A Philosophical Introduction to Language Models

Part II: The Way Forward

Raphaël Millière
Department of Philosophy
Macquarie University
raphael.milliere@mq.edu.eu

Cameron Buckner
Philosophy Department
University of Houston
cjbuckner@uh.edu

Abstract

In this paper, the second of two companion pieces, we explore novel philosophical questions raised by recent progress in large language models (LLMs) that go beyond the classical debates covered in the first part. We focus particularly on issues related to interpretability, examining evidence from causal intervention methods about the nature of LLMs' internal representations and computations. We also discuss the implications of multimodal and modular extensions of LLMs, recent debates about whether such systems may meet minimal criteria for consciousness, and concerns about secrecy and reproducibility in LLM research. Finally, we discuss whether LLM-like systems may be relevant to modeling aspects of human cognition, if their architectural characteristics and learning scenario are adequately constrained.

1. Introduction

The maturation of connectionist models in the form of deep neural networks not only revives long-standing philosophical issues but also introduces new ones that await exploration (Buckner 2019). This development notably includes the progress of large language models (LLMs) like GPT-4 (OpenAI 2023*a*), whose impressive performance on complex linguistic and cognitive tasks raises philosophically rich questions. In a companion paper, we discussed the significance of LLMs in relation to classical problems from the philosophy of artificial intelligence, cognitive science, and linguistics (Millière & Buckner 2024). In this paper, we turn to relatively new issues raised by the progress of LLMs. We focus in particular on three sets of questions. In section 2, we ask whether and how we can gain an understanding of the internal mechanisms of LLMs beyond their behavioral performance. In section 3, we discuss philosophical questions about achievements that may be just over the horizon of current research: focusing on multimodality, agency, consciousness, and reproducibility. Finally, in section 4, we bring these strands together by interrogating the status of LLMs as models of human and animal cognition.

2. Mechanistic understanding and intervention methods

As noted in Part I, LLMs have made rapid progress on producing human-like linguistic behavior in many different scenarios that were challenging for previous methods in artificial intelligence. They can

readily produce grammatically-correct and generally semantically-coherent text. They can respond flexibly to a wide range of questions, including a variety of challenging aptitude and proficiency tests. They can even generate functional blocks of computer code in various programming languages, as well as code in visual markup languages that produce coherent images. As we also noted, however, there remain significant concerns that these achievements may have less profound explanations than they first appear. While we have argued that skeptical interpretations appealing to mere memorization cannot account for the full range of behaviors exhibited by these models, our previous analysis did not provide a positive account of the mechanisms that might enable LLMs to achieve such breakthrough performance. Providing such an account is challenging due to familiar concerns about the opacity of neural networks, which is exacerbated by LLMs' relatively novel architecture, their enormous number of adjustable parameters, and the sheer magnitude of their training data. To address these challenges, the research community is rapidly developing new methods of analysis, which we explore in this section.

2.1. The trouble with benchmarks

The most common way to evaluate LLMs is to pit them against a human baseline, or against each other, on *benchmarks* – standardized tests or sets of tasks designed to assess specific capabilities of these models. Benchmarks are designed to provide a quantitative assessment of model performance in various domains, facilitating a direct comparison of their abilities in a controlled and systematic way. Unfortunately, benchmarking methods are plagued by limitations that hamper their reliability and adequacy to arbitrate dispute about the capacities of LLMs. These limitations include *saturation*, *gamification*, *contamination*, and *lack of construct validity* (which, as we shall see, are not all independent concerns, but are interrelated).

New natural language processing (NLP) benchmarks tend to saturate at an accelerating pace, meaning that LLMs quickly surpass the human baseline (Kiela et al. 2021, Ott et al. 2022). This is particularly striking for so-called 'natural language understanding' (NLU) benchmarks. Taken at face value, saturation on these benchmarks would suggest that the best-performing NLP models 'understand' language better than humans themselves. However, such success is often marred by independent examples of obvious failures, suggesting that performance saturation is not reliable evidence that LLMs actually surpass humans on the cognitive ability or aptitude (the 'target construct') that the benchmark was designed to assess.[1]

Several factors explain benchmark saturation. One such factor is the gamification of benchmarks. Intense competition for best model performance on leaderboards – also known as 'SOTA' (state-of-the-art) chasing – ultimately undermines the very purpose of behavioral evaluation. It results in models optimized to improve scores on proxy metrics targeted at benchmarks rather than the attribute the benchmark was designed to assess. This can be seen as an example of Goodhart's Law, commonly stated as the idea that when a measure becomes a target, it thereby ceases to be a good measure (Goodhart 1975). In the context of benchmark creation, statistical relationships identified from empirical observation and used to construct proxy metrics measured by test items tend to break down when actively exploited and optimized (Manheim & Garrabrant 2018). This results in models that may overperform on a benchmark, yet lack the capacity supposed to be measured by that benchmark, at least to the degree suggested by their performance.[2]

[1] In the case of NLU, the target construct – language 'understanding' – is itself ill-defined. See our discussion of semantic competence in Part I (Millière & Buckner 2024) and our discussion of construct validity below.

[2] See Gururangan et al. (2018) for an example of how artifacts in benchmark construction can amplify the divergence between proxy metrics and target construct as models saturate the benchmark.

Another phenomenon that contributes to benchmark saturation is data contamination. This occurs when test data – including benchmark items and their solutions – leak into the training data, either by chance or by design. With LLMs trained on internet-scaled data, contamination is increasingly difficult to avoid because benchmarks are widely reproduced and discussed online (OpenAI 2023*a*). Detecting contamination is a challenge because simple variations of test items, such as paraphrases, can easily evade common detection measures based on string matching (Yang, Chiang, Zheng, Gonzalez & Stoica 2023). This calls into question the significance of some LLMs' reported performance on certain tests. For example, there is evidence that GPT-4's training data was contaminated with solutions to problems from the competitive programming contest Codeforces, such that it performs much worse than reported on old or recent problems that did not make it into its training data (Roberts et al. 2023).

A broader concern with LLM benchmarks pertains to the construct validity of their target. Construct validity refers to the degree to which a test accurately measures the theoretical construct it purports to assess. Standardized tests designed for humans are (ideally) meaningful because they are conceived to measure some underlying skill or capacity, such that good test performance should generalize to relevant real-world situations. However, establishing construct validity for LLM benchmarks is more challenging. Constructs targeted by common benchmarks, such as 'understanding' or 'reasoning,' are often abstract, multifaceted, and implicitly defined with reference to human psychology. Simply using test items validated for human subjects will not do, not only because of the potential for contamination, but also because validity is conditional upon background theoretical assumptions about test subjects. In other words, background assumptions about human cognition on which human-centric tests are premised can impact their applicability to LLMs. As a result, an LLM achieving good performance on a validated test designed to measure a capacity ϕ in humans may not constitute adequate evidence that the LLM has ϕ.

As an example, consider the contentious case of Theory of Mind (ToM) – the cognitive ability to understand and attribute mental states to oneself and others, and to recognize that these mental states may differ from one's own. Using classic verbal false belief tasks adapted from the developmental psychology literature on ToM, Kosinski (2023) found that GPT-3 performed comparably to nine-year-old children. However, Ullman (2023) complicates this picture by showing that introducing minor conceptual variations on these test items, while preserving the requirement for false belief inference, significantly degrades LLM performance. This disparity suggests that GPT-3's impressive performance on original test items does not provide adequate evidence of ToM. This could certainly be due to some form of data contamination – after all, there is hardly any doubt that false belief tasks feature in GPT-3's training data. But Ullman (2023) also suggests that even if LLMs matched human performance on challenging variations of classic false belief tasks, we should be careful not to jump to conclusions about their putative ToM aptitude. Indeed, tests designed with human cognitive processes in mind warrant inferences about the target construct given plausible assumptions about the mechanisms that drive human performance on these tests. These assumptions may not translate to artificial systems like LLMs without independent motivation. For example, the developmental trajectory and cognitive architecture of human ToM may lend itself to simulation-based or causal models of other minds that generalize flexibly. In contrast, an LLM could potentially achieve similar task performance by memorizing and interpolating patterns in training data, without engaging in the same underlying reasoning. Thus, the construct validity of ToM tests depends not just on input-output mappings, but on the nature of the algorithms and representations that generate those outputs. We have independent reasons to posit Theory of Mind as a core socio-cognitive capacity in humans, based on its early emergence, specificity to agents, and dissociability from general intelligence. Analogous supplementary evidence might be needed to corroborate claims about emergent ToM in LLMs.

Most of the limitations of LLM benchmarking are inherent to behavioral evaluation more generally. This point is often highlighted with reference to the distinction between *performance* and *competence* (Chomsky 1965, Firestone 2020). The distinction is most commonly invoked by skeptics about LLMs to suggest that performance success, such as high scores of benchmarks, need not reflect the competence associated with the underlying construct (Kiela et al. 2021). Humans may achieve good performance because they exercise some cognitive capacity ϕ (e.g., ToM), while LLMs may achieve similar performance through completely different means (e.g., memorization of linguistic patterns that correlate with common ToM evaluations). The core concern here is that input-output mappings provide insufficient evidence about the *mechanisms* that lead from input to output in a given system. Consequently, serious research on the capacities of LLMs should go beyond behavioral evaluation, and seek to understand how they process information mechanistically.

It should be noted that the performance-competence distinction cuts both ways. In humans, it is common to disregard performance failures as evidence of a lack of competence, because such failures may be explained away by contingent constraints on cognitive function (e.g., limitations of working memory or attention). Most famously, Chomskyan linguists argue that performance errors are mostly irrelevant to linguistic competence, because they are the result of processing limitations or external factors rather than a reflection of the underlying linguistic system.[3] Many cognitive scientists have called into question the existence of such an absolute gap between performance and competence (Tomasello 2009, Christiansen & Chater 2016). Another concern is that hypotheses about performance limitations need independent empirical support (lest they be invoked ad hoc to preserve a classical model in the face of empirical disconfirmation), but this independent evidence is rarely provided even in the case of humans, and direct investigations of performance factors can clash with competence theories (Franks 1995, Tomasello 2009, Theakston et al. 2001). Nonetheless, it is intriguing to consider how the dissociation between performance and competence might also apply to LLMs in both directions. That is, just as LLMs' impressive performance on benchmarks may always not accurately reflect their true competence, one might also consider that the existence of some performance errors do not always conclusively establish the absence of such competence.

2.2. Mechanistic explanation

In science, mechanistic explanations aim to reveal the causal structure underlying a phenomenon of interest by describing the organized entities and activities that are responsible for producing or maintaining that phenomenon (Machamer et al. 2000). More precisely, a mechanistic explanation identifies the component parts of a mechanism, characterized by their properties and capacities, the causal interactions between these component parts, and the organization of the parts and activities such that they give rise to the phenomenon. Such explanations stand in contrast to purely descriptive or phenomenological models that simply *re-describe* the phenomenon itself, as well as covering-law explanations that explain by subsuming the phenomenon under empirically discovered regularities or governing laws. Mechanisms explain by revealing *how* the phenomenon arises from the causal structure of the system, not merely describing empirical regularities in which it partakes. As such, mechanistic explanations reveal opportunities for manipulation and control over the phenomenon in a way that descriptive or law-based explanations do not.

Understanding the behavior of a simple system like a mechanical clock is easy enough – one can simply open it up and observe the mechanism at work. This is not so straightforward with more

[3]There is much debate about how to delineate the performance/competence distinction, as well as some confusion which is partly due to somewhat inconsistent characterizations in Chomsky's work itself. For an enlightening exposition of the distinction in linguistics, see Dupre (2022). For a discussion of its relevance to the NLP and LLMs, see Dupre (2021) and Millière (n.d.).

complex systems, like the weather, the brain, or artificial neural networks. Neural networks are often described as 'black boxes' precisely because the causal mechanisms that explain their behavior seem opaque to scrutiny. As we emphasized in Part I (Millière & Buckner 2024), simply staring at the learning objective, architecture, or parameters of LLMs will reveal neither how they exhibit their remarkable performance on challenging tasks, nor what functional capacities can be meaningfully ascribed to them. In principle, one could even provide a complete mathematical description of an LLM as a giant composite function, consisting in an absurdly complex sequence of linear and nonlinear transformations across many layers; but such a description, on its own, would be useless to provide a genuine explanation of the network's behavior in specific contexts. The 'black box' metaphor underscores this chasm: it highlights the difficulty to trace precise causal pathways in the network through which specific inputs are transformed into specific outputs. This is why merely *re-describing* what an LLM does in terms of next-token prediction or matrix multiplication – what we called the 'Re-description Fallacy' in Part I – cannot possibly settle philosophical debates that are fundamentally about causal mechanisms.

The search for causal mechanisms has become central across the life sciences and cognitive science. Like their artificial counterparts, biological neural networks are 'black boxes'; yet neuroscientists are engaged in the project of uncovering multilevel mechanisms underlying psychological capacities and nervous system functions (Craver 2007). At the molecular and cellular levels, they describe mechanisms of protein synthesis, gene expression, and synaptic transmission; at an intermediate level are mechanisms of neuron spiking and oscillation as well as mechanisms of development and synaptic plasticity; at higher levels are mechanisms underlying learning, memory, reasoning, and other psychological capacities.

Mechanisms explain by opening up the causal 'black box' linking cause and effect – revealing the internal entities, activities and organization that transmit causal influence through the system. The notion of intervention is central to this explanatory project. In the philosophy of science, interventionism holds that causal relationships are best understood in terms of what would change under interventions or manipulations to parts of the system (Woodward 2005). More specifically, X is considered a direct cause of Y if and only if there exists a possible intervention on X that will change Y (or the probability distribution of Y) when all other variables in the system are held fixed. This relationship is asymmetric – intervening on Y does not change X if the causal arrow truly points from X to Y. Interventionism eschews regularity or correlational notions of causation, recognizing that a system's behavior depends on more than merely observing regular successions of events. In the context of mechanistic explanation, interventions involve altering specific parts of a posited mechanism piecemeal in order to learn about their causal contribution to the target phenomenon. For example, pharmacology targets specific receptor mechanisms or cell signaling pathways; brain stimulation techniques activate or inhibit activity in restricted brain areas; optogenetics uses light to control neurons genetically modified to express light-sensitive channels; and knockout models remove genes hypothesized to be critical for mechanisms.

A similar interventionist approach can be applied to unravel causal mechanisms in artificial neural networks like LLMs. Of course, there are major disanalogies between biological nervous systems evolved over millions of years and artificial neural networks designed by human engineers. Nevertheless, the motivation of interventionist research is similar: to achieve explanatory understanding by revealing multilevel mechanisms, not merely observing input-output patterns. Like neuroscientists, computer scientists can aim for explanatory understanding linking particular components of neural networks and patterns of internal activations to specific capacities, such as translating between languages or answering arithmetic questions. This explanatory project is often described as *mechanistic interpretability*. In a broad sense, mechanistic interpretability could be characterized as the search for mechanistic explanations of the behavior of deep neural networks, including LLMs. As

we shall see, however, the phrase is often used in a more restrictive sense, to denote a particular set of theoretical assumptions and intervention methods used to achieve mechanistic explanations in machine learning.[4]

2.3. Opening up the black box

There are three main methodological approaches to investigate the inner structure of neural networks: probing, attribution, and *bona fide* causal intervention. Probing involves training a separate supervised classifier, known as a diagnostic probe, to predict certain properties (e.g., part-of-speech tags, dependency relations) from the model's internal activations (Alain & Bengio 2018, Hupkes et al. 2018). High accuracy in decoding a particular linguistic feature F from a probe tuned to a pattern of activation A in a subset of the network can be though to provide evidence that A is sensitive to F, and provide information about its presence or absence for downstream processing. However, a probe's successful prediction of a certain linguistic feature from a model's activations does not necessarily mean that the feature plays a causal role in the model's behavior (Belinkov 2022). In addition, probes can pick up on spurious correlations, thereby failing to distinguish between genuine representation and incidental associations (Hewitt & Liang 2019).

Probing can tell us that some information is likely present in the activations of the system, but not that it is in fact used for a particular purpose or function in generating the system's outputs. Indeed, the mere presence of usable information does not demonstrate that it is actually exploited. By way of analogy, suppose you throw block letters spelling a word into a pond. With the right apparatus, you might be able to decode the word's identity from the ripples formed by letters over the pond's surface; but this does not provide evidence that the pond represents (let alone 'understands') the word in any meaningful sense. It merely shows that the block letters created surface patterns whose origin could be recovered by a sufficiently powerful decoding method.

Methods that do not involve training a separate classifier – also known as nonparametric methods – have been explored as an alternative to probing. For example, one can directly investigate the weights (or attention scores) that attention heads place on different parts of the input sequence. A high attention score on a particular word might be thought to provide evidence that the head is playing a role in processing the semantic or syntactic role of that word in the rest of the sentence. This approach to interpretability falls within the broader and somewhat loose umbrella of attribution methods in deep learning. Attribution methods assign importance scores to input features to explain individual model predictions; in other words, they are meant to identify parts of the input that are most influential for the output. While attention patterns in Transformers intuitively provide clues about the relevance of each input token to model predictions, simply analyzing these patterns in a given layer of the network only offers limited explanatory power (Chefer et al. 2021). In particular, visualizing attention patterns is thoroughly insufficient to draw strong conclusions the flow of information through the network. If we want to understand what information the model represents in the input, and how that information is processed through step-by-step computations across layers to drive output predictions, we need to intervene directly on the network to reveal its causal structure.

2.4. Interventions on neural networks

To assert that a certain activation pattern in a model genuinely represents a given feature, mainstream philosophical theories of representation require that three criteria be satisfied: (a) the activation

[4]The lack of a common lexicon in interventionist research on neural networks is rather unfortunate – many different labels exist for similar ideas and methods, with 'interpretability' being a particularly confusing term (Lipton 2018). In what follows, we will attempt to give a cohesive overview of this fragmented literature.

pattern should carry information about the feature; (b) the activation pattern should influence the model's behavior in a task-relevant way, and (c) the model should be capable of misrepresenting the feature (Harding 2023). Standard probing methods only provide evidence for the first criterion. Showing that some information about a feature is *actually* used by the LLM to generate its outputs requires suitable *interventions* on patterns of activation encoding that information. Changes in model behavior caused by such interventions should be consistent with the hypothesis that the model represents the target feature, but should not be explainable by mere perturbations from the training set. Accordingly, an increasingly large body of work uses targeted intervention methods to establish causal relationships between language models' internal representations and their behavior.

The simplest kind of intervention on neural networks – both artificial and biological – is an ablation. In the context of artificial neural networks, ablation involves disabling or eliminating individual neurons or groups of neurons within a trained model to observe the resulting changes in behavior. By ablating neurons and observing the resulting change in performance metrics such as classification accuracy or reconstruction error, researchers can determine how much each neuron or group of neurons contributes to the network's overall function. Neurons that are critical to network performance will result in a substantial decline in metrics when ablated.

Indiscriminate ablations, common in the early days of connectionist research, involved disabling nodes at random. This approach was aimed at demonstrating general properties of neural networks. One key finding from such studies was the concept of graceful degradation: like biological brains, the performance of neural networks tends to decline in a gradual, rather than abrupt, manner when parts of the network are damaged or disabled (Sejnowski & Rosenberg 1987, Smolensky 1988).

Targeted ablations, on the other hand, involve disabling specific nodes or modules believed to serve distinct representational or functional roles within the network. By analyzing the network's altered behavior, researchers could infer the impact of the disabled nodes, thereby gaining insights into their hypothesized functions (Meyes et al. 2019). This method mirrors a common approach in neuroscience where the study of natural or induced brain lesions helps to unravel the functions of specific brain areas.

However, both indiscriminate and targeted ablation studies are somewhat limited in their ability to fully uncover representational or functional roles in neural networks. This limitation is partly due to the nature of how information is represented within these networks. Traditional views centered around *localized* representations, positing that specific neurons or small groups of neurons could be responsible for representing distinct, complex stimuli or concepts. The classic example in neuroscience is the hypothetical 'grandmother cell' – a neuron that would activate exclusively in response to the mental image or concept of one's grandmother. While intuitively appealing, this model of localized representation has largely been abandoned in favor of a distributed model (Plaut & McClelland 2010, Barwich 2019).

In practice, individual neurons or small clusters of neurons in neural networks often encode information about multiple concepts (Smolensky 1986, Rumelhart et al. 1987, Henighan et al. 2023). This is consistent with distributed representations: concepts are represented by patterns of activity that are spread across a larger number of neurons within the network. When components of distributed representations overlap, the same neurons might simultaneously represent multiple features – they are 'polysemantic'. This property is studied under the name of 'superposition'; for example, Elhage et al. (2022) demonstrate that superposition can allow networks to linearly represent many more features than they have neurons or dimensions, at the cost of some interference between features. The distributed model of representations in neural networks reflects a more integrated and holistic approach to neural processing, where information is not stored in isolation but as part of a dynamic, interconnected system.

The shift from localized to distributed representation models has significant implications for interpreting the results of ablation studies. In a distributed system, disabling a node or a set of nodes affects more than just the immediate functionalities those nodes are associated with. It impacts the network-wide interplay of neural activities. Therefore, determining the specific effects of localized damage on the overall behavior of the network becomes increasingly complex (Jonas & Kording 2017).

Fortunately, the level of control we have over artificial neural networks allows for more sophisticated causal interventions that improve on localized ablations by allowing us to 'edit' distributed representations in ways consistent with representational or functional hypotheses, and verifying corresponding changes in behavior. For example, Giulianelli et al. (2018) used a probe to identify activations associated with subject-verb number agreement, then modified these activations to improve the model's performance on a subject-verb agreement task. Such interventions provide more robust evidence about the causal role of specific internal features in a model's behavior.

A more sophisticated approach called 'iterative nullspace projection' was developed by Ravfogel et al. (2020). Iterative nullspace projection can determine whether some particular information is causally involved in a neural network's predictions by identifying and removing that information from distributed neural representations, and then assessing the consequence on model behavior. The approach involves iteratively projecting neural representations onto the nullspaces of a probe to remove detectable information about user-defined target concepts.

More specifically, the first step is to train linear probe on internal neural representations to predict values of the concept of interest. For example, probes could be trained to classify grammatical number from the hidden state of a language model. The nullspace of a probe is the subspace consisting of all activations for which the probe makes the same constant prediction. Projecting representations onto these nullspaces removes information that linearly correlates with concept values while preserving unrelated information. After projecting onto the first probe's nullspace, a second probe is trained on the transformed representations to predict the target concept. This process is repeated iteratively, with projections onto each new probe's nullspace removing additional detectable information about the target concept. Finally, the network makes predictions using the fully projected representations where linear information about the target concept has putatively been eliminated. If performance degrades, the target concept can be inferred to be causally significant for model performance, suggesting that the network exploits it for its original computations.

For example, Ravfogel et al. (2021) investigated whether information about relative clause boundaries is used by language models like BERT to predict subject-verb agreement across a relative clause. They trained a set of linear probes to predict whether a word is inside a relative clause, based on the model's contextual word embeddings. Each probe defines a direction in the representation space that separates words inside relative clauses from words outside relative clauses. Collectively, these directions span a 'feature subspace' that contains relative clause information. The orthogonal complement of this subspace is the nullspace which does not contain information useful for predicting whether a word is inside a relative clause. Then, iterative nullspace projection is used to generate 'counterfactual representations' for the masked verb token. The representation is projected into the relative clause feature subspace, and then flipped to the opposite side of the separating hyperplane, either towards the side containing relative words ('positive counterfactual') or away from that side ('negative counterfactual'). This process minimally modifies the representation to incorrectly encode that the word is inside or outside a relative clause. Finally, the effect of the counterfactual representations on the model's number agreement predictions is measured. If swapping in the positive counterfactual increases error rate and the negative counterfactual decreases error rate, this alignment

with predictions from linguistic theory suggests the model uses relative clause boundary information appropriately for agreement, confirming the causal effect (fig. 1).

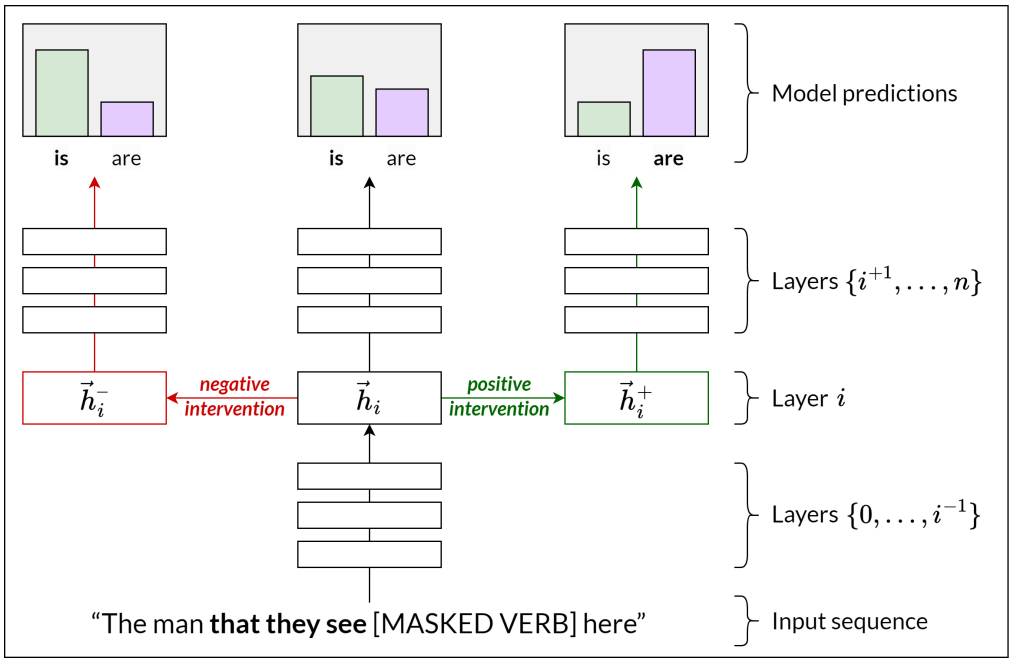

Figure 1 | **Iterative nullspace projection**. Given the representation $\vec{h}_i$ of a masked word in layer i of a Transformer model, a probe is trained to predict relative clause boundaries. The probe's nullspace, which encodes information not relevant to relative clause boundaries, is identified. Two counterfactual representations, $\vec{h}_i^-$ and $\vec{h}_i^+$, are derived by projecting $\vec{h}_i$ onto the nullspace and then performing negative and positive interventions, respectively, along the probe's decision boundary. $\vec{h}_i^-$ encodes that the word is outside a relative clause, while $\vec{h}_i^+$ encodes that it is inside a relative clause, with other information preserved. The model's predictions and using these counterfactual representations are compared to its original prediction to assess the causal effect of the relative clause boundary information on the model's behavior in number agreement.

This method is directly inspired by counterfactual approaches to causal explanation in philosophy of science (Woodward 2005). Such approaches aim to isolate the causal contribution of some factor X to an outcome Y by minimally altering the value or presence of information about X in the system, while holding all other factors fixed. This allows assessment of whether and how much the factor X is exploited causally by the system to produce Y. The removal of detectable information about a target variable through iterative nullspace projection is analogous to a hypothetical intervention that breaks the links between that variable and the system while leaving other causal mechanisms intact. The comparison between original model outputs and outputs based on projected representations missing information about X mirrors the evaluation of interventionist counterfactuals, which aims to answer questions such as "What would happen to Y if an intervention prevented information about X from influencing the mechanism?".

2.5. Mechanistic interpretability

Intervention methods such as iterative nullspace projection can help us determine whether some target concept, such as a particular syntactic feature, is represented by a language model and causally efficacious in its behavior. But this is insufficient to understand language models and other neural networks the way we understand classical computer programs. Indeed, the guiding ideal

of interpretability research is to model the internal causal structure of neural networks, such that we can explain some behavior of interest in terms of a series of computational steps applied on the input. Mechanistic interpretability refers to this concerted effort to reverse engineer the internal computations performed by artificial neural networks. Rather than solely focusing on interpreting individual model predictions, it aims to provide a detailed and systematic understanding of how the model transforms inputs to outputs (Elhage et al. 2021). The overarching goal of mechanistic interpretability is to open up the 'black box' of neural networks by providing human-intelligible descriptions of the functional modules that drive the emergence of model behaviors.

Like systems neuroscience, mechanistic interpretability specifically targets the algorithmic level of analysis (Marr 1982, Lindsay & Bau 2023). An algorithmic explanation elucidates the flow of information through a network and the sequence of operations performed upon it. Crucially, it abstracts away from engineering (or biological) details, instead focusing on the computations performed by the system. Algorithmic explanations support counterfactual inferences about how changing inputs or network components would affect computations and outputs. This involves identifying structural elements like motifs and circuits that are reusable, such that manipulating them produces systematic effects across many inputs and conditions.

Mechanistic interpretability specifically seeks to reverse-engineer neural networks in terms of learned *features* and reusable *circuits* that operate on those features. Features refer to human-interpretable properties of the input data that the model represents internally. For computer vision models, such features might be edges, textures, or shapes. For language models, features may correspond to part-of-speech tags, named entities, or semantic relationships. By studying a network's internal activations, researchers aim to determine which features are encoded where.

Circuits, in turn, are chains of operations that detect certain input features and transform them into output features. The goal is to decompose an entire neural network into a hierarchy of understandable features and circuits that process information step-by-step. This modular view would explain how trained models transform inputs to outputs via learned features and program-like circuits, providing a form of algorithmic understanding.[5]

More formally, a given neural network can be modeled as a graph (sometimes called a computational graph) whose nodes correspond to components of the network at some level of granularity – such as attention heads or individual neurons. The directed edges between nodes represent the flow of information and computations in the neural network. The connectivity defined in the computational graph needs to faithfully represent the actual computation flow in the neural network. In this model, a circuit is a subgraph within the overall computational graph that implements some specific behavior or functionality of interest. The goal of representing the neural network as a computational graph is to systematically analyze it and localize particular circuits that are causally responsible for certain model behaviors.

Once candidate circuits are found, their functionality must be validated through causal interventions. This often involves surgically replacing components of suspected circuits and observing the effect on model outputs. Validated circuits can then be composed into a hierarchical understanding of

[5] In fact, it is even possible to *program* a Transformer from scratch using a specialized programming language like RASP (Restricted Access Sequence Processing Language). RASP allows mapping the basic components of a Transformer, like attention and feed-forward layers, into simple primitives that can be composed into programs. These RASP programs can then be compiled into the weights of a Transformer network that implements the specified computation (Weiss et al. 2021). More recently, methods have been developed to train Transformers whose weights are constrained to implement human-interpretable RASP-like programs. These Transformer Programs can be learned end-to-end from data and decompiled back into discrete, readable code (Friedman et al. 2023). Such approaches provide a more direct way to understand the computations of a Transformer in terms of modular algorithmic components.

the succession of transformations enacted by the model. The resulting mechanistic explanation should provide significant behavioral control, for instance allowing targeted editing of model computations.

The picture that emerges from mechanistic interpretability research is that Transformer models can be viewed as containing parallel processing streams (also known as 'residual streams'), one at each input token position (fig. 2). While each residual stream encodes information in a very high-dimensional vector space, attention heads at each layer operate over much smaller (low-dimensional) subspaces of the stream that may not overlap with one another. Making use of these disjoint subspaces allows Transformers to route information about tokens and their dependencies dynamically across layers and positions. Specifically, each stream is functionally analogous to a addressable memory, in which attention heads can write to and read from subspaces of the main embedding space. Such information may include, for example, syntactic dependencies between tokens in the input sequence.

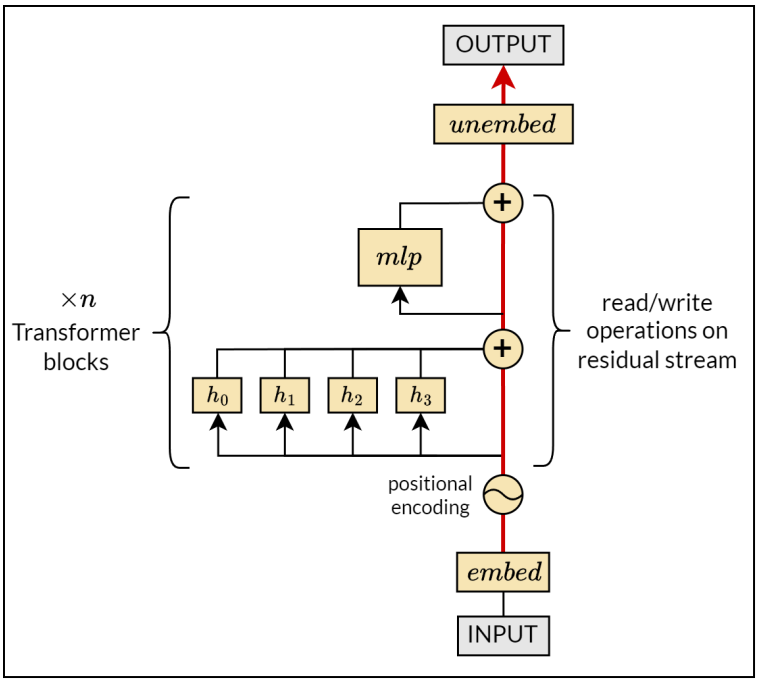

Figure 2 | **The residual stream view of the Transformer.** Each input token is first embedded into a dense vector representation and combined with a positional encoding injecting information about their position in the input sequence. This forms the initial state of the "residual stream" (depicted by the red arrow), which flows through the entire network. Each Transformer block, consisting of multi-head self-attention and a multi-layer perceptron (MLP), reads from the residual stream, transforms the representation, and writes the result back into the stream via residual connections. This process is repeated across multiple Transformer blocks. Finally, the output of the residual stream is 'unembedded' to map the transformed representation back to the original token space. In this view, the Transformer's components are seen as operators that successively refine the residual representation.

A common intervention method used in mechanistic interpretability research is activation patching (Zhang & Nanda 2023), also known as causal tracing (Meng et al. 2023) and interchange intervention (Geiger et al. 2021) (fig. 3). The basic method involves three steps. First, the neural network must be run on an original input that relates to some behavior of interest (e.g., answering factual questions). For example, the original input might be "The capital of France is...", with the expected output being "Paris". Importantly, the activations of the network during the forward pass on this original input are cached for later use. The second step is to run the model again with an alternative input that introduces a key variation on the original input that changes the behavior (output). For example, an

alternative input might be “The capital of Germany is...”, with the expected output being “Berlin”. Finally, the alternative input is run again but a specific component of the network’s activation is swapped for its cached value from the original forward pass – an intervention known as ‘patching’, because it patches the alternative forward pass with a component of the original forward pass. The effect of patching a component’s activation is then evaluated by comparing the model’s performance in the regular forward pass on the alternative input versus the patched forward pass on the same input. For example, the metrics used could be the probability the model assigns to the original, correct token (“Paris”), or the difference in logits between the correct and incorrect tokens (“Paris” vs “Berlin”). Intuitively, changing the input hurts model performance on the expected behavior, while patching activations from the original run helps restore it. So if patching a particular component leads to a significant restoration of performance (e.g., an increase in the probability of “Paris” being the output), it suggests that component is important for the model’s behavior on the task (i.e. the component not only contains information on the target feature, but that information is causally implicated in producing the desired result). By iterating this procedure over many components of the computational graph, such as attention heads in a Transformer model, activation patching aims to identify the key circuits that enable behaviors like factual recall or logical reasoning.

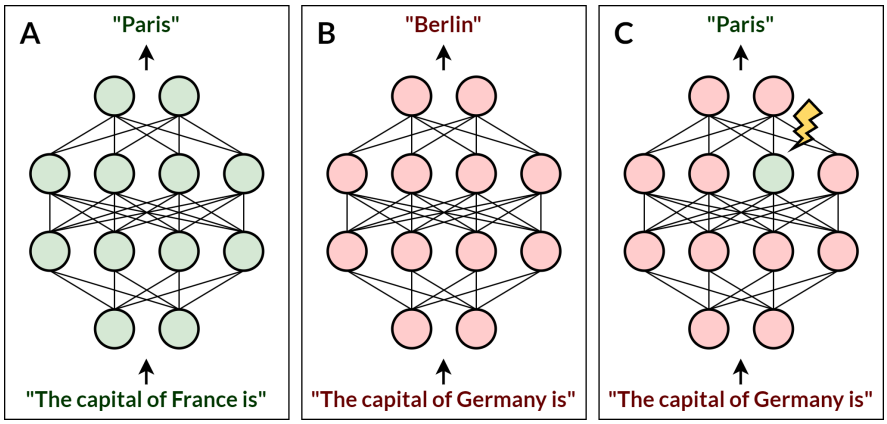

Figure 3 | **Activation patching. A.** In the original forward pass, the model takes as input the prompt “The capital of France is” and outputs the correct answer “Paris”. The model activations from this forward pass are cached. **B.** In the alternative forward pass, the prompt is changed to “The capital of Germany is”. The model now outputs “Berlin” as the answer. **C.** Activation patching is applied in a third forward pass. The model is given the alternative prompt “The capital of Germany is” once again, but a specific component of the model has its activations replaced (*patched*) with those from the original forward pass on the France prompt. This causes the model to output “Paris” instead of “Berlin”, despite being given the Germany prompt. The restoration of the original output through patching the activations of a particular model component provides evidence that this component encodes information that is causally implicated in the target behavior.

The mechanistic interpretability framework has been applied to many different problems, and has started shedding light on some abilities of Transformers. It is worth noting that much of this research is painstaking work conducted with toy models that are easier to interpret. While these toy models are based on the Transformer architecture, they may differ from LLMs in terms of architectural details, learning objective, dataset, and of course size (parameter count). Nonetheless, efforts are underway to automate and scale mechanistic interpretability techniques to bona fide LLMs, with promising initial results (Wu, Geiger, Potts & Goodman 2023, Conmy et al. 2023, Syed et al. 2023). In what follows, we will briefly illustrate the fruitfulness of the mechanistic interpretability program through three case studies.

2.5.1. Case study 1: Induction heads

A canonical example of circuit discovery in language models through the tools of mechanistic interpretability is that of so-called 'induction heads' Olsson et al. (2022). These are specialized attention heads that emerge through training even in very small models, and implement a form of pattern completion, allowing models to repeat or generalize sequences based on similar context patterns.

Specifically, induction heads use a 'prefix matching' attention pattern to look back over previous tokens and detect if any match the current token. Rather than relying merely on memorized statistics about which tokens tend to follow others, they flexibly attend to whichever prior token is most similar to the current one based on learned representations. If a previous token is sufficiently similar, the induction head will attend to the next token after it. The head then increases the probability of that attended next token, effectively predicting that the current sequence will continue like the previous matched sequence. For example, if the input contains the sequence "...the cat sat on the mat. The cat...," the induction head matches the second "cat" token to the first one, attends to "sat" from the first sequence, and increases the likelihood of outputting "sat" again. This allows Transformer models to repeat sequences and generalize patterns.

Importantly, the computations performed by induction head circuits do not rely on bigram statistics memorized from the training data; rather, they operate over abstract patterns in the input sequence (prompt), even if the latter does not contain familiar strings. As such, induction head circuitry can be seen as an instance of what Shea (2023) called non-content-specific computations in neural networks: computations that apply the same procedure or algorithm irrespective of the specific content represented at input and output. Specifically, when the network is processing a sequence `[A][B]...[A]`, the induction head circuit can be characterized algorithmically as storing the value of the first `[A]` token in a specific subspace of the residual stream – which we may call the `previous_token` subspace – at the position of the `[B]` token. Importantly, this operation occurs whatever that specific value (content) of `[A]` might be. Functionally, the `previous_token` subspace in which this information is stored operates rather like a variable in a classical symbolic program. Its value is accessed at the next layer by the second part of the induction head circuitry, which reads the content of the `previous_token` subspace in the residual stream at the position of the `[B]` (fig. 4). This is somewhat analogous to the indirect addressing mechanism that enables variable binding in classical systems: an attention head stores information about the token that precedes another token in a dedicated subspace of the residual stream (what we called the `previous_token` subspace), such that a distinct attention head in a later layer may retrieve it for downstream processing. This kind of circuit satisfies the distinction between storage and use that is crucial to computation over bound variables (Smolensky 1988, Gallistel & King 2011).[6]

Circuits using induction heads may be an important building block for the advanced in-context learning abilities exhibited by LLMs. Even in toy models, for instance, if the context contains "...the cat sat on the mat. The dog...", an induction head can generalize that "dog" signals repeating the sequence, despite never seeing "dog" next to "sat" during training. Research shows that Transformer models excel at discovering complex abstract patterns in context and extrapolating from them Mirchandani et al. (2023).

2.5.2. Case study 2: Modular addition

Mechanistic interpretability is not just helpful to investigate how trained neural networks process information through algorithmic circuits, but also how they *learn* such algorithms during the course

[6] For a discussion of induction heads circuitry as implementing a form of variable binding, and the implications of this view for the debate about compositionality in connectionist models, see Millière (forthcoming).

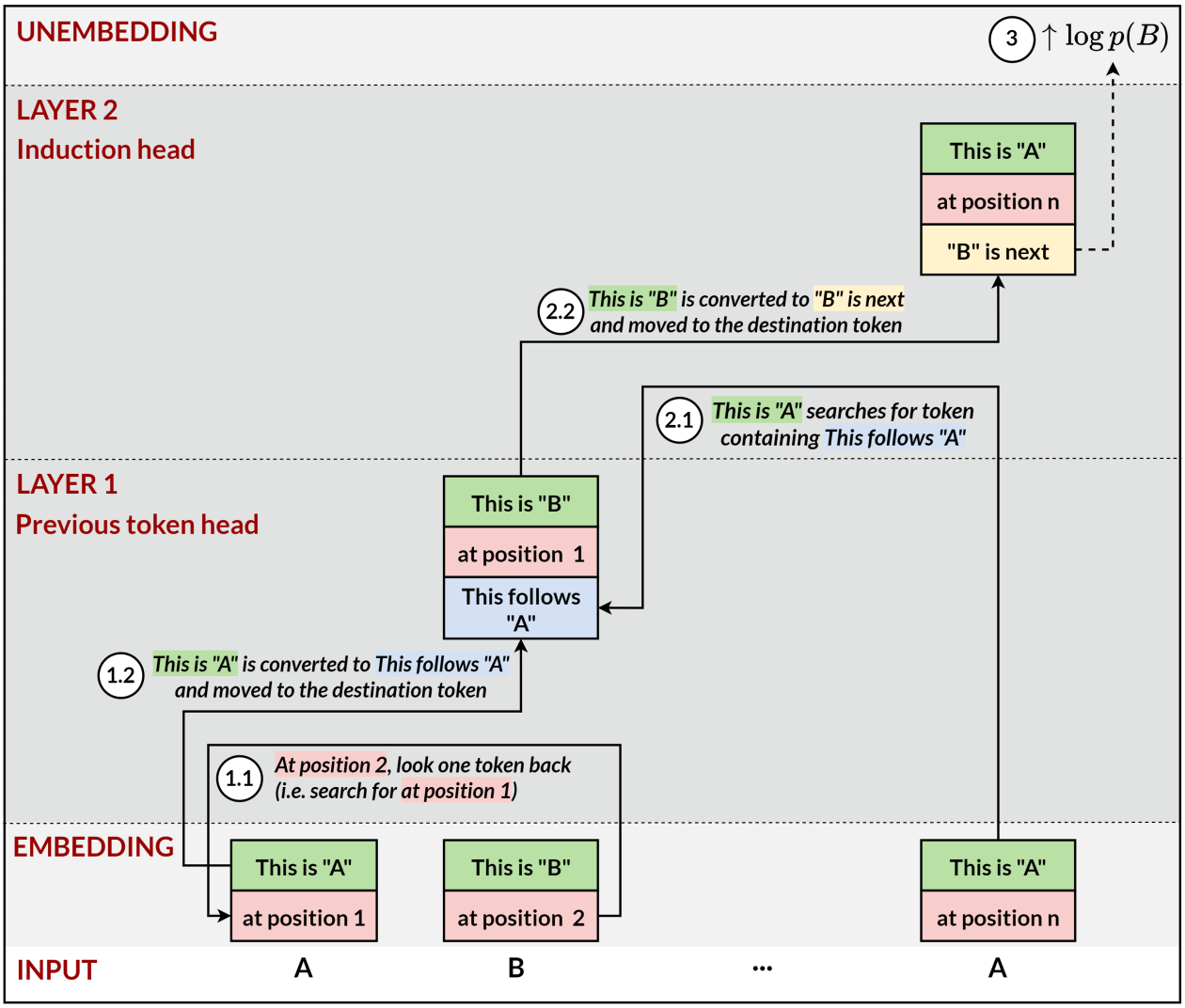

Figure 4 | **A schematic illustration of the induction head circuit in a two-layer Transformer model.** At the embedding stage, each token from the input sequence is encoded as a vector, together with information about its position in the sequence. The first layer contains an attention head – known as the *previous token head* – that acquired a specialized function during training. When processing token `[B]` at position 2, the previous token head does the following: (1.1) it attends to the previous token at position 1; (1.2) It writes the identity of this preceding token to a dedicated subspace of the residual stream at the current position (position 2), effectively storing the information "the token before me is `[A]`". Layer 2 contains another specialized attention head known as the *induction head*. When processing the second instance of `[A]` at position n, the induction head does the following: (2.1) it queries the residual stream for information in the 'previous token' subspace matching the current token's identity; (2.2) having located this previous token information in the residual stream at position 2, it retrieves the identity of the token at that position (`[B]`), then writes this identity to a dedicated subspace of the residual stream at the current position (position n), effectively storing the information "predict that the next token will be `[B]`". (3) The unembedding layer maps information in the 'next token' subspace at position n to an increased logit for `[B]` at position $n+1$, which translates to an increased log likelihood of `[B]` being predicted as the next token.

of training. Studying the training dynamics of neural networks is important, because it can provide

insights into learning transition phases that are highly relevant to ongoing debates about the capacities of LLMs. Thus, Nanda et al. (2022) investigated the puzzling phenomenon of grokking, where neural networks trained on algorithmic tasks with regularization initially overfit to the training data but later suddenly generalize after many training steps. As a case study, the authors trained small Transformer models on modular addition tasks. They find these networks exhibit grokking, initially overfitting but later learning to generalize.

To understand this phenomenon, they reverse engineered the mechanisms learned by these networks using techniques from mechanistic interpretability. That found that the network learns to perform modular addition tasks by mapping inputs onto rotations in the plane and composing those rotations using trigonometric identities (fig. 5). This clever algorithm, dubbed 'Fourier multiplication,' allows the network to perform addition modulo the prime.

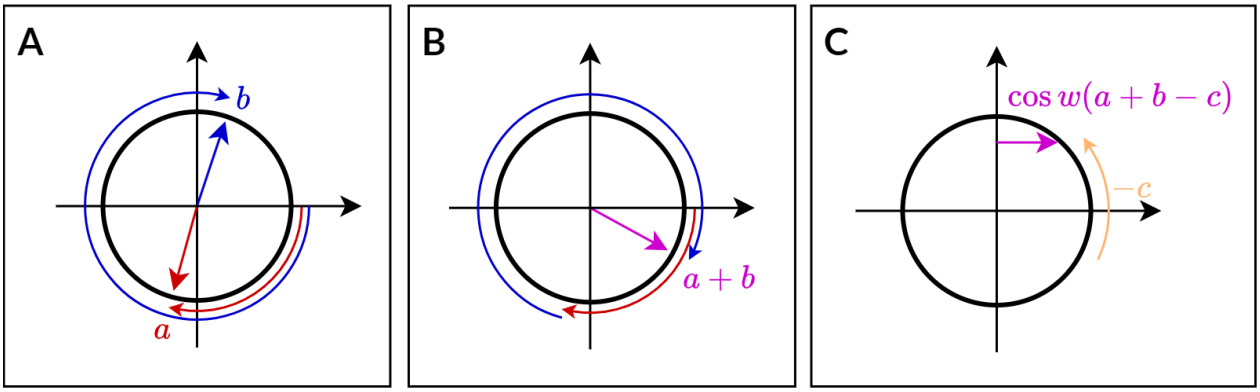

Figure 5 | **A learned algorithm for modular addition** (figure adapted from Nanda et al. (2022)). **A. Embedding projection.** Given two input numbers a and b in the modular addition $a + b \equiv c \mod P$, the model uses its embedding matrix to project each number onto a corresponding rotation around the unit circle. The embedding matrix essentially memorizes a mapping between each possible input number and a specific rotation amount, converting the numbers into geometric representations. **B. Rotation composition.** The model composes the two rotations generated for a and b. This step effectively adding the two rotation amounts together, resulting in a new, single rotation that represents the sum $a + b$ in modular arithmetic. In modular arithmetic, numbers 'wrap around' after exceeding the modulus P, so if $a + b$ is greater than P, the resulting rotation will correspond to $a + b \mod P$, which is the remainder when $a + b$ is divided by P. **C. Output decoding.** To produce the output logits (raw scores used for next-token prediction), the model considers each possible result c (ranging from 0 to $P - 1$) and performs a reverse rotation by $-c$. This step essentially checks, for each c, whether undoing the rotation by c results in a rotation that matches the one representing $a + b \mod P$. The output c that produces the rotation most closely matching the $a + b \mod P$ rotation is assigned the highest logit. This works because the trained model ensures that the correct c satisfying $a + b \equiv c \mod P$ will undo the rotation by exactly the right amount to point back to the $a + b \mod P$ rotation. The trigonometric functions cosine and sine are used to implement these rotations and achieve the desired result mathematically using angle addition identities, but conceptually, the algorithm is based on representing numbers as rotations and composing these rotations together.

Using this understanding, Nanda et al. defined new progress metrics allowing them to study the training dynamics of the models as they learn to perform modular addition algorithmically. They identified three distinct learning phases, each marked by continuous progress on certain metrics:

1. *Memorization phase*: In the early part of training, the models fit to the training data by simply memorizing input-output pairs. Performance on the test set remains low while performance on the training set increases rapidly, indicating the models are overfitting.
2. *Circuit formation phase*: After memorization, there is a transition period where the models internalize the algorithm for modular addition using trigonometric identities and rotations (the 'Fourier multiplication' circuit). However, performance on the test set remains low, implying memorization components still persist.
3. *Cleanup phase* Finally, weight decay drives the removal of the initial memorization components. Performance on the test set abruptly improves to match performance on the training set at the end of this phase, corresponding to the 'grokking' transition in generalization capability.

The entire phenomenon is thus not a sudden onset of generalization ability, but rather a gradual amplification of structured mechanisms encoded in the weights, followed by pruning of unnecessary components. Importantly, these findings speak against the skeptical view of LLMs discussed in Part I (Millière & Buckner 2024), according to which they are analogous to giant lookup tables with memorized input-output pairs. While the initial training phase does support that view, subsequent phase transitions show that Transformer models are perfectly capable to learn general, rule-like algorithms to solve tasks, including tasks as rigid and well-defined as arithmetic problems.

2.5.3. Case study 3: World models

As we discussed in the complement paper (Millière & Buckner 2024), there is an ongoing debate about whether LLMs possess world models. While behavioral evaluations are generally not helpful to settle this debate, mechanistic interpretability shows promise to uncover what the LLMs and related Transformer models represent internally.

To investigate whether Transformer models trained to predict sequences can learn interpretable world representations rather than mere surface statistics, Li et al. (2023) focus on the board game Othello as a simplified yet non-trivial domain. They trained a GPT variant (Othello-GPT) to sequentially predict tokens representing Othello board positions. The only inputs to the model are move sequences derived from game transcript; no explicit game rules or board structure are provided. After training on championship games and synthetic games, Othello-GPT recommends legal moves with high accuracy, suggesting it has learned more than surface statistics. Furthermore, nonlinear probes reliably predict board states from model activations, implying that a nonlinear representation of the board state had emerged during training. To validate the probes' accuracy, the authors performed intervention experiments that modify Othello-GPT's internal activations to reflect altered board states. The model's subsequent move predictions change accordingly, confirming the causal role of these latent board state representations.

In a follow-up study, Nanda et al. (2023) discovered that rather than representing the board state (e.g., representing each board tile as `black`, `white`, or `empty`), Othello-GPT actually encodes tiles relative to the current player (as `player`, `opponent`, or `empty`). By re-orienting probes to classify this player-centric representation, Nanda et al. demonstrated that the board state is in fact *linearly* encoded with high accuracy in the network, contrary to Li et al. (2023)'s claim that the board state is only encoded non-linearly (fig. 6). They further demonstrated behavioral control by conducting simple vector arithmetic interventions to alter the model's encoding of board states and change predictions accordingly. Hazineh et al. (2023) found similar evidence that information about board state is encoded in a simple, linear way in the deeper layers of Othello-GPT models. Like Nanda et al. (2023), they decoded a representation corresponding to tiles marked `player`, `opponent`, or `empty`, which aligns well with the role of the model in alternating between playing as white or black. To test whether these internal representations play a causal role in the model's predictions, they also

intervened by manipulating activations to trick the model about the state of the board. Through visualizing effects on predicted logits and comparing distributional similarity of logit outputs, they demonstrated layers in which this internal representation steers next-move predictions. The internal representation appears fully developed and utilized in middle layers of deeper models, while shallow models fail to use the representation causally.

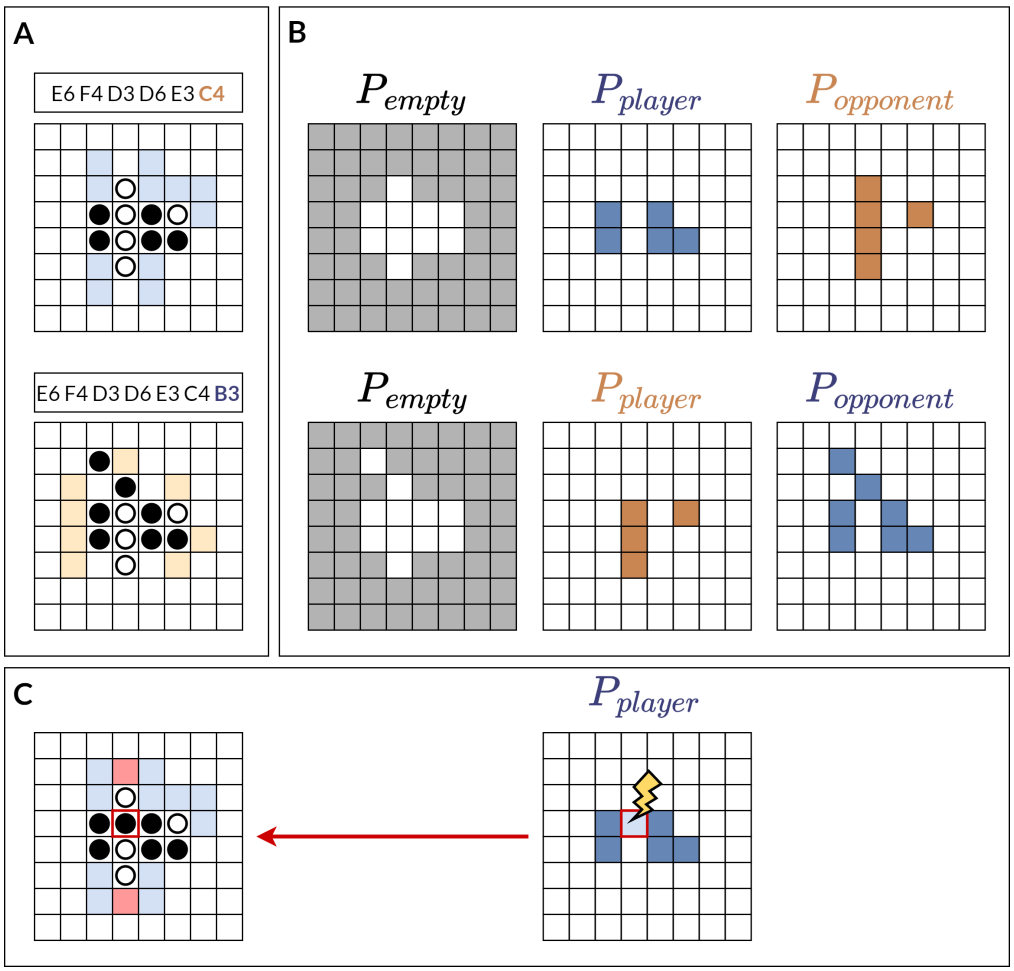

Figure 6 | **Emergent representations of the board state in Othello-GPT** (adapted from Nanda et al. (2023)). The model represents board states relative to the current player. **A.** The ground truth board states at two consecutive time steps. Colored tiles show legal moves for the current player (light blue for BLACK and light orange for WHITE). **B.** The board states at the same two time steps decoded by linear probes. The probes are trained to classify the board relative to the current player: `empty` for empty tiles, `player` for tiles occupied by the current player, and `opponent` for tiles occupied by the current player's opponent. Note that the `player` and `opponent` colors are flipped between the two time steps, as the current player changes. **C.** We can intervene on the model's internal board state representation by pushing an empty tile's representation in the direction of the `player` vector, to make the model represent that tile as occupied by the current player. This simple linear intervention is sufficient to alter Othello-GPT's move predictions (red tiles on the left), demonstrating that the linearly represented board state causally determines the model's outputs.

These findings align with the *linear representation hypothesis*, according to which high-level concepts or features are represented linearly as directions in a neural network's activation space. This hypothesis is often taken to be central to the agenda of mechanistic interpretability (Elhage et al.

2022). Indeed, if features correspond to directions, we can in principle extract and understand them by projecting activations onto those directions. This should allow us to decompose the high-dimensional activation space of the network into interpretable components. We can also check whether the weights of downstream circuits align with a feature vector and thus conclude that the model is using this feature. Linear directions provide a global foothold across the whole model to reason about a specific feature. Furthermore, manipulating model behavior via causal intervention becomes significantly simpler when features are linear directions. We can predict model behavior under counterfactual input features by pushing model activations along feature directions. In the case of Othello-GPT, we can linearly decode information about the state of the board from activations – namely, which tiles are occupied by the current player's pieces as opposed to the opponent's pieces, and which tiles are empty. The fact that this information is linearly decodable in a given layer makes it easy to manipulate with simple vector arithmetic by pushing activations along feature directions. It should be noted that the linear representation hypothesis can be spelled out in slightly different ways, depending on how one technically defines what it means for a concept or feature to be linearly represented in the network (Park, Choe & Veitch 2023). In addition, there is room for disagreement about the theoretical grounding of this hypothesis. While linear decodability does make interpretability easier, this does not entail that complex neural networks do not also encode information non-linearly and cannot make sure of such information in downstream processing.

In Part I, we defined world models in the context of LLMs as internal representations of the world that allows them to parse and generate language that is consistent with real-world knowledge and dynamics (Millière & Buckner 2024). Studies on Othello-GPT provide tentative evidence that Transformer models can acquire world models in this sense, at least in toy domains.[7] While Othello-GPT is not a language model properly speaking, it generates legal game moves in written form, and interventionist experiments reveal that it generates such moves on the basis of linearly decodable representations of the board state. To what extent can we generalize these findings to actual LLMs? This is a challenging question. A reasonable assumption is that Othello-GPT acquires a world model – an internal representation of the game world – because this is useful to improve its predictions of legal moves past a certain threshold. However, neural networks are notorious in their ability to achieve high accuracy scores on problems by mastering surface statistics; they tend to learn shortcuts (e.g., shallow heuristics) to achieve good performance on their learning objective until they hit a bottleneck. Furthermore, acquiring world models about the real world, rather than an extremely simple game world, is presumably costly in terms of training dynamics; it would require a substantial reorganization of the model's internal structure, akin to the phase transition undergone by Nanda et al. (2022)'s model to 'grok' modular addition, albeit on a much broader scale.

It may well be that LLMs do undergo such phase transitions during training, and acquire representations akin to world models at least in some limited domains. This might be required to unlock certain abilities such as commonsense reasoning about intuitive physics, which in turn would further reduce the loss (i.e., improve next-token prediction performance) in certain contexts. For this to happen, two conditions must presumably be satisfied: (a) the text-based training data should provide enough information to induce the relevant components of world models; (b) the learning pressure to reduce loss on the next-token prediction objective should be sufficient to push the model to acquire such representations given long enough training trajectories. Neither condition is antecedently guaranteed, and there is currently limited evidence that they apply to most existing LLMs. In particular, interventionist evidence regarding the most capable LLMs such as GPT-4, whose behavior on certain tasks is most

[7] For example, similar results have been found by studying Chess-GPT, a Transformer model trained on chess moves (Karvonen 2024).

consistent with the acquisition of world models, is still sorely lacking.[8] As far as behavioral evidence is concerned, lingering failure modes on out-of-distribution tasks cast doubt on the hypothesis that current LLMs acquire sophisticated and robust world models (McCoy et al. 2023, Yildirim & Paul 2023).

There are nonetheless ongoing efforts to assess the existence of world models in LLMs using the tools of mechanistic interpretability. For example, Li et al. (2021) used probing and intervention methods on models fine-tuned on the Alchemy and TextWorld datasets. Alchemy contains sequences of instructions for manipulating colored liquids in beakers (i.e., instructions for performing fictional chemical experiments and their outcomes), while TextWorld consists of textual transcripts of navigation in simulated worlds. Li et al. designed probes to test if the models' contextual token embeddings encode and track the state of entities mentioned in the discourse. For example, in Alchemy the probe tries to determine if a representation encodes that a beaker is empty after its contents are drained. Li et al. used these probes to test whether intervening on the decoded entity representations would change model behavior. Specifically, they constructed two discourses x_1 and x_2 that describe draining liquid from two different beakers, b_1 and b_2, resulting in one empty beaker per discourse. After encoding each discourse, they created a synthetic representation C_{mix} by taking the encoding C_1 of discourse x_1 and replacing the vector representations corresponding to the initial description of beaker b_2 with those from C_2. Although C_{mix} does not correspond to any real textual input, it implicitly represents a situation in which both b_1 and b_2 are empty. When generating text conditioned on C_{mix}, they found that the model generates instructions that are more often consistent with both beakers being empty compared to generating from C_1 or C_2 alone. However, the generated instructions from C_{mix} are still not always fully consistent with the implicit state, suggesting the induced representation is approximate rather than perfect. By editing the entity representations to model a new state not seen in the actual training data or prompt, and observing changes in the model's outputs that are often consistent with this new state, Li et al. provided tentative evidence that the model can induce an approximate implicit representation of the state of discourse entities purely from text. Note that producing text that is consistent with the hypothesis-driven manipulation of the LLMs activation space is not something that the system was trained to do. Furthermore, the method was found to be robust against 'sanity checks' to confirm that the intervention itself is not wholly responsible for injecting appropriate information into the system, such as finding negative results when attempting the intervention method on similarly-complex Transformer architecture with randomized weights. However, the generality of these findings remains uncertain, as the experiments focused on narrow domains with simple objects and dynamics. More systematic testing would be needed to determine how well they generalize to more complex and open-ended settings.

2.6. Interpretability and causal abstraction

The project of mechanistic interpretability can be seen through the lens of causal abstraction (Geiger et al. 2021). Causal abstraction is a theoretical framework that aims to provide an interpretable high-level causal explanation of the behavior of a complex system, such as a neural network, that is consistent with the low-level causal structure of that system.

The core intuition is that the variables of a low-level causal model can be clustered into sets and aligned with the variables of a high-level causal model. The high-level model provides an abstract characterization of the low-level model if aligned high-level and low-level variables have equivalent causal roles and information content. This equivalence is experimentally verified with interventions

[8] This is partly due to the fact that the weights of state-of-the-art LLMs are not publicly released, which precludes interpretability research from anyone other than researchers in the team that trains them. We will come back to this issue in section 3.3.

on both levels that produce the same counterfactual behavior. More formally, an alignment between a low-level model $\mathcal{L}$ and high-level model $\mathcal{H}$ is a partition of the low-level variables into sets, with each set aligned to a high-level variable. A translation function maps low-level variable values to high-level values (Geiger et al. 2023). The alignment is causally consistent if, for any low-level intervention that has a corresponding high-level intervention under the translation function, intervening on both models produces equivalent results after translation. If an alignment is causally consistent, then $\mathcal{H}$ is said to be a constructive causal abstraction of $\mathcal{L}$.

Any program or algorithm can be represented as a causal model (Icard 2017). In the context of interpretability research in deep learning, the low-level causal model is a neural network, containing many interconnected nodes and weights that determine its function. The high-level causal model proposed by mechanistic interpretability researchers offer hypotheses about the abstract computations and algorithms implemented by the network. Aligning groups of neurons and weights to single variables in the high-level model allows interpreting their collective causal role. The causal consistency condition ensures the high-level causal model faithfully captures the causal mechanisms embodied by the low-level neural network model. Causal abstraction thus enables the development of human-interpretable high-level causal models that accurately explain the reasoning and computations inside opaque neural networks.

A key method for assessing causal abstraction is interchange interventions, where neural representations created for one input are swapped into the model when it is processing another input. If the low-level neural model and high-level algorithm have the same counterfactual behavior under aligned interchange interventions, that provides evidence that the alignment witnesses a causal abstraction. Many of the methods described above, such as activation patching, can be used as interchange interventions. In fact, iterative nullspace projection can also be formalized within a causal abstraction framework (Geiger et al. 2023). Thus, causal abstraction is a useful framework to unify interventionist approaches to interpretability.

In practice, the project of mechanistic interpretability with large neural networks like LLMs can be seen as aiming for *approximate* causal abstraction, which relaxes the criteria for alignment (Beckers et al. 2020). The degree of abstraction can be quantified as the proportion of aligned interchange interventions that have equivalent effects in the high-level causal model and low-level causal model or target neural network (Geiger et al. 2023). When interchange intervention accuracy is 100%, the high-level model exactly abstracts the neural network. Otherwise, the high-level model approximately abstracts the neural network approximately, to the degree quantified by interchange intervention accuracy. Approximate abstraction can provide an interpretable high-level explanation that still reflects the causal structure of the target neural network with a degree of faithfulness suitable for genuine explanation.

2.7. Biological plausibility of decoded computations

Even if the preceding evidence regarding the complex and relevant functional roles attributed to LLM activation vectors are accepted, readers may be frustrated at the Transformer architecture's apparent lack of biological inspiration or plausibility. Whereas other deep learning architectures like DCNNs for computer vision were substantially inspired by biological investigations into cortical tissues and have since even been further aligned with data from functional neuroscience (Buckner 2019), Transformers arose almost independently of the neuroscience of language processing. Moreover, the key-query-value division on which self-attention depends was inspired by an analogy to database theory in computer science rather than theory in psychology or cognitive science. What, we might wonder, could possibly correspond to the mathematical operations of self-attention in the brain, even at a gross functional grain of analysis?

Nevertheless, numerous imaging and alignment studies have suggested that some families of Transformers are also excellent predictors of activation patterns in human language processing areas, such as Schrimpf et al. (2021), Caucheteux et al. (2021). Such alignment is defended even by researchers who concede that state-of-the-art Transformers like GPT-4 nevertheless lack substantial amounts of grounded world knowledge and other kinds of understanding often attributed to language comprehension, such as of complex semantic intentions or discourse representations. This has led some researchers to argue that such alignment demonstrates that left temporal areas of the brain associated with linguistic processing may play a limited word prediction role, arguing for a narrower construal of the language faculty than has traditionally attributed to such areas in cognitive neuroscience (Mahowald et al. 2023). Others have been led to draw even more ambitious comparisons between the architecture of self-attention and the brain; for example, Whittington et al. (2022) have argued that operations hypothesized to be implemented by medial temporal lobe tissues in computational neuroscience – specifically, place and grid cell coding between the entorhinal cortex and hippocampus – can be seen as mathematically equivalent to self-attention operations in Transformers. Whether this abstract mathematical equivalence supports closer mechanistic alignment between Transformers and human neural tissues remains to be seen, but may yet be seen to soften some of the initial skepticism that such alignment is outright implausible.

3. Newer philosophical questions

The previous section summarized some of the strongest mechanistic evidence to date that current-generation LLMs can, at least in principle, acquire 'world models' – structure-preserving representations of the state of the world described in text inputs that are causally efficacious on the model's ability to generate text outputs consistent with that world state. Overall, this evidence weighs against what we characterized in Part I as our null hypothesis, the claim that LLMs can be adequately described as giant lookup tables that merely retrieve sequences memorized during training. Nevertheless, traditional LLMs limited to next-token prediction over linguistic inputs have significant limitations that preclude naive comparisons to human or animal cognition. Research on generative modeling is fast-paced, however. Newer approaches have already moved past traditional LLMs, enhancing their architecture with multimodal capabilities and/or modular designs in which the text-generating aspect is simply one component of a more complex system. In this section, we address newer philosophical questions that may arise from these recent developments in LLM research. We also explore more difficult questions raised by LLM's unprecedented abilities (such as whether they meet tentative criteria for consciousness), as well as new worries about their scientific legitimacy precipitated by their unprecedented scale and the secrecy under which they are developed.

3.1. LLMs and modular architectures

LLMs exhibit impressive performance on traditional natural language processing tasks, where they have increasingly superseded task-specific models. However, there is only so much one can expect from a system exclusively trained to predict sequences of linguistic tokens. One promising idea to expand the capacities of LLMs is to augment or integrate them with components subserving other basic faculties – such as perception, imagination, and planning. There are two ways to execute this vision. The first involves modifying the Transformer-based architecture of traditional LLMs to allow them to process inputs in multiple modalities (e.g., text and images). The second involves incorporating LLMs as modules within a composite system, which may include multiple neural networks as well as classical symbolic algorithms. Each of these strategies has their own benefits and tradeoffs, but they are not mutually exclusive; a vision-language model designed to process images and text can itself be integrated with other module.

Language modeling plays a key role in these experiments. When Transformer architectures are augmented to receive visual input in addition to linguistic input, for example, their abilities to parse images is critically informed by linguistic concepts and dependencies induced from next-token prediction. By contrast, when traditional text-only LLMs are integrated within an ensemble of models, language acts as a common ground or 'universal API' for these models to communicate with each other without sharing differentiable parameters (Zeng et al. 2022).[9] In either case, language modeling forms the backbone of more capable systems.

Efforts to move beyond text-only LLMs have converged in around two trends in deep learning research: multimodality and so-called 'agent' systems. In the rest of this section, we will discuss each of these trends and the philosophical questions they raise.

3.1.1. Multimodality

While the Transformer architecture was initially applied to language modeling tasks, it has been adapted and expanded to tackle other modalities. For example, Vision Transformers can process visual inputs through a clever trick that involves splitting images into 'patches' that can be fed to the model as sequential tokens, as one would with linguistic tokens (Dosovitskiy et al. 2021). Subsequent research efforts have explored different ways to hybridize the Vision Transformer architecture with other architectures to leverage their respective strengths.

In the Taming Transformers architecture for image generation (Esser et al. 2021), the authors created a modular architecture in which raw images were first transformed into abstract feature maps by a deep convolutional neural network; those abstracted feature maps were then passed to a Transformer for encoding, and finally passed to a generative deep neural network for image reconstruction. The authors of this paper argue that their system can combine the ability of deep convolutional neural networks to extract abstract local features and the ability of Transformers to recognize long-distance compositional dependencies in inputs. As they put it,

> Our key insight to obtain an effective and expressive model is that, taken together, convolutional and Transformer architectures can model the compositional nature of our visual world: We use a convolutional approach to efficiently learn a codebook of context-rich visual parts and, subsequently, learn a model of their global compositions. The long-range interactions within these compositions require an expressive Transformer architecture to model distributions over their constituent visual parts. (Esser et al. 2021, p. 12874)

This combination of strengths allowed them to achieve significant performance gains at the time over other generative architectures like GANs, especially regarding holistic and relational aspects of an image's composition. Notably, the transformer's strengths in these regards can help neural network models address some classical criticisms of the previous generation of artificial neural networks, as noted in Part I (Millière & Buckner 2024); such modular networks demonstrate that these benefits can be provided to other domains like image labeling and generation as well.

An entire generation of multimodal models has been based on the insight that representing linguistic and visual information in a joint latent space is helpful to parse the structure of images. For example, OpenAI's CLIP – which stands for Contrastive Language–Image Pre-training – learns from a

[9] An API (Application Programming Interface) is a set of rules and protocols that allows different software components to communicate and interact with each other. By analogy, language can serve as a 'universal API' for generative models by providing a common interface to exchange information without being part of the same neural network.

massive dataset comprising millions of image-text pairs, enabling it to predict the most relevant text description for a given image without being explicitly optimized for this task (Radford et al. 2021). CLIP has two main components: an image encoder and a text encoder. These encoders transform images and text into a shared high-dimensional space where linguistic and visual representations can be compared directly. The image encoder is typically a modified version of a Vision Transformer that encodes images into vectors, and the text encoder is also a Transformer model similar to LLMs that encodes text snippets into vectors in the very same vector space. CLIP learns these vector representations through *contrastive learning*, which involves contrasting similar and dissimilar pairs of images and captions. Each training step involves a batch of image-caption pairs that are encoded into vectors by their respective encoders; CLIP's contrastive learning objective consists in maximizing the similarity of image vectors with the text vectors corresponding to their matching captions, while minimizing the similarity of image vectors with the text vectors corresponding to non-matching captions. After training, CLIP can parse and categorize images it has never seen before. For example, it can accurately identify objects, actions, or scenes in new images based on the learned associations between images and text during training. Beyond simple classification, CLIP can be used to generate new captions for images or find images that match a given caption.

CLIP was a stepping stone towards multimodal models that descend from LLMs, known as vision-language models (VLMs), which can receive both images and text as input and generate text like as LLMs (Alayrac et al. 2022). VLMs can answer question about images, and the best of them demonstrate a sophisticated ability to parse visual content. For example, GPT-4V, the VLM version of GPT-4, achieves strong performance on image captioning and visual question answering benchmarks, and shows some ability for commonsense reasoning about visual inputs (OpenAI 2023*b*, Yang, Li, Lin, Wang, Lin, Liu & Wang 2023, Wu, Wang, Yang, Zheng, Zhang, Zhao & Qin 2023).

Systems that can generate images from text description, like Stable Diffusion (Rombach et al. 2022) or DALL-E (Ramesh et al. 2022), also build on the same technical foundation. CLIP's text encoder plays a critical role in these systems, by converting text descriptions into vectors that can be used to condition the image generation process. Images are generated by iteratively denoising latent representations – a process known as *latent denoising diffusion*. At each step of the denoising process, the model considers both the current state of the image being generated, and the semantic direction provided by the vector from CLIP's text encoder. This ensures that the generated images are not only visually coherent but also semantically aligned with the text prompt. In other words, cross-modal alignment between the linguistic and visual domains achieved through the encoding of text descriptions in a shared latent space is fundamental to image generation models' ability to generate outputs whose content and composition adhere to the semantic and syntactic structure of the input. In fact, text-space encodings can serve as the common medium for breakthrough results in a variety of other modalities, include text-to-audio, text-to-code, and text-to-action (for a review, see Gozalo-Brizuela & Garrido-Merchán 2023).

There is a direct lineage between the Transformer architecture of LLMs, the Transformer architecture of CLIP's text encoder, and both advanced VLMs like GPT-4V and advanced text-to-image generation models like DALL-E. Nevertheless, seemingly subtle differences in architecture and training objective matter greatly to the performance of these systems. For example, VLMs and image generation models based on CLIP's contrastive text encoder have been found to suffer from significant limitations in their ability to process the compositional structure of text prompts and images appropriately. In particular, these models often have difficulty correctly associating attributes with their corresponding objects when multiple objects are present in an image. They also struggle to accurately interpret spatial prepositions describing relative positions of objects, properly account for specified numbers of objects, and handle negation terms that exclude certain attributes or objects from a scene (Zhang &

Lewis 2023, Hsieh et al. 2023, Tong et al. 2023, Kamath et al. 2023*b*, Lewis et al. 2023, Murphy et al. 2024).

However, these failures should be interpreted carefully. They do not seem symptomatic of a general limitation of the Transformer architecture; as discussed in Part I, text-only Transformer-based LLMs are much better at compositional generalization Millière & Buckner (2024). Rather, the compositional failures observed in many multimodal models can be attributed in large part to limitations of the contrastive learning objective used to train their text encoders (Yuksekgonul et al. 2022, Kamath et al. 2023*a*). Contrastive learning trains the model to match images to captions in a way that does not require preserving detailed compositional and syntactic information from the text. In essence, text encoders trained in this way treat linguistic inputs like 'bags of words', discarding critical aspects of sentence structure and word order. In fact, recent work has shown that replacing the contrastively-trained text component of a multimodal model with one that has instead been pre-trained on image captioning can significantly improve the model's compositional abilities (Tschannen et al. 2023). Image captioning models can excel at understanding attributional and relational information in text compared to CLIP-style contrastive models – outperforming the latter by large margins on benchmarks that test sensitivity to word order, object attributes, and spatial relations. Importantly, these models all use a Transformer architecture. The key difference is in how the text component is trained, either with a contrastive objective that ignores detailed sentence structure, or an image captioning objective that preserves it. The fact that simply changing the training objective while holding the architecture type fixed leads to such dramatic differences in compositional performance provides compelling evidence that the Transformer itself is not the root issue. This is a useful cautionary tale about the interpretation of failure modes in deep learning.

3.1.2. 'Agent' systems

Influential theorists like Goyal & Bengio (2022) have speculated that integrating Transformer-based language models in composite systems can provide other benefits to other modules. For example, they speculate that common causal variables tend to correspond to words in natural language, and so a language model that biased an image classifying module or a reinforcement-trained agent module would be more likely to focus on causally-relevant features of its training set (which they summarize with the hypothesis "that semantic variables are also causal variables", Goyal & Bengio (2022, p. 14)). These variables include words referring to agents or subjects of sentences, actions (often captured by verbs), objects of agents' actions (often captured by noun phrases in the direct object position), and modalities or properties of objects (often captured by adjectives and adverbs). From this perspective, natural languages are a kind of cultural technology that help us focus on the most important aspects of our environment (cf. Clark 1998), and systems that draw on language models to label incoming sensor data and formulate plans of action would thus be more likely to focus on causally relevant and robust properties in the environment and less likely to focus on artifacts and chimeras.

Many such architectures are already being implemented by industry and academic research groups, often inspired by the old Vygotskyan hypothesis in cognitive science that simulated inner speech is a powerful medium of thought that reliably develops in humans between the ages of 3-6, when children begin to demonstrate distinctively human success on problems of causal, logical, and social inference (Vygotsky 1987, Carruthers 2002, Lupyan 2012, Colas et al. 2021). To give an early example of such a system, Google Robotics has implemented an 'inner monologue' agent (Huang et al. 2022) that integrates a scene descriptor, success detector, and planning module together with a human interlocutor in a closed feedback loop. This agent can use a language model to interpret instructions given to it by a human, label visual input from sensors, formulate internal plans in terms of a constrained action language, and verify the success or failure of plan components by

deploying a success detector on sensor input. Even more ambitious agent architectures deploying internal monologues are possible, such as using text encodings to drive internal simulations of expected perceptual output, which could then be used for long-term planning and counterfactual 'imagination' of possible scenarios – such as text-driven planning rollouts of the sort seen in DeepMind's Imagination-Augmented Agents (I2A) architecture (Racanière et al. 2017, Buckner 2023).

These developments can be seen as specific steps on a broader journey from narrowly language-focused Transformer models to multi-modal Transformer-based 'agents', which many theorists see as the natural next step in deep-learning-based artificial intelligence. According to this frame, LLMs were an important breakthrough in making DNNs more compositional and in particular allowing them to tackle language-scaffolded effects in thought and action, but they are only one component in a full agent architecture. Moreover, the standard context of deployment for LLMs – continual next-word completion – lacks the stability and impetus of rational agency. This paradigm requires massive, ecologically-invalid datasets or at least deliberately-harvested instruction-tuning or reinforcement learning, and standard LLMs cannot gather their own ongoing training data by interacting with the world directly. Those trying to build Transformer-based agents seek to address these shortcomings by embedding LLMs in broader architectures and decision loops that allow them to generate their own goals (so-called 'autotelic' agents, Colas et al. 2022), formulate plans to pursue them, verify their success using a variety of modalities, and learn from continual self-driven exploration of a world.

There are three different kinds of 'agent' systems that build on the success of standard LLMs: (a) language agents that make use of external databases or tools; (b) embodied models that control a robotic body from natural language instructions with an LLM; and (c) multimodal models trained end-to-end to process actions in addition to text (Fig. 7).

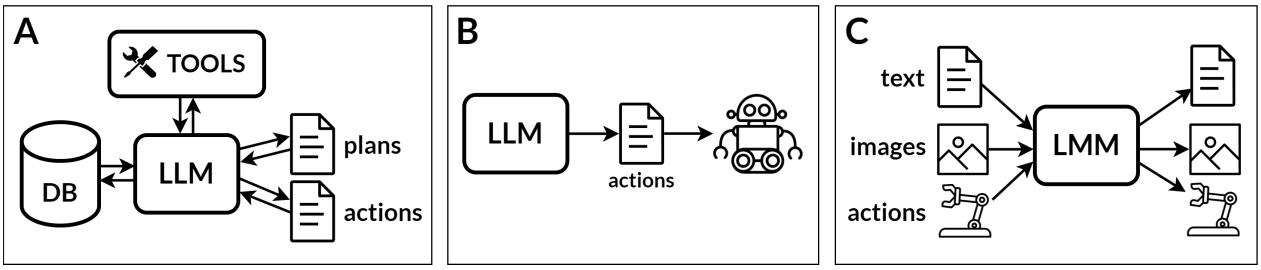

Figure 7 | **Three kinds of agent systems based on LLMs or multimodal models. A.** A modular language agent system compose of a LLM that can interacts with (i) an external database (DB) to store and retrieve long-term knowledge or 'memories,' (ii) text files to store and retrieve plans and actions, and (iii) external tools that can be used through function calls. **B.** A robotic system that makes use of a LLM to translate high-level natural language instructions into low-level policies to plan and act in the real world. **C.** A large multimodal model (LMM) that can not only receive and produce text, but also images and even action tokens to control a robotic body.

Language agents are systems with specialized modules orchestrated by calls a LLM (see Wang, Ma, Feng, Zhang, Yang, Zhang, Chen, Tang, Chen, Lin, Zhao, Wei & Wen 2023 for a review). These modules may include a memory component, a planning component, and an action component. The memory component extends the input context window of the LLM, which acts as a short-term memory buffer, with an external long-term storage (like a text file or a structured database). For example, Reflexion combines a sliding context window with persistent memory banks (Shinn et al. 2023). The LLM can write to and read from the external memory, as well as summarize multiple memories into high-level insights to inform planning and action. A planning component can structure the reasoning process to accomplish goals. For example, ReAct utilizes action-observation cycles to refine plans

based on environmental outcomes (Yao et al. 2023). Finally, an action component can interpret the agent's decisions into concrete outputs and behaviors.

A striking example of this approach is the 'generative agents' of Park, O'Brien, Cai, Morris, Liang & Bernstein (2023). The architecture has three main components: the memory stream stores a comprehensive natural language record of the agent's experiences, with a retrieval model combining relevance, recency and importance to surface the most pertinent records; 'reflection' synthesizes memories into higher-level inferences that guide behavior over time; and planning translates conclusions from reflection and the current environment into high-level action plans that are recursively decomposed into detailed behaviors. Park et al. demonstrate the effectiveness of this approach in a Sims-like sandbox game environment called Smallville, inhabited by 25 unique language agents. These agents exhibit emergent social behaviors like information diffusion, relationship formation, and coordination (e.g. for spontaneously organizing a Valentine's Day party), based solely on initial seed prompts. Another impressive example is Voyager, a language agent powered by GPT-4 that is able to continuously explore, develop skills, and make discoveries in the open-ended 3D world of Minecraft without any human intervention (Wang, Xie, Jiang, Mandlekar, Xiao, Zhu, Fan & Anandkumar 2023). The key components of Voyager consist of an automatic curriculum module that proposes a stream of progressively more complex tasks optimized for exploration; a skill library for developing and composing complex, reusable behaviors; and an iterative prompting mechanism that leverages execution feedback and self-verification to iteratively refine GPT-4's code generation until a task is successfully completed.

A concurrent trend consists in augmenting LLMs with external tools that can be manipulated through generated code or calls to APIs. One strategy consists in fine-tuning LLMs to enable them to learn in context when API calls are helpful in an entirely self-supervised manner, requiring only a handful of demonstrations per tool (Schick et al. 2023). Augmenting LLMs with external storage and tools effectively creates hybrid neurosymbolic systems that use language and/or code as a universal interface between connectionist and symbolic components. In turn, this goes a long way towards addressing some remaining limitations of pure LLMs; for example, using tools such as a code interpreter or Wolfram Alpha significantly improves the performance of LLMs on challenging math problems (Zhou, Wang, Lu, Shi, Luo, Qin, Lu, Jia, Song, Zhan & Li 2023, Davis & Aaronson 2023).

The second category of agent systems encompasses regular LLMs embedded in a system that include a physical robotic body roaming the real world. This approach combines the linguistic competence of LLMs with pre-trained robotic skills without requiring special modifications to the LLM component itself. For example, SayCan takes as input a high-level natural language instruction that describes a task for the robot to perform (Ahn et al. 2022). It then scores candidate low-level skills based on the probability that the skill's description is relevant for the high-level goal – as determined by the LLM component – and the probability from a learned value function that the skill can successfully execute in the current state. By multiplying these probabilities, the method selects skills that are both useful for the goal and possible to achieve. The process repeats by appending selected skills until termination. SayCan can be implemented in a robot with vision-based manipulation skills trained via behavioral cloning and reinforcement learning, to perform real-world long-horizon tasks in a complex environment such as a kitchen.

The third category of agent system consists in Transformer-based models trained to receive multimodal inputs (including text) and able to produce actions as outputs. Unlike text-only LLMs embedded in robotic systems, these models are trained end-to-end to generate action commands. DeepMind's Gato (Reed et al. 2022) is designed to be a 'generalist agent' that operates in a multi-modal, multi-task architecture on a generalist learning policy – which allows the same system to play Atari games, caption images, chat with humans, manipulate a real robot arm to stack blocks,

and much more. In a similar spirit, Zitkovich et al. (2023) propose a method for incorporating Transformer-based vision-language models directly into robotic manipulation policies. The key idea is to represent robotic actions as special text tokens that can be generated as part of the model's text output. The resulting model, RT-2, is pre-trained on image, text and action captioning datasets scraped from the internet, then fine-tuned on robotics datasets to output tokenized actions. The unified text-based output space of RT-2 is a key advantage over prior works, enabling it to successfully generalize to novel instructions after training.

The evolution of LLMs towards 'agent' systems capable of executing instructions, forming plans, and performing actions in the real world raises interesting philosophical questions. One pertains to the grounding problem: are embodied, LLM-based agents more likely to meet the requirements for referential grounding? Mollo & Millière (2023) provide reasons to doubt that this is the case for systems, like SayCan, that simply make calls to a distinct text-based LLM component – where the latter is not pre-trained or fine-tuned on world-involving tasks. However, systems trained end-to-end to generate actions like Gato or RT-2 might deserve a different treatment. As Transformer-based models expand beyond text-only training to process any information that can be serialized – including images, videos, joint torques, button presses, etc. – it becomes less obvious that they should be deemed intrinsically incapable of inducing normative world-involving functions, even if they are not explicitly fine-tuned from human feedback. These systems also raise obvious question about agency itself: is the term 'agent' (let alone 'autonomous agent') a misnomer when it comes to systems that do not fundamentally have intrinsic goals or motivations? The answer largely depends on how thick one's notion of agency is. LLM-based language agents are often described as generating, storing and retrieving goals, as well as breaking them down into sub-goals and individual actions. However, their behavior remains ultimately grounded in a human-written prompt. As such, they fail to meet the requirements for pure agential autonomy.

3.2. Consciousness

LLMs can eloquently converse about any topic sufficiently prevalent in their training data – including consciousness. Indeed, they are perfectly capable of generating compelling first-person reports that look like self-ascriptions of subjective experiences. In humans, such reports are generally taken at face value; while introspection can be unreliable (Irvine 2021), verbal reports typically provide (defeasible) evidence about someone's experience. This intuition breaks down when it comes to LLMs. No matter how convincing their fluent mimicry of experience reports might be, it cannot be taken as prima facie evidence that they are conscious. Imitating patterns of human language use is precisely what LLMs are trained to do. Nonetheless, if LLMs can acquire emergent capabilities in the process of learning from a next-token prediction objective, then we ought to wonder whether current or future systems with a similar architecture could, in principle, acquire the capacity for conscious experience. Like other debates about the putative cognitive capacities of LLMs, this issue cannot be settled merely through armchair speculation, but calls for careful empirical research informed by mature theories.

While consciousness may seem like a particularly high bar to clear for machines that still clearly lack some of the hallmarks of human intelligence, it is worth nothing that it need not correlate with sophisticated cognitive abilities. Indeed, there is a rather broad consensus that many non-human animal species that presumably lack such abilities have the capacity for consciousness – not only mammals (Seth et al. 2005), but more largely vertebrates including reptiles, birds, and fish (Cabanac et al. 2009, Merker 2005), and perhaps even invertebrates including mollusks and arthropods (Barron & Klein 2016, Godfrey-Smith 2021). Unlike animals, however, AI systems do not share a common evolutionary history or physical substrate with humans. As such, considerations about phylogeny or

similar neurophysiological features that are central to the discussion of animal consciousness do not straightforwardly apply to LLMs.

The hypothesis that artificial neural networks running on silicon chips could be conscious is premised upon computational functionalism – the view that what makes a physical system conscious is not its particular physical makeup, but whether it implements an appropriate set of computations (Chalmers 1995). If computational functionalism is true, biological brains and computers could in principle implement the same consciousness-enabling computations using different physical machinery. This assumption is not uncontroversial. Biological nervous systems exhibit complex dynamics involving global electrical patterns that modulate fine-grained neural signaling. Biochemical mechanisms and metabolic processes in the brain may impose specific constraints on the realization of conscious mental states (Cao 2022, Godfrey-Smith 2016). Artificial neural networks implemented on digital computing hardware bear little resemblance to biological nervous systems at this level of granularity, and it may be that the kind of silicon substrates used by modern computers cannot support consciousness even in principle.

If computational functionalism holds, however, then it should be possible in principle to identify the computational correlates of consciousness in biological brains, and assess whether similar computations are implemented in systems like LLMs. Given the lack of a complete and widely accepted scientific theory of consciousness, there is currently no definitive test or criteria to determine if an AI system is conscious. It is nonetheless possible to make qualified claims about the plausibility of consciousness in AI systems based on computational markers of biological consciousness derived from cognitive neuroscience research (Aru et al. 2023, Butlin et al. 2023, LeDoux et al. 2023).

Butlin et al. (2023) survey prominent neuroscientific theories of consciousness that are compatible with computational functionalism. From these theories, they derive a list of 'indicator properties' – features or mechanisms that the theories associate with consciousness. The more indicator properties an AI system exhibits, the more likely it is to be conscious. Prominent theories of consciousness include, among others: recurrent processing theory, which proposes that recurrence within neural networks distinguishes conscious from unconscious processing (Lamme 2006); global workspace theory, which claims that consciousness arises when information is broadcast to widespread networks from a limited-capacity workspace (Baars 1993, Dehaene & Naccache 2001); and higher-order theories, which hold that consciousness depends on higher-order representation of lower-level activity (Carruthers & Gennaro 2023, Brown et al. 2019).

The authors cite evidence that current DNN architectures, including Transformer-based LLMs, fail to satisfy key indicator properties of consciousness associated with these theories. For example, the global workspace theory of consciousness requires the existence of modules operating in parallel, feeding into and taking inputs from a global workspace with limited capacity to create an information bottleneck. It is tempting to view the residual stream of Transformer-based LLMs as loosely analogous to a global workspace that 'broadcasts' information to dedicated attention modules. However, the bottleneck requirement is arguably not satisfied: the residual stream has the same dimensionality from input to output, and attention heads work with much lower-dimensional spaces. In addition, there is no genuine recurrent processing in a single forward pass through the model: an attention head in one layer can only store information for downstream layers to process, as opposed to making it globally available to all other attention heads.

This example highlights the challenge of reaching strong conclusions about the computations that current model architectures may or may not implement to vindicate claims about whether they satisfy 'indicator properties' of consciousness. Attention heads in Transformers do compress information in subspaces of the residual stream that are globally available for downstream layers to read from, albeit only within the forward pass. However, some variations of autoregressive

Transformers enable recurrent processing that might, in principle, satisfy at least some of the core computational requirements identified by leading theories of consciousness (Giannou et al. 2023, Hutchins et al. 2022, Bulatov et al. 2022). Ultimately, as Butlin et al. (2023) note, most of the indicator properties extracted form these theories could likely be implemented in AI systems using current methods and architectural tweaks, although no existing system seems a strong candidate for consciousness in this respect. As with the assessment of other psychological capacities, more work is required to bridge low-level descriptions of algorithmic circuits in LLMs and high-level descriptions of their architectural features and behavior with intermediate-level descriptions of functional building blocks.

3.3. Secrecy and the reproducibility crisis

Much like the brains of humans and animals, LLMs today are large, opaque, and produce strikingly sophisticated behavior; this has led some researchers to suggest that we need a new 'science of machine behavior' to develop appropriate instruments and methods of analysis to study them (Rahwan et al. 2019). In developing these methods, we should be careful to import general lessons that were hard won in other sciences of behavior, such as human and comparative psychology. As mentioned in the companion paper, these sciences have from the beginning been fraught with biases like anthropomorphism and anthropofabulation, and as a result have hit numerous speed bumps in their history. For example, early comparative psychology relied too much on the method of anecdotes, which led to numerous dead ends and overinterpretation of behaviors which might have simply been statistical flukes (Thomas 1998). This led modern comparative psychology to emphasize the need for reproducible experimental conditions on model organisms that could be raised in controlled conditions for all labs. Several areas of social, comparative, and developmental psychology are today also facing 'replication crises', where well-known psychological effects and mechanisms that were thought to be well-confirmed and taught routinely in textbooks have been found to be based on statistical artifacts that were reified through questionable methodological practices, such as p-hacking or outright data fraud (Wiggins & Christopherson 2019, Frank et al. 2017). This has similarly led to calls for pre-registration of all experimental protocols, training stimuli, datasets, and analysis methods to be published alongside with experimental results, so that other researchers can confirm findings on their own and scrutinize experimental materials directly to explain any discrepancies (Beran 2018, Frank et al. 2017).

Unfortunately, research on LLMs seems set to repeat and even exacerbate these problematic practices from the history of psychology. Most of the highest-performing LLMs are the results of enormous investments in high-quality datasets for training, labor-intensive fine-tuning, and proprietary architecture tweaks. These datasets and training protocols are regarded as valuable trade secrets that are closely guarded by companies like Google and OpenAI who seek to control market share by outperforming their competitors. As a result, many of the most alluring behavioral results, such as those reported in Bubeck et al. (2023), are essentially anecdotes about systems which are unavailable to outside researchers (especially by interventionist methods) and by design irreproducible by other research groups. For commercial reasons, the results are frequently hyped on social media and in press releases, often beyond the scientific merit of the underlying findings. Even before the advent of LLMs, methodologically-focused researchers worried about the lack of reproducibility in deep learning research (Henderson et al. 2018), which has only been exacerbated in the age of LLMs. Most academic research has been conducted on models that are publicly available for academics such as BERT, which has led to the field being metonymically dubbed 'BERTology' (Rogers et al. 2020). Prominent researchers on LLMs have already called for disregarding analyses based on anecdotal evidence and urged for early adoption of best practices for reproducibility in architecture, data, training methods, and analyses (Frank 2023, Sellam et al. 2022). There have also been developments

of large, collaborative, open-access benchmarking methods such as BIG-bench to evaluate LLMs in a more controlled and objective manner (Srivastava et al. 2023). At the same time, empirical results may suggest that certain discontinuous gains in performance in LLMs can only be achieved at extremes of scale and computation power (Kaplan et al. 2020, Wei et al. 2022). If explainability and safety research is to address the largest systems which are likely to be deployed by large technology companies like OpenAI, Google, and Microsoft, researchers cannot naively hope that research focused on smaller open-access models will always generalize. Future researchers will have to learn how to strike a balance between the scientific need for reproducibility and the practical need to understand the latest achievements of large, proprietary, closed research efforts.

4. The status of LLMs as cognitive models

We started this two-part paper with a question: Are LLMs more than approximate lookup tables? In other words, do they merely memorize and regurgitate common patterns in their training data through content-specific transitions without inducing more sophisticated representations and computations? Our survey of behavioral and mechanistic evidence supports a broadly negative answer (with some qualifications). LLMs can and do induce complex mechanisms that enable them to perform challenging reasoning tasks better than any other domain-general computer program. As they get more capable, they tend to approach or match human performance and error patterns on many such tasks (Webb et al. 2023, Dasgupta et al. 2023, Han et al. 2024, Suri et al. 2024). This raises intriguing questions about the status of LLMs and their derivatives – including large multimodal models and modular architectures – as potential models of aspects of human cognition (McGrath et al. 2023, Millière forthcoming). We will conclude this overview by briefly addressing these questions in light of the preceding discussion.

4.1. The allure of 'alien intelligence'

A refrain that has recently appeared in many popular science articles, blogs, and social media posts holds that LLMs may reflect the discovery of an entirely new kind of 'alien' intelligence that is fundamentally unlike our own. This may be seen as a mere statement of fact – an optimistic appraisal that the linguistic behavior already exhibited by state-of-the-art LLMs is obviously intelligent, coupled with a pessimistic appraisal that the underlying processes producing this behavior are much at all like ours. With respect to most of the philosophical questions raised above, however, it reflects an almost complete dodge and a return to a naive behaviorism about intelligence by use of an eye-catching label. In any particular case, we need to ensure that before we accept the use of such a label as a descriptor of some scientifically-important LLM behavior that we have scientifically-respectable methods to assess that behavior along the important dimensions reviewed in Part I.

As Nickles (2020) points out, this language first began to appear as a descriptor of deep learning performance well before the current boom in LLMs, especially regarding new machine-learning-based methods to analyze data in high-complexity sciences like biology and fundamental physics. There, the language of 'alien' intelligence was conceptually tied to older philosophical dreams of rationalists like Descartes to create a science that was free from the shackles of human perspective and bias – an objective science free from the foibles of human 'powers and faculties', as Hume put it. The hope kindled by 'alien intelligence' here is that by automating the process of data collection and analysis and relaxing requirements on transparency and intelligibility, "nonhuman sensors and neutral algorithms can replace these human 'powers and faculties'" with alien agents that are wholly "data-driven and increasingly are able to construct their own models and search strategies". Indeed, there is some reason to think that breakthrough systems like AlphaFold, which have nearly solved the

highly-complex protein folding problem that eluded human microbiologists for more than a century, achieved their results precisely by discovering inscrutable features that are too complex for humans to understand (Buckner 2020, Kieval 2023). Whether one buys the talk of alien intelligence and inscrutable features in this context, however, the concept derives its utility from pragmatic scientific goals that humans largely accept – the ability to predict and perhaps control natural phenomena.

When this notion of 'alien intelligence' is applied to the linguistic behavior of LLMs, however, the standards relevant to assessing its utility are far less clear. We suggested in Part I that there are many reasons to be skeptical of purely behavioral criteria for intelligence when applied to products of modern LLMs, for they may in any instance be mere Blockheads that, like a student who has merely memorized the answer key, are simply regurgitating information contained in their training sets. A purely behavioral criteria for intelligence that makes no mention of learning efficiency or representational properties should also be rejected, for it gives us no way to say that two systems that display the same behavior differ in intelligence, even when they use very different procedures to achieve the same goal. We also reviewed reasons in section 2 above to be skeptical that implicit assumptions on which benchmarks or other measures are based could be trusted, without evidence of deeper mechanistic or algorithmic correspondences with human subjects. As such, we should resist the application of 'alien intelligence' to LLMs until this term can be explicated in a way that addresses these concerns and outlines a scientifically-responsible set of practices for evaluating internal performance differences amongst various LLMs.

4.2. A plausible middle ground

Summarizing the previous discussion leads to the following conclusions. Firstly, DNNs like Transformers have have an astounding capacity for memorization and learn from enormous training sets. As a result, any particular behavior they exhibit could be mere regurgitation or approximate retrieval, rather than demonstrating human-like processing or understanding of the task. This is what we characterized as our null hypothesis in Part I (Millière & Buckner 2024). Secondly, however, not all LLM behaviors can be explained by this null hypothesis. Interventionist methods suggest that, at least in some cases, their behavior is caused by robust representations of task variables.

Thirdly, these models do not yet generalize as well out of distribution as humans or some classical models. Like many other ANNs trained by gradient descent, LLMs exhibit varying degrees of abstraction in their generalization behavior, depending on the inductive biases of their neural network architecture and the exploitable informational structure of their training data. They may exhibit weak systematicity or islands of strongly systematic behavior, but not fully classical systematic generalization to very different task stimuli. While some 'non-content-specific' processing does occur in LLMs (Shea 2023), their behavior still falls short of the full consistency and generality exhibited by humans.

This limitation is likely due to the fact that most LLMs only model formal aspects of human linguistic competence (Mahowald et al. 2023). Their designers make little attempt to model a human-like cognitive architecture, which might include multiple semi-independent modules corresponding to domain-general psychological faculties that enable and maintain consistency in humans – such as perception, memory, imagination, attention, social cognition, metacognition, reflection, and long-term planning. Without these additional components, LLMs remain limited in their ability to match human-level systematicity across diverse contexts. The lack of such a comprehensive agent architecture probably also partly explains why LLMs learn so much less efficiently than humans, requiring orders of magnitude more training data before achieving comparable performance.

Considering available evidence, a middle-ground position emerges regarding the relevance of LLMs to human cognition. Current-generation LLMs may serve as inefficient but useful tools for reverse-

engineering partial models of the processes or algorithms that humans use to generate linguistic behavior (Zhou, Bradley, Littwin, Razin, Saremi, Susskind, Bengio & Nakkiran 2023). While it would be far-fetched to consider LLMs adequate computational models of the human mind, let alone human learning, they are not entirely irrelevant to human cognition and intelligence.

LLMs may capture partial models of human cognition by learning representations and computations similar to those used by humans, at least in specific domains. These learned representations and computations can be flexibly combined using operations on continuous vectors, enabling LLMs to generate novel behaviors. The resulting behaviors can reproduce not only the surface-level linguistic patterns of human language use but also, to a limited extent, the underlying systematicity and generalization ability characteristic of human cognition. It is worth noting that no previous AI technology has been able to achieve this level of fidelity to human linguistic and cognitive behavior. However, it is also important to acknowledge that LLMs often still fall short of human efficiency and flexibility when generalizing to novel stimuli.

To be clear, even this middle-ground position remains speculative. We have reviewed evidence from intervention methods suggesting that LLMs can represent syntactic and semantic properties of their utterances, but the degree of success achieved by current interventions is difficult to interpret as definitive. Weaker evidence from interchange interventions can often be obtained for alternative representational interpretations of networks, and these various interpretations may be mutually exclusive. Simply regarding the strongest interpretation as the 'correct' representational interpretation of a trained LLM requires further justification.

This issue reflects a more general fact about language and language use: philosophers of language and linguists have long noted that words and sentences in typical dialogues are inherently ambiguous and vague. It may be that only when sentences are produced and controlled by agents with stable viewpoints and communicative intentions do they acquire determinate meanings. Current generations of LLMs likely lack such agential stability, as readers can verify for themselves by observing the malleability of their responses. It is easy to prompt the current generation of chatbots to recant previously affirmed statements and defend inconsistent positions by simply suggesting that they are wrong, confused, or offensive. In short, while LLMs may accumulate information about the same syntactic and semantic properties as humans and even combine that information in flexible ways to create novel outputs, they nevertheless lack the agential stability required to imbue their utterances with determinate meanings that remain stable over time.

5. Conclusion

In this two-part paper, we have sought to provide a systematic overview of philosophical issues raised by the rapid progress of LLMs. In Part I, we discussed the significance of LLMs in relation to classical debates in the philosophy of artificial intelligence, mind and language. We argued that LLMs have made important headway on problems that challenged previous connectionist approaches, such as modeling compositional and systematic aspects of language use. However, we also highlighted limitations that preclude hasty comparisons between LLMs and human competence.

In Part II, we turned to newer philosophical questions raised by the current state of the art in language modeling research. We reviewed philosophically-grounded interventionist methods that aim to uncover the causal mechanisms underlying LLM performance, such as the existence of algorithmically meaningful internal representations and computations. This growing body of work suggests that state-of-the-art LLMs implement important aspects of the abstractness, systematicity, and generalizability of human cognitive processes, although they still fall short in terms of efficiency, completeness, and agency. We also discussed emerging trends in LLM research, including the

development of multimodal models and the integration of LLMs into broader 'agent' architectures. These approaches attempt to address some of the weaknesses of traditional LLMs by creating AI systems that can learn from self-exploration, maintain internal memories, and formulate plans based on their interactions with real or virtual environments. While it remains unclear whether current LLMs satisfy proposed computational markers of consciousness, these advancements open up the possibility of future systems exhibiting increasingly sophisticated forms of intelligence.

Finally, we proposed a nuanced perspective on the status of LLMs as partial models of human cognition. While current LLMs are far from full-fledged computational models of the human mind, carefully designed experiments informed by cognitive science and philosophy may provide genuine insights into some of the representations and algorithms that enable specific aspects of natural language use in humans. For example, training LLMs on ecologically valid data in developmentally plausible learning scenarios, rather than internet-scale corpora, could constrain hypotheses about the mechanisms underlying language acquisition and processing. However, it is crucial to stress the limitations of current LLM-based systems as models of human cognition as a whole. They lack the architectural components that enable the consistency and generality of human cognition across diverse domains. Moreover, while LLMs can flexibly recombine semantic and syntactic information to produce novel outputs, they plausibly lack the agential stability and communicative intentions to reliably anchor the meaning of their utterances. Nevertheless, the fidelity with which they can emulate specific aspects of human linguistic and cognitive behavior, when trained and evaluated using scientifically rigorous methods, is already far beyond previous AI approaches. This suggests that LLMs could serve as valuable tools for investigating targeted research questions in cognitive science, provided that their limitations are carefully considered and their performance is not overinterpreted.

Research on LLMs is a highly active and fast-paced endeavor at the intersection of artificial intelligence, cognitive science, and philosophy. Our overview suggests that this line of research is not 'hitting a wall' (Marcus 2022), but rather that its prospects as a tool to study distinctively human forms of cognition continue to grow. Philosophy can provide valuable conceptual clarity and theoretical guidance in interpreting the behavior and capabilities of LLMs. We hope that this two-part paper will not only bring attention to the philosophical significance of LLMs, but also convince researchers across disciplines – from computer science to cognitive science – of the value of engaging with philosophical perspectives in their work. At the same time, we have cautioned against the trend towards increasingly closed, irreproducible, and proprietary research on LLMs, which risks compromising the scientific integrity and social responsibility of this field. Sustaining genuine progress in understanding LLMs will require a commitment to open and interdisciplinary research practices. Only then can we hope to fully grasp the implications of LLMs and their successors for the long-term project of illuminating the nature of intelligence – both artificial and natural.[10]

Glossary

ablation Ablation refers to the process of removing or disabling parts of a neural network in order to study the impact on the network's performance. It allows researchers to determine which components of a network contribute most to its functionality. As a causal intervention method, it is somewhat rudimentary and has largely been superseded by more sophisticated techniques like interchange interventions. 7

addressable memory An addressable memory is a critical component of a classical computer architecture that enables a computing system to store and retrieve information. It consists of a set of memory locations, each of which has a unique address. The address acts as a 'handle' that allows

[10]We are grateful to Jim Garson and Charles Rathkopf for their comments on previous drafts of this work.

the computational machinery to access the contents of a specific memory location directly, without having to search through the entire memory. Adressable memory enables variable binding – the process of associating a specific value with a variable, so that the value can be retrieved and used in computations by referring to the variable. In an addressable memory, the address of a memory location serves as a variable, while the contents at that memory location represent the value bound to that variable. By using the address as a symbol for the variable, the computational machinery can retrieve the value associated with the variable (stored at that memory address), and can also update the value by writing a new value to the same memory location. This architecture allows for complex data structures and computations. Notably, it enables indirect addressing, where the value stored at a memory location is itself an address pointing to another memory location. Unlike classical computer architectures, DNNs like Transformer-based LLMs do not have explicit addresses and memory locations. Yet interpretability research suggests that they develop an analogous mechanism through the residual stream. In a Transformer, the residual stream is the main vector that flows through the network at each token position, being iteratively updated by each layer. The residual stream is a high-dimensional vector space, and different components of the model, such as attention heads, can use different subspaces of this vector space to store and communicate information. An attention head reading and writing to a particular subspace of the residual stream is analogous to reading and writing to a particular memory address, even though distinctions between addressable subspaces are learned rather than hardcoded in the architecture. 11, 37

attribution Attribution methods refer to a set of techniques used in deep learning to determine which parts of the input data a model relies on most when making predictions or decisions. The goal is to explain what aspects of the input have the greatest influence on a trained model's outputs. Attribution methods originated in computer vision, where they have been used to highlight the portions of an image that a neural network considers most important for its predictions. For example, when classifying an image as containing a dog, an attribution method could identify that the pixels representing the dog's face were the key factor driving the classification. By visualizing these important input features, attribution methods aim to provide insight into the model's classification process. More recently, the notion of attribution has been applied to other domains such as NLP. In NLP, attribution methods can highlight the words or phrases in a text input that have the biggest impact on a language model's predictions. This allows for better interpretability of language models and can reveal whether they are basing their outputs on relevant words or irrelevant/spurious correlations. Importantly, attribution methods only identify which input features are most important for a model's predictions, but do not explain how those features are internally encoded by the model. Other interpretability techniques such as probing and causal intervention methods can provide deeper insight into the model's learned representations and computations. 6

bigram A bigram is a sequence of two adjacent elements from a string, such as two consecutive words or tokens. Bigram models predict the next token based on the previous token, essentially memorizing common pairs like "United States" or "New York". In contrast, induction heads do not merely memorize bigram statistics, but implement instead a general algorithm: find previous occurrences of the current token, attend to the token that came next, and predict that same token is likely to come next again (`[A][B] ... [A] → [B]`). This computation is *not* content-specific; it does not depend on the specific identities of tokens `[A]` and `[B]`. It works for arbitrary token pairs, not just memorized common bigrams. In fact, induction heads can complete repeated sequences of random tokens whose bigram statistics the model could not have possibly memorized during training. 13

causal intervention Traditional methods for interpreting deep learning models, such as probing, can only establish correlational relationships between a model's internal representations and properties of interest. Such correlational methods risk yielding 'false positives' where a probe detects information that is not causally relevant to the model's behavior. In contrast, causal intervention methods aim to establish the causal role that a representation plays in a model's computation. The key idea is to directly manipulate or intervene on a model's representations and observe the effect on the model's output. If intervening to modify information about a property in the representation changes the model's predictions in a systematic way, this provides evidence that the model was relying on that information to make predictions. One popular approach known as *interchange intervention* works in the following way: given two inputs, swap the model's intermediate representations for those inputs and observe whether this changes the model's final output in an interpretable way. This allows researchers to test whether the model's representations encode the kind of modular, compositional structure that would be expected if the model is performing a systematic computation to solve the task. 6, 34

GAN A Generative Adversarial Network (GAN) is a machine learning architecture that consists of two networks – the *generator* and the *discriminator* – competing against each other to generate new, synthetic data that closely resembles real data. The generator's goal it so produce fake data (e.g., fake images) that looks as close to real data as possible, while the discriminator's goal is to distinguish between the real data and the fake data created by the generator. The generator and discriminator are trained simultaneously. The generator learns to create increasingly realistic fake data to fool the discriminator, while the discriminator continuously learns to better identify the differences between real and fake data. As the training progresses, the generator becomes better at creating realistic data, and the discriminator becomes more skilled at detecting fake data. The result of this competitive process is a generator that can create new, synthetic data that is very similar to real data. GANs have been used in various applications, such as generating realistic images. 22

graceful degradation The capacity of neural networks to maintain functionality despite the impairment or loss of some components. This concept emerged from the study of how neural networks respond to interventions such as ablations, where nodes within the network are selectively disabled. Unlike abrupt failure, graceful degradation is characterized by a gradual decline in performance, illustrating the network's robustness and resilience to partial damage. This phenomenon is akin to how biological brains, when faced with injury or loss of neurons, often continue to function, albeit at a reduced efficiency. The discovery of graceful degradation in neural networks underscores their parallel with biological systems and highlights a fundamental property of distributed processing systems, where information processing and network functionality are not localized to a single node but are distributed across the network. 7

grokking Grokking is a phenomenon in neural networks where, after a period of overfitting the training data and performing poorly on held-out test data, the model abruptly transitions to generalizing well, with test performance rapidly improving to match the training performance. This sudden onset of generalization, which occurs after the model has already overfit, distinguishes grokking from typical learning curves. Grokking relies on the model being regularized, typically through weight decay (a technique where the model weights are gradually shrunk towards zero, favoring simpler solutions), and trained on a limited dataset; without these conditions, the rapid shift to generalization does not occur. The term 'grokking' evokes the idea of the network suddenly 'getting it' after a period of stagnation. Mechanistically, grokking is hypothesized to occur when the regularization pressure eventually causes the model to shift from a brittle, overfitted solution to a more parsimonious, generalizable one after the memorization becomes

too costly. Grokking highlights the complex, non-monotonic generalization behavior that can arise in deep learning. 15

key-query-value In the attention mechanism of a Transformer, each input word embedding is projected into three different vectors - a query vector (Q), a key vector (K), and a value vector (V). This is done using three separate trainable weight matrices (W_q, W_k, W_v) for each attention head. The purpose of these three vectors is to enable the Transformer to selectively focus on or 'attend to' the most relevant parts of the input sequence when a given attention head is processing each token. At a given attention head, each token's query vector encodes the information that should be queried from other tokens, while key vectors encodes the information about each token that is relevant to the query. To determine which tokens are most relevant to each token in the input, the Transformer takes the dot product of each query vector with all the key vectors. This produces an 'attention score' for every token indicating how closely it matches what the query is looking for at that particular attention head. Finally, the value vector encodes which information should be retrieved from each token, which is then weighted by the attention score. Through this mechanism, each attention head processing a given input token probes all the other tokens for relevant information (using Q and K vectors), and then construct a new embedding (using the V vectors) that integrates that information based on relevance. This allows attention heads to selectively capture different relationships and dependencies between words, even across long distances. 20

nonparametric In the context of experiments on LLMs, nonparametric methods are used for analyzing or extracting information from the trained model without the need to train additional parameters. Unlike traditional probing techniques that involve training a separate classifier (probe) to interpret the model's internal activations, nonparametric approaches seek to infer linguistic or other properties directly from the existing components of the trained model. These methods often focus on pairwise importance scores, such as attention weights or distances between token representations, to derived insights about the internal structure of the model's representations and computations. 6

nullspace In the context of iterative nullspace projection, the nullspace of a probe contains representation directions that are not useful for that probe to make its prediction. Projecting onto the nullspace removes information the probe uses, but maintains encoding in the orthogonal directions. 8, 20

probing Probing is a method to analyze and interpret the information encoded in artificial neural networks such as LLMs. The basic idea is to train a classifier, called a 'probing classifier', to predict some property from the model's internal representations, such as part-of-speech tags or syntactic dependencies. More formally, let f be the target neural network model that maps input text x to some output y. This model generates intermediate representations $f_l(x)$ at some layer l. The probing classifier g is a separate model (typically, a linear classifier) that maps these intermediate representations $f_l(x)$ to linguistic properties z. The probing classifier g is trained on a labeled dataset of $\{x, z\}$ pairs to predict property z from representation $f_l(x)$. The key assumption is that if the probing classifier g achieves high accuracy in predicting property z, then the representations $f_l(x)$ must encode useful information about that linguistic property. Thus, probing aims to shed light on what kind of information is encoded in the representations learned by the target model f at different layers. There are several limitations and open questions in the probing paradigm, which includes issues related to the choice of probing classifier architecture, the selection of proper control tasks and baselines for comparison, disentangling the influence of the training dataset vs the model architecture. Crucially, probing is a correlational rather than

causal method; it can reveal if some information is encoded in a network, but it cannot by itself reveal whether the network actually makes use of that information for behavior. Causal claims can be established on firmer ground by complementing probing with interventional methods that modify the information encoded by the network and assess downstream effects of such interventions on model behavior. 6, 34, 35

residual stream The residual stream is the central pathway through which information flows in a Transformer model. It is a vector at each token position that is iteratively updated by each layer as it passes through the model. Specifically, the residual stream starts as the sum of the token embeddings and positional embeddings. This initial vector representation is then fed into the first layer. Each layer reads from the residual stream, applies attention heads and MLP blocks to update the information, and then writes its output back into the residual stream by adding it to the previous values. This process repeats at each layer, with the residual stream accumulating information at each step. The final value of the residual stream is what gets mapped to output logits to predict the next token. Importantly, the residual stream provides a direct pathway for information to travel from any layer to any later layer, and attention heads can also route information dynamtically between residual streams across distinct token positions. The fact that attention heads and MLP layers can write and read information in subspaces of residual streams suggests that they effectively function like an addressable memory, which enables sophisticated mechanisms like induction head circuits. 11, 13, 14, 28, 34

References

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. J., Jeffrey, K., Jesmonth, S., Joshi, N. J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M. & Zeng, A. (2022), 'Do As I Can, Not As I Say: Grounding Language in Robotic Affordances'.

Alain, G. & Bengio, Y. (2018), 'Understanding intermediate layers using linear classifier probes'.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A. & Simonyan, K. (2022), 'Flamingo: A Visual Language Model for Few-Shot Learning', *Advances in Neural Information Processing Systems* **35**, 23716–23736.

Aru, J., Larkum, M. E. & Shine, J. M. (2023), 'The feasibility of artificial consciousness through the lens of neuroscience', *Trends in Neurosciences* .

Baars, B. J. (1993), *A Cognitive Theory of Consciousness*, Cambridge University Press.

Barron, A. B. & Klein, C. (2016), 'What insects can tell us about the origins of consciousness', *Proceedings of the National Academy of Sciences* **113**(18), 4900–4908.

Barwich, A.-S. (2019), 'The Value of Failure in Science: The Story of Grandmother Cells in Neuroscience', *Frontiers in Neuroscience* **13**.

Beckers, S., Eberhardt, F. & Halpern, J. Y. (2020), Approximate Causal Abstractions, *in* 'Proceedings of The 35th Uncertainty in Artificial Intelligence Conference', PMLR, pp. 606–615.

Belinkov, Y. (2022), 'Probing Classifiers: Promises, Shortcomings, and Advances', *Computational Linguistics* **48**(1), 207–219.

Beran, M. (2018), 'Replication and Pre-Registration in Comparative Psychology', *International Journal of Comparative Psychology* **31**(0).

Brown, R., Lau, H. & LeDoux, J. E. (2019), 'Understanding the Higher-Order Approach to Consciousness', *Trends in Cognitive Sciences* **23**(9), 754–768.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T. & Zhang, Y. (2023), 'Sparks of Artificial General Intelligence: Early experiments with GPT-4'.

Buckner, C. (2019), 'Deep learning: A philosophical introduction', *Philosophy Compass* **14**(10), e12625.

Buckner, C. (2023), *From Deep Learning to Rational Machines: What the History of Philosophy Can Teach Us about the Future of Artificial Intelligence*, Oxford University Press, Oxford, New York.

Bulatov, A., Kuratov, Y. & Burtsev, M. (2022), 'Recurrent Memory Transformer', *Advances in Neural Information Processing Systems* **35**, 11079–11091.

Butlin, P., Long, R., Elmoznino, E., Bengio, Y., Birch, J., Constant, A., Deane, G., Fleming, S. M., Frith, C., Ji, X., Kanai, R., Klein, C., Lindsay, G., Michel, M., Mudrik, L., Peters, M. A. K., Schwitzgebel, E., Simon, J. & VanRullen, R. (2023), 'Consciousness in Artificial Intelligence: Insights from the Science of Consciousness'.

Cabanac, M., Cabanac, A. J. & Parent, A. (2009), 'The emergence of consciousness in phylogeny', *Behavioural Brain Research* **198**(2), 267–272.

Cao, R. (2022), 'Multiple realizability and the spirit of functionalism', *Synthese* **200**(6), 506.

Carruthers, P. (2002), 'The cognitive functions of language', *Behavioral and Brain Sciences* **25**(6), 657–674.

Carruthers, P. & Gennaro, R. (2023), Higher-Order Theories of Consciousness, *in* E. N. Zalta & U. Nodelman, eds, 'The Stanford Encyclopedia of Philosophy', fall 2023 edn, Metaphysics Research Lab, Stanford University.

Caucheteux, C., Gramfort, A. & King, J.-R. (2021), 'GPT-2's activations predict the degree of semantic comprehension in the human brain'.

Chalmers, D. J. (1995), Absent Qualia, Fading Qualia, Dancing Qualia, *in* T. Metzinger, ed., 'Conscious Experience', Ferdinand Schoningh, pp. 309–328.

Chefer, H., Gur, S. & Wolf, L. (2021), Transformer Interpretability Beyond Attention Visualization, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 782–791.

Chomsky, N. (1965), *Aspects of the Theory of Syntax*, Cambridge, MA, USA: MIT Press.

Christiansen, M. H. & Chater, N. (2016), *Creating Language: Integrating Evolution, Acquisition, and Processing*, MIT Press.

Clark, A. (1998), Magic Words: How Language Augments Human Computation, *in* P. Carruthers & J. Boucher, eds, 'Language and Thought: Interdisciplinary Themes', Cambridge University Press, pp. 162–183.

Colas, C., Karch, T., Moulin-Frier, C. & Oudeyer, P.-Y. (2021), 'Language as a Cognitive Tool: Dall-E, Humans and Vygotskian RL Agents'.

Colas, C., Karch, T., Sigaud, O. & Oudeyer, P.-Y. (2022), 'Autotelic Agents with Intrinsically Motivated Goal-Conditioned Reinforcement Learning: A Short Survey', *Journal of Artificial Intelligence Research* **74**, 1159–1199.

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S. & Garriga-Alonso, A. (2023), 'Towards Automated Circuit Discovery for Mechanistic Interpretability'.

Craver, C. F. (2007), *Explaining the Brain: Mechanisms and the Mosaic Unity of Neuroscience*, Oxford University Press, Clarendon Press, New York : Oxford University Press,.

Dasgupta, I., Lampinen, A. K., Chan, S. C. Y., Sheahan, H. R., Creswell, A., Kumaran, D., McClelland, J. L. & Hill, F. (2023), 'Language models show human-like content effects on reasoning tasks'.

Davis, E. & Aaronson, S. (2023), 'Testing GPT-4 with Wolfram Alpha and Code Interpreter plug-ins on math and science problems'.

Dehaene, S. & Naccache, L. (2001), 'Towards a cognitive neuroscience of consciousness: Basic evidence and a workspace framework', *Cognition* **79**(1), 1–37.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houlsby, N. (2021), 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'.

Dupre, G. (2021), '(What) Can Deep Learning Contribute to Theoretical Linguistics?', *Minds and Machines* **31**(4), 617–635.

Dupre, G. (2022), 'Realism and Observation: The View from Generative Grammar', *Philosophy of Science* **89**(3), 565–584.

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. (2022), 'Toy models of superposition', *Transformer Circuits Thread* .

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. (2021), 'A mathematical framework for transformer circuits', *Transformer Circuits Thread* .

Esser, P., Rombach, R. & Ommer, B. (2021), Taming Transformers for High-Resolution Image Synthesis, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 12873–12883.

Firestone, C. (2020), 'Performance vs. competence in human–machine comparisons', *Proceedings of the National Academy of Sciences* **117**(43), 26562–26571.

Frank, M. C. (2023), 'Large language models as models of human cognition'.

Frank, M. C., Bergelson, E., Bergmann, C., Cristia, A., Floccia, C., Gervain, J., Hamlin, J. K., Hannon, E. E., Kline, M., Levelt, C., Lew-Williams, C., Nazzi, T., Panneton, R., Rabagliati, H., Soderstrom, M., Sullivan, J., Waxman, S. & Yurovsky, D. (2017), 'A Collaborative Approach to Infant Research: Promoting Reproducibility, Best Practices, and Theory-Building', *Infancy* **22**(4), 421–435.

Franks, B. (1995), 'On Explanation in the Cognitive Sciences: Competence, Idealization, and the Failure of the Classical Cascade', *The British Journal for the Philosophy of Science* **46**(4), 475–502.

Friedman, D., Wettig, A. & Chen, D. (2023), 'Learning Transformer Programs'.

Gallistel, R. C. & King, A. P. (2011), *Memory and the Computational Brain: Why Cognitive Science Will Transform Neuroscience*, John Wiley & Sons.

Geiger, A., Lu, H., Icard, T. & Potts, C. (2021), Causal Abstractions of Neural Networks, *in* 'Advances in Neural Information Processing Systems', Vol. 34, Curran Associates, Inc., pp. 9574–9586.

Geiger, A., Potts, C. & Icard, T. (2023), 'Causal Abstraction for Faithful Model Interpretation'.

Giannou, A., Rajput, S., Sohn, J.-y., Lee, K., Lee, J. D. & Papailiopoulos, D. (2023), 'Looped Transformers as Programmable Computers'.

Giulianelli, M., Harding, J., Mohnert, F., Hupkes, D. & Zuidema, W. (2018), Under the Hood: Using Diagnostic Classifiers to Investigate and Improve how Language Models Track Agreement Information, *in* 'Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP', Association for Computational Linguistics, Brussels, Belgium, pp. 240–248.

Godfrey-Smith, P. (2016), 'Mind, Matter, and Metabolism', *The Journal of Philosophy* **113**(10), 481–506.

Godfrey-Smith, P. (2021), *Metazoa: Animal Life and the Birth of the Mind*, Picador.

Goodhart, C. (1975), 'Problems of monetary management: The U.K. experience', *Papers in Monetary Economics* **1**, 1–20.

Goyal, A. & Bengio, Y. (2022), 'Inductive biases for deep learning of higher-level cognition', *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **478**(2266), 20210068.

Gozalo-Brizuela, R. & Garrido-Merchán, E. C. (2023), 'A survey of Generative AI Applications'.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R. & Smith, N. A. (2018), 'Annotation Artifacts in Natural Language Inference Data'.

Han, S. J., Ransom, K. J., Perfors, A. & Kemp, C. (2024), 'Inductive reasoning in humans and large language models', *Cognitive Systems Research* **83**, 101155.

Harding, J. (2023), 'Operationalising Representation in Natural Language Processing'.

Hazineh, D. S., Zhang, Z. & Chiu, J. (2023), 'Linear Latent World Models in Simple Transformers: A Case Study on Othello-GPT'.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D. & Meger, D. (2018), 'Deep Reinforcement Learning That Matters', *Proceedings of the AAAI Conference on Artificial Intelligence* **32**(1).

Henighan, T., Carter, S., Hume, T., Elhage, N., Lasenby, R., Fort, S., Schiefer, N. & Olah, C. (2023), 'Superposition, memorization, and double descent', *Transformer Circuits Thread* .

Hewitt, J. & Liang, P. (2019), Designing and Interpreting Probes with Control Tasks, *in* 'Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)', Association for Computational Linguistics, Hong Kong, China, pp. 2733–2743.

Hsieh, C.-Y., Zhang, J., Ma, Z., Kembhavi, A. & Krishna, R. (2023), 'SugarCrepe: Fixing Hackable Benchmarks for Vision-Language Compositionality', *Advances in Neural Information Processing Systems* **36**.

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., Sermanet, P., Brown, N., Jackson, T., Luu, L., Levine, S., Hausman, K. & Ichter, B. (2022), 'Inner Monologue: Embodied Reasoning through Planning with Language Models'.

Hupkes, D., Veldhoen, S. & Zuidema, W. (2018), 'Visualisation and 'Diagnostic Classifiers' Reveal How Recurrent and Recursive Neural Networks Process Hierarchical Structure', *Journal of Artificial Intelligence Research* **61**, 907–926.

Hutchins, D., Schlag, I., Wu, Y., Dyer, E. & Neyshabur, B. (2022), 'Block-Recurrent Transformers', *Advances in Neural Information Processing Systems* **35**, 33248–33261.

Icard, T. F. (2017), From programs to causal models, *in* 'Proceedings of the 21st Amsterdam Colloquium', pp. 35–44.

Irvine, E. (2021), 'Developing Dark Pessimism Towards the Justificatory Role of Introspective Reports', *Erkenntnis* **86**(6), 1319–1344.

Jonas, E. & Kording, K. P. (2017), 'Could a Neuroscientist Understand a Microprocessor?', *PLOS Computational Biology* **13**(1), e1005268.

Kamath, A., Hessel, J. & Chang, K.-W. (2023*a*), Text encoders bottleneck compositionality in contrastive vision-language models, *in* 'The 2023 Conference on Empirical Methods in Natural Language Processing'.

Kamath, A., Hessel, J. & Chang, K.-W. (2023*b*), 'What's "up" with vision-language models? Investigating their struggle with spatial reasoning'.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J. & Amodei, D. (2020), 'Scaling Laws for Neural Language Models'.

Karvonen, A. (2024), 'Chess-GPT's Internal World Model', https://adamkarvonen.github.io/machine_learning/2024/
world-models.html.

Kiela, D., Bartolo, M., Nie, Y., Kaushik, D., Geiger, A., Wu, Z., Vidgen, B., Prasad, G., Singh, A., Ringshia, P., Ma, Z., Thrush, T., Riedel, S., Waseem, Z., Stenetorp, P., Jia, R., Bansal, M., Potts, C. & Williams, A. (2021), 'Dynabench: Rethinking Benchmarking in NLP'.

Kosinski, M. (2023), 'Theory of Mind Might Have Spontaneously Emerged in Large Language Models'.

Lamme, V. A. F. (2006), 'Towards a true neural stance on consciousness', *Trends in Cognitive Sciences* **10**(11), 494–501.

LeDoux, J., Birch, J., Andrews, K., Clayton, N. S., Daw, N. D., Frith, C., Lau, H., Peters, M. A. K., Schneider, S., Seth, A., Suddendorf, T. & Vandekerckhove, M. M. P. (2023), 'Consciousness beyond the human case', *Current Biology* **33**(16), R832–R840.

Lewis, M., Nayak, N. V., Yu, P., Yu, Q., Merullo, J., Bach, S. H. & Pavlick, E. (2023), 'Does CLIP Bind Concepts? Probing Compositionality in Large Image Models'.

Li, B. Z., Nye, M. & Andreas, J. (2021), Implicit Representations of Meaning in Neural Language Models, *in* 'Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)', Association for Computational Linguistics, Online, pp. 1813–1827.

Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H. & Wattenberg, M. (2023), 'Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task'.

Lindsay, G. W. & Bau, D. (2023), 'Testing methods of neural systems understanding', *Cognitive Systems Research* **82**, 101156.

Lipton, Z. C. (2018), 'The mythos of model interpretability', *Communications of the ACM* **61**(10), 36–43.

Lupyan, G. (2012), Chapter Seven - What Do Words Do? Toward a Theory of Language-Augmented Thought, *in* B. H. Ross, ed., 'Psychology of Learning and Motivation', Vol. 57 of *The Psychology of Learning and Motivation*, Academic Press, pp. 255–297.

Machamer, P., Darden, L. & Craver, C. F. (2000), 'Thinking about Mechanisms', *Philosophy of Science* **67**(1), 1–25.

Mahowald, K., Ivanova, A. A., Blank, I. A., Kanwisher, N., Tenenbaum, J. B. & Fedorenko, E. (2023), 'Dissociating language and thought in large language models: A cognitive perspective'.

Manheim, D. & Garrabrant, S. (2018), 'Categorizing Variants of Goodhart's Law', https://arxiv.org/abs/1803.04585v4.

Marcus, G. (2022), 'Deep Learning Is Hitting a Wall'.

Marr, D. (1982), *Vision: A Computational Approach*, Freeman & Co.

McCoy, R. T., Yao, S., Friedman, D., Hardy, M. & Griffiths, T. L. (2023), 'Embers of Autoregression: Understanding Large Language Models Through the Problem They are Trained to Solve'.

McGrath, S., Russin, J., Pavlick, E. & Feiman, R. (2023), 'How Can Deep Neural Networks Inform Theory in Psychological Science?'.

Meng, K., Bau, D., Andonian, A. & Belinkov, Y. (2023), 'Locating and Editing Factual Associations in GPT'.

Merker, B. (2005), 'The liabilities of mobility: A selection pressure for the transition to consciousness in animal evolution', *Consciousness and Cognition* **14**(1), 89–114.

Meyes, R., Lu, M., de Puiseau, C. W. & Meisen, T. (2019), 'Ablation Studies in Artificial Neural Networks'.

Millière, R. (forthcoming), 'Philosophy of Cognitive Science in the Age of Deep Learning', *WIREs Cognitive Science* .

Millière, R. (n.d.), Language Models as Models of Language, *in* R. Nefdt, G. Dupre & K. Stanton, eds, 'The Oxford Handbook of the Philosophy of Linguistics', Oxford University Press, Oxford.

Millière, R. & Buckner, C. (2024), 'A Philosophical Introduction to Language Models – Part I: Continuity With Classic Debates'.

Mirchandani, S., Xia, F., Florence, P., Ichter, B., Driess, D., Arenas, M. G., Rao, K., Sadigh, D. & Zeng, A. (2023), 'Large Language Models as General Pattern Machines'.

Mollo, D. C. & Millière, R. (2023), 'The Vector Grounding Problem'.

Murphy, E., de Villiers, J. & Morales, S. L. (2024), 'A Comparative Investigation of Compositional Syntax and Semantics in DALL-E 2'.

Nanda, N., Chan, L., Lieberum, T., Smith, J. & Steinhardt, J. (2022), Progress measures for grokking via mechanistic interpretability, *in* 'The Eleventh International Conference on Learning Representations'.

Nanda, N., Lee, A. & Wattenberg, M. (2023), 'Emergent Linear Representations in World Models of Self-Supervised Sequence Models'.

Nickles, T. (2020), 'Alien Reasoning: Is a Major Change in Scientific Research Underway?', *Topoi* **39**(4), 901–914.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S. & Olah, C. (2022), 'In-context learning and induction heads', *Transformer Circuits Thread* .

OpenAI (2023*a*), 'GPT-4 Technical Report'.

OpenAI (2023*b*), 'GPT-4V(ision) System Card'.

Ott, S., Barbosa-Silva, A., Blagec, K., Brauner, J. & Samwald, M. (2022), 'Mapping global dynamics of benchmark creation and saturation in artificial intelligence', *Nature Communications* **13**(1), 6793.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P. & Bernstein, M. S. (2023), 'Generative Agents: Interactive Simulacra of Human Behavior'.

Park, K., Choe, Y. J. & Veitch, V. (2023), 'The Linear Representation Hypothesis and the Geometry of Large Language Models'.

Plaut, D. C. & McClelland, J. L. (2010), 'Locating object knowledge in the brain: Comment on Bowers's (2009) attempt to revive the grandmother cell hypothesis', *Psychological Review* **117**(1), 284–288.

Racanière, S., Weber, T., Reichert, D., Buesing, L., Guez, A., Jimenez Rezende, D., Puigdomènech Badia, A., Vinyals, O., Heess, N., Li, Y., Pascanu, R., Battaglia, P., Hassabis, D., Silver, D. & Wierstra, D. (2017), Imagination-Augmented Agents for Deep Reinforcement Learning, *in* 'Advances in Neural Information Processing Systems', Vol. 30, Curran Associates, Inc.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. & Sutskever, I. (2021), Learning Transferable Visual Models From Natural Language Supervision, *in* 'Proceedings of the 38th International Conference on Machine Learning', PMLR, pp. 8748–8763.

Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J.-F., Breazeal, C., Crandall, J. W., Christakis, N. A., Couzin, I. D., Jackson, M. O., Jennings, N. R., Kamar, E., Kloumann, I. M., Larochelle, H., Lazer, D., McElreath, R., Mislove, A., Parkes, D. C., Pentland, A. S., Roberts, M. E., Shariff, A., Tenenbaum, J. B. & Wellman, M. (2019), 'Machine behaviour', *Nature* **568**(7753), 477–486.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. (2022), 'Hierarchical Text-Conditional Image Generation with CLIP Latents'.

Ravfogel, S., Elazar, Y., Gonen, H., Twiton, M. & Goldberg, Y. (2020), 'Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection'.

Ravfogel, S., Prasad, G., Linzen, T. & Goldberg, Y. (2021), Counterfactual Interventions Reveal the Causal Effect of Relative Clause Representations on Agreement Prediction, *in* 'Proceedings of the 25th Conference on Computational Natural Language Learning', Association for Computational Linguistics, Online, pp. 194–209.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-maron, G., Giménez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M. & de Freitas, N. (2022), 'A Generalist Agent', *Transactions on Machine Learning Research* .

Roberts, M., Thakur, H., Herlihy, C., White, C. & Dooley, S. (2023), 'Data Contamination Through the Lens of Time'.

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.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. (2022), High-Resolution Image Synthesis with Latent Diffusion Models, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', arXiv, pp. 10684–10695.

Rumelhart, D. E., Mcclelland, J. L. & Group, P. R. (1987), *Parallel Distributed Processing, Volume 1: Explorations in the Microstructure of Cognition: Foundations*, MIT Press.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N. & Scialom, T. (2023), 'Toolformer: Language Models Can Teach Themselves to Use Tools'.

Schrimpf, M., Blank, I. A., Tuckute, G., Kauf, C., Hosseini, E. A., Kanwisher, N., Tenenbaum, J. B. & Fedorenko, E. (2021), 'The neural architecture of language: Integrative modeling converges on predictive processing', *Proceedings of the National Academy of Sciences* **118**(45), e2105646118.

Sejnowski, T. J. & Rosenberg, C. R. (1987), 'Parallel Networks that Learn to Pronounce English Text', *Complex System* **1**, 145–168.

Sellam, T., Yadlowsky, S., Wei, J., Saphra, N., D'Amour, A., Linzen, T., Bastings, J., Turc, I., Eisenstein, J., Das, D., Tenney, I. & Pavlick, E. (2022), 'The MultiBERTs: BERT Reproductions for Robustness Analysis'.

Seth, A. K., Baars, B. J. & Edelman, D. B. (2005), 'Criteria for consciousness in humans and other mammals', *Consciousness and Cognition* **14**(1), 119–139.

Shea, N. (2023), 'Moving beyond content-specific computation in artificial neural networks', *Mind & Language* **38**(1), 156–177.

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K. & Yao, S. (2023), 'Reflexion: Language Agents with Verbal Reinforcement Learning'.

Smolensky, P. (1986), Neural and conceptual interpretation of PDP models, *in* 'Parallel Distributed Processing: Explorations in the Microstructure, Vol. 2: Psychological and Biological Models', MIT Press, Cambridge, MA, USA, pp. 390–431.

Smolensky, P. (1988), 'On the proper treatment of connectionism', *Behavioral and Brain Sciences* **11**(1), 1–23.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., Lewkowycz, A., Agarwal, A., Power, A., Ray, A., Warstadt, A., Kocurek, A. W., Safaya, A., Tazarv, A., Xiang, A., Parrish, A., Nie, A., Hussain, A., Askell, A., Dsouza, A., Slone, A., Rahane, A., Iyer, A. S., Andreassen, A. J., Madotto, A., Santilli, A., Stuhlmüller, A., Dai, A. M., La, A., Lampinen, A., Zou, A., Jiang, A., Chen, A., Vuong, A., Gupta, A., Gottardi, A., Norelli, A., Venkatesh, A., Gholamidavoodi, A., Tabassum, A., Menezes, A., Kirubarajan, A., Mullokandov, A., Sabharwal, A., Herrick, A., Efrat, A., Erdem, A., Karakaş, A., Roberts, B. R., Loe, B. S., Zoph, B., Bojanowski, B., Özyurt, B., Hedayatnia, B., Neyshabur, B., Inden, B., Stein, B., Ekmekci, B., Lin, B. Y., Howald, B., Orinion, B., Diao, C., Dour, C., Stinson, C., Argueta, C., Ferri, C., Singh, C., Rathkopf, C., Meng, C., Baral, C., Wu, C., Callison-Burch, C., Waites, C., Voigt, C., Manning, C. D., Potts, C., Ramirez, C., Rivera, C. E., Siro, C., Raffel, C., Ashcraft, C., Garbacea, C., Sileo, D., Garrette, D., Hendrycks, D., Kilman, D., Roth, D., Freeman, C. D., Khashabi, D., Levy, D., González, D. M., Perszyk, D., Hernandez, D., Chen, D., Ippolito, D., Gilboa, D., Dohan, D., Drakard, D., Jurgens, D., Datta, D., Ganguli, D., Emelin, D., Kleyko, D., Yuret, D., Chen, D., Tam, D., Hupkes, D., Misra, D., Buzan, D., Mollo, D. C., Yang, D., Lee, D.-H., Schrader, D., Shutova, E., Cubuk, E. D., Segal, E., Hagerman, E., Barnes, E., Donoway, E., Pavlick, E., Rodolà, E., Lam, E., Chu, E., Tang, E., Erdem, E., Chang, E., Chi, E. A., Dyer, E., Jerzak, E., Kim, E., Manyasi, E. E., Zheltonozhskii, E., Xia, F., Siar, F., Martínez-Plumed, F., Happé, F., Chollet, F., Rong, F., Mishra, G., Winata, G. I., de Melo, G., Kruszewski, G., Parascandolo, G., Mariani, G., Wang, G. X., Jaimovitch-Lopez, G., Betz, G., Gur-Ari, G., Galijasevic, H., Kim, H., Rashkin, H., Hajishirzi, H., Mehta, H., Bogar, H., Shevlin, H. F. A., Schuetze, H., Yakura, H., Zhang, H., Wong, H. M., Ng, I., Noble, I., Jumelet, J., Geissinger, J., Kernion, J., Hilton, J., Lee, J., Fisac, J. F., Simon, J. B., Koppel, J., Zheng, J., Zou, J., Kocon, J., Thompson, J., Wingfield, J., Kaplan, J., Radom, J., Sohl-Dickstein, J., Phang, J., Wei, J., Yosinski, J., Novikova, J., Bosscher, J., Marsh, J., Kim, J., Taal, J., Engel, J., Alabi, J., Xu, J., Song, J., Tang, J., Waweru, J., Burden, J., Miller, J., Balis, J. U., Batchelder, J., Berant, J., Frohberg, J., Rozen, J., Hernandez-Orallo, J., Boudeman, J., Guerr, J., Jones, J., Tenenbaum, J. B., Rule, J. S., Chua, J., Kanclerz, K., Livescu, K., Krauth, K., Gopalakrishnan, K., Ignatyeva, K., Markert, K., Dhole, K., Gimpel, K., Omondi, K., Mathewson, K. W., Chiafullo, K., Shkaruta, K., Shridhar, K., McDonell, K., Richardson, K., Reynolds, L., Gao, L., Zhang, L., Dugan, L., Qin, L., Contreras-Ochando, L., Morency, L.-P., Moschella, L., Lam, L., Noble, L., Schmidt, L., He, L., Oliveros-Colón, L., Metz, L., Senel, L. K., Bosma, M., Sap, M., Hoeve, M. T., Farooqi, M., Faruqui, M., Mazeika, M., Baturan, M., Marelli, M., Maru, M., Ramirez-Quintana, M. J., Tolkiehn, M., Giulianelli, M., Lewis, M., Potthast, M., Leavitt, M. L., Hagen, M., Schubert, M., Baitemirova, M. O., Arnaud, M., McElrath, M., Yee, M. A., Cohen, M., Gu, M., Ivanitskiy, M., Starritt, M., Strube, M., Swędrowski, M., Bevilacqua, M., Yasunaga, M., Kale, M., Cain, M., Xu, M., Suzgun, M., Walker, M., Tiwari, M., Bansal, M., Aminnaseri, M., Geva, M., Gheini, M., T, M. V., Peng, N., Chi, N. A., Lee, N., Krakover, N. G.-A., Cameron, N., Roberts, N., Doiron, N., Martinez, N., Nangia, N., Deckers, N., Muennighoff, N., Keskar, N. S., Iyer, N. S., Constant, N., Fiedel, N., Wen, N., Zhang, O., Agha, O., Elbaghdadi, O., Levy, O., Evans, O., Casares, P. A. M., Doshi, P., Fung, P., Liang, P. P., Vicol, P., Alipoormolabashi, P., Liao, P., Liang, P., Chang, P. W., Eckersley, P., Htut, P. M., Hwang, P., Miłkowski, P., Patil, P., Pezeshkpour, P., Oli, P., Mei, Q., Lyu, Q., Chen, Q., Banjade, R., Rudolph, R. E., Gabriel, R., Habacker, R., Risco, R., Millière, R., Garg, R., Barnes, R., Saurous, R. A., Arakawa, R., Raymaekers, R., Frank, R., Sikand, R., Novak, R., Sitelew, R., Bras, R. L., Liu, R., Jacobs, R., Zhang, R., Salakhutdinov, R., Chi, R. A., Lee, S. R., Stovall, R., Teehan, R., Yang, R., Singh, S., Mohammad, S. M., Anand, S., Dillavou, S., Shleifer, S., Wiseman, S., Gruetter, S., Bowman, S. R., Schoenholz, S. S., Han, S., Kwatra, S., Rous, S. A., Ghazarian, S., Ghosh, S., Casey, S., Bischoff, S., Gehrmann, S., Schuster, S., Sadeghi, S., Hamdan, S., Zhou, S., Srivastava, S., Shi, S., Singh, S., Asaadi, S., Gu, S. S., Pachchigar, S., Toshniwal, S., Upadhyay, S., Debnath, S. S., Shakeri, S., Thormeyer, S., Melzi, S., Reddy, S., Makini, S. P., Lee, S.-H., Torene, S., Hatwar, S., Dehaene, S., Divic, S., Ermon, S., Biderman, S., Lin, S., Prasad, S., Piantadosi,

S., Shieber, S., Misherghi, S., Kiritchenko, S., Mishra, S., Linzen, T., Schuster, T., Li, T., Yu, T., Ali, T., Hashimoto, T., Wu, T.-L., Desbordes, T., Rothschild, T., Phan, T., Wang, T., Nkinyili, T., Schick, T., Kornev, T., Tunduny, T., Gerstenberg, T., Chang, T., Neeraj, T., Khot, T., Shultz, T., Shaham, U., Misra, V., Demberg, V., Nyamai, V., Raunak, V., Ramasesh, V. V., Prabhu, V. U., Padmakumar, V., Srikumar, V., Fedus, W., Saunders, W., Zhang, W., Vossen, W., Ren, X., Tong, X., Zhao, X., Wu, X., Shen, X., Yaghoobzadeh, Y., Lakretz, Y., Song, Y., Bahri, Y., Choi, Y., Yang, Y., Hao, Y., Chen, Y., Belinkov, Y., Hou, Y., Hou, Y., Bai, Y., Seid, Z., Zhao, Z., Wang, Z., Wang, Z. J., Wang, Z. & Wu, Z. (2023), 'Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models', *Transactions on Machine Learning Research* .

Suri, G., Slater, L. R., Ziaee, A. & Nguyen, M. (2024), 'Do large language models show decision heuristics similar to humans? A case study using GPT-3.5.', *Journal of Experimental Psychology: General* **153**(4), 1066–1075.

Syed, A., Rager, C. & Conmy, A. (2023), 'Attribution Patching Outperforms Automated Circuit Discovery'.

Theakston, A. L., Lieven, E. V. M., Pine, J. M. & Rowland, C. F. (2001), 'The role of performance limitations in the acquisition of verb-argument structure: An alternative account', *Journal of Child Language* **28**(1), 127–152.

Thomas, R. K. (1998), Lloyd Morgan's Canon, *in* G. Greenberg & M. M. Haraway, eds, 'Comparative Psychology: A Handbook', Vol. 894, Garland Publishing Co, New York, pp. 156–163.

Tomasello, M. (2009), *Constructing a Language*, Harvard University Press.

Tong, S., Jones, E. & Steinhardt, J. (2023), 'Mass-Producing Failures of Multimodal Systems with Language Models', *Advances in Neural Information Processing Systems* **36**, 29292–29322.

Tschannen, M., Kumar, M., Steiner, A., Zhai, X., Houlsby, N. & Beyer, L. (2023), 'Image Captioners Are Scalable Vision Learners Too', *Advances in Neural Information Processing Systems* **36**.

Ullman, T. (2023), 'Large Language Models Fail on Trivial Alterations to Theory-of-Mind Tasks'.

Vygotsky, L. S. (1987), 'Thinking and Speech', *The Collected Works of L. S. Vygotsky* **1**, 39–285.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L. & Anandkumar, A. (2023), 'Voyager: An Open-Ended Embodied Agent with Large Language Models'.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z. & Wen, J.-R. (2023), 'A Survey on Large Language Model based Autonomous Agents'.

Webb, T., Holyoak, K. J. & Lu, H. (2023), 'Emergent analogical reasoning in large language models', *Nature Human Behaviour* pp. 1–16.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J. & Fedus, W. (2022), 'Emergent Abilities of Large Language Models'.

Weiss, G., Goldberg, Y. & Yahav, E. (2021), Thinking Like Transformers, *in* 'Proceedings of the 38th International Conference on Machine Learning', PMLR, pp. 11080–11090.

Whittington, J. C. R., Warren, J. & Behrens, T. E. J. (2022), 'Relating transformers to models and neural representations of the hippocampal formation'.

Wiggins, B. J. & Christopherson, C. D. (2019), 'The replication crisis in psychology: An overview for theoretical and philosophical psychology', *Journal of Theoretical and Philosophical Psychology* **39**(4), 202–217.

Woodward, J. (2005), *Making Things Happen: A Theory of Causal Explanation*, Oxford University Press, USA.

Wu, Y., Wang, S., Yang, H., Zheng, T., Zhang, H., Zhao, Y. & Qin, B. (2023), 'An Early Evaluation of GPT-4V(ision)'.

Wu, Z., Geiger, A., Potts, C. & Goodman, N. D. (2023), 'Interpretability at Scale: Identifying Causal Mechanisms in Alpaca'.

Yang, S., Chiang, W.-L., Zheng, L., Gonzalez, J. E. & Stoica, I. (2023), 'Rethinking Benchmark and Contamination for Language Models with Rephrased Samples'.

Yang, Z., Li, L., Lin, K., Wang, J., Lin, C.-C., Liu, Z. & Wang, L. (2023), 'The Dawn of LMMs: Preliminary Explorations with GPT-4V(ision)'.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. & Cao, Y. (2023), 'ReAct: Synergizing Reasoning and Acting in Language Models'.

Yildirim, I. & Paul, L. A. (2023), 'From task structures to world models: What do LLMs know?', *Trends in Cognitive Sciences* .

Yuksekgonul, M., Bianchi, F., Kalluri, P., Jurafsky, D. & Zou, J. (2022), When and Why Vision-Language Models Behave like Bags-Of-Words, and What to Do About It?, *in* 'The Eleventh International Conference on Learning Representations'.

Zeng, A., Attarian, M., Ichter, B., Choromanski, K., Wong, A., Welker, S., Tombari, F., Purohit, A., Ryoo, M., Sindhwani, V., Lee, J., Vanhoucke, V. & Florence, P. (2022), 'Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language'.

Zhang, F. & Nanda, N. (2023), 'Towards Best Practices of Activation Patching in Language Models: Metrics and Methods'.

Zhang, K. & Lewis, M. (2023), Evaluating CLIP's Understanding on Relationships in a Blocks World, *in* '2023 IEEE International Conference on Big Data (BigData)', pp. 2257–2264.

Zhou, A., Wang, K., Lu, Z., Shi, W., Luo, S., Qin, Z., Lu, S., Jia, A., Song, L., Zhan, M. & Li, H. (2023), 'Solving Challenging Math Word Problems Using GPT-4 Code Interpreter with Code-based Self-Verification'.

Zhou, H., Bradley, A., Littwin, E., Razin, N., Saremi, O., Susskind, J., Bengio, S. & Nakkiran, P. (2023), 'What Algorithms can Transformers Learn? A Study in Length Generalization'.

Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P. R., Salazar, G., Ryoo, M. S., Reymann, K., Rao, K., Pertsch, K., Mordatch, I., Michalewski, H., Lu, Y., Levine, S., Lee, L., Lee, T.-W. E., Leal, I., Kuang, Y., Kalashnikov, D., Julian, R., Joshi, N. J., Irpan, A., Ichter, B., Hsu, J., Herzog, A., Hausman, K., Gopalakrishnan, K., Fu, C., Florence, P., Finn, C., Dubey, K. A., Driess, D., Ding, T., Choromanski, K. M., Chen, X., Chebotar, Y., Carbajal, J., Brown, N., Brohan, A., Arenas, M. G. & Han, K. (2023), RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control, *in* '7th Annual Conference on Robot Learning'.