Complexity-Theoretic Limits on the Promises of Artificial Neural Network Reverse-Engineering

Federico Adolfi (fede.adolfi@bristol.ac.uk)
University of Bristol, UK & Ernst Strüngmann Institute for Neuroscience, Germany

Martina G. Vilas (martina.vilas@esi-frankfurt.de)
Department of Computer Science, Goethe University Frankfurt & Ernst Strüngmann Institute for Neuroscience, Germany

Todd Wareham (harold@mun.ca)
Department of Computer Science, Memorial University of Newfoundland, Canada

Abstract

Emerging folklore in the cognitive sciences suggests that interpretability techniques to reverse-engineer artificial neural networks (ANNs) could speed up discovery and theory-building. For many researchers in psychology, linguistics, neuroscience, and artificial intelligence (AI), the full observability and perturbability of ANNs trained on complex tasks affords a shortcut to domain insights, cognitive theories, neurocognitive models, application improvement, and user safety. Folklore intuitions, however, are typically disconnected from other relevant knowledge. Here we examine these intuitions formally by drawing relevant connections to computational complexity theory. We model interpretability queries computationally and analyze their resource demands for biological/artificial high-level cognition. We prove mathematically that, contrary to folklore, basic circuit-finding queries in classic ANNs are already infeasibly demanding to answer even approximately. We discuss how interdisciplinary integration can mitigate this disconnect and situate the broader implications for the cognitive sciences, the philosophy of AI-fueled discovery, and AI ethics.

Keywords: meta-theory; high-level cognition; reverse-engineering; interpretability; circuit finding; computational modeling; artificial neural networks; theoretical computer science; computational complexity; artificial intelligence; neuroscience; psychology; philosophy.

The truth is in there, but so are falsities, and it's hard to tell them apart.

S.A. Dana Scully
(slightly rephrased; 1993)

Introduction

In both basic and applied science, the possibility of reverse-engineering optimized neural networks represents a potential methodological shortcut with invaluable consequences. Folklore across disciplines frames it as a novel way to gain knowledge: since trained artificial neural networks (ANNs) are fully transparent, the insights they contain can be extracted with interpretability techniques that hold potential to scale up to larger systems solving complex real-world tasks.

There is a heightened sense of optimism in the cognitive sciences. Researchers in psychology and linguistics hope to leverage this methodology to discover domain theories of cognition by analyzing the outputs (see Ivanova, 2023) or internals (see Pavlick, 2023) of large models trained on vast quantities of text. Among neuroscientists, there is excitement about the opportunities that the full perturbability and observability of ANNs might open up for reverse-engineering candidate models. These properties would help circumvent longstanding obstacles in producing mechanistic hypotheses for neurocognitive function (see Lindsay & Bau, 2023; Lindsay, 2024) and controlling neural activity (e.g., Tuckute et al., 2023). Artifical Intelligence (AI) engineers are exploring whether and how reverse-engineering might help diagnose problems with applications, improve architectures, deal with safety issues, and potentially discover domain insights where learned solutions exceed human capabilities (Raghu & Schmidt, 2020; Räuker, Ho, Casper, & Hadfield-Menell, 2023; Adadi & Berrada, 2018). Some philosophers of science believe this synergy amounts to a novel epistemic perspective (Crook & Kästner, 2023; Boge, 2022). Ostensibly, it provides researchers and engineers with a vantage point from where to pursue basic and applied goals with less friction.

However insightful, these folklore intuitions about the impact of technology on the problems we care about are arguably best construed as conjectures. Commonsense notions of what makes scientific problems easier or harder are often intuitively deceptive (e.g., Adolfi, Wareham, & van Rooij, 2023; Adolfi & van Rooij, 2023; van Rooij, Evans, Müller, Gedge, & Wareham, 2008). This is because they are typically disconnected from other, possibly well-established, knowledge (see Green, 2019). Here we set out to draw relevant connections between folklore on ANN reverse-engineering and formal knowledge in Theoretical Computer Science by assessing the real-world feasibility of these intuitions through the lens of Computational Complexity Theory (see Garey & Johnson, 1979; van Rooij, Blokpoel, Kwisthout, & Wareham, 2019). This paper[1] initiates the study of the resource demands of reverse-engineering the internals of ANNs for natural/artificial high-level cognition.

Overview. In **[§1] Inner Interpretability and its Folklore**, we introduce the subfield tasked with reverse-engineering the internals of trained ANNs and survey intuitions about its promises across the cognitive sciences: (I) the benefits of full observability and perturbability for neurocognitive modeling, (II) the potential for extracting domain insights from trained networks, and (III) the feasibility of scaling up reverse-engineering to large networks solving complex tasks. We contribute a **[§2] Conceptualization and Formalization** of basic interpretability queries as computational problems: given a trained ANN, find a subcircuit responsible for cer-

[1] See also Appendix: osf.io/fs4ym/?view_only=dbe3ec16033b43cfb8715ba967eaf6c5

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

tain behaviors. Then we explain how our **[§3] Computational Complexity Analyses** pin down folklore intuitions to assess them against the intrinsic difficulty of these interpretability queries. We present mathematical proofs of **[§4] Complexity-Theoretic Results**: contrary to the prevailing folklore, basic circuit finding queries in classic, fully observable and perturbable neural networks are already infeasibly demanding to solve not only optimally but even approximately. Finally, we offer a **[§5] Discussion** where we situate our findings in current debates on the impact of AI interpretability for the problems of interest to psychologists, linguists, neuroscientists, and computer scientists, and draw out their broader implications for the philosophy of AI-fueled discovery, and AI engineering, safety and ethics.

§1 Inner Interpretability and its Folklore

AI is increasingly concerned with understanding how the internals of a learned system support particular aspects of machine behavior (e.g., Meng, Bau, Andonian, & Belinkov, 2022; Geva, Caciularu, Wang, & Goldberg, 2022; Vilas, Schaumlöffel, & Roig, 2023). *Inner interpretability* is an emerging subfield tasked with reverse-engineering trained ANNs (Räuker et al., 2023; Gilpin et al., 2019) and enabling "a kind of anatomy of neural networks" (Voss, Goh, et al., 2021, see Figure 1). In this section we survey folklore about its promises that is shared across the cognitive sciences. We will later investigate whether these commonsense intuitions align with formal analyses.

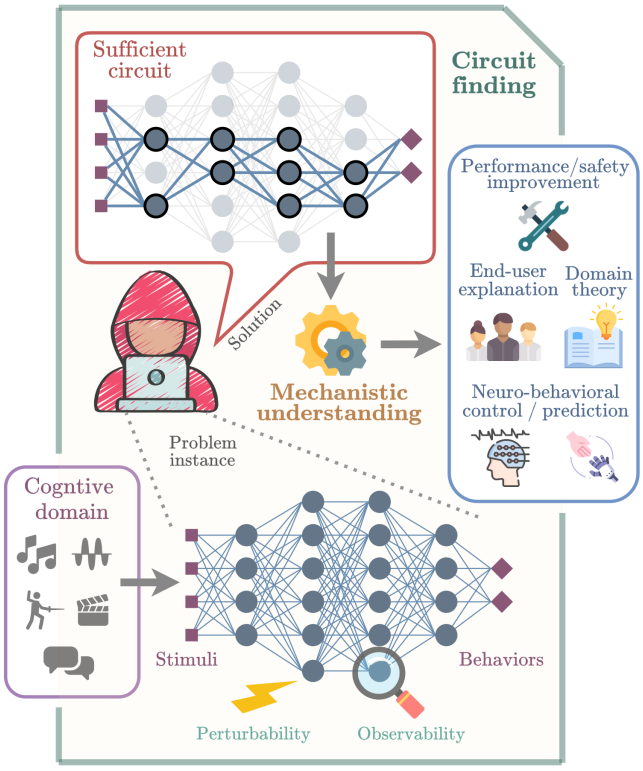

Figure 1: Schematic of the problem of reverse-engineering neural networks (bottom) trained in a domain of interest (left), the interpretability task of *circuit finding* (top), and its intended application in various disciplines (right).

I. Exploiting full observability and perturbability. The see-through nature of ANNs, in contrast to most complex systems of scientific interest, intuitively suggests that reverse-engineering goals should be less challenging to attain. "[E]xperiments can be carried out on ANNs easier", more "thoroughly and quickly [...], than on real brains [...] because ANNs are fully observable and perturbable" (Lindsay & Bau, 2023). "Proving causality [...] is much easier [...] due to the ease with which [they] can be perturbed and lesioned" (Lindsay, 2024). "[N]euroscientists might give a great deal to have the access to weights that those of us studying [ANNs] get for free" (Voss, Cammarata, et al., 2021). "[T]he field of interpretable AI [has] identified several methods that can find the neural features responsible for network outputs" (Lindsay, 2024). These methods are thought to present computational and cognitive (neuro)scientists with transformative means to construct candidate neurocognitive models and theories (e.g., Pavlick, 2023).

II. Extracting latent insights. The synergy of ANN optimization and post-hoc inner interpretability is thought to "offer the potential to discover emergent computations and mechanisms not directly built into them" (Lindsay, 2024), and to extract "domain insights by thoroughly interpreting high-performing AI systems" (Räuker et al., 2023). Some cognitive scientists believe that ANNs with human-like performance (cf. van Rooij et al., 2023) hold a latent "theory [that is] certainly in there" (e.g., Piantadosi, 2023). In a critique of this position, Kodner, Payne, and Heinz (2023) ask "[b]ut what does it mean for a theory to be hidden in a black box?" On one account, an explanation is a cognitive artifact that is 'efficiently queriable'; for instance, models are often referred to as "a way of making an explanation tractable" (Cao & Yamins, 2023). This means that extracting these supposed insights from ANNs via inner interpretability, if at all realistic, should be feasible with real-world resources.

III. Scaling up interpretability. ANN applications (e.g., natural language processing) have "consistently progressed by consuming more and more data [...] and a steady increase in model size" (Kodner et al., 2023). Performance-wise, "[l]arge hidden representation size [is often] consistently better" and especially important for domain-general models (Rogers, Kovaleva, & Rumshisky, 2020). For the cognitive sciences, the "benefit of the ANN approach is that it can face [i.e., scale up to] the complexity of real-world stimuli and behavior" (Lindsay & Bau, 2023). Accordingly, "[i]nterpretability techniques should scale to large models" (Räuker et al., 2023), "which often contain hundreds of layers and billions or trillions of parameters" (Olsson et al., 2022). "This trend raises concerns about computational complexity" (Rogers et al., 2020) and environmental impact (Luccioni, Jernite, & Strubell, 2023). There is a preoccupation with the obstacle of "dimensionality and scale" (Voss, Cammarata, et al., 2021) in the presence of exponential search spaces. "Even when all parameter values are available [...], it is not straight-

forward to map these to model behavior, and this problem is only exacerbated as model size increases" (Kodner et al., 2023). Yet, "[s]mall networks and simple tasks [...] are often used for testing" (Räuker et al., 2023). This mix of optimism and preoccupation with scale has not been matched by efforts to understand complexity-theoretic properties of interpretability problems (i.e., whether and how scaling up might be feasible). We address this gap in the following sections.

§2 Conceptualization and Formalization

By modeling reverse-engineering tasks at the computational level (see Marr & Poggio, 1976), we ensure that our results yield properties intrinsic to the research queries (see van Rooij et al., 2019) and hence generalize across all possible (and relevant) interpretability methods and all their possible implementations involving humans and/or machines.

§2.1 Neural network architecture

We begin by defining the architecture that the interpretability queries studied here are relative to, and the notion of (sub)circuit therein.

Definition 1 (Multi-layer perceptron)**.** [Adapted from Barceló, Monet, Pérez, & Subercaseaux, 2020]. A *multi-layer perceptron* (MLP) is a neural network model $\mathcal{M}$, with $\hat{L}$ layers, defined by sequences of weight matrices $(\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_{\hat{L}})$, $\mathbf{W}_i \in \mathbb{Q}^{d_{i-1} \times d_i}$, bias vectors $(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{\hat{L}})$, $\mathbf{b}_i \in \mathbb{Q}^{d_i}$, and (element-wise) ReLU functions

$$(f_1, f_2, \ldots, f_{\hat{L}-1}), \quad \texttt{relu}(x) := \max(0, x).$$

The final function $f_{\hat{L}}$ is, w.l.o.g., the binary step function $\texttt{step}(x) := 1$ if $x \geq 0$, otherwise 0. The computation rules for $\mathcal{M}$ are given by

$$\mathbf{h}_i := f_i(\mathbf{h}_{i-1}\mathbf{W} + \mathbf{b}_i), \quad \mathbf{h}_0 := \mathbf{x}$$

where $\mathbf{x}$ is the input. The output of $\mathcal{M}$ on $\mathbf{x}$ is defined as $\mathcal{M}(\mathbf{x}) := \mathbf{h}_{\hat{L}}$. The graph $G_{\mathcal{M}} = (V, E)$ of $\mathcal{M}$ has a vertex for each component of each $\mathbf{h}_i$. All vertices in layer i are connected by edges to all vertices of layer $i+1$, with no intra-layer connections. Edges carry weights according to $\mathbf{W}_i$, and vertices carry the components of $\mathbf{b}_i$ as biases. The size of $\mathcal{M}$ is defined as $|\mathcal{M}| := |V|$. (See Figure 1, bottom).

MLPs can be found in most major ANN architectures currently in use (e.g., transformers, albeit activation functions may vary; see Strobl, 2023) for all input modalities.

Definition 2 (Circuit)**.** A *circuit* $\mathcal{C}$ of a multi-layer perceptron $\mathcal{M}$ is defined by a subset of vertices $V' \subseteq V \in G_{\mathcal{M}}$. The circuit includes all connections in $G_{\mathcal{M}}$ for all $v \in V'$, and its size is defined as $|\mathcal{C}| := |V'|$. Figure 1 (top) shows a schematic.

§2.2 Interpretability queries

We formalize inner interpretability tasks as computational problems. We focus on and formalize *circuit finding* queries: the problem facing a researcher searching for a *sufficient circuit* responsible for a type of model behaviour in response to inputs in a domain of interest (see Figure 1). "We may be trying to comprehensively study a model, [or we might try] to study neurons we've determined related to some narrower aspect of model behavior" (Voss, Cammarata, et al., 2021).

Circuit finding. Mechanistic investigative strategies typically involve decomposition and localization (see Ross, 2021) and follow these initial steps: "(i) describing a behavior whose neural circuit mechanisms we seek to understand, (ii) identifying which neurons are involved..." (Olsen & Wilson, 2008). Furthermore, "[a] more compact description of [trained ANNs] is a goal for both machine learning and neuroscience" (Lindsay, 2021). "[N]etworks can often be pruned [heavily] with little to no loss in performance [...], to increase interpretability" (Räuker et al., 2023; see Voita, Talbot, Moiseev, Sennrich, & Titov, 2019). This conceptualization (Figure 1) leads to the following formalizations[2].

Problem 1. (*decision version*)
MINIMUMLOCALLYSUFFICIENTCIRCUIT (MLSC)

Input: A multi-layer perceptron $\mathcal{M}$, an input $\mathbf{x}$, and an integer $u \leq |\mathcal{M}|$.

Output: <YES> if there is a circuit $\mathcal{C}$ in $\mathcal{M}$ of size $|\mathcal{C}| \leq u$ that produces the same output on $\mathbf{x}$ as $\mathcal{M}$. Otherwise, <NO>.

Problem 2. (*decision version*)
MINIMUMGLOBALLYSUFFICIENTCIRCUIT (MGSC)

Input: A multi-layer perceptron $\mathcal{M}$ and an integer $u \leq |\mathcal{M}|$.

Output: <YES> if there is a circuit $\mathcal{C}$ in $\mathcal{M}$ of size $|\mathcal{C}| \leq u$ that produces the same output on every possible input as $\mathcal{M}$. Otherwise, <NO>.

§3 Computational Complexity Analyses

Our proofs of hardness and inapproximability build on concepts and techniques from classical (Garey & Johnson, 1979) and parameterized (Downey & Fellows, 2013) computational complexity theory. Our modeling choices ensure that these analyses yield lower-bounds on the complexity of real-world interpretability queries. Here we give definitions that form the basis of the proof techniques deployed later.

Definition 3 (Polynomial-time tractability)**.** An algorithm is said to run in *polynomial-time* if the number of steps it performs is $O(n^c)$, where n is a measure of the input size and c is some constant. A problem Π is said to be *tractable* if it has a *polynomial-time algorithm*. P denotes the class of such problems.

Parameterized complexity. Consider a more fine-grained look at the *sources of complexity* of problems. The following is a relaxation of the notion of tractability, where unreason-

[2] We focus on *decision* (yes/no output) rather than *solution* (subcircuit output) versions of the circuit finding problems because procedures to solve the latter can be used to solve the former, thus allowing any hardness results for the former to propagate to the latter.

able resource demands are allowed as long as they are constrained to a set of problem parameters.

Definition 4 (Fixed-parameter tractability)**.** Let $\mathcal{P}$ be a set of problem parameters. A problem $\mathcal{P}$-Π is *fixed-parameter tractable* relative to $\mathcal{P}$ if there exists an algorithm that computes solutions to instances of $\mathcal{P}$-Π of any size n in time $f(\mathcal{P}) \cdot n^c$, where c is a constant and $f(\cdot)$ some computable function. FPT denotes the class of such problems and includes all problems in P.

In Table 1 we describe the problem parameters we study later.

Parameter description	Notation	
	Model (given)	*Circuit (requested)*
Number of layers	$\hat{L}$	$\hat{l}$
Maximum layer width	$\hat{L}_w$	$\hat{l}_w$
Total number of units	$\hat{U} = \lvert\mathcal{M}\rvert \leq \hat{L} \cdot \hat{L}_w$	$\lvert\mathcal{C}\rvert = \hat{u}$
Number of input units	$\hat{U}_I$	$\hat{u}_I$
Number of output units	$\hat{U}_O$	$\hat{u}_O$
Maximum weight	$\hat{W}$	$\hat{w}$
Maximum bias	$\hat{B}$	$\hat{b}$

Table 1: Problem parameters. Note the colored parameters can artificially bound the input size if bounded. We avoid this in analyses.

Hardness and reductions. Most proof techniques in this work involve reductions between computational problems.

Definition 5 (Reducibility)**.** A problem Π_1 is *polynomial-time reducible* to Π_2 if there exists a polynomial-time algorithm (*reduction*) that transforms instances of Π_1 into instances of Π_2 such that solutions for Π_2 can be transformed in polynomial-time into solutions for Π_1. This implies that if a tractable algorithm for Π_2 exists, it can be used to solve Π_1 tractably. *Fpt-reductions* transform an instance (x,k) of some problem parameterized by k into an instance (x',k') of another problem, with $k' \leq g(k)$, in time $f(k) \cdot p(|x|)$ where p is a polynomial. These reductions analogously transfer fixed-parameter tractability results between problems.

The results in the next sections are conditional on two conjectures with extensive theoretical and empirical support. Intractability statements build on these as follows.

Conjecture 1. P $\neq$ NP.

Definition 6 (Polynomial-time intractability)**.** The class NP contains all problems in P and more. Assuming Conjecture 1, *NP-hard* problems lie outside P. These problems are considered *intractable* because they cannot be solved in polynomial-time (unless Conjecture 1 is false; see Fortnow, 2009).

Conjecture 2. FPT $\neq$ W[1].

Definition 7 (Fixed-parameter intractability)**.** The class W[1] contains all problems in the class FPT and more. Assuming Conjecture 2, *W[1]-hard* parameterized problems lie outside FPT. These problems are considered *fixed-parameter intractable*, relative to a given parameter set, because no fixed-parameter tractable algorithm can exist to solve them (unless Conjecture 2 is false; see Downey & Fellows, 2013).

Computational problems of known complexity. We will construct reductions from the following two intractable problems (see Garey & Johnson, 1979).

Problem 3. CLIQUE (CQ)
Input: An undirected graph $G = (V,E)$ and a positive integer k.
Output: <YES> if G has a *clique* of size at least k; that is, a subset $V' \subseteq V$, $|V'| \geq k$, such that for all pairs $v,v' \in V'$, $(v,v') \in E$. Otherwise, <NO>.

Problem 4. VERTEXCOVER (VC)
Input: An undirected graph $G = (V,E)$ and a positive integer k.
Output: <YES> if G contains a *vertex cover* of size at most k; that is, a subset $V' \subseteq V$, $|V'| \leq k$, such that for all $(u,v) \in E$, at least one of u or v is in V'. Otherwise, <NO>.

§4 Complexity-Theoretic Results

In this section we present our findings on the resource demands of general and parameterized circuit finding queries, for optimal and approximate problem versions. (Here we present proof sketches; see Online Appendix for full proofs).

§4.1 Intractability of optimal solutions

The first result is that the queries studied here are intrinsically infeasible in general, in that they demand more resources (i.e., space or time) than possible in the real world.

Theorem 1. MLSC is NP-hard.

Theorem 2. MGSC is NP-hard.

Each of these theorems can be proven using two different reductions (given in proof sketches A and B). Both these reductions are used to prove different problem properties later.

Proof A (sketch). Given an instance $\langle G = (V,E), k \rangle$ of CLIQUE, we construct an MLP (Definition 1) with $|V| + |E| + 2$ neurons spread across four layers, as shown in Figure 2. Weights and biases are set such that there is a *sufficient circuit* of size $k(k-1)/2 + k + 2$ in the latter if and only if there is a *clique* of size k in the former. ■

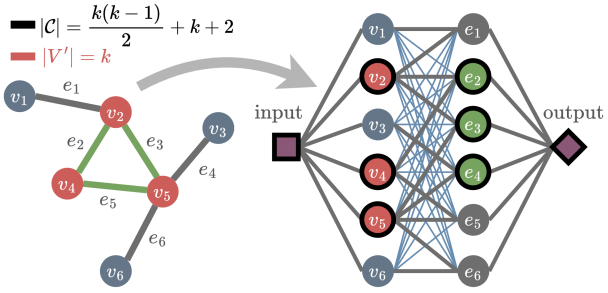

Figure 2: Schematic of the reduction from CLIQUE (left).

Proof B (sketch). Given an instance $\langle G = (V,E), k \rangle$ of VERTEXCOVER, construct an MLP (Definition 1) using the ReLU logic gates described in Barceló et al., 2020 (Lemma 13), with $|V| + 2|E| + 2$ neurons spread across five layers, as shown in Figure 3. Weights and biases are set such that there is a *sufficient circuit* of size $2|E| + k + 2$ in the latter if and only if there is a *vertex cover* of size k in the former. ■

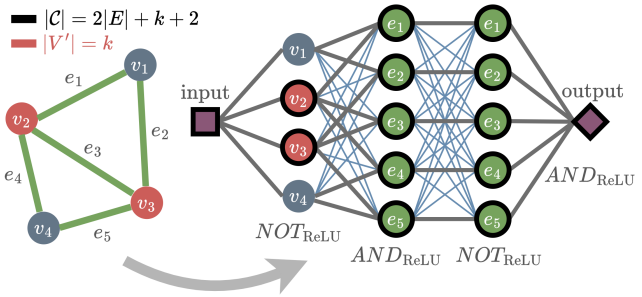

Figure 3: Schematic of the reduction from VERTEXCOVER (left).

§4.2 Parameterized intractability

The preceding theorems establish the intractability of the general problems but leave open the possibility that these unreasonable resource demands might be contained within certain problem parameters (see Table 1). However, the following theorems establish the fixed-parameter intractability of MLSC and MGSC relative to any combination of parameters in the set $\mathcal{P} = \{\hat{L}_w, \hat{U}_I, \hat{U}_O, \hat{W}, \hat{B}, \hat{l}, \hat{l}_w, \hat{u}, \hat{u}_I, \hat{u}_O, \hat{w}, \hat{b}\}$.

Theorem 3. $\mathcal{P}$-MLSC is W[1]-hard.

Theorem 4. $\mathcal{P}$-MGSC is W[1]-hard.

Proof (sketch). In the instances of MLSC and MGSC constructed in proofs sketch A of Theorem 1 and Theorem 2 (Figure 2), all $p \in \mathcal{P}$ are constants or functions of k in the given instance of CLIQUE. The result then follows from the fact that $\{k\}$-CLIQUE is W[1]-hard (Downey & Fellows, 2013). ■

§4.3 Intractability of approximate solutions

Although we proved that computing optimal solutions is intractable, it is still conceivable that we could devise tractable procedures to obtain approximate solutions that are useful in practice. Consider two natural notions of approximation.

Multiplicative approximation. For a minimization problem Π, let $OPT_\Pi(I)$ be an optimal solution for Π, $A_\Pi(I)$ be a solution for Π returned by an algorithm A, and $m(OPT_\Pi(I))$ and $m(A_\Pi(I))$ be the values of these solutions.

Definition 8 (Multiplicative approximation algorithm)**.** [Ausiello et al., 1999, Def. 3.5]. Given a minimization problem Π, an algorithm A is a *multiplicative* ε*-approximation algorithm* for Π if for each instance I of Π, $m(A_\Pi(I)) - m(OPT_\Pi(I)) \leq \varepsilon \times m(OPT_\Pi(I))$.

It would be ideal if one could obtain approximate solutions for a problem Π that are arbitrarily close to optimal if one is willing to allow extra algorithm runtime.

Definition 9 (Multiplicative approximation scheme)**.** [Adapted from Ausiello et al., 1999, Def. 3.10]. Given a minimization problem Π, a *polynomial-time approximation scheme* (PTAS) for Π is a set $\mathcal{A}$ of algorithms such that for each integer $k > 0$, there is a $\frac{1}{k}$-approximation algorithm $A^k_\Pi \in \mathcal{A}$ that runs in time polynomial in $|I|$.

Unfortunately, the following results establish the intractability of arbitrarily-precise multiplicative approximation.

Theorem 5. MLSC cannot have a PTAS.

Theorem 6. MGSC cannot have a PTAS.

Proof (sketch). The result is obtained by showing that the reduction from VERTEXCOVER used in proof sketch B (Figure 3) of Theorem 1 and Theorem 2 is also an *L-reduction*, following Arora, Lund, Motwani, Sudan, & Szegedy, 1998 (Theorem 1.2.2). ■

Additive approximation. It would be useful to have guarantees that an approximation algorithm for our problems returns solutions at most a fixed distance away from optimal.

Definition 10 (Additive approximation algorithm)**.** [Adapted from Ausiello et al., 1999, Def. 3.3]. An algorithm A_Π for a problem Π is a *d-additive approximation algorithm* (d-AAA) if there exists a constant d such that for all instances x of Π the error between the value $m(\cdot)$ of an optimal solution $optsol(x)$ and the output $A_\Pi(x)$ is such that $| m(optsol(x)) - m(A_\Pi(x)) | \leq d$.

This would ensure errors cannot get impractically large. Alas, the following theorems rule out this possibility.

Theorem 7. MLSC cannot have a tractable d-AAA.

Theorem 8. MGSC cannot have a tractable d-AAA.

Proof (sketch). We combine ideas in Proof A (reduction from CLIQUE) of Theorems 1-2 with a padding technique in Garey & Johnson, 1979. Given an MLP constructed as in Figure 2, we build a larger instance by creating $d+1$ part copies and connecting them such that obtaining an approximate solution for the larger instance would imply obtaining an optimal solution for the smaller one, from which a solution to the CLIQUE instance can be extracted (Figure 4). Since the existence of a tractable additive approximation algorithm would contradict the hardness of CLIQUE, no such algorithm is possible. ■

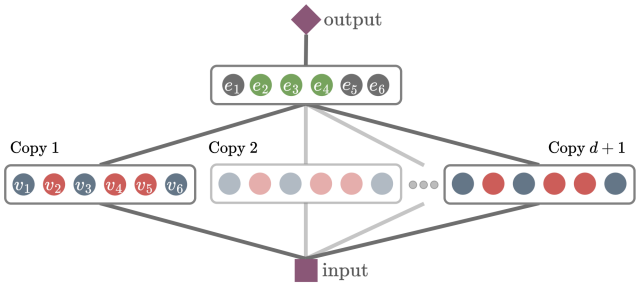

Figure 4: Schematic of the padding strategy to prove Theorems 7-8.

Approximation alternatives. Finally, let us consider three other types of polynomial-time approximability that may be acceptable in situations where always getting the correct output for an input is not required: (1) algorithms that always run in polynomial time and produce the correct output for a given input in all but a small number of cases (Hemaspaandra & Williams, 2012); (2) algorithms that always run in polynomial time and produce the correct output for a given input with high probability (Motwani & Raghavan, 1995); and (3) algorithms that run in polynomial time with high probability but are always correct (Gill, 1977).

Theorem 9. Neither MLSC nor MGSC are tractably approximable in senses (1–3).

Proof (sketch). The result holds conditional on strongly-believed or established complexity-class conjectures due to the proven NP-hardness of MLSC and MGSC (Theorems 1-2) and the reasoning in Wareham, 2022 (Result E). ■

§5 Discussion

Imagine, implausibly, if just about anything of interest could be learned efficiently from data using ANN architectures (cf. van Rooij et al., 2023), and there were no limits on the amount of quality data we could obtain (cf. Birhane, Prabhu, Han, & Boddeti, 2023) and no environmental cost of training (cf. Luccioni & Hernandez-Garcia, 2023). The following recipe is tempting: train ANNs to solve interesting real-world problems, and then exploit their transparency to reverse-engineer their inner workings and extract domain insights. Could we ease and speed up discovery and theory-building in this way?

Folklore in the cognitive sciences suggests the insights must surely be 'in there' (viz. in trained ANNs) and they can be extracted with *inner interpretability* techniques. Yet our findings suggest even separating basic things like relevant circuits from other stuff also 'in there' seems to be infeasible to do reliably for increasingly larger networks. In what sense then can ANN reverse-engineering represent a shortcut? We have some understanding that finding theories 'out here' — bottom-up experimentation on biological brains (Adolfi & van Rooij, 2023) and top-down theorizing (Rich, de Haan, Wareham, & van Rooij, 2021) — face similar barriers. Compare this with the ANN-interpretability promises described in the introductory sections. On the forward-engineering side, we know that scaling up the training of ANNs to perform ever more complex human-like or -level tasks runs up against intractability barriers (van Rooij et al., 2023). On the reverse-engineering side, the hardness and inapproximability theorems proven here provide preliminary evidence that inner interpretability queries, even very basic ones, are no exception.

Contrary to folklore intuitions, full perturbability and full observability do not appear to translate directly into efficient queriability or controllability. That these research activities are rendered possible in ANNs does not mean that the scientific problems get easier or the expected knowledge gains are brought about (see Adolfi, 2023, for a generalizing framework). In this narrow sense, the value of perturbability and observability might be overstated and/or a distraction.

Our findings hence temper notions of ANN reverse-engineering as a shortcut to build neurocognitive models (see e.g., Lindsay, 2024; Lindsay & Bau, 2023) or to extract psychological/linguistic theories (see e.g., Piantadosi, 2023, and Kodner et al., 2023 for critique) and application domain insights (Adadi & Berrada, 2018), or as an overall novel and more efficient epistemic perspective (Crook & Kästner, 2023). We echo calls in other fields to question the idea that there are shortcuts to theory (Devezer, 2023). If our results hold more generally, they would indicate a sobering conclusion: there exist no such shortcuts 'in there'; theoretical insights *in there* — if there are any (there might be no *there* there) — could be just as hard to discover as those *out here*.

The disconnect we observe here between knowledge in adjacent disciplines and folklore intuitions among cognitive scientists is not uncommon. We stress the importance of disciplinary diversity and interdisciplinary integration (Green & Andersen, 2019; Bender, Beller, & Nersessian, 2015) to recover the relevant connections. The field of Inner Interpretability can leverage existing knowledge in older fields (Vilas, Adolfi, Poeppel, & Roig, 2024). The risk of overlooking relevant disciplinary links is that unexamined folklore intuitions about what makes problems easier, or investigative strategies feasible, can end up driving research programs dominated by a technology lottery (see Hooker, 2020).

Interpretability efforts currently rely on poorly understood heuristics to solve poorly understood computational problems (see Krishnan, 2020; Vilas et al., 2024). When the latter are intrinsically intractable (e.g., our results, those in Barceló et al., 2020), the landscape of empirical results can get messy. This is due, in part, to compounding errors which our inapproximability results suggest cannot be kept small. Indeed, "many popular interpretability methods produce estimates [...] that are not better than a random designation..." (Hooker, Erhan, Kindermans, & Kim, 2019). While our work is preliminary regarding the broader scope and limits of ANN interpretability, it suggests that tractable methods must be sought relative to additional knowledge about the structure of the problems and/or further restrictions on the architectures. Future work could exploit knowledge of how constraints (e.g., to the input domain, architecture, parameter space) affect the intrinsic complexity of interpretability queries to design algorithms capable of answering them efficiently.

Turning to AI ethics, post-hoc reverse-engineering methods could be useful to tackle issues of user safety and environmental efficiency (see Räuker et al., 2023; and Krishnan, 2020 for critique). Still, we caution against framing this path as a *shortcut* to making systems safe or efficient. Relying on interpretability techniques to defer these issues until after training would not seem to render them more feasibly solvable. Grappling head-on with issues of training dataset curation (Birhane et al., 2023), energy demands and carbon emissions (Luccioni et al., 2023) are plausible alternatives.

Acknowledgments

FA and MV are supported by the Ernst Strüngmann Foundation. TW is supported by NSERC Discovery grant 228104-2015.

References

Adadi, A., & Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, *6*, 52138–52160.

Adolfi, F. (2023). *Computational Meta-Theory in Cognitive Science: a theoretical computer science framework* (PhD thesis, University of Bristol). Retrieved from https://hdl.handle.net/1983/c3702d1d-143c-40cc-987e-f2160ea74ac3

Adolfi, F., & van Rooij, I. (2023). Resource demands of an implementationist approach to cognition. In *Proceedings of the 21st International Conference on Cognitive Modeling*.

Adolfi, F., Wareham, T., & van Rooij, I. (2023). A computational complexity perspective on segmentation as a cognitive subcomputation. *Topics in Cognitive Science*, *15*(2), 255-273.

Arora, S., Lund, C., Motwani, R., Sudan, M., & Szegedy, M. (1998). Proof verification and the hardness of approximation problems. *Journal of the ACM*, *45*(3), 501–555. doi: 10.1145/278298.278306

Ausiello, G., Marchetti-Spaccamela, A., Crescenzi, P., Gambosi, G., Protasi, M., & Kann, V. (1999). *Complexity and Approximation*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Barceló, P., Monet, M., Pérez, J., & Subercaseaux, B. (2020). Model Interpretability through the lens of Computational Complexity. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 15487–15498). Curran Associates, Inc.

Bender, A., Beller, S., & Nersessian, N. J. (2015). Diversity as Asset. *Topics in Cognitive Science*, *7*(4), 677–688. doi: 10.1111/tops.12161

Birhane, A., Prabhu, V., Han, S., & Boddeti, V. N. (2023). *On Hate Scaling Laws For Data-Swamps.* arXiv. (arXiv:2306.13141)

Boge, F. J. (2022). Two Dimensions of Opacity and the Deep Learning Predicament. *Minds and Machines*, *32*(1), 43–75. doi: 10.1007/s11023-021-09569-4

Cao, R., & Yamins, D. (2023). Explanatory models in neuroscience, Part 2: Functional intelligibility and the contravariance principle. *Cognitive Systems Research*, 101200. doi: 10.1016/j.cogsys.2023.101200

Crook, B., & Kästner, L. (2023). *Don't Fear the Bogeyman: On Why There is No Predic:on-Understanding Trade-Off for Deep Learning in Neuroscience.* PhilSciArchive.

Devezer, B. (2023). *There are no shortcuts to theory* (preprint). MetaArXiv. doi: 10.31222/osf.io/umkan

Downey, R. G., & Fellows, M. R. (2013). *Fundamentals of parameterized complexity*. London: Springer.

Fortnow, L. (2009). The status of the P versus NP problem. *Communications of the ACM*, *52*(9), 78–86.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability*. W.H. Freeman.

Geva, M., Caciularu, A., Wang, K. R., & Goldberg, Y. (2022). *Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space.* arXiv. (arXiv:2203.14680)

Gill, J. (1977). Computational Complexity of Probabilistic Turing Machines. *SIAM Journal on Computing*, *6*(4), 675–695. doi: 10.1137/0206049

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2019). *Explaining Explanations: An Overview of Interpretability of Machine Learning.* arXiv. (arXiv.1806.00069)

Green, S. (2019). Science and common sense: perspectives from philosophy and science education. *Synthese*, *196*(3), 795–818. doi: 10.1007/s11229-016-1276-9

Green, S., & Andersen, H. (2019). Systems science and the art of interdisciplinary integration. *Systems Research and Behavioral Science*, *36*(5), 727–743. doi: 10.1002/sres.2633

Hemaspaandra, L. A., & Williams, R. (2012). SIGACT News Complexity Theory Column 76: an atypical survey of typical-case heuristic algorithms. *ACM SIGACT News*, *43*(4), 70–89. doi: 10.1145/2421119.2421135

Hooker, S. (2020). *The Hardware Lottery.* arXiv. (arXiv:2009.06489)

Hooker, S., Erhan, D., Kindermans, P.-J., & Kim, B. (2019). A Benchmark for Interpretability Methods in Deep Neural Networks. In *Advances in Neural Information Processing Systems* (Vol. 32).

Ivanova, A. A. (2023). *Running cognitive evaluations on large language models: The do's and the don'ts.* arXiv. (arXiv:2312.01276)

Kodner, J., Payne, S., & Heinz, J. (2023). *Why Linguistics Will Thrive in the 21st Century: A Reply to Piantadosi (2023).* LingBuzz.

Krishnan, M. (2020). Against Interpretability: a Critical Examination of the Interpretability Problem in Machine Learning. *Philosophy & Technology*, *33*(3), 487–502. doi: 10.1007/s13347-019-00372-9

Lindsay, G. W. (2021). Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future. *Journal of Cognitive Neuroscience*, *33*(10), 2017–2031. doi: 10.1162/jocn_a_01544

Lindsay, G. W. (2024). Grounding neuroscience in behavioral changes using artificial neural networks. *Current opinion in neurobiology*, *84*, 102816.

Lindsay, G. W., & Bau, D. (2023). Testing methods of neural systems understanding. *Cognitive Systems Research*, *82*, 101156. doi: 10.1016/j.cogsys.2023.101156

Luccioni, A. S., & Hernandez-Garcia, A. (2023). *Counting Carbon: A Survey of Factors Influencing the Emissions of Machine Learning.* arXiv. (arXiv:2302.08476)

Luccioni, A. S., Jernite, Y., & Strubell, E. (2023). *Power Hungry Processing: Watts Driving the Cost of AI Deployment?* arXiv. (arXiv:2311.16863)

Marr, D., & Poggio, T. (1976). From understanding computation to understanding neural circuitry. *A.I. Memo*, *357*(3), 1–22.

Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, *35*, 17359–17372.

Motwani, R., & Raghavan, P. (1995). *Randomized algorithms*. Cambridge; New York: Cambridge University Press.

Olsen, S. R., & Wilson, R. I. (2008). Cracking neural circuits in a tiny brain: New approaches for understanding the neural circuitry of Drosophila. *Trends in Neurosciences*, *31*, 512–520.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., ... Olah, C. (2022). *In-context Learning and Induction Heads.* arXiv. (arXiv:2209.11895)

Pavlick, E. (2023, June). Symbols and grounding in large language models. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *381*(2251), 20220041. doi: 10.1098/rsta.2022.0041

Piantadosi, S. (2023). *Modern language models refute Chomsky's approach to language.* LingBuzz. (LingBuzz Published In:)

Raghu, M., & Schmidt, E. (2020). *A Survey of Deep Learning for Scientific Discovery.* arXiv. (arXiv:2003.11755)

Rich, P., de Haan, R., Wareham, T., & van Rooij, I. (2021). How hard is cognitive science? In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 43, pp. 3034–3040).

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, *8*, 842–866. doi: 10.1162/tacl_a_00349

Ross, L. N. (2021). Causal Concepts in Biology: How Pathways Differ from Mechanisms and Why It Matters. *The British Journal for the Philosophy of Science*, *72*(1), 131–158. (Publisher: The University of Chicago Press) doi: 10.1093/bjps/axy078

Räuker, T., Ho, A., Casper, S., & Hadfield-Menell, D. (2023). *Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks.* arXiv. (arXiv:2207.13243)

Strobl, L. (2023). *Average-Hard Attention Transformers are Constant-Depth Uniform Threshold Circuits.* arXiv. (arXiv:2308.03212)

The X-Files. (1993). *Season 1: Young at heart.* ("The truth is out there, but so are lies" — Dana Scully)

Tuckute, G., Sathe, A., Srikant, S., Taliaferro, M., Wang, M., Schrimpf, M., ... Fedorenko, E. (2023). Driving and suppressing the human language network using large language models. *bioRxiv*. (doi.org/10.1101/2023.04.16.537080)

van Rooij, I., Blokpoel, M., Kwisthout, J., & Wareham, T. (2019). *Cognition and intractability: A guide to classical and parameterized complexity analysis*. Cambridge University Press.

van Rooij, I., Evans, P., Müller, M., Gedge, J., & Wareham, T. (2008). Identifying sources of intractability in cognitive models: An illustration using analogical structure mapping. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 30, pp. 15–20).

van Rooij, I., Guest, O., Adolfi, F., Haan, R. d., Kolokolova, A., & Rich, P. (2023). *Reclaiming AI as a theoretical tool for cognitive science.* PsyArXiv. doi: 10.31234/osf.io/4cbuv

Vilas, M. G., Adolfi, F., Poeppel, D., & Roig, G. (2024). An Inner Interpretability Framework for AI Inspired by Lessons from Cognitive Neuroscience. In *Proceedings of the 41th international conference on machine learning.* PMLR. (Accepted)

Vilas, M. G., Schaumlöffel, T., & Roig, G. (2023). Analyzing vision transformers for image classification in class embedding space. In *Advances in neural information processing systems* (Vol. 36, pp. 40030–40041).

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., & Titov, I. (2019). Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 5797–5808). Florence, Italy: Association for Computational Linguistics. doi: 10.18653/v1/P19-1580

Voss, C., Cammarata, N., Goh, G., Petrov, M., Schubert, L., Egan, B., ... Olah, C. (2021). Visualizing Weights. *Distill*, e00024.007.

Voss, C., Goh, G., Cammarata, N., Petrov, M., Schubert, L., & Olah, C. (2021). Branch Specialization. *Distill*, e00024.008.

Wareham, T. (2022). *Creating Teams of Simple Agents for Specified Tasks: A Computational Complexity Perspective.* arXiv. (arXiv:2205.02061)