Comment on: *Transformers As Approximations of Solomonoff Induction*

Tolga Topal [1*]

[1]Emergent Cognition Research Initiative,
https://emergentcognition.org.

Corresponding author(s). E-mail(s): tolga.topal@protonmail.com;

Abstract

In this short communication, we comment, bring additional reflection and perspective on: *Transformers As Approximations of Solomonoff Induction* [1].

The pursuit goal has two points which are interlinked: 1) foster research in the direction of Algorithmic Information Theory, 2) conditioned on the former, increase explainability and interpretation of neural-based architectures e.g.: Transformers.

We close this correspondence with possible research extension paths.

Keywords: Solomonoff Induction, Algorithmic Probability, Algorithmic Complexity, Algorithmic Information Theory, Explainability, Interpretability

1 Introduction

As artificial intelligence(AI) permeates all layers of our societies, the need for understanding what some of these technologies are doing is growing stronger.

To what extend can we understand neural networks[1], neural-based architectures such as Transformers?
The question is explored in: *Could a neuroscientist understand a microprocessor?* [2]

Using techniques from neuroscience, they observe that *in silico* based general purpose computing device has commonalities with the human brain. Among other things we extract the following points:

- Studying both transistors and neurons individually results in limited insight extraction; which echoes the famous formula from Hebbian learning: "Neurons that fire together wire together" is instead illustrated by the experiments,
- Even though not articulated in the paper in the following way, the distributed nature of the brain is the main discrepancy between an *in silico* and *in vivo* computing device[2],
- Last but not least, the main focus of our argument concerning: determining the relation between outputs and inputs is "very difficult" - developed hereafter.

Some closing remarks, on the design aspect of a computing device. The architecture might hold even more weight than what it has commonly been attributed to. Indeed, universality in computing is a property much desired and can be achieved through various means.
To remain with [2], complexity can rise through unintended/unforeseen/innocuous mechanisms, for instance, we can observe Turing-completeness through the use of specific instructions of compiled code/artifacts e.g.: JOP(Jump-Oriented Programming)[3],[4],[5].

Another example [6], which brings and attempts to tackle the complex subject of consciousness by considering a specific architecture: *Finite State Machines with Feedback: An Architecture Supporting Minimal Machine Consciousness* [6].

At the same time, the pace at which technology is evolving has triggered research initiatives and paths revolving around the concept of XAI.[3]
The main components of eXplainable AI (commonly abbreviated as XAI) are:

- Transparency,
- Interpretability,
- Explainability

We can identify at least three different axis to further and deepen our understanding of these boxes; of which we briefly present:

[1] Also referred to as Artificial Neural Networks.
[2] Using this formula does not presume nor entail a computationalist point of view.
[3] The *Alignment Problem* is acknowledged but out of current scope.

Mechanistic Interpretability (MI)

The development of *Mechanistic Interpretability* (MI) [7] has shed some light into the inner workings of the neural-based architectures. Similarly to how a scanning electron microscope proceeds, we are able to zoom into the inner workings of how these boxes operate; and we need this type of approaches and tools. As formulated by Dr. B. Bent: *MI can be seen as the neuroscience of NNs*[4].

Theory of Computation (ToC)

Among all models of computation, since its inception the *Turing machine*(TM) has been the one with very likely the biggest penetration factor into the research landscape. Even though, its embodiment as physical a computing device(e.g.: {micro}processor) is more close to a finite-state machine[5], TM remains the closest with respect to other models of computation.

Outlined earlier, we saw that from an empirical perspective, that deriving inputs from observed outputs is very difficult endeavor.
Nevertheless, to extend from a theoretical point, it is known that from their *universal approximation property* [8], neural networks map inputs-to-outputs. Finding out the inverse process, more precisely the exact inputs for a given set of weights is a NP-hard problem. This is linked to the Halting Problem(HP) [9] in general, and, to Rice's theorem more finely [10].

Information Theory / Algorithmic Information Theory

In third and our main point of concern is to consider how information flows through the layers, network(s). This can be achieved through *information theory* and to some extent with *algorithmic information theory*. The latter being explored in the paper of our reply. Given the nature of Solomonoff's Induction, they hypothesize and argue that Transformers pervasiveness across fields could be linked to the fact they instantiate Solomonoff's Induction or *SolInd* in their nomenclature.

2 Background and Original/Initial Hypotheses

In this section, we narrow down and briefly recall the main objects of study:

- Solomonoff's Induction also known as *Algoritmic Probability* or *Levin's semi-measure*,
- The Transformer neural architecture(TNA),
- Can - and in the positive, to what extent, a TNA instantiate the *universal prior*?

The format of this section, will be as follows, first, we recall some basic information pertaining to our subject matter.
Second, we put an excerpt of the paper we are replying to ([1]), followed by our remark(s); which can either enforce or bring contrast to their position.

[4] https://www.coursera.org/specializations/explainable-artificial-intelligence-xai
[5] The limitations are due to the fact that we cannot replicate infinite time and space.

Solomonoff's Induction

We start by recalling the algorithmic probability definition:

Definition 1 (Algorithmic Probability)

$$AP(s) = \sum_{p:U(p)=s} \frac{1}{2^{|p|}} \tag{1}$$

Which reads as, the *algorithmic probability* or *Levin's semi-measure*, of a string s is the probability of generating that string s by a random program resulting from a fair toss coin and halts, with respect to a Turing machine, usually the Universal Turing machine.
As for the alphabet, it is common practice to use the dyadic alphabet: $\Sigma = \{0, 1\}$.

We also recall that in the long run, Solomonoff's Induction dominates all other measures [11].

Transformers

The Transformers have been introduced in the paper: *Attention is All you Need* [12]. Initially, their purpose was to address a performance bottleneck issue in the field of natural language processing(NLP).

However, later developments showed that there is no stopping TNAs. This is in the sense that when exposed to enough data, they are capable of performing on par, and in some cases outperform "classical" algorithms in computer vision tasks [13].

Recall of *Self Attention* mechanism mathematically:

Definition 2 (Attention)

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

With:

- Q, query vector,
- K, key vector,
- V, value vector.

In the **Background** section [1], the following point is raised:

> SolInd's speed of converging on any given sequence depends on the choice of UTM - if a sequence x has a shorter description on M than on some other UTM N , then M will make less prediction error on x than N will. This has limited importance in the limit, though it may be a vital consideration in practice.

Remark 1 The use of a binary alphabet has implications both from a technical and philosophical [14] perspectives. Technically, we know that we can encode without loss of information any data. However, and as expressed in the paper [1], if we are interested on the practical side of things, the encoding choice has significant weight.

Remark 2 The previous remark is correlated with the choice of the Turing machine. Entering the landscape, is the *invariance theorem*; which up to an additive constant, tells us that it does not really matter which Turing machine to select. Similarly, to the previous remark, in practice, we argue that this factor matters.
To illustrate our point about this dialectic between theory and practice, we can refer to satisfiability modulo theories problems(SATs). Theory tells us that the difficulty is NP-Hard, however, in practice, SAT solvers perform well; there seems to be an underlying structure to the problems that we are interested in [15].

Hypothesis 1: *For general problems, is there a similar pattern? Could the computational space of programs we are interested in be a more localized/reduced one?*

In section **3.1 Reasoning**:

> Our main consideration is in comparing Transformer models to SolInd, since they currently represent an unopposed state-of-the-art in almost all fields of machine learning and artificial intelligence; however, the intention is to introduce a way of understanding every method by which continuations of sequences can be predicted, including all those invented to the present day and all those which might be invented in the future. Indeed, it is necessary to compare Transformers' approximations of SolInd to other prediction methods if we are to understand how they achieve their superior performance.

Remark 3 In a recent work: *SuperARC: An Agnostic Test for Narrow, General, and Super Intelligence Based On the Principles of Recursive Compression and Algorithmic Probability* [16]. The authors investigate and propose a custom intelligence benchmark and metrics AIT aware. They evaluate frontier models which are often composed of decoder-only Transformers. Rooted in algorithmic probability, they observe that these models performs do not stem from first principles but from memorization, search and powerful pattern matching.

In section **4.1 Universality**:

> One key result on which our hypothesis relies is the idea that Transformers - and NNs in general - can implement arbitrary programs (equivalently, can emulate arbitrary TMs). That Recurrent Neural Networks (RNNs) can emulate arbitrary TMs was first shown in 1992 by Siegelmann and Sontag [12].

Remark 4 Although we agree with the previous, the following work: *Attention is Turing Complete* [17] is more in line/tailored with the Transformers. In order to prove Turing completeness of the *attention* mechanism, they construct their proof around the following key points, which pertains to this remark and the next:

- An arbitrary precision is required. However, hardware implementation uses fixed precision. The interplay between fixed precision and positional encoding leads to the universality property loss. From an operational point of view, the positional encoding is required to track the order of processed tokens.
- Additionally, the use of *residual connections* is required for their proof. The question as if it is a necessary condition or not is still open. We propose a hypothesis below in *Remark* 8 in the context of Chomsky's hierarchy.
  Additionally, the use of *residual connections* in TNAs which are a declination of *skip connections* allows to smooth the loss function landscape by avoiding to fall into an chaotic attractor point [18]. In closing on the *residual connections*, another notion that could be of interest to explore is that of: *super weights* and *super outliers* [19]. These are weights that have orders of magnitude more impact on the accuracy of the model.

In section **4.2 Decomposition**:

> One well-supported hypothesis in this space is the lottery ticket hypothesis [5]: that randomly-initialised networks contain subnetworks that can be achieve sim- ilar performance to the whole network if trained in isolation, having been given random values that quickly reach a high level of performance. This implies that such "winning tickets" exist as part of any random initialisation of any NN, for every function that that NN is capable of emulating. Further, since a RNN can emulate a UTM with as few as 1058 neurons [12] - many fewer than modern deep NNs - most functions can be reasonably considered to be emulable by most NNs.

Remark 5 Another body of work considers that randomly initialized inputs are biased toward capturing simple functions [20].

> In other words, there is good reason to believe that randomly-initialised net- works contain something close to a representation of any arbitrary TM, the most relevant of which can be found during training and given increased weight over other subnetworks. This seems to closely mirror SolInd's process of induction.

Remark 6 This can be brought close to and explored through the lens of the *superposition* concept of mechanistic interpretability [21].

In section **5.1 Limits of stochastic gradient descent**:

> After training on tasks at varying levels of the Chomsky hierarchy, Delétang et al. [4] found that Transformers can generalise well to regular tasks, but suffer greater performance penalties on tasks that are further up on the Chomsky hierarchy - eventually leading to recursively enumerable tasks, which correspond to those that can be solved by Turing

machines. Other kinds of NNs tested exhibited similar behaviour. The authors speculate that this may be due to the limitations of stochastic gradient descent as a training technique - when weights are gradually changed to approach an ideal value, small imperfections can accumulate.

Remark 7 A first remark, regarding the hypothesis on the potential limitations of stochastic gradient descent(SGD) is that, 1) SGD has an implicit simplicity bias(SB) which may lead to reduced feature rich capable classifier [22]. In addition, SB impacts and hinders the generalization capabilities of a network [23]. At the same time one should also consider the fact that non-adaptive optimizers such as SGD may lead to better generalization; specifically for SGD, it seeks flatter minima [24].

Remark 8 This remark echoes/extends the one presented in *Remark* 4 in the following sense. We know that *partial recursive functions* are equivalent to Turing machines that may or not halt; thus having the most expressive power. This class of functions is equipped with the following properties [25][1 (Enumerability), 7-9 (Recursive Functions), 15 (Turing Machines and Recursive Functions)]:

- All initial functions (zero, successor and projection),
- Composition,
- Primitive recursion,
- Unbounded search μ-operator.

In *Remark* 4, we have outlined the potential role of *residual connections* in acquiring the Turing-completeness property. More specifically, the "`+xi , +ai , +yi , and +pi summands in the definition of the single-layer encoder.`" Based on the presented elements, we formulate the following Hypothesis 2:
Can the previous summands be used to help implement the identity function - which can be expressed using one of the initial projection functions?
In the positive, this would make it/them a necessary condition to reach Turing-completeness.

Remark 9 We question the speculation concerning potential limitations of stochastic gradient descent(SGD). To this end, we put forward arguments [26], embedded in the framework of algorithmic information theory(AIT) and specifically Kolmogorov complexity. It is a reasonable assumption that TNAs(e.g.: LLMs) are trained using SGD or one of its variants. Thus, using the development conducted in [26], shows convergence of SGD with randomly initialized inputs (for the technicality of the proof, the author sets a stopping for when the accuracy is sufficiently high). This could constitute an element weakening the claim about the potential limitations of SGD. Leading us, to another culprit e.g.: the untrained network prior, [22],[23].
In closing this point, combined to the previous *Remark* 8 would weaken the proposed speculation.

3 Discussion & Conclusion

The artificial intelligence landscape is advancing at fast pace. The dominating paradigm fueling most the current advances: *Deep Learning (DL) / Deep Neural Networks (DNNs)* and some of its derivatives e.g.: Transformers based neural architectures e.g.: *Large Language Models (LLMs)* with their different incarnations.

All the same, our position is within the framework of {algorithmic, information} theory. We are witnessing and uprising of this paradigm which has been confined to the theoretical realm for quite some time. Moreover, its practicality or the lack thereof could probably be linked to the uncomputable nature of its main measures: algorithmic {complexity, probability}. We thus have chosen to reply to a work exploring one of this aspect and expanding upon it.

In this exercise, we started with a quick recall of the cornerstones of our study i.e.: *Transformers* and *Solomonoff's Induction*. Then, we replied to a number of the elements advanced in *Transformers As Approximations of Solomonoff Induction* [1]. Based on the presented argument, we took a position that would either: 1) enforce i.e. strengthen it with an additional perspective, 2) weaken i.e. bring some contrast to their proposed reading.

Furthermore, through our remarks, we express two hypotheses: (2) and (8).
The former asking whether there could be a "computational space" where most of the problems we are interested in would hypothetically be represented. Tangentially or more, this is not without recalling the topics of 1) the halting problem and enumeration of small Turing machines [27],[28] – which can be interpreted as most machines halt "quickly" or never halt, and 2) the statistical mechanics-based interpretation[29][§8.1] of DNNs – from which we can observe that, the space of interested functions(from an operational point of view) to be smaller.

The second hypothesis is interested in computational universality and asks, whether there is a possible connection between the properties(specifically the identity function) of a *partial recursive function* and the *residual connections* in a Transformer.

Penultimately, within the subject matter and derived from our remarks, we present some potential research extensions based in current literature and are interlinked, of which:

- Sparse coding, e.g.: *Emergence of simple-cell receptive field properties by learning a sparse code for natural images* [30],
- Distributed organization (architecture, memory), e.g.: *Attention approximates sparse distributed memory* [31]
- Mechanistic interpretability e.g.: *Toy Models of Superposition* [21], *Superposition, memorization, and double descent* [32].

Finally, in light of the presented information, *Solomonoff's Induction* or *Universal prior* can be thought of, as a descriptive selection model rooted in algorithmic complexity.

4 Declarations

4.1 Funding

Not applicable

4.2 Conflict of interest/Competing interests

Author: Tolga Topal declares having not conflict of interests.

4.3 Ethics approval and consent to participate

Not applicable

4.4 Consent for publication

Not applicable

4.5 Data availability

Not applicable

4.6 Materials availability

Not applicable

4.7 Code availability

Not applicable

4.8 Author contribution

Lead author: Tolga Topal work done while also being affiliated to KAUST.

References

[1] Young, N., Witbrock, M.: Transformers as approximations of solomonoff induction. In: Mahmud, M., Doborjeh, M., Wong, K., Leung, A.C.S., Doborjeh, Z., Tanveer, M. (eds.) Neural Information Processing, pp. 16–25. Springer, Singapore (2025)

[2] Jonas, E., Kording, K.P.: Could a neuroscientist understand a microprocessor? PLOS Computational Biology **13**(1), 1–24 (2017) https://doi.org/10.1371/journal.pcbi.1005268

[3] Bletsch, T., Jiang, X., Freeh, V.W., Liang, Z.: Jump-oriented programming: a new class of code-reuse attack. In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security. ASIACCS '11, pp. 30–40. Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/1966913.1966919 . https://doi.org/10.1145/1966913.1966919

[4] Dolan, S.: Mov is turing-complete. Technical report, University of Cambridge, Computer Laboratory (2013). Accessed: 2025-04-05. https://web.archive.org/web/20190331191157/https://www.cl.cam.ac.uk/%7esd601/papers/mov.pdf Accessed 2025-04-05

[5] Rojas, R.: Conditional branching is not necessary for universal computation in von neumann computers. JUCS - Journal of Universal Computer Science **2**(11), 756–768 (1996) https://doi.org/10.3217/jucs-002-11-0756 https://doi.org/10.3217/jucs-002-11-0756

[6] Wiedermann, J., Leeuwen, J.: Finite state machines with feedback: An architecture supporting minimal machine consciousness. In: Computing with Foresight and Industry: 15th Conference on Computability in Europe, CiE 2019, Durham, UK, July 15–19, 2019, Proceedings, pp. 286–297. Springer, Berlin, Heidelberg (2019). https://doi.org/10.1007/978-3-030-22996-2_25

[7] Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., Olah, C.: A mathematical framework for transformer circuits. Transformer Circuits Thread (2021). https://transformer-circuits.pub/2021/framework/index.html

[8] Kratsios, A.: The universal approximation property. Ann. Math. Artif. Intell. **89**(5-6), 435–469 (2021) https://doi.org/10.1007/S10472-020-09723-1

[9] Turing, A.M.: On computable numbers, with an application to the entscheidungsproblem. Proc. London Math. Soc. **s2-42**(1), 230–265 (1937) https://doi.org/10.1112/PLMS/S2-42.1.230

[10] Barak, B.: Introduction to Theoretical Computer Science: index (2023). https://introtcs.org/

[11] Schmidhuber, J.: Discovering neural nets with low kolmogorov complexity and high generalization capability. Neural Networks **10**(5), 857–873 (1997) https://doi.org/10.1016/S0893-6080(96)00127-X

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 5998–6008 (2017). https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[13] Topal, T.: What fuels transformers in computer vision? Unraveling vit's advantages (2024). https://www.grin.com/document/1437625

[14] Sterkenburg, T.F.: The foundations of solomonoff prediction. Studenttheses.uu.nl (2013) https://doi.org/https://studenttheses.uu.nl/handle/20.500.12932/12946

[15] Ansótegui, C., Bonet, M.L., Giráldez-Cru, J., Levy, J., Simon, L.: Community structure in industrial sat instances. Journal of Artificial Intelligence Research **66**(Vol. 66), 443–472 (2019) https://doi.org/10.1613/jair.1.11741

[16] Hernández-Espinosa, A., Ozelim, L., Abrahão, F.S., Zenil, H.: SuperARC: An Agnostic Test for Narrow, General, and Super Intelligence Based On the Principles of Recursive Compression and Algorithmic Probability (2025). https://arxiv.org/abs/2503.16743

[17] Pérez, J., Barceló, P., Marinkovic, J.: Attention is turing-complete. Journal of Machine Learning Research **22**(75), 1–35 (2021)

[18] Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T.: Visualizing the Loss Landscape of Neural Nets (2018). https://arxiv.org/abs/1712.09913

[19] Yu, M., Wang, D., Shan, Q., Reed, C.J., Wan, A.: The Super Weight in Large Language Models (2025). https://arxiv.org/abs/2411.07191

[20] Dingle, K., Camargo, C.Q., Louis, A.A.: Input–output maps are strongly biased towards simple outputs. Nature Communications **9**(1) (2018) https://doi.org/10.1038/s41467-018-03101-6

[21] Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., Olah, C.: Toy Models of Superposition (2022).

https://arxiv.org/abs/2209.10652

[22] Mingard, C., Valle-PÃ©rez, G., Skalse, J., Louis, A.A.: Is sgd a bayesian sampler? well, almost. Journal of Machine Learning Research **22**(79), 1–64 (2021)

[23] Shah, H., Tamuly, K., Raghunathan, A., Jain, P., Netrapalli, P.: The Pitfalls of Simplicity Bias in Neural Networks (2020). https://arxiv.org/abs/2006.07710

[24] Wilson, A.C., Roelofs, R., Stern, M., Srebro, N., Recht, B.: The marginal value of adaptive gradient methods in machine learning. CoRR **abs/1705.08292** (2017) 1705.08292

[25] Boolos, G.S., Burgess, J.P., Jeffrey, R.C.: Computability and Logic, 5th edn. Cambridge University Press, ??? (2007)

[26] Schwartzman, G.: SGD Through the Lens of Kolmogorov Complexity (2022). https://arxiv.org/abs/2111.05478

[27] Calude, C.S., Dumitrescu, M.: A probabilistic anytime algorithm for the halting problem. Comput. **7**(2-3), 259–271 (2018) https://doi.org/10.3233/COM-170073

[28] Calude, C.S., Dumitrescu, M.: A statistical anytime algorithm for the halting problem. Comput. **9**(2), 155–166 (2020) https://doi.org/10.3233/COM-190250

[29] Martin, C.H., Mahoney, M.W.: Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning (2018). https://arxiv.org/abs/1810.01075

[30] Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature **381**(6583), 607–609 (1996) https://doi.org/10.1038/381607a0

[31] Bricken, T., Pehlevan, C.: Attention approximates sparse distributed memory. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. NIPS '21. Curran Associates Inc., Red Hook, NY, USA (2021)

[32] Henighan, T., Carter, S., Hume, T., Elhage, N., Lasenby, R., Fort, S., Schiefer, N., Olah, C.: Superposition, memorization, and double descent. Transformer Circuits Thread (2023)