{i,k}\alpha_k + v_i$. As i and $\alpha \in \mathcal{S}$ were arbitrary, this implies that $B\rho(A\alpha + z) = C\alpha + v$ for all $\alpha \in \mathcal{S}$, thereby concluding the proof of the lemma. □

With Lemma 5.5, we can now prove Lemma 5.4.

Proof of Lemma 5.4. We begin by proving the following claim:

Claim: There exists a set

$$\mathcal{S} = \{w^s, w^{s+1}, w^{s+2}, \ldots, w^{s+n}\} \subset \{w^1, w^2, \ldots, w^t\}$$

for some $s \in \mathbb{N}$ and $n \in \mathbb{N}$, a matrix $M \in \mathbb{R}^{1 \times N_0}$ and a $z \in \mathbb{R}$ such that, for all $w \in \mathcal{S}$, we have $\varphi(w) = Mw + z$ and so that $|\mathcal{S}| \geq 3m$.

To see the validity of this claim, we proceed inductively by showing that there are sets $\mathcal{S}_\ell \subset \{w^1, w^2, \ldots, w^t\}$, matrices $M^\ell \in \mathbb{R}^{N_\ell \times N_0}$ and vectors $z^\ell \in \mathbb{R}^{N_\ell}$ for $\ell = 1, \ldots, L$ such that

(i) $|\mathcal{S}_\ell| \geq 3m \cdot (N_\ell + 1) \cdots (N_{L-1} + 1)$,

(ii) $\mathcal{S}_\ell = \{w^{s_\ell}, w^{s_\ell+1}, \ldots, w^{s_\ell+n_\ell}\}$ for some $s_\ell, n_\ell \in \mathbb{N}$.

(iii) $\varphi(w) = W^L \rho W^{L-1} \rho W^{L-2} \ldots W^{\ell+1} \rho(M^\ell w + z^\ell)$ whenever $w \in \mathcal{S}_\ell$, where the W^i are affine maps and ρ is applied coordinatewise.

The induction base is obvious by taking $\mathcal{S}_1 = \mathcal{W}, M^1 = W^1$ and $z^1 = b^1$. The induction step will follow with the help of Lemma 5.5. Indeed, assuming the existence of $\mathcal{S}_\ell$, M^ℓ and z^ℓ for some $\ell < L$, we apply Lemma 5.5 with $B = A^{\ell+1}, A = M^\ell, R = \mathcal{S}_\ell$ and $w = z^\ell$ to obtain some set $\mathcal{S}_{\ell+1}$, a matrix $M^{\ell+1}$ and a vector $v^{\ell+1}$ for which $A^{\ell+1}\rho(M^\ell w + z^\ell) = M^{\ell+1}w + v^{\ell+1}$ for $w \in \mathcal{S}_{\ell+1}$, and thus $W^{\ell+1}\rho(M^\ell w + z^\ell) = M^{\ell+1}w + z^{\ell+1}$, where we set $z^{\ell+1} = v^{\ell+1} + b^{\ell+1}$. With the completed induction in hand, the proof of the claim follows by setting $\mathcal{S} = \mathcal{S}_L, s = s_L, n = n_L, M = M^L$ and $z = z^L$.

Using the claim, we can now complete the proof of Lemma 5.4. Indeed, define the disjoint sets $\mathcal{S}^>$, $\mathcal{S}^<$ as follows:

$$\mathcal{S}^> = \{w \in \mathcal{S} \mid g(\varphi(w)) \geq 1/2\}, \quad \mathcal{S}^< = \{w \in \mathcal{S} \mid g(\varphi(w)) < 1/2\}$$

For any $w \in \mathcal{S}$, we have $\varphi(w) = Mw + z$. Furthermore, any such w has $w_2 = w_3 = \cdots = w_N = 0$. Therefore, $\varphi(w) = M_{1,1}w_1 + z$. In particular, $g \circ \varphi$ restricted to $\mathcal{S}$ is monotonic in the first coordinate of vectors in $\mathcal{S}$. This implies that

$$\mathcal{S}^> = \{w^{k_1}, w^{k_1+1}, w^{k_1+2}, \ldots, w^{k_1+t_1-1}\}, \quad \mathcal{S}^< = \{w^{k_2}, w^{k_2+1}, w^{k_2+2}, \ldots w^{k_2+t_2-1}\}$$

for some k_1 and k_2 and t_1, t_2 with $t_1 + t_2 = |\mathcal{S}| \geq 3m$. Furthermore, by 3 and the fact that the range of f is the set $\{0, 1\}$, we must have $f(w^i) = 1$ for all even i and $f(w^i) = 0$ for all odd i or $f(w^i) = 0$ for all even i and $f(w^i) = 1$ for all odd i. We will consider these two cases separately

Case 1: $f(w^i) = 1$ for all even i and $f(w^i) = 0$ for all odd i. We define the sets

$$\mathcal{S}^{E,<} = \{w^i \mid w^i \in \mathcal{S}^<, \ i \text{ even}\}, \quad \mathcal{S}^{O,>} = \{w^i \mid w^i \in \mathcal{S}^>, \ i \text{ odd}\}$$

For $w \in \mathcal{S}^{E,<}$, we have $f(w) = 1$ and $g(\varphi(w)) < 1/2$, whence we obtain $|g(\varphi(w)) - f(w)| \geq 1/2$. Similarly, for $w \in \mathcal{S}^{O,>}$ we have $f(w) = 0$ and $g(\varphi(w)) \geq 1/2$ and we thus obtain $|g(\varphi(w)) - f(w)| \geq 1/2$. We set $\mathcal{U} = \mathcal{S}^{E,>} \cup \mathcal{S}^{O,<}$ and conclude that for any $w \in \mathcal{U}$ we have $|f(w) - g(\varphi(w))| \geq 1/2$.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

The claim about the cardinality of $\mathcal{U}$ follows by noting that $|\mathcal{S}^{E,<}| \geq \lceil (t_1 - 1)/2 \rceil$ and that $|\mathcal{S}^{O,>}| \geq \lceil (t_2 - 1)/2 \rceil$. Therefore, (using the disjointedness of $\mathcal{S}^{E,>}$ and $\mathcal{S}^{O,<}$)

$$\begin{aligned} |\mathcal{U}| = |\mathcal{S}^{E,>}| + |\mathcal{S}^{O,<}| &\geq \lceil (t_1 - 1)/2 \rceil + \lceil (t_2 - 1)/2 \rceil \\ &\geq \lceil (t_1 - 1 + t_2 - 1)/2 \rceil = \lceil (t_1 + t_2)/2 \rceil - 1 \geq \lceil 3m/2 \rceil - 1 \geq m \end{aligned} \tag{5.4}$$

Case 2: $f(w^i) = 0$ for all even i and $f(w^i) = 1$ for all odd i. The proof here is similar to that of Case 1. This time however, we define the sets

$$\mathcal{S}^{E,>} = \{w^i \mid w^i \in \mathcal{S}^>,\ i \text{ even}\}, \quad \mathcal{S}^{O,<} = \{w^i \mid w^i \in \mathcal{S}^>,\ i \text{ odd}\}$$

An analogous argument to the above allows us to conclude that $|g(\varphi(w)) - f(w)| \geq 1/2$ for all $w \in \mathcal{U}$, where this time $\mathcal{U} = \mathcal{S}^{E,>} \cup \mathcal{S}^{O,<}$. The argument that $|\mathcal{U}| \geq m$ is identical to (5.4) except we replace references to $\mathcal{S}^{E,<}$ with $\mathcal{S}^{E,>}$ and references to $\mathcal{S}^{O,>}$ with $\mathcal{S}^{O,<}$. □

5.3. Proof of Theorem 2.2

We require two further lemmas specific to the proof of Theorem 2.2. These are stated as Lemmas 5.6 and 5.7.

Lemma 5.6. *For $\gamma \in (1,2)$, define the probability distribution $\mathcal{P} = \{\mathrm{p}_j\}_{j=1}^{\infty}$ on $\mathbb{N}$ by $\mathrm{p}_{2j-1} = \mathrm{p}_{2j} = \frac{1}{2} C_\zeta(\gamma) j^{-\gamma}$, for $j \in \mathbb{N}$, where $C_\zeta(\gamma) := \left(\sum_{j=1}^{\infty} j^{-\gamma} \right)^{-1}$ is a normalising factor.*

Fix $\theta \in \mathbb{N}$ and let $X_1, X_2, \ldots, X_\theta$ be i.i.d. random variables in $\mathbb{N}$ distributed according to $\mathcal{P}$. Next, consider the random set whose elements are the values of $X_1, X_2, \ldots, X_\theta$ and enumerate it as $S = \{Z_1, Z_2, \ldots, Z_N\}$ with $Z_1 < Z_2 < \cdots < Z_N$ (note that N, the number of distinct elements of S, is an integer-valued random variable such that $N \leq \theta$). Then, setting $c_1 = (1 - e^{-C_\zeta(\gamma)})/2$ and $c_2 = C_\zeta(\gamma)/(\gamma - 1)$, we have

(i) $\mathbb{P}\left(N \geq c_1 \theta^{1/\gamma}\right) \geq 1 - c_1^{-2} \theta^{-(2/\gamma - 1)}$,

(ii) $\mathbb{P}(\max S \leq n) \geq 1 - c_2\, \theta \lfloor n/2 \rfloor^{1-\gamma}$, *for all $n \in \mathbb{N}$, and*

(iii) $\mathbb{P}\left(\sum_{j=1}^{N-1} \chi_{\{Z_{j+1} - Z_j \text{ odd}\}} \leq n/5 \;\middle|\; N = n \right) \leq e^{-n/100}$, *for all integers n such that $10 \leq n \leq \theta$.*

Proof. Throughout this proof, we will use the convention that for a random variable $Y : \Omega \to \mathcal{E}$ the notation $\{Y = \mu\}$ for $\mu \in \Omega$ means the set $\{\tau \in \Omega \mid Y(\tau) = \mu\}$.

For item (i), define the random variable M_θ to be the number of different unique values taken by the random variables $\lceil X_1/2 \rceil, \ldots, \lceil X_\theta/2 \rceil$ and note that $\mathbb{P}(N < \beta) \leq \mathbb{P}(M_r < \beta)$, for $\beta \in \mathbb{R}$. Now, as the random variables $\lceil X_j/2 \rceil$, $j = 1, \ldots, r$, are i.i.d. and distributed according to the zeta distribution with parameter γ, it follows from [82, Lemmas 4, 3] that $\mathbb{E}[M_\theta] > (1 - e^{-C_\zeta(\gamma)}) \theta^{1/\gamma}$ and $\sigma^2 := \mathrm{Var}[M_\theta] \leq \mathbb{E}[M_\theta] \leq \theta$, and hence Chebyshev's inequality yields

$$\begin{aligned} &\mathbb{P}\left(N < \frac{1 - e^{-C_\zeta(\gamma)}}{2} \theta^{1/\gamma} \right) \leq \mathbb{P}\left(M_r < \frac{1 - e^{-C_\zeta(\gamma)}}{2} \theta^{1/\gamma} \right) \\ &\leq \mathbb{P}\left(|M_\theta - \mathbb{E}[M_\theta]| > \frac{1 - e^{-C_\zeta(\gamma)}}{2\sigma} \theta^{1/\gamma} \cdot \sigma \right) \leq \left(\frac{1 - e^{-C_\zeta(\gamma)}}{2\sigma} \theta^{1/\gamma} \right)^{-2} \leq \frac{4\theta^{-(2/\gamma - 1)}}{(1 - e^{-C_\zeta(\gamma)})^2}, \end{aligned}$$

which implies item (i).

The proof of item (ii) is simple. Note that $\{\max S \leq n\} = \bigcap_{j=1}^{r} \{X_j \leq n\}$ and, for each j,

$$\mathbb{P}(X_j \leq n) = \sum_{j=1}^{n} \mathrm{p}_j \geq \sum_{j=1}^{\lfloor n/2 \rfloor} C_\zeta(\gamma) j^{-\gamma} \geq 1 - C_\zeta(\gamma) \int_{\lfloor n/2 \rfloor}^{\infty} t^{-\gamma} \mathrm{d}t \geq 1 - \frac{C_\zeta(\gamma)}{\gamma - 1} \lfloor n/2 \rfloor^{1-\gamma},$$

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

and hence, as the X_j are independent,

$$\mathbb{P}(\max S \le n) = \mathbb{P}(X_j \le n)^\theta \ge \left(1 - \frac{C_\zeta(\gamma)}{\gamma - 1}\lfloor n/2\rfloor^{1-\gamma}\right)^\theta \ge 1 - \frac{C_\zeta(\gamma)}{\gamma - 1}\theta\lfloor n/2\rfloor^{1-\gamma}$$

where the last inequality follows by Bernoulli's inequality.

Item (iii) is somewhat more involved. We start by outlining the strategy: the set S may contain pairs of the form $(Z_j, Z_{j+1}) = (2i-1, 2i)$, that is, an odd natural number followed by the next even one. We will condition on the set of j where (Z_j, Z_{j+1}) is such a pair, as well as the specific value of Z_j.

More precisely, for fixed sets $\mathcal{I}$ and $\mathcal{J}$ with $|\mathcal{I}| = |\mathcal{J}|$, enumerated by

$$\mathcal{I} = \{i_1, i_2, \dots, i_m\} \text{ and } \mathcal{J} = \{j_1, j_2, \dots, j_m\},$$

let $\mathcal{A} = \{1, \dots, N\} \setminus \big(\mathcal{J} \cup (\mathcal{J}+1)\big)$ where $\mathcal{J}+1 := \{j+1 \mid j \in \mathcal{J}\}$. We will condition on the event $F_{\mathcal{I},\mathcal{J}}$ which occurs precisely when $N = n$, $(Z_{j_\ell}, Z_{j_\ell+1}) = (2i_\ell - 1, 2i_\ell)$ for $\ell \in \{1, 2, \dots, m\}$, and, on the indices in $\mathcal{A}$, the set S contains no odd–even pairs, that is, $(Z_a, Z_{a+1}) \notin \{(2i-1, 2i) \mid i \in \mathbb{N}\}$ for all $a \in \mathcal{A}$ with $a < n$ and $(Z_{a-1}, Z_a) \notin \{(2i-1, 2i) \mid i \in \mathbb{N}\}$ for all $a \in \mathcal{A}$ with $a > 1$. With varying $\mathcal{I}$ and $\mathcal{J}$, these sets $F_{\mathcal{I},\mathcal{J}}$ partition the event $\{N = n\}$.

The intuition behind this construction is as follows: conditional on $F_{\mathcal{I},\mathcal{J}}$, whenever $j \in \mathcal{J}$ we have $Z_{j+1} - Z_j = 1$ and hence $\chi_{\{Z_{j+1}-Z_j \text{ odd}\}} = 1$. Thus for sets $\mathcal{J}$ with $|\mathcal{J}| \ge n/5$, we are done. If instead $|\mathcal{J}|$ is small, then $|\mathcal{A}|$ will be relatively large. For $a \in \mathcal{A}$, we will argue that every Z_a has equal probability of being an odd number or the even number following it, owing to the assumption that $\mathrm{p}_{2i-1} = \mathrm{p}_{2i}$ and the assumption that if $a < n$ then $(Z_a, Z_{a+1}) \notin \{(2i-1, 2i) \mid i \in \mathbb{N}\}$ and if $a > 1$ then $(Z_{a-1}, Z_a) \notin \{(2i-1, 2i) \mid i \in \mathbb{N}\}$.

This will allow us to conclude that the indicator random variables $\chi_{\{Z_a \text{ odd}\}}$ for $a \in \mathcal{A}$ are independent symmetric Bernoulli random variables (that is to say, they take the values 1 and 0 each with probability 1/2). The desired bound will follow by an application of Hoeffding's inequality.

We are now ready to present the formal proof. If $\theta < 10$ there is nothing to prove, so assume that $\theta \ge 10$ and fix an n such that $10 \le n \le \theta$. Consider arbitrary sets $\mathcal{I} \subset \mathbb{N}$ and $\mathcal{J} \subset \{1, \dots, n-1\}$ so that

$$m := |\mathcal{I}| = |\mathcal{J}| < n \text{ and } \mathcal{J} \cap (\mathcal{J}+1) = \varnothing, \tag{5.5}$$

and define $\mathcal{A} := \{1, \dots, N\} \setminus \big(\mathcal{J} \cup (\mathcal{J}+1)\big)$. Enumerate $\mathcal{I} = \{i_1, \dots, i_m\}$ with $i_1 < \cdots < i_m$, $\mathcal{J} = \{j_1, \dots, j_m\}$ with $j_1 < \cdots < j_m$, and $\mathcal{A} = \{a_1, \dots, a_{n-2m}\}$ with $a_1 < \cdots < a_{n-2m}$ and define the event

$$F_{\mathcal{I},\mathcal{J}} = \{N = n\} \cap \bigcap_{\ell=1}^{m} \{(Z_{j_\ell}, Z_{j_\ell+1}) = (2i_\ell - 1, 2i_\ell)\} \cap \bigcap_{\substack{a \in \mathcal{A}, a < n \\ i \in \mathbb{N}}} \{(Z_a, Z_{a+1}) \ne (2i-1, 2i)\}$$
$$\cap \bigcap_{\substack{a \in \mathcal{A}, 1 < a \le n \\ i \in \mathbb{N}}} \{(Z_{a-1}, Z_a) \ne (2i-1, 2i)\}.$$

Note that, for every $n \in \mathbb{N}$, we have

$$\{N = n\} = \bigcup_{\substack{\mathcal{I} \subset \mathbb{N}, \mathcal{J} \subset \{1,\dots,n-1\} \\ \text{satisfying (5.5)}}} F_{\mathcal{I},\mathcal{J}}, \tag{5.6}$$

that is, the events $F_{\mathcal{I},\mathcal{J}}$ for different $\mathcal{I}$ and $\mathcal{J}$ partition the event $\{N = n\}$, and thus our strategy will be to prove the bound $\mathbb{P}\left(\sum_{j=1}^{N-1} \chi_{\{Z_{j+1}-Z_j \text{ odd}\}} \le n/5 \;\middle|\; F_{\mathcal{I},\mathcal{J}}\right) \le e^{-n/100}$ for each of these events.

The argument relies on bounding from below the number of indices j such that $Z_{j+1} - Z_j$ is odd. For $j \in \mathcal{J}$, this will be easy, as $Z_{j+1} - Z_j = 2i_j - (2i_j - 1) = 1$ is always odd, by definition of $F_{\mathcal{I},\mathcal{J}}$. For $j \in \mathcal{A}$, we will need the following claim which we prove last.

Claim: For any $\mathcal{I}$, $\mathcal{J}$ and $\mathcal{A}$ as above, the indicator random variables $\chi_{\{Z_a \text{ odd}\}}$, $a \in \mathcal{A}$, conditional on $F_{\mathcal{I},\mathcal{J}}$ are independent symmetric Bernoulli variables.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

Armed with the claim, the counting argument is as follows. Note that, on the event $F_{\mathcal{I},\mathcal{J}}$, for $k \in \{1,\dots,n-2m-1\}$ such that $a_{k+1} > a_k + 1$, we have that $\{Z_{a_k},\dots,Z_{a_{k+1}}\} = \{Z_{a_k}, 2i_t - 1, 2i_t, 2i_{t+1} - 1, 2i_{t+1},\dots,2i_{t+s-1} - 1, 2i_{t+s-1}, Z_{a_{k+1}}\}$ for some $t \in \{1,2,\dots,m\}$ and where $s = |\mathcal{J} \cap \{a_k,\dots,a_{k+1}-1\}|$. Hence,

$$\begin{aligned}\sum_{\ell=a_k}^{a_{k+1}-1} \chi_{\{Z_{\ell+1}-Z_\ell \text{ odd}\}} &\geq \chi_{\{2i_t-1-Z_{a_k} \text{ odd}\}} + \sum_{\ell=0}^{s-1} \chi_{\{(2i_{t+\ell})-(2i_{t+\ell}-1) \text{ odd}\}} + \chi_{\{Z_{a_{k+1}}-2i_{t+s-1} \text{ odd}\}} \\ &= \chi_{\{Z_{a_k} \text{ even}\}} + |\mathcal{J} \cap \{a_k,\dots,a_{k+1}-1\}| + \chi_{\{Z_{a_{k+1}} \text{ odd}\}} \\ &\geq |\mathcal{J} \cap \{a_k,\dots,a_{k+1}-1\}| + \chi_{\{Z_{a_{k+1}}-Z_{a_k} \text{ even}\}}, \end{aligned} \tag{5.7}$$

where we used the simple observation that $\chi_{\{Z_{a_k} \text{ even}\}} + \chi_{\{Z_{a_{k+1}} \text{ odd}\}} \geq \chi_{\{Z_{a_{k+1}}-Z_{a_k} \text{ even}\}}$. This motivates defining random variables E_{a_k} with $k \in \{1,\dots,n-2m-1\}$ conditioned on the event $F_{\mathcal{I},\mathcal{J}}$ according to

$$E_{a_k} = \begin{cases} 1, & Z_{a_k+1} - Z_{a_k} \text{ is odd} \\ 0, & Z_{a_k+1} - Z_{a_k} \text{ is even} \end{cases}, \quad \text{for } k \text{ s.t. } a_{k+1} = a_k + 1, \text{ and}$$

$$E_{a_k} = \begin{cases} 0, & Z_{a_{k+1}} - Z_{a_k} \text{ is odd} \\ 1, & Z_{a_{k+1}} - Z_{a_k} \text{ is even} \end{cases}, \quad \text{for } k \text{ s.t. } a_{k+1} > a_k + 1,$$

which, as a consequence of the Claim, are themselves independent symmetric Bernoulli random variables. Thus, writing $U := \sum_{k=1}^{N-1} \chi_{\{Z_{k+1}-Z_k \text{ odd}\}}$, on the event $F_{\mathcal{I},\mathcal{J}}$ we have

$$\begin{aligned} U &= \sum_{\ell < a_1 \text{ or } \ell \geq a_{n-2m}} \chi_{\{Z_{\ell+1}-Z_\ell \text{ odd}\}} + \sum_{k=1}^{n-2m-1} \sum_{\ell=a_k}^{a_{k+1}-1} \chi_{\{Z_{\ell+1}-Z_\ell \text{ odd}\}} \\ &\geq |\mathcal{J} \cap \{1,\dots,a_1-1\}| + |\mathcal{J} \cap \{a_{n-2m},\dots,n\}| \\ &\qquad + \sum_{\substack{1 \leq k \leq n-2m-1 \\ a_{k+1} = a_k + 1}} \chi_{\{Z_{a_k+1}-Z_{a_k} \text{ odd}\}} + \sum_{\substack{1 \leq k \leq n-2m-1 \\ a_{k+1} > a_k + 1}} \sum_{\ell=a_k}^{a_{k+1}-1} \chi_{\{Z_{\ell+1}-Z_\ell \text{ odd}\}} \\ &\geq |\mathcal{J} \cap \{1,\dots,a_1-1\}| + |\mathcal{J} \cap \{a_{n-2m},\dots,n\}| \\ &\qquad + \sum_{\substack{1 \leq k \leq n-2m-1 \\ a_{k+1} = a_k + 1}} E_{a_k} + \sum_{\substack{1 \leq k \leq n-2m-1 \\ a_{k+1} > a_k + 1}} \left(|\mathcal{J} \cap \{a_k,\dots,a_{k+1}-1\}| + E_{a_k} \right) \\ &= |\mathcal{J}| + \sum_{k=1}^{n-2m-1} E_{a_k} = m + \sum_{k=1}^{n-2m-1} E_{a_k}, \end{aligned} \tag{5.8}$$

where the second inequality is due to (5.7) and the penultimate equality follows from the observation that $|\mathcal{J} \cap \{a_k,\dots,a_{k+1}-1\}| = 0$ whenever $a_{k+1} = a_k + 1$.

Now, for sets $\mathcal{I} \subset \mathbb{N}$ and $\mathcal{J} \subset \{1,\dots,n-1\}$ satisfying (5.5) as well as $m = |\mathcal{I}| = |\mathcal{J}| \leq n/5$, we have that (5.8) implies $U \geq \sum_{k=1}^{n-2m-1} E_{a_k}$, which together with Hoeffding's inequality yields

$$\begin{aligned} \mathbb{P}\left(U \leq n/5 \,\middle|\, F_{\mathcal{I},\mathcal{J}} \right) &\leq \mathbb{P}\left(\sum_{k=1}^{n-2m-1} E_{a_k} \leq n/5 \,\middle|\, F_{\mathcal{I},\mathcal{J}} \right) \\ &\leq \exp\left(-2\left(\frac{1}{2} - \frac{n/5}{n-2m-1} \right)^2 (n-2m-1) \right) \leq \exp\left(-n/100\right) \end{aligned}$$

where in the last inequality we used $n - 2m - 1 \geq n/2$ (recall that $n \geq 10$). On the other hand, in the case when $m = |\mathcal{I}| = |\mathcal{J}| > n/5$ we have $\mathbb{P}\left(U \leq n/5 \,\middle|\, F_{\mathcal{I},\mathcal{J}} \right) = 0$ directly from (5.8).

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

Therefore, we have shown that for any $\mathcal{I}$, $\mathcal{J}$ satisfying (5.5), $\mathbb{P}\Big(U \le n/5 \;\Big|\; F_{\mathcal{I},\mathcal{J}}\Big) \le \exp(-n/100)$ and so using (5.6)

$$\mathbb{P}\Big(U \le n/5, N = n\Big) = \sum_{\substack{\mathcal{I}\subset\mathbb{N},\mathcal{J}\subset\{1,\dots,n-1\}\\ \text{satisfying (5.5)}}} \mathbb{P}\Big(U \le n/5 \;\Big|\; F_{\mathcal{I},\mathcal{J}}\Big)\mathbb{P}(F_{\mathcal{I},\mathcal{J}})$$
$$\le \exp(-n/100)\,\mathbb{P}\left(\bigcup_{\substack{\mathcal{I}\subset\mathbb{N},\mathcal{J}\subset\{1,\dots,n-1\}\\ \text{satisfying (5.5)}}} F_{\mathcal{I},\mathcal{J}}\right) = \exp(-n/100)\,\mathbb{P}(N=n),$$

which yields the desired bound after dividing both sides by $\mathbb{P}(N=n)$.

It remains to prove the Claim. To this end, fix n, $\mathcal{I} = \{i_1 < \ldots < i_m\}$, $\mathcal{J} = \{j_1 < \ldots < j_m\}$ and $\mathcal{A} = \{a_1 < \ldots < a_{n-2m}\}$ satisfying (5.5). Then, conditional on $F_{\mathcal{I},\mathcal{J}}$ we can write $Z_a = 2\lceil Z_a/2\rceil - \chi_{\{Z_a \text{ odd}\}}$, for $a \in \mathcal{A}$, where the $\chi_{\{Z_a \text{ odd}\}}$ are random variables taking values in $\{0,1\}$ and the $\lceil Z_a/2\rceil$ are random variables taking values in $\mathbb{N}\setminus\mathcal{I}$ and moreover $\lceil Z_{a_1}/2\rceil < \ldots < \lceil Z_{a_{n-2m}}/2\rceil$. Now, for a set $\mathcal{U} = \{u_1 < \ldots < u_{n-2m}\} \subset \mathbb{N}\setminus\mathcal{I}$ denote $F_{\mathcal{U}} = \bigcap_{j=1}^{n-2m}\{\lceil Z_{a_j}/2\rceil = u_j\}$ so that for any $b \in \{0,1\}^{n-2m}$

$$\begin{aligned}
&\mathbb{P}\left(\{\chi_{\{Z_{a_1} \text{ odd}\}} = b_1, \ \ldots\ , \chi_{\{Z_{a_{n-2m}} \text{ odd}\}} = b_{n-2m}\} \;\middle|\; F_{\mathcal{I},\mathcal{J}}\right)\\
&\quad= \sum_{\mathcal{U}\subset\mathbb{N}\setminus\mathcal{I}} \mathbb{P}\left(\{\chi_{\{Z_{a_1} \text{ odd}\}} = b_1, \ \ldots\ , \chi_{\{Z_{a_{n-2m}} \text{ odd}\}} = b_{n-2m}\} \;\middle|\; F_{\mathcal{I},\mathcal{J}} \cap F_{\mathcal{U}}\right)\mathbb{P}(F_{\mathcal{U}} \mid F_{\mathcal{I},\mathcal{J}})\\
&\quad= \sum_{\mathcal{U}\subset\mathbb{N}\setminus\mathcal{I}} \mathbb{P}\left(\{\chi_{\{Z_{a_1} \text{ odd}\}} = 0, \ \ldots\ , \chi_{\{Z_{a_{n-2m}} \text{ odd}\}} = 0\} \;\middle|\; F_{\mathcal{I},\mathcal{J}} \cap F_{\mathcal{U}}\right)\mathbb{P}(F_{\mathcal{U}} \mid F_{\mathcal{I},\mathcal{J}}) \qquad (5.9)\\
&\quad= \mathbb{P}\left(\{\chi_{\{Z_{a_1} \text{ odd}\}} = 0, \ \ldots\ , \chi_{\{Z_{a_{n-2m}} \text{ odd}\}} = 0\} \;\middle|\; F_{\mathcal{I},\mathcal{J}}\right),
\end{aligned}$$

where in (5.9) we used the fact that $p_{2j-1} = p_{2j}$, for all $j \in \mathbb{N}$. It hence follows that the $\chi_{\{Z_{a_j} \text{ odd}\}}$, $1 \le j \le n-2m$, conditional on $F_{\mathcal{I},\mathcal{J}}$ are independent symmetric Bernoulli variables, establishing the Claim and thus completing the proof. □

Lemma 5.7. *Fix an even $K \in \mathbb{N}$ and let $\{\alpha_j\}_{k=1}^K$ be such that $0 < \alpha_{k+1} < \alpha_k < 1$ for all $1 \le k \le K-1$. Furthermore, let $N_0 \in \mathbb{N}$. Then there exists a neural network $\psi : \mathbb{R}^{N_0} \to \mathbb{R}$ with the ReLU non-linearity $\rho(t) = \max\{0,t\}$ such that*

$$\psi(x) = \begin{cases} 0 & \text{whenever } x_1 \in [\alpha_k, \alpha_{k-1}] \text{ with } k \equiv 2 \mod 4\\ 1 & \text{whenever } x_1 \in [\alpha_k, \alpha_{k-1}] \text{ with } k \equiv 0 \mod 4 \end{cases}, \qquad \text{for all } x \in \mathbb{R}^{N_0} \text{ and } k \in \{2,3,\ldots,K\}. \tag{5.10}$$

Proof. We may w.l.o.g. assume that K is divisible by 4. Indeed, if K is not divisible by 4, we can extend the sequence $\{\alpha_k\}_{k=1}^K$ by adjoining two new elements (say $\alpha_K/2$ and $\alpha_K/4$) at the end of the sequence. We additionally set $\alpha_{K+1} = 0$ for convenience. Now, for $\ell \in \{1,\ldots,K/4\}$, define the single-layer neural network

$$\psi_\ell(x) = (\alpha_{4\ell-2} - \alpha_{4\ell-1})^{-1}\left(\rho(\alpha_{4\ell-2} - x_1) - \rho(\alpha_{4\ell-1} - x_1)\right)$$
$$- (\alpha_{4\ell} - \alpha_{4\ell+1})^{-1}\left(\rho(\alpha_{4\ell} - x_1) - \rho(\alpha_{4\ell+1} - x_1)\right), \quad \text{for } x \in \mathbb{R}^{N_0}.$$

One now easily verifies that $\psi_\ell(x) = 1$ whenever $x_1 \in [\alpha_{4\ell}, \alpha_{4\ell-1}]$ and $\psi_\ell(x) = 0$ whenever $x_1 \in \mathbb{R}\setminus(\alpha_{4\ell+1}, \alpha_{4\ell-2})$. Hence, setting $\psi(x) = \sum_{k=1}^{K/4}\psi_\ell(x)$ yields the desired network. □

We are now in a position to prove Theorem 2.2:

Proof of Theorem 2.2. We begin by defining the sets $\mathcal{C}_1$ and $\mathcal{C}_2$. Let $\mathcal{C}_1 = \{f_a : \mathbb{R}^d \to [0,1] \mid a \in [1/2,1]\}$, where f_a is defined as in (5.2). Since all norms on finite dimensional vector spaces are equivalent, let $D > 0$ be such that $\|\cdot\| \le D\|\cdot\|_1$. To define the set of distributions, we first set $\delta = \epsilon/(2D)$. For each

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

$\kappa \in [1/4, 3/4]$, define the distribution $\mathcal{D}_\kappa$ on $[0,1]^{N_0}$

$$X \sim \mathcal{D}_\kappa \quad \iff \quad \mathbb{P}(X = x) = \begin{cases} \mathrm{p}_k & \text{if } x = x^{k,\delta} \\ 0 & \text{otherwise} \end{cases}.$$

where $\mathrm{p}_{2j-1} = \mathrm{p}_{2j} = C_\zeta(3/2) j^{-3/2}$ for $j \in \mathbb{N}$ and $x^{k,\delta}$ is defined according to (5.1). We set $\mathcal{C}_2 = \{\mathcal{D}_\kappa \mid \kappa \in [1/4, 3/4]\}$.

Let c_1, c_2 and $C_\zeta(3/2)$ be the constants defined in Lemma 5.6 with γ set to $3/2$. We choose the constant C so that each of the following hold:

$$C \geq 4^3 c_1^{-6}, \tag{5.11}$$

$$C \geq 200 \log(8)^{3/2} c_1^{-3/2}, \text{ and} \tag{5.12}$$

$$C \geq 4 \cdot (8c_2)^2. \tag{5.13}$$

Fix $a \in [1/2, 1]$ so that $f_a \in \mathcal{C}_1$ and $\kappa \in [1/4, 3/4]$ so that $\mathcal{D}_\kappa \in \mathcal{C}_2$. Let $\mathcal{T} = \{x^1, \dots, x^r\}$ and $\mathcal{V} = \{y^1, \dots, y^s\}$ be the random multisets drawn from this distribution as in the statement of the theorem. Then by the definition of the distribution $\mathcal{D}_\kappa$, we can write (after removing repetitions and reordering) $\mathcal{T} \cup \mathcal{V}$ as $S := \mathcal{T} \cup \mathcal{V} = \{x^{Z_1,\delta}, x^{Z_2,\delta}, x^{Z_3,\delta}, \dots, x^{Z_N,\delta}\}$ where the random variable N satisfying $N \leq r + s$ is the number of unique elements in $\mathcal{T} \cup \mathcal{V}$ and where $Z_1 < Z_2 < \cdots < Z_N$. For shorthand, we also set $z^j = x^{Z_j,0}$ for $j = 1, 2, \dots, N$.

Since $C/2 \geq 2 \cdot (8c_2)^2$ (by (5.13)) and $C(r \vee s)^2/(2p^2) \geq 4^3 c_1^{-6}/2 \geq 2$ (by (5.11) and the facts that $(r \vee s)/p \geq 1$ and $c_1^{-1} \geq 1$) we obtain

$$\frac{C(r \vee s)^2}{p^2} = \frac{C(r \vee s)^2}{2p^2} + \frac{C(r \vee s)^2}{2p^2} \geq 2 \cdot \frac{(8c_2)^2 (r \vee s)^2}{p^2} + 2 \geq 2 \left\lceil \left(\frac{8c_2(r \vee s)}{p} \right)^2 + 1 \right\rceil - 2$$

and thus item (ii) of Lemma 5.6 with $\gamma = 3/2$ yields

$$\begin{aligned} \mathbb{P}\left(\max\{k \in \mathbb{N} \mid x^{k,\delta} \in \mathcal{T} \cup \mathcal{V}\} \leq \left\lceil \frac{C(r \vee s)^2}{\mathrm{p}^2} \right\rceil \right) &= \mathbb{P}\left(Z_N \leq \left\lceil \frac{C(r \vee s)^2}{\mathrm{p}^2} \right\rceil \right) \\ &\geq \mathbb{P}\left(Z_N \leq 2 \left\lceil \left(\frac{8c_2(r \vee s)}{p} \right)^2 + 1 \right\rceil - 2 \right) \geq 1 - \frac{c_2(r+s)}{\left\lfloor \left\lceil \left(\frac{8c_2(r \vee s)}{p} \right)^2 + 1 \right\rceil - 1 \right\rfloor} \\ &\geq 1 - \frac{c_2(r+s)}{(8c_2(r \vee s)/\mathrm{p})} \geq 1 - \mathrm{p}/4. \end{aligned} \tag{5.14}$$

Writing $N_{\mathrm{prod}} := (N_1 + 1) \cdots (N_{L-1} + 1)$, by the Assumptions (2.3) and (5.12), we obtain

$$\lfloor c_1 (r+s)^{2/3} \rfloor \geq \lfloor C^{2/3} c_1 q N_{\mathrm{prod}} \rfloor \geq \lfloor 200^{2/3} \log(8) q N_{\mathrm{prod}} \rfloor \geq 30 q N_{\mathrm{prod}}.$$

Therefore, we can apply item (iii) of Lemma 5.6 to see that

$$\begin{aligned} \mathbb{P}\Big(\sum_{i=1}^{N-1} \chi_{\{f_a(z^{i+1}) \neq f_a(z^i)\}} > 6qN_{\mathrm{prod}} \Big) &= \mathbb{P}\Big(\sum_{i=1}^{N-1} \chi_{\{Z_{i+1} - Z_i \text{ odd}\}} > 6qN_{\mathrm{prod}} \Big) \\ &\geq \sum_{n = \lfloor c_1 (r+s)^{\frac{2}{3}} \rfloor}^{r+s} \mathbb{P}\Big(\sum_{i=1}^{n-1} \chi_{\{Z_{i+1} - Z_i \text{ odd}\}} > \frac{n}{5} \,\Big|\, N = n \Big) \mathbb{P}(N = n) \\ &\geq \sum_{n = \lfloor c_1 (r+s)^{\frac{2}{3}} \rfloor}^{r+s} \exp\left(-\frac{n}{100} \right) \mathbb{P}(N = n) \\ &\geq \left[1 - \exp\left(- \left\lfloor \frac{c_1 (r+s)^{\frac{2}{3}}}{100} \right\rfloor \right) \right] \cdot \mathbb{P}(N \geq \lfloor c_1 (r+s)^{2/3} \rfloor) \end{aligned} \tag{5.15}$$

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

where the application of Lemma 5.6 is justified by the bound $\lfloor c_1(r+s)^{2/3} \rfloor \geq 30qN_{\text{prod}} \geq 10$ and the initial equality in the first line is justified by the fact that $f_a(z^i)$ depends only on the parity of i, a fact itself readily seen from the definition of f_a and z^i.

Now, by differentiating it is easy to see that the function $p \mapsto p \log(8/p)$ is increasing on $(0, 1)$. Hence for $p < 1$, we have $p^{-2} \log(8) > p^{-1} \log(8) > \log(8/p)$ and so combining this with (2.3) and (5.12) gives

$$\left\lfloor \frac{c_1(r+s)^{2/3}}{100} \right\rfloor \geq \left\lfloor \frac{c_1 C^{2/3} \mathrm{p}^{-2}}{100} \right\rfloor \geq \left\lfloor \frac{200^{2/3} \mathrm{p}^{-2} \log(8)}{100} \right\rfloor \geq \mathrm{p}^{-2} \log(8) - 1 \geq \log(8/\mathrm{p}) - 1. \tag{5.16}$$

Furthermore, using item (i) of Lemma 5.6 with $\gamma = 3/2$, we obtain $\mathbb{P}\big(N \geq c_1(r+s)^{2/3}\big) \geq 1 - c_1^{-2}(r+s)^{-1/3} \geq 1 - \mathrm{p}/4$, where the final bound follows because $r + s \geq Cp^{-3}$ (which, in turn, is due to the Assumption (2.3)) and (5.11). Using this result together with (5.16) in (5.15) yields

$$\mathbb{P}\Big(\sum_{i=1}^{N-1} \chi_{\{f_a(z^{i+1}) \neq f_a(z^i)\}} > 6qN_{\text{prod}} \Big) > (1 - e\mathrm{p}/8)\,(1 - \mathrm{p}/4) > 1 - \mathrm{p}/2 \tag{5.17}$$

Combining (5.14) and (5.17), we see that the probability that both

$$\max\{k \in \mathbb{N} \mid x^{k,\delta} \in \mathcal{T} \cup \mathcal{V}\} \leq \left\lceil \frac{C(r \vee s)^2}{\mathrm{p}^2} \right\rceil \quad \text{and} \quad \sum_{i=1}^{N-1} \chi_{\{f_a(z^{i+1}) \neq f_a(z^i)\}} > 6qN_{\text{prod}} \tag{5.18}$$

occur is at least $1 - (\mathrm{p}/4 + \mathrm{p}/2) > 1 - \mathrm{p}$. We will now proceed to show that each of (i) through (iii) listed as in the statement of Theorem 2.2 hold assuming that this event occurs.

Proof of (i): Success – great generalisability

To see that $\mathcal{T}, \mathcal{V} \in \mathcal{S}^f_{\varepsilon((r\vee s)/\mathrm{p})}$, note that (5.12) and $c_1^{-1} \geq 1$ yields $C^2 t^2 \geq (4\lceil t \rceil + 3)(4\lceil t \rceil + 4)$ for all $t \geq 1$. Applying this inequality with $t = C((r \vee s)/\mathrm{p})^2 \geq 1$, we deduce that

$$\begin{aligned} \varepsilon\left\lceil \frac{C(r\vee s)}{\mathrm{p}} \right\rceil = C^{-2} \left(\frac{C(r\vee s)^2}{\mathrm{p}^2} \right)^{-2} &\leq \left[\left(4 \left\lceil \frac{C(r\vee s)^2}{\mathrm{p}^2} \right\rceil + 3 \right) \left(4 \left\lceil \frac{C(r\vee s)^2}{\mathrm{p}^2} \right\rceil + 4 \right) \right]^{-1} \\ &= \varepsilon' \left(\left\lceil \frac{C(r\vee s)^2}{\mathrm{p}^2} \right\rceil \right), \end{aligned} \tag{5.19}$$

where $\varepsilon'(n) = [(4n+3)(4n+4)]^{-1}$. Therefore because we assume that $\max\{k \in \mathbb{N} \mid x^{k,\delta} \in \mathcal{T} \cup \mathcal{V}\} \leq \left\lceil \frac{C(r\vee s)^2}{\mathrm{p}^2} \right\rceil$, Lemma 5.2 yields $\mathcal{T}, \mathcal{V} \subset \{x^{1,\delta}, \ldots, x^{\lceil C(r\vee s)^2/\mathrm{p}^2, \delta \rceil}\} \in \mathcal{S}^{f_a}_{\varepsilon'(\lceil C(r\vee s)^2/\mathrm{p}^2 \rceil)} \subset \mathcal{S}^{f_a}_{\varepsilon(C(r\vee s)/\mathrm{p})}$.

The construction of ϕ satisfying (2.5) is immediate: we take ϕ to be the neural network $\tilde{\varphi}$ defined in Lemma 5.3. We conclude that $\phi(x) = f_a(x)$ for all $x \in \mathcal{T} \cup \mathcal{V}$ (this establishes (2.5)). Because $\phi(x) = f_a(x)$ for all $x \in \mathcal{T}$ and because $\mathcal{R} \in \mathcal{CF}_r$ we conclude that $\mathcal{R}\left(\{\phi(x^j)\}_{j=1}^r, \{f(x^j)\}_{j=1}^r\right) = 0$. Thus (2.4) holds, completing the proof of (i).

Proof of (ii): Any successful NN in $\mathcal{NN}_{\mathbf{N},L}$ – regardless of architecture – becomes universally unstable

Our next task will be to show that if $\hat{\phi} \in \mathcal{NN}_{\mathbf{N},L}$ and $g{:}\mathbb{R} \to \mathbb{R}$ is monotonic, then there is a subset $\tilde{\mathcal{T}} \subset \mathcal{T} \cup \mathcal{V}$ of the combined training and validation set of size $|\tilde{\mathcal{T}}| \geq q$, such that there exist uncountably many universal adversarial perturbations $\eta \in \mathbb{R}^d$ so that for each $x \in \tilde{\mathcal{T}}$ Eq. (2.6) applies.

To this end, note that (5.18) implies that there exist natural numbers $k_1 < k_2 < \ldots < k_{6qN_{\text{prod}}}$ such that $z_1^{k_i} > z_1^{k_{i+1}}$ and $f_a(z^{k_i}) \neq f_a(z^{k_{i+1}})$ for all $i \in \{1, \ldots, 6qN_{\text{prod}} - 1\}$. Moreover, by the definition of $\mathcal{T}$, $\mathcal{V}$ and S, there exist m_i such that $z_1^{k_i} = x_1^{m_i,\delta}$ and such that $x^{m_i,\delta} \in \mathcal{T} \cup \mathcal{V}$. For such i and any $\omega \in [0, \delta \wedge \varepsilon((r \vee s)/\mathrm{p}))$, we define the vectors $w^{i,\omega} = z^{k_i} + \omega e_1$. We also define the sets $\mathcal{W}^\omega := \{w^{i,\omega} \mid i \in \{1, \ldots, 6qN_{\text{prod}}\}\}$.

Because of the definition of $x^{k,0}$ given in (5.1) and the definition of z^{k_i}, we have $z_2^{k_i} = z_3^{k_i} = \cdots = z_d^{k_i} = 0$ and $z^{k_i} = x^{m_i,0}$. In particular, $\{z^{k_i} \mid i \in \{1, \ldots, 6qN_{\text{prod}}\}\} = \{x^{m_i,0} \mid i \in \{1, \ldots, 6qN_{\text{prod}}\}\} \in \mathcal{S}^{f_a}_{\varepsilon((r\vee s)/\mathrm{p})}$ where we have used Lemma 5.2 and the bound (5.19). Since $\|z^{k_i} - w^{i,\omega}\|_\infty = \omega < \varepsilon((r \vee s)/\mathrm{p})$, we conclude that $f_a(z^{k_i}) = f_a(w^{i,\omega})$ for $i \in \{1, \ldots, 6qN_{\text{prod}}\}$. Thus, $f_a(w^{i,\omega}) = f_a(z^{k_i}) \neq f_a(z^{k_{i+1}}) = f_a(w^{i+1,\omega})$ for $i \in \{1, \ldots, 6qN_{\text{prod}} - 1\}$.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

We can now use Lemma 5.4 to conclude that for each $\omega \in [0, \delta \wedge \varepsilon((r \vee s)/\mathrm{p}))$ there exists a set $\mathcal{I}^\omega$ and a set $\mathcal{U}^\omega \subset \mathcal{W}^\omega$ with the following properties:

(1) $\mathcal{I}^\omega \subset \{1, 2, \dots, 6qN_{\mathrm{prod}}\}$

(2) $\mathcal{U}^\omega = \{w^{i,\omega} \mid i \in \mathcal{I}^\omega\}$

(3) For all $w \in \mathcal{U}^\omega$, $|g(\hat{\phi}(w)) - f_a(w)| \geq 1/2$.

(4) $|\mathcal{U}^\omega| \geq 2q$.

By the pigeonhole principle and the finiteness of $\{1, 2, \dots, 6qN_{\mathrm{prod}}\}$, there exists an uncountable set $\Omega \subset [0, \delta \wedge \varepsilon((r \vee s)/\mathrm{p}))$ such that for all $\omega \in \Omega$, $\mathcal{I}^\omega$ is independent of ω. Let $\mathcal{I}$ denote this common value and let $\mathcal{I}_E := \{i \mid i \in \mathcal{I}, m_i \text{ even}\}$ and $\mathcal{I}_O := \{i \mid i \in \mathcal{I}, m_i \text{ odd}\}$. Note that $|\mathcal{I}| \geq 2q$; otherwise, $|\mathcal{U}^\omega| < 2q$ for some ω. Therefore, at least one of $|\mathcal{I}_E| \geq q$ or $|\mathcal{I}_O| \geq q$: we now split into two cases depending on which of these two sets has cardinality at least q.

Case 1: $|\mathcal{I}_E| \geq q$.

In this case, we choose $\tilde{\mathcal{T}} = \{x^{m_i,\delta} \mid i \in \mathcal{I}_E\}$. For each $\omega \in \Omega$, define $\eta^\omega = (\omega, -\delta, 0, \dots, 0) \in \mathbb{R}^d$ and $\mathcal{H} = \{\eta^\omega \mid \omega \in \Omega\}$. Then the set $\mathcal{H}$ is uncountable, for each $i \in \mathcal{I}_E$ and $\omega \in \Omega$ we have $x^{m_i,\delta} + \eta^\omega = w^{i,\omega}$, $|g(\hat{\phi}(x^{m_i,\delta} + \eta^\omega)) - f_a(x^{m_i,\delta} + \eta^\omega)| = |g(\hat{\phi}(w^{i,\omega})) - f_a(w^{i,\omega})| \geq 1/2$ and $\|\eta\| \leq D\|\eta^\omega\|_1 = D(\omega + \delta) \leq 2D\delta \leq \epsilon$. Furthermore, $|\mathrm{supp}(\eta^\omega)| = 2$. We conclude that (2.6) holds.

Case 2: $|\mathcal{I}_O| \geq q$

In this case, we choose $\tilde{\mathcal{T}} = \{x^{m_i,\delta} \mid i \in \mathcal{I}_O\}$. For each $\omega \in \Omega$, define $\eta^\omega = (\omega, 0, 0, \dots, 0) \in \mathbb{R}^d$ and $\mathcal{H} = \{\eta^\omega \mid \omega \in \Omega\}$. Then the set $\mathcal{H}$ is uncountable, for each $i \in \mathcal{I}_O$ and $\omega \in \Omega$ we have $x^{m_i,\delta} + \eta^\omega = w^{i,\omega}$, $|g(\hat{\phi}(x^{m_i,\delta} + \eta^\omega)) - f_a(x^{m_i,\delta} + \eta^\omega)| = |g(\hat{\phi}(w^{i,\omega})) - f_a(w^{i,\omega})| \geq 1/2$ and $\|\eta^\omega\| \leq D\|\eta^\omega\|_1 = D\omega \leq D\delta \leq \epsilon$. Furthermore, $|\mathrm{supp}(\eta^\omega)| = 1$. We conclude that (2.6) holds.

Proof of (iii): Other stable and accurate NNs exist

Finally, we must show the existence of ψ, which we do with the help of Lemma 5.7. To this end, we set $K = \lceil C((r \vee s)/\mathrm{p})^2 \rceil$ and define $\{\alpha_j\}_{j=1}^{2K}$ by $\alpha_{2k-1} = x_1^{k,\delta} + \varepsilon((r \vee s)/\mathrm{p})$, $\alpha_{2k} = x_1^{k,\delta} - \varepsilon((r \vee s)/\mathrm{p})$ for $k = 1, \dots, K$. We first claim that $0 < \alpha_{2K} < \alpha_{2K-1} < \cdots < \alpha_2 < \alpha_1 < 1$.

Because $C \geq 4^3$, $p \leq 1$ and $(r \vee s) \geq 1$ we have

$$\alpha_1 = \frac{a}{2-\kappa} + \frac{\mathrm{p}^4}{C^4(r \vee s)^4} \leq \frac{1}{(2-3/4)} + \frac{1}{C^4} < 1$$

and similarly we obtain $2\lceil C((r \vee s)/\mathrm{p})^2 \rceil + 1 - \kappa \leq 2(C((r \vee s)/\mathrm{p})^2) + 2 - \kappa \leq 4(C((r \vee s)/\mathrm{p})^2)$. Therefore,

$$\begin{aligned}\alpha_{2K} = \frac{a}{2\lceil C((r \vee s)/\mathrm{p})^2 \rceil + 1 - \kappa} - \frac{\mathrm{p}^4}{C^4(r \vee s)^4} &\geq \frac{a}{4C((r \vee s)/\mathrm{p})^2} - \frac{\mathrm{p}^4}{C^4(r \vee s)^4} \\ &\geq \frac{\mathrm{p}^2}{8C(r \vee s)^2} - \frac{\mathrm{p}^2}{4^{12}C(r \vee s)^2} > 0\end{aligned}$$

A simple calculation also shows that for each $j = 1, \dots, K-1$

$$x_1^{j,\delta} - x_1^{j+1,\delta} = \frac{a}{(j+2-\kappa)(j+1-\kappa)} \geq \frac{a}{(K+1-\kappa)(K-\kappa)} \geq [2(K+1-\kappa)(K-\kappa)]^{-1}.$$

On the other hand, once again employing the result that $C^2t^2 \geq (4\lceil t \rceil + 3)(4\lceil t \rceil + 4)$, for all $t \geq 1$, (which is a consequence of (5.12)) with $t = C((r \vee s)/\mathrm{p})^2$ we obtain

$$2\varepsilon((r \vee s)/\mathrm{p}) = 2C^{-2}(C((r \vee s)/\mathrm{p})^2)^{-2} \leq 2[(4K+3)(4K+4)]^{-1} < [2(K+1-\kappa)(K-\kappa)]^{-1}.$$

We therefore conclude that $\alpha_{2j-1} > \alpha_{2j} = x_1^{j,\delta} - \varepsilon((r \vee s)/\mathrm{p}) > x_1^{j+1,\delta} + \varepsilon((r \vee s)/\mathrm{p}) = \alpha_{2j+1}$, and thus the conditions to apply Lemma 5.7 are met.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

Now, let ψ be the network provided by Lemma 5.7 with this sequence $\{\alpha_j\}_{j=1}^{2K}$. Because of the definition of α_j and the conclusion of Lemma 5.7 we have

$$\psi(x) = \begin{cases} 0 & \text{if } x_1 \in [x_1^{k,\delta} - \varepsilon((r\vee s)/\mathrm{p}), x_1^{k,\delta} + \varepsilon((r\vee s)/\mathrm{p})] \text{ and } k \text{ is odd} \\ 1 & \text{if } x_1 \in [x_1^{k,\delta} - \varepsilon((r\vee s)/\mathrm{p}), x_1^{k,\delta} + \varepsilon((r\vee s)/\mathrm{p})] \text{ and } k \text{ is even} \end{cases}$$

Moreover, because of Lemma 5.2, the fact that the value of $f_a(x)$ depends only on x_1 and the bound (5.19)

$$f_a(x) = \begin{cases} 0 & \text{if } x_1 \in [x_1^{k,\delta} - \varepsilon((r\vee s)/\mathrm{p}), x_1^{k,\delta} + \varepsilon((r\vee s)/\mathrm{p})] \text{ and } k \text{ is odd} \\ 1 & \text{if } x_1 \in [x_1^{k,\delta} - \varepsilon((r\vee s)/\mathrm{p}), x_1^{k,\delta} + \varepsilon((r\vee s)/\mathrm{p})] \text{ and } k \text{ is even} \end{cases}$$

In particular, $\psi(x) = f_a(x)$ whenever $x_1 \in [x_1^{k,\delta} - \varepsilon((r\vee s)/\mathrm{p}), x_1^{k,\delta} + \varepsilon((r\vee s)/\mathrm{p})]$ for any $k \in \{1, 2, \ldots, K\}$.

To see that $\psi(x) = f_a(x)$ for all $x \in \mathcal{B}^{\infty}_{\varepsilon((r\vee s)/\mathrm{p})}(\mathcal{T}\cup\mathcal{V})$, note that, for every $x \in \mathcal{B}^{\infty}_{\varepsilon((r\vee s)/\mathrm{p})}(\mathcal{T}\cup\mathcal{V})$, there exists an $x^{k,\delta} \in \mathcal{T}\cup\mathcal{V}$ such that $\|x^{k,\delta} - x\|_\infty \le \varepsilon((r\vee s)/\mathrm{p})$. Then, by the assumption that $\max\{\ell \in \mathbb{N} \mid x^{\ell,\delta} \in \mathcal{T}\cup\mathcal{V}\} \le \lceil C((r\vee s)/\mathrm{p})^2\rceil$ occurs in (5.18), we have $k \le K$, and so $x_1 \in [x_1^{k,\delta} - \varepsilon((r\vee s)/\mathrm{p}), x_1^{k,\delta} + \varepsilon((r\vee s)/\mathrm{p})]$. But we have already shown that for such x, $\psi(x) = f_a(x)$. Thus, the proof of the theorem is complete. □

5.4. Tools from the SCI hierarchy used for Theorem 3.5

In order to formalise the non-computability result stated in Theorem 3.5, we shall summarise appropriate definitions and ideas on the 'SCI hierarchy' [11–13, 25, 30, 43, 59, 60]. The material in this section very closely follows the definitions and presentation in [9] with slight adaptations made owing to the different focus of this paper. Working with the SCI hierarchy and general algorithms allows us to show the non-computability is independent of both the underlying computational model (e.g., a Turing machine, BSS machine) and local minima as in Remark 3.7.

It also allows us to easily make non-computability statements applicable to both deterministic and randomised algorithms. We include the ensuing discussion to ensure that this paper is self-contained.

5.4.1. Computational problems

We start by defining a *computational problem* [11]:

Definition 5.8 (Computational problem). *Let Ω be some set, which we call the input set, and Λ be a set of complex-valued functions on Ω such that for $\iota_1, \iota_2 \in \Omega$, then $\iota_1 = \iota_2$ if and only if $f(\iota_1) = f(\iota_2)$ for all $f \in \Lambda$. We call Λ an evaluation set. Let $(\mathcal{M}, d_{\mathcal{M}})$ be a metric space, and finally let $\Xi : \Omega \to \mathcal{M}$ be a function which we call the solution map. We call the collection $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ a computational problem.*

The set Ω is essentially the set of objects that give rise to the various instances of our computational problem. The solution map $\Xi : \Omega \to \mathcal{M}$ is what we are interested in computing. Finally, the set Λ is the collection of functions that provide us with the information we are allowed to read. As a simple example, if we were considering matrix inversion then Ω might be a collection of invertible matrices, Ξ would be the matrix inversion map taking Ω to the set of matrices and Λ would consist of functions that allow us to access entries of the input matrices.

In the slightly more complicated context of a computational problem, the neural network problem formulated in Section 3 can be understood as per the following:

Definition 5.9 (Neural network computational problem). *Fix $d, r \in \mathbb{N}$, a classification function $f{:}\mathbb{R}^d \to \{0, 1\}$, neural network layers and dimensions L and $\mathbf{N} = (N_L = 1, N_{L-1}, \ldots, N_1, N_0 = d)$, respec-*

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

tively, as well as $\epsilon, \hat{\epsilon}$ *and a cost function* $\mathcal{R} \in \mathcal{CF}_r^{\epsilon,\hat{\epsilon}}$. *The neural network computational problem*

$$\{\Xi^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}, \Omega^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}, \mathcal{M}^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}, \Lambda^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}\}$$

is defined as follows:

(1) The input set $\Omega^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}$ *is the collection of all* $\mathcal{T}$ *with* $\mathcal{T} = \{x^1, \dots, x^r\}$ *a finite subset of* $\mathbb{R}^d$ *such that* $\mathcal{T} \in \mathcal{S}^f_{\varepsilon'(K)}$ *with* $\varepsilon'(n) := [(4n+3)(4n+4)]^{-1}$.

(2) The metric space $\mathcal{M}^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}$ *is set to* $\mathbb{R}^r$ *with the distance function induced by* $\|\cdot\|_*$ *where* $* = 1, 2$ *or* ∞ *as per the statement of Theorem 3.5.*

(3) The solution map $\Xi^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}$ *is given by the following: for a training set* $\mathcal{T}$, *we let*

$$\mathcal{A}^{\epsilon}_{\mathcal{T}} := \operatorname*{argmin}_{\epsilon,\, \varphi \in \mathcal{NN}_{\mathbf{N},L}} \mathcal{R}\left(\{\varphi(x^j)\}_{j=1}^r, \{f(x^j)\}_{j=1}^r\right),$$

and then $\Xi^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}(\mathcal{T}) = \{\phi(x^j)\}_{i=1}^r$ *for* $\phi \in \mathcal{A}^{\epsilon}_{\mathcal{T}}$. *Note that* Ξ *is potentially multivalued if* $\mathcal{A}^{\epsilon}_{\mathcal{T}}$ *has more than one element – this will not be a problem for our theory and will be explained further in Remark 5.15.*

(4) The set $\Lambda^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}$ *is given by*

$$\Lambda^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)} = \{f^{j,k}\}_{j=1,k=1}^{j=d,k=r}, \tag{5.20}$$

where $f^{j,k}(\mathcal{T}) = x^k_j$ *gives access to the jth coordinate of the kth vector of the training set.*

To reduce the burden on notation, we will abbreviate

$$\{\Xi^{\mathcal{NN}}, \Omega^{\mathcal{NN}}, \mathcal{M}^{\mathcal{NN}}, \Lambda^{\mathcal{NN}}\} = \{\Xi^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}, \Omega^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}, \mathcal{M}^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}, \Lambda^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}\}$$

where there is no ambiguity surrounding the parameters $f, r, \epsilon, \mathcal{R}, \mathbf{N}, L$.

Remark 5.10 (Existence of a neural network). It may not be a priori obvious that the set $\mathcal{A}^{\epsilon}_{\mathcal{T}}$ is non-empty and thus $\Xi^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}(\mathcal{T})$ is well defined. In fact, this is an immediate consequence the fact that the cost function $\mathcal{R}$ is a member of $\mathcal{CF}_r^{\epsilon,\hat{\epsilon}}$ defined in (3.4) and the definition of $\operatorname{argmin}_{\epsilon}$ given in (3.3). In particular, the existence of an approximate minimiser is guaranteed since $\mathcal{R}$ is bounded from below.

5.4.2. Algorithms

In this section, we shall describe the algorithms that are designed to approximate the solution map Ξ in a computational problem $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$. We shall start with deterministic general algorithms:

Definition 5.11 (General Algorithm). *Given a computational problem* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$, *a general algorithm is a mapping* $\Gamma : \Omega \to \mathcal{M} \cup \{NH\}$ *such that, for every* $\iota \in \Omega$, *the following conditions hold:*

(i) there exists a non-empty subset of evaluations $\Lambda_\Gamma(\iota) \subset \Lambda$, *and, whenever* $\Gamma(\iota) \neq NH$, *we have* $|\Lambda_\Gamma(\iota)| < \infty$,

(ii) the action of Γ *on* ι *is uniquely determined by* $\{f(\iota)\}_{f \in \Lambda_\Gamma(\iota)}$,

(iii) for every $\iota' \in \Omega$ *such that* $f(\iota') = f(\iota)$ *for all* $f \in \Lambda_\Gamma(\iota)$, *it holds that* $\Lambda_\Gamma(\iota') = \Lambda_\Gamma(\iota)$.

Remark 5.12 (The purpose of a general algorithm: universal impossibility results). The purpose of a general algorithm is to have a definition that will encompass any model of computation and that will allow impossibility results to become universal. Given that there are several non-equivalent models of computation, impossibility results will be shown with this general definition of an algorithm.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

Remark 5.13 (The power of a general algorithm). General algorithms are extremely powerful computational models with every Turing or BSS machine a general algorithm but the converse does not hold. Thus, a non-computability result proven using general algorithms is strictly stronger than one proven only for Turing machines or BSS machines.

In particular, general algorithms are more powerful than any Turing machine or BSS machine, or even such a machine with access to an oracle that provides an approximate minimiser

$$\phi \in \operatorname*{argmin}_{\tilde{\phi} \in \mathcal{NN}_{\mathbf{N},L}}{}_{\epsilon} \mathcal{R}\left(\{\tilde{\phi}(x^j)\}_{j=1}^r, \{f(x^j)\}_{j=1}^r\right)$$

for every inexact input provided to the algorithm, or an oracle that detects when an algorithm has encountered local minima. It is for this reason that we stated in Remark 3.7 that local minima were not relevant to Theorem 3.5.

Remark 5.14 (The non-halting output NH). The non-halting 'output' NH of a general algorithm may seem like an unnecessary distraction given that a general algorithm is just a mapping, which is strictly more powerful than a Turing or a BSS machine. However, the NH output is needed when the concept of a general algorithm is extended to a randomised general algorithm (RGA). A technical remark about NH is also appropriate, namely that $\Lambda_\Gamma(\iota)$ is allowed to be infinite in the case when $\Gamma(\iota) = \mathrm{NH}$. This is to allow general algorithms to capture the behaviour of a Turing or a BSS machine not halting by virtue of requiring an infinite amount of input information.

Owing to the presence of the special non-halting 'output' NH, we have to extend the metric $d_{\mathcal{M}}$ on $\mathcal{M} \times \mathcal{M}$ to $d_{\mathcal{M}}: \mathcal{M} \cup \{\mathrm{NH}\} \times \mathcal{M} \cup \{\mathrm{NH}\} \to \mathbb{R}_{\geq 0}$ in the following way:

$$d_{\mathcal{M}}(x,y) = \begin{cases} d_{\mathcal{M}}(x,y) & \text{if } x, y \in \mathcal{M} \\ 0 & \text{if } x = y = \mathrm{NH} \\ \infty & \text{otherwise.} \end{cases} \tag{5.21}$$

Definition 5.11 is sufficient for defining a RGA, which is the only tool from the SCI theory needed in order to prove Theorem 3.5.

Remark 5.15 (Multivalued functions). When dealing with optimisation problems, one needs a framework that can handle multiple solutions. As the set-up above does not allow Ξ to be multivalued, we need some slight changes. We allow Ξ to be multivalued, even though a general algorithm is assumed not to be. For $\iota \in \Omega$, we define $\mathrm{dist}_{\mathcal{M}}(\Xi(\iota), \Gamma(\iota)) := \inf_{x \in \Xi(\iota)} d_{\mathcal{M}}(x, \Gamma(\iota))$. That is to say, the error that Γ is assumed to incur in trying to compute $\Xi(\iota)$ is the best (infimum) of all possible errors across all values of $\Xi(\iota)$.

One final definition that is useful is that of the *minimum amount of input information*, defined if Λ is countable. Although this definition has its own uses in other work on the SCI hierarchy, in the context of this paper it will only be useful to address a technicality in the next section.

Definition 5.16 (Minimum amount of input information). *Given the computational problem* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$*, where* $\Lambda = \{f_k \mid k \in \mathbb{N},\ k \leq |\Lambda|\}$ *and a general algorithm* Γ*, we define the minimum amount of input information* $T_\Gamma(\iota)$ *for* Γ *and* $\iota \in \Omega$ *as*

$$T_\Gamma(\iota) := \sup\{m \in \mathbb{N} \mid f_m \in \Lambda_\Gamma(\iota)\}.$$

Note that, for ι *such that* $\Gamma(\iota) = NH$*, the set* $\Lambda_\Gamma(\iota)$ *may be infinite (see Definition 5.11), in which case* $T_\Gamma(\iota) = \infty$.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

5.4.3. Randomised algorithms

In many contemporary fields of mathematics of information such as DL, the use of randomised algorithms is widespread. We therefore need to extend the concept of a general algorithm to a *randomised random algorithm*.

Definition 5.17 (Randomised general algorithm). *Given a computational problem* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$, *where* $\Lambda = \{f_k \mid k \in \mathbb{N},\ k \leq |\Lambda|\}$, *a RGA is a collection* X *of general algorithms* $\Gamma : \Omega \to \mathcal{M} \cup \{NH\}$, *a sigma-algebra* $\mathcal{F}$ *on* X, *and a family of probability measures* $\{\mathbb{P}_\iota\}_{\iota \in \Omega}$ *on* $\mathcal{F}$ *such that the following conditions hold:*

(Pi) *For each* $\iota \in \Omega$, *the mapping* $\Gamma_\iota^{\text{ran}}{:}(X, \mathcal{F}) \to (\mathcal{M} \cup \{NH\}, \mathcal{B})$ *defined by* $\Gamma_\iota^{\text{ran}}(\Gamma) = \Gamma(\iota)$ *is a random variable, where* $\mathcal{B}$ *is the Borel sigma-algebra on* $\mathcal{M} \cup \{NH\}$.

(Pii) *For each* $n \in \mathbb{N}$ *and* $\iota \in \Omega$, *we have* $\{\Gamma \in X \mid T_\Gamma(\iota) \leq n\} \in \mathcal{F}$.

(Piii) *For all* $\iota_1, \iota_2 \in \Omega$ *and* $E \in \mathcal{F}$ *so that, for every* $\Gamma \in E$ *and every* $f \in \Lambda_\Gamma(\iota_1)$, *we have* $f(\iota_1) = f(\iota_2)$, *it holds that* $\mathbb{P}_{\iota_1}(E) = \mathbb{P}_{\iota_2}(E)$.

It is not immediately clear whether condition (Pii) for a given RGA $(X, \mathcal{F}, \{\mathbb{P}_\iota\}_{\iota \in \Omega})$ *holds independently of the choice of the enumeration of* Λ. *This is indeed the case, but we shall not show this here (see [9] for further information).*

Remark 5.18 (Assumption (Pii)). Note that (Pii) in Definition 5.17 is needed in order to ensure that the minimum amount of input information (i.e., the amount of input information the algorithm makes use of) also becomes a valid random variable. More specifically, for each $\iota \in \Omega$, we define the random variable

$$T_{\Gamma^{\text{ran}}}(\iota){:}X \to \mathbb{N} \cup \{\infty\} \text{ according to } \Gamma \mapsto T_\Gamma(\iota).$$

Assumption (Pii) ensures that this is indeed a random variable.

As the minimum amount of input information is typically related to the complexity of an algorithm, one would be dealing with a rather exotic probabilistic model if $T_{\Gamma^{\text{ran}}}(\iota)$ were not a random variable. Indeed, note that the standard models of randomised algorithms (see [5]) can be considered as RGAs (in particular, they will satisfy (Pii)).

Remark 5.19 (The purpose of a randomised general algorithm: universal lower bounds). As for a general algorithm, the purpose of a RGA is to have a definition that will encompass every model of computation, which will allow lower bounds and impossibility results to be universal. Indeed, randomised Turing and BSS machines can be viewed as RGAs.

We will, with a slight abuse of notation, also write RGA for the family of all RGAs for a given a computational problem and refer to the algorithms in RGA by Γ^{ran}. With the definitions above, we can now make probabilistic version of the strong breakdown epsilon as follows.

Definition 5.20 (Probabilistic strong breakdown epsilon). *Given a computational problem* $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$, *where* $\Lambda = \{f_k \mid k \in \mathbb{N},\ k \leq |\Lambda|\}$, *we define the probabilistic strong breakdown epsilon* $\epsilon_{\mathbb{PB}}^{\text{s}}{:}[0, 1) \to \mathbb{R}$ *according to*

$$\epsilon_{\mathbb{PB}}^{\text{s}}(\text{p}) = \sup\{\epsilon \geq 0,\ \mid \forall\, \Gamma^{\text{ran}} \in \text{RGA}\ \exists\, \iota \in \Omega \text{ such that } \mathbb{P}_\iota(\text{dist}_{\mathcal{M}}(\Gamma_\iota^{\text{ran}}, \Xi(\iota)) > \epsilon) > \text{p}\},$$

where $\Gamma_\iota^{\text{ran}}$ *is defined in (Pi) in Definition 5.17.*

Note that the probabilistic strong breakdown epsilon is not a single number but a function of p. Specifically, it is the largest ϵ so that the probability of failure with at least ϵ-error is greater than p.

5.4.4. Inexact input and perturbations

Suppose we are given a computational problem $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$, and that $\Lambda = \{f_j\}_{j \in \beta}$, where β is some index set that can be finite or infinite. Obtaining f_j may be a computational task on its own, which is

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

exactly the problem in most areas of computational mathematics. In particular, for $\iota \in \Omega$, $f_j(\iota)$ could be the number $e^{\frac{\pi}{j}i}$ for example. Hence, we cannot access $f_j(\iota)$, but rather $f_{j,n}(\iota)$ where $f_{j,n}(\iota) \to f_j(\iota)$ as $n \to \infty$. In this paper, we will be interested in the case when this can be done with error control. In particular, we consider $f_{j,n} : \Omega \to \mathbb{D}_n + i\mathbb{D}_n$, where $\mathbb{D}_n := \{k\, 2^{-n} \mid k \in \mathbb{Z}\}$, such that

$$\|\{f_{j,n}(\iota)\}_{j\in\beta} - \{f_j(\iota)\}_{j\in\beta}\|_\infty \leq 2^{-n}, \quad \forall \iota \in \Omega. \tag{5.22}$$

We will call a collection of such functions Δ_1-information for the computational problem. Formally, we have the following.

Definition 5.21 (Δ_1-information). *Let $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ be a computational problem with $\Lambda = \{f_j\}_{j\in\beta}$. Suppose that, for each $j \in \beta$ and $n \in \mathbb{N}$, there exists an $f_{j,n} : \Omega \to \mathbb{D}_n + i\mathbb{D}_n$ such that (5.22) holds. We then say that the set $\hat{\Lambda} = \{f_{j,n} \mid j \in \beta, n \in \mathbb{N}\}$ provides Δ_1-information for $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$.*

We can now define what we mean by a computational problem with Δ_1-information.

Definition 5.22 (Computational problem with Δ_1-information). *Given $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ with $\Lambda = \{f_j\}_{j\in\beta}$, the corresponding computational problem with Δ_1-information is defined as:*

$$\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1} := \{\tilde{\Xi}, \tilde{\Omega}, \mathcal{M}, \tilde{\Lambda}\},$$

where

$$\tilde{\Omega} = \left\{\tilde{\iota} = \left\{(f_{j,1}(\iota), f_{j,2}(\iota), f_{j,3}(\iota), \dots)\right\}_{j\in\beta} \mid \iota \in \Omega,\ f_{j,n} : \Omega \to \mathbb{D}_n + i\mathbb{D}_n \text{ satisfy (5.22)}\right\}, \tag{5.23}$$

$\tilde{\Xi}(\tilde{\iota}) = \Xi(\iota)$, and $\tilde{\Lambda} = \{\tilde{f}_{j,n}\}_{j,n\in\beta\times\mathbb{N}}$, where $\tilde{f}_{j,n}(\tilde{\iota}) = f_{j,n}(\iota)$. Given an $\tilde{\iota} \in \tilde{\Omega}$, there is a unique $\iota \in \Omega$ for which $\tilde{\iota} = \left\{(f_{j,1}(\iota), f_{j,2}(\iota), f_{j,3}(\iota), \dots)\right\}_{j\in\beta}$ (by Definition 5.8). We say that this $\iota \in \Omega$ corresponds to $\tilde{\iota} \in \tilde{\Omega}$.

Remark 5.23. Note that the correspondence of a unique ι to each $\tilde{\iota}$ in Definition 5.22 ensures that $\tilde{\Xi}$ and the elements of $\tilde{\Lambda}$ are well defined.

One may interpret the computational problem $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1} = \{\tilde{\Xi}, \tilde{\Omega}, \mathcal{M}, \tilde{\Lambda}\}$ as follows. The collection $\tilde{\Omega}$ is the family of all sequences approximating the inputs in Ω. For an algorithm to be successful for $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1}$, it must work for all $\tilde{\iota} \in \tilde{\Omega}$, that is, for any sequence approximating ι.

Remark 5.24 (Oracle tape/node providing Δ_1-information). For impossibility results, we use general algorithms and RGAs (as defined below), and thus, due to their generality, we do not need to specify how the algorithms read the information.

The next proposition serves as the key building block for Theorem 3.5 and is proven in [[9], Proposition 9.5]. Note that the proposition is about arbitrary computational problems and is hence also a tool for demonstrating lower bounds on the breakdown epsilon for general computational problems.

Proposition 5.25. *Let $\{\Xi, \Omega, \mathcal{M}, \Lambda\}$ be a computational problem with $\Lambda = \{f_k \mid k \in \mathbb{N}, k \leq |\Lambda|\}$ countable. Suppose that $\iota^0 \in \Omega$ and that $\{\iota_n^1\}_{n=1}^\infty$ is a sequence in Ω so that the following conditions hold:*

(Pa) For every $k \leq |\Lambda|$ and for all $n \in \mathbb{N}$, we have $|f_k(\iota_n^i) - f_k(\iota^0)| \leq 1/4^n$.

(Pb) There is a $\kappa > 0$ such that $\inf_{\upsilon^1\in\Xi(\iota_n^1),\upsilon^2\in\Xi(\iota^0)} d_{\mathcal{M}}(\upsilon^1, \upsilon^2) \geq \kappa$.

Then the computational problem $\{\Xi, \Omega, \mathcal{M}, \Lambda\}^{\Delta_1}$ satisfies $\epsilon^{\mathrm{s}}_{\mathbb{P}\mathrm{B}}(\mathrm{p}) \geq \kappa/2$ for $\mathrm{p} \in [0, 1/2)$.

5.5. Stating Theorem 3.5 in the SCI language - Proposition 5.26

A slightly stronger formal statement of Theorem 3.5 in the SCI language is now as follows.

Proposition 5.26. *There is an uncountable collection $\mathcal{C}_1$ of classification functions f as in (2.1) – with fixed $d \geq 2$ – such that for*

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

(1) any neural network dimensions $\mathbf{N} = (N_L = 1, N_{L-1}, \dots, N_1, N_0 = d)$ *with* $L \geq 2$,

(2) any $r \geq 3(N_1 + 1) \cdots (N_{L-1} + 1)$,

(3) and any $\epsilon > 0$, $\hat{\epsilon} \in (0, 1/2)$ *and cost function* $\mathcal{R} \in \mathcal{CF}_r^{\epsilon,\hat{\epsilon}}$.

There is an uncountable collection $\mathcal{C}_3$ *of disjoint subsets of* $\Omega^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}$ *so that for each* $\hat{\Omega} \in \mathcal{C}_3$ *the computational problem*

$$\{\Xi^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}, \hat{\Omega}, \mathcal{M}^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}, \Lambda^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}\}^{\Delta_1}$$

has breakdown epsilon $\epsilon^{\mathrm{s}}_{\mathbb{PB}}(\mathrm{p}) \geq 1/4 - \hat{\epsilon}/2$, *for all* $\mathrm{p} \in [0, 1/2)$.

To see that Proposition 5.26 implies Theorem 3.5, assume that Proposition 5.26 holds and suppose that Γ is a randomised algorithm (in either the BSS or Turing models). The existence of ϕ stated in Theorem 3.5 is guaranteed as per the discussion in Remark 5.10. Furthermore, Γ is also a RGA and hence we can consider Γ restricted to $\hat{\Omega}$ for each $\hat{\Omega} \in \mathcal{C}_3$. Since the computational problem $\{\Xi^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}, \hat{\Omega}, \mathcal{M}^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}, \Lambda^{\mathcal{NN}}_{f,r,\epsilon,\mathcal{R},(\mathbf{N},L)}\}^{\Delta_1}$ has $\epsilon^{\mathrm{s}}_{\mathbb{PB}}(\mathrm{p}) \geq 1/4 - \hat{\epsilon}/2 > 1/4 - 3\hat{\epsilon}/4$, for all $\mathrm{p} \in [0, 1/2)$ there must exist a training set $\mathcal{T} = \mathcal{T}(\hat{\Omega})$ with $\mathcal{T} = \{x^1, x^2, \dots, x^r\}$ for which

$$\mathbb{P}\Big(\|\{\Gamma_{\mathcal{T}}(x^j)\}_{j=1}^r - \{\phi(x^j)\}_{j=1}^r\|_* \geq 1/4 - 3\hat{\epsilon}/4 \Big) > \mathrm{p},$$

for any $\phi \in \operatorname{argmin}_{\varphi \in \mathcal{NN}_{\mathbf{N},L}, \epsilon} \mathcal{R}\left(\{\varphi(x^j)\}_{j=1}^r, \{f(x^j)\}_{j=1}^r\right)$ (this is itself a consequence of Remark 5.15).

We now choose $\mathcal{C}_2 = \{\mathcal{T}(\hat{\Omega}) \mid \hat{\Omega} \in \mathcal{C}_3\}$. Because $\mathcal{C}_3$ is an uncountable collection of disjoint sets, $\mathcal{C}_2$ is uncountable and thus Theorem 3.5 follows.

5.6. *Proof of Proposition 5.26 and Theorem 3.5*

As demonstrated in the previous section, to prove Theorem 3.5 it suffices to prove Proposition 5.26. We begin by starting the following useful lemma:

Lemma 5.27. *Recall the set-up of Proposition 5.26 and the vectors* $x^{k,\delta}$ *defined in (5.1). For any* $\delta \in (0, \varepsilon'(r))$ *and arbitrary*

$$\phi \in \operatorname{argmin}_{\varphi \in \mathcal{NN}_{\mathbf{N},L}, \epsilon} \mathcal{R}\left(\{\varphi(x^{j,\delta})\}_{j=1}^r, \{f_a(x^{j,\delta})\}_{j=1}^r\right) \tag{5.24}$$

we have $|\phi(x^{k,\delta}) - f_a(x^{k,\delta})| \leq \hat{\epsilon}$ *for all* $k \in \{1, \dots, r\}$.

Proof. By Lemma 5.3, there exists a neural network $\tilde{\varphi} \in \mathcal{NN}_{\mathbf{N},L}$ with $\tilde{\varphi}(x^{k,\delta}) = f_a(x^{k,\delta})$ for all k. In particular, $\mathcal{R}\left(\{\tilde{\varphi}(x^{j,\delta})\}_{j=1}^r, \{f(x^{j,\delta})\}_{j=1}^r\right) = 0$. Thus, by (5.24) and the definition of the approximate argmin as in (3.3), we must have that

$$\mathcal{R}\left(\{\phi(x^{j,\delta})\}_{j=1}^r, \{f(x^{j,\delta})\}_{j=1}^r\right) \leq \epsilon$$

and the conclusion of the claim follows because $\mathcal{R} \in \mathcal{CF}_r^{\epsilon,\hat{\epsilon}}$ as defined in (3.4).

□

Now that we have proven Lemma 5.27, we are ready to prove Proposition 5.26.

Proof of Proposition 5.26. As in the proof of Theorem 2.2, we begin by defining the sets $\mathcal{C}_1$ and $\mathcal{C}_3$. Let $\mathcal{C}_1 = \{f_a : \mathbb{R}^d \to [0, 1] \mid a \in [1/2, 1]\}$, where f_a is defined as in (5.2). Fix $a \in [1/2, 1]$ and $\kappa \in [1/4, 3/4]$, define $\mathcal{T}_\delta^\kappa := \{x^{1,\delta}, x^{2,\delta}, \dots, x^{r,\delta}\}$ where the values $x^{i,\delta}$ (each depending on κ and a) are defined in (5.1). We define $\hat{\Omega}^\kappa := \{\mathcal{T}_\delta^\kappa \mid \delta \in [0, \epsilon'(r))$. By Lemma 5.2, we have $\mathcal{T}_\delta^\kappa \in \mathcal{S}^{f_a}_{\varepsilon'(r)}$ so that $\hat{\Omega}^\kappa \subset \Omega^{\mathcal{NN}}$. Note also that noting that the $\hat{\Omega}^\kappa$ are disjoint as an immediate consequence of (5.1). Finally, we set $\mathcal{C}_3 := \{\hat{\Omega}^\kappa \mid \kappa \in [1/4, 3/4]\}$, .

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

Now we have defined $\mathcal{C}_1$ and $\mathcal{C}_3$, we will show that for any $\kappa \in [1/4, 3/4]$ the computational problem

$$\{\Xi^{\mathcal{NN}}, \hat{\Omega}^{\kappa}, \mathcal{M}^{\mathcal{NN}}, \Lambda^{\mathcal{NN}}\}^{\Delta_1}$$

has breakdown epsilon $\epsilon^{\mathrm{s}}_{\mathbb{P}_{h\mathrm{B}}}(\mathrm{p}) \geq 1/4 - \hat{\epsilon}/2$, for all $\mathrm{p} \in [0, 1/2)$. This will be done using Proposition 5.25. We will define $\iota^0 := \mathcal{T}_0^{\kappa}$ and $\iota_n^1 := \mathcal{T}_{4^{-n}}^{\kappa}$.

By (5.1), we see that $\|x^{j,4^{-n}} - x^{j,0}\|_{\infty} \leq 4^{-n}$ for $j = 1, 2, \ldots, r$. Hence (recalling the definition of $\Lambda^{\mathcal{NN}}$), property (Pa) from Proposition 5.25 holds.

Fix $n \in \mathbb{N}$ sufficiently large and let ϕ_0 and ϕ_n be arbitrary neural networks so that

$$\begin{aligned}\phi_0 &\in \operatorname*{argmin}_{\epsilon} \left(\{\varphi(x^{j,0})\}_{j=1}^{r}, \{f_a(x^{j,0})\}_{j=1}^{r}\right)\\ \phi_n &\in \operatorname*{argmin}_{\epsilon} \left(\{\varphi(x^{j,4^{-n}})\}_{j=1}^{r}, \{f_a(x^{j,4^{-n}})\}_{j=1}^{r}\right).\end{aligned} \tag{5.25}$$

By Lemma 5.4 and the assumption that $|\mathcal{T}_0^{\kappa}| = r \geq 3(N_1 + 1) \cdots (N_{L-1} + 1)$, we conclude that

$$\max_{j=1,2,\ldots,r} |\phi_0(x^{j,0}) - f_a(x^{j,0})| \geq 1/2.$$

By contrast, Lemma 5.27 shows that $\max_{j=1,2,\ldots,r} |\phi_n(x^{j,4^{-n}}) - f_a(x^{j,4^{-n}})| \leq \hat{\epsilon}$. Combining these two results and the fact that $f_a(x^{j,0}) = f_a(x^{j,4^{-n}})$ for each $j = 1, 2, \ldots, r$ yields

$$\max_{j=1,2,\ldots,r} |\phi_0(x^{j,0}) - \phi_n(x^{j,4^{-n}})| \geq 1/2 - \hat{\epsilon}.$$

Therefore, since both the ℓ^1 and ℓ^2 norms are bounded from below by the ℓ^{∞} norm and ϕ_0 and ϕ_n were chosen arbitrarily according to (5.25), we have $\inf_{\upsilon^1 \in \Xi(\iota_n^1), \upsilon^2 \in \Xi(\iota^0)} d_{\mathcal{M}}(\upsilon^1, \upsilon^2) \geq 1/2 - \hat{\epsilon}$ where $d_{\mathcal{M}}$ is the ℓ^* norm with $* = 1, 2$ or ∞. Hence, property (Pb) from Proposition 5.25 holds with $\kappa = 1/2 - \hat{\epsilon}$, thereby concluding the proof. □

Financial support. ACH acknowledges support from the Simons Foundation Award No. 663281 granted to the Institute of Mathematics of the Polish Academy of Sciences for the years 2021–2023, from a Royal Society University Research Fellowship, and from the Leverhulme Prize 2017.

Competing interest. The autors declare no competing interests.

References

[1] Adcock, B. & Dexter, N. (2021) The gap between theory and practice in function approximation with deep neural networks. *SIAM J. Math. Data Sci.* **3**(2), 624–655.
[2] Adcock, B. & Hansen, A. C. (2021) *Compressive Imaging: Structure, Sampling, Learning*, Cambridge University Press.
[3] Akhtar, N. & Mian, A. (2018) Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* **6**, 14410–14430.
[4] Antun, V., Renna, F., Poon, C., Adcock, B. & Hansen, A. C. (2020) On instabilities of deep learning in image reconstruction and the potential costs of AI. *Proc. Natl. Acad. Sci. USA* **117**(48), 30088–30095.
[5] Arora, S. & Barak, B. (2009) *Computational Complexity – A Modern Approach*, Princeton University Press.
[6] Bastounis, A., Campodonico, P., van der Schaar, M., Adcock, B. & Hansen, A. C. (2024) On the consistent reasoning paradox of intelligence and optimal trust in AI: The power of 'i don't know. *CoRR*. arXiv: 2408.02357.
[7] Bastounis, A., Cucker, F. & Hansen, A. C. (2023) When can you trust feature selection? – i: A condition-based analysis of lasso and generalised hardness of approximation. arXiv: 2312.11425.
[8] Bastounis, A., Gorban, A. N., Hansen, A. C., et al. (2023) The boundaries of verifiable accuracy, robustness, and generalisation in deep learning. In: Iliadis, L., Papaleonidas, A., Angelov, P. & Jayne, C. (eds.), *Artificial Neural Networks and Machine Learning – ICANN*, Springer Nature, Cham, pp. 530–541.
[9] Bastounis, A., Hansen, A. C. & Vlačić, V. (2021) The extended smale's 9th problem – on computational barriers and paradoxes in estimation, regularisation, computer-assisted proofs and learning. arXiv: 2110.15734.
[10] Beerens, L. & Higham, D. J. (2023) Adversarial ink: Componentwise backward error attacks on deep learning. *IMA J. Appl. Math.* **89**(1), 175–196.
[11] Ben-Artzi, J., Colbrook, M. J., Hansen, A. C., Nevanlinna, O. & Seidel, M. (2020) Computing spectra – on the solvability complexity index hierarchy and towers of algorithms. arXiv: 1508.03280.
[12] Ben-Artzi, J., Hansen, A. C., Nevanlinna, O. & Seidel, M. (2015) New barriers in complexity theory: On the solvability complexity index and the towers of algorithms. *C. R. Math.* **353**(10), 931–936.
[13] Ben-Artzi, J., Marletta, M. & Rösler, F. (2022) Computing the sound of the sea in a seashell. *Found. Comput. Math.* **22**, 697–731.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

[14] Ben-Tal, A., Ghaoui, L. El & Nemirovski, A. (2009) *Robust Optimization, Princeton Series in Applied Mathematics*, Princeton University Press.
[15] Ben-Tal, A. & Nemirovski, A. (2000) Lectures on modern convex optimization: Analysis, algorithms, and engineering applications. https://www2.isye.gatech.edu/.
[16] Ben-Tal, A. & Nemirovski, A. (2000) Robust solutions of linear programming problems contaminated with uncertain data. *Math. Program.* **88**(3), 411–424.
[17] Bishop, E. (1967) *Foundations of Constructive Analysis, McGraw-Hill Series in higher mathematics*. McGraw-Hill.
[18] Blum, L., Shub, M. & Smale, S. (1989) On a theory of computation and complexity over the real numbers: *NP*- completeness, recursive functions and universal machines. *Bull. Am. Math. Soc.* **21**(1), 1–46.
[19] Bungert, L., Trillos, N. García & Murray, R. (2023) The geometry of adversarial training in binary classification. *Inform. Inference: J. IMA* **12**(2), 921–968.
[20] Carlini, N. & Wagner, D. (2018) Audio adversarial examples: Targeted attacks on speech-to-text. In: *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, pp. 1–7.
[21] Choi, C. (2021) 7 revealing ways AIs fail: Neural networks can be disastrously brittle, forgetful, and surprisingly bad at math. *IEEE Spectrum.* 21st of September.
[22] Choi, C. (2022) Some AI systems may be impossible to compute. *IEEE Spectrum*. 30th of March.
[23] Colbrook, M. (2022) On the computation of geometric features of spectra of linear operators on hilbert spaces. *Found. Comput. Math.* **24**(3), 723–804.
[24] Colbrook, M. J. (2021) Computing spectral measures and spectral types. *Commun. Math. Phys.* **384**(1), 433–501.
[25] Colbrook, M. J., Antun, V. & Hansen, A. C. (2022) The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and smale's 18th problem. *Proc. Natl. Acad. Sci. USA* **119**(12), e2107151119.
[26] Colbrook, M. J. & Hansen, A. C. (2022) The foundations of spectral computations via the solvability complexity index hierarchy. *J. Eur. Math. Soc.* **25**(12), 4639–4718.
[27] E. Commission, (2021) Europe fit for the digital age: https://digital-strategy.ec.europa.eu/en/news/europe-fit-digital-age-commission-proposes-new-rules-and-actions-excellence-and-trust-artificial. *Press Release*.
[28] Cucker, F. & Smale, S. (1999) Complexity estimates depending on condition and round-off error. *J. ACM* **46**(1), 113–184.
[29] DeVore, R., Hanin, B. & Petrova, G. (2021) Neural network approximation. *Acta Numer.* **30**, 327–444.
[30] Doyle, P. & McMullen, C. (1989) Solving the quintic by iteration. *Acta Math.* **163**(3-4), 151–180.
[31] Fawzi, A., Fawzi, H. & Fawzi, O. (2018) Adversarial vulnerability for any classifier. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Red Hook, NY, USA*. Curran Associates Inc, pp. 1186–1195.
[32] Fawzi, A., Dezfooli, S. M. Moosavi & Frossard, P. (2017) The robustness of deep networks - A geometric perspective. *IEEE Signal Proc. Mag.* **34**(6), 1350–62.
[33] Fefferman, C., Hansen, A. C. & Jitomirskaya, S. (2022) Computational mathematics in computer assisted proofs. American Institute of Mathematics Workshops. American Institute of Mathematics. https://aimath.org/pastworkshops/compproofsvrep.pdf.
[34] Fefferman, C. & Klartag, B. (2009) Fitting a C^m-smooth function to data II. *Rev. Mat. Iberoam.* **25**(1), 49–273.
[35] Fefferman, C. L. & Klartag, B. (2009) Fitting a C^m-smooth function to data. *I. Ann. Math.* **169**(1), 315–346.
[36] Finlayson, S. G., Bowers, J. D., Ito, J., Zittrain, J. L., Beam, A. L. & Kohane, I. S. (2019) Adversarial attacks on medical machine learning. *Science* **363**(6433), 1287–1289.
[37] Gazdag, L. E. & Hansen, A. C. (2022) Generalised hardness of approximation and the SCI hierarchy – On determining the boundaries of training algorithms in AI. arXiv: 2209.06715.
[38] Goodfellow, I., Bengio, Y. & Courville, A. (2016) *Deep Learning*, MIT Press. http://www.deeplearningbook.org.
[39] Goodfellow, I., Shlens, J. & Szegedy, C. (2015) Explaining and harnessing adversarial examples. In: *International Conference on Learning Representations*.
[40] Gottschling, N. M., Antun, V., Hansen, A. C. & Adcock, B. (2025) The troublesome kernel: On hallucinations, no free lunches, and the accuracy-stability tradeoff in inverse problems. *SIAM Rev.* **67**(1), 73–104.
[41] Gottschling, N. M., Campodonico, P., Antun, V. & Hansen, A. C. (2023) On the existence of optimal multi-valued decoders and their accuracy bounds for undersampled inverse problems. arXiv: 2311.16898.
[42] Hamon, R., Junklewitz, H. & Sanchez, I. (2020) Robustness and explainability of artificial intelligence - From technical to policy solutions. *Publ. Office European Union*.
[43] Hansen, A. C. (2011) On the solvability complexity index, the n-pseudospectrum and approximations of spectra of operators. *J. Amer. Math. Soc.* **24**(1), 81–124.
[44] Hansen, A. C. & Nevanlinna, O. (2016) Complexity issues in computing spectra, pseudospectra and resolvents. *Banach Cent.* **112**, 171–194.
[45] Hansen, A. C. & Roman, B. (2021) *Structure and Optimisation in Computational Harmonic Analysis: On Key Aspects in Sparse Regularisation*, Springer International Publishing, Cham, pp. 125–172.
[46] He, K., Zhang, X., Ren, S. & Sun, J. (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.
[47] Heaven, D. (2019) Why deep-learning AIs are so easy to fool. *Nature* **574**(7777), 163–166.
[48] Higham, C. F. & Higham, D. J. (2019) Deep learning: An introduction for applied mathematicians. *SIAM Rev.* **61**, 860–891.
[49] Huang, Y., et al. (2018) Some investigations on robustness of deep learning in limited angle tomography. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 145–153.

https://doi.org/10.1017/S0956792525100193 Published online by Cambridge University Press

[50] Ilyas, A. & and, etal (2019) Adversarial examples are not bugs, they are features. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS, December 8–14, Vancouver, BC, Canada*, pp. 125–136.
[51] Ko, K. (1991) *Complexity theory of real functions*. Birkhäuser.
[52] LeCun, Y., Bengio, Y. & Hinton, G. (2015) Deep learning. *Nature* **521**(7553), 436–444.
[53] Liu, Z. N. D. & Hansen, A. C. (2024) Do stable neural networks exist for classification problems? – A new view on stability in ai. arXiv: 2401.07874.
[54] Lovasz, L. (1987) *An Algorithmic Theory of Numbers, Graphs and Convexity*. In: *CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics*.
[55] Maas, A. L., Hannun, A. Y., A., Y. & Ng, etal (2013) Rectifier nonlinearities improve neural network acoustic models. In: *Proc. icml*, Vol. **30**. Citeseer, p. 3.
[56] Madry, A., Makelov, A., Schmidt, L., Tsipras, D. & Vladu, A. (2018) Towards deep learning models resistant to adversarial attacks. In: *International Conference on Learning Representations*.
[57] Matiyasevich, Y. V. (1993) *Hilbert's Tenth Problem*. MIT Press.
[58] McKinney, S. & and, et al (2020) International evaluation of an AI system for breast cancer screening. *Nature* **577**(7788), 89–94.
[59] McMullen, C. (1987) Families of rational maps and iterative root-finding algorithms. *Ann. Math.* **125**(3), 467–493.
[60] McMullen, C. (1988) Braiding of the attractor and the failure of iterative algorithms. *Invent. Math.* **91**(2), 259–272.
[61] Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O. & Frossard, P. (2017) Universal adversarial perturbations. In: *IEEE Conf. On Computer Vision and Pattern Recognition*, pp. 86–94.
[62] Moosavi-Dezfooli, S., Fawzi, A. & Frossard, P. (2016) Deepfool: A simple and accurate method to fool deep neural networks. In: *CVPR*. IEEE Computer Society, pp. 2574–2582.
[63] Niyogi, P., Smale, S. & Weinberger, S. (2011) A topological view of unsupervised learning from noisy data. *SIAM J. Comput.* **40**(3), 646–663.
[64] Owhadi, H., Scovel, C. & Sullivan, T. (2015) Brittleness of Bayesian inference under finite information in a continuous world. *Electron. J. Stat.* **9**(1), 1–79.
[65] Owhadi, H., Scovel, C. & Sullivan, T. J. (2015) On the brittleness of Bayesian inference. *SIAM Rev.* **57**(4), 566–582.
[66] Papyan, V., Han, X. Y. & Donoho, D. L. (2020) Prevalence of neural collapse during the terminal phase of deep learning training. *Proc. Natl. Acad. Sci.* **117**(40), 24652–24663.
[67] Pinkus, A. (1999) Approximation theory of the MLP model in neural networks. *Acta Numer.* **8**, 143–195.
[68] Poonen, B. (2014) *Undecidable Problems: A Sampler. Interpreting Gödel: Critical Essays*, Cambridge University Press, pp. 211–241.
[69] Shafahi, A., Huang, W., Studer, C., Feizi, S. & Goldstein, T. (2019) Are adversarial examples inevitable? In: *International Conference on Learning Representations (ICLR)*.
[70] Shalev-Shwartz, S. & Ben-David, S. (2014) *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, USA.
[71] Smale, S. (1998) Mathematical problems for the next century. *Math. Intell.* **20**, 7–15.
[72] Smith, P. (2013) *An Introduction to Gödel's Theorems. Cambridge Introductions to Philosophy*. 2nd edn. Cambridge University Press.
[73] Sutton, O. J., Zhou, Q., Tyukin, I. Y., Gorban, A. N., Bastounis, A. & Higham, D. J. (2023) How adversarial attacks can disrupt seemingly stable accurate classifiers. arXiv preprint arXiv: 2309.03665.
[74] Szegedy, C., Zaremba, W., Sutskever, I., et al. (2014) Intriguing properties of neural networks. In: *Int. Conf. on Learning Representations*
[75] Turing, A. M. (1936) On computable numbers, with an application to the entscheidungsproblem. *Proc. Lond. Math. Soc.* **S2-42**(1), 230.
[76] Turing, A. M. (1950) I.-Computing machinery and intelligence. *Mind* **LIX**(236), 433–460.
[77] Tyukin, I., Higham, D. & Gorban, A. (2020) On adversarial examples and stealth attacks in artificial intelligence systems. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–6.
[78] Tyukin, I. Y., Higham, D. J., Bastounis, A., Woldegeorgis, E. & Gorban, A. N. (2023) The feasibility and inevitability of stealth attacks. *IMA J. Appl. Math.* **89**(1), 44–84.
[79] Wang, S., Si, N., Blanchet, J. & Zhou, Z. (2023) On the foundation of distributionally robust reinforcement learning. arXiv: 2311.09018.
[80] Weinberger, S. (2004) *Computers, Rigidity, and Moduli: The Large-Scale Fractal Geometry of Riemannian Moduli Space*, Princeton University Press, USA.
[81] Wind, J. S., Antun, V. & Hansen, A. C. (2023) Implicit regularization in ai meets generalized hardness of approximation in optimization – sharp results for diagonal linear networks. arXiv: 2307.07410.
[82] Zakrevskaya, N. S. & Kovalevskii, A. P. (2001) One-parameter probabilistic models of text statistics. *Sib. Zh. Ind. Mat.* **4**, 142–153.

Cite this article: Bastounis A., Hansen A. and Vlačić V. The mathematics of adversarial attacks in AI – why deep learning is unstable despite the existence of stable neural networks. *European Journal of Applied Mathematics*, https://doi.org/10.1017/S0956792525100193