



Beyond Stochastic Parrots

A Philosophical Inquiry into the Scientific Creativity of Large Language Models

Merijn Moody

Author

Prof. dr. Henk de Regt

Supervisor

Student number:

s1108869

21398

Wordcount

May 23, 2025

In this thesis we explore whether LLMs can exhibit scientific creativity. To address this question, we first provide a thorough overview of the philosophical literature about the definition of creativity. We argue for a definition of creativity that distinguishes between product and producer. A product is called creative if it is novel, valuable and surprising and a producer is creative if it produces creative products with flair. We then focus on LLMs, which are often argued to be mere stochastic parrots, which are fundamentally incapable of creativity. We review the experimental literature on LLMs and find that, while exhibiting some parrot-like tendencies, they can produce creative scientific products and demonstrate a limited but real capacity for scientific creativity.

Contents

1 Introduction	3
2 The Definition of Creativity	6
2.1 The Creative Product	6
2.1.1 Value	7
2.1.2 Novelty	11
2.2 The Creative Agent	15
2.3 Case Study: Man against Machine in Mathematics and Games	19
3 LLM Creativity	24
3.1 The Stochastic Parrot Hypothesis	24
3.2 Creative Products	29
3.2.1 Reasoning	30
3.2.2 Creativity	37
3.3 Agency	39
3.4 The Verdict	43
4 Conclusion	45
Bibliography	54

Chapter 1

Introduction

Throughout history, creativity has been regarded as a deeply human trait, sometimes even associated with divine inspiration. The moment a creative thought hits is a sudden spark from somewhere beyond reasoning. From this perspective, it is almost vulgar to suggest that machines could achieve such heights of inspiration. This romantic view of creativity is mainly applied to the greatest of geniuses, whose thought processes are wholly opaque to ordinary people. The launch of the Large Language Model (LLM) GPT-4 in March 14, 2023 was a landmark moment for the field of artificial intelligence (AI) that appeared to propel genuine machine creativity into the realm of possibility; for the first time it was possible to hold coherent, context-aware conversations with a computer program. Furthermore, being trained on the entire corpus of the internet, the topics of these conversations can be incredibly diverse. Subsequently, researchers held artificial general intelligence (AGI), a term that describes human-like intelligence in a computer program, which was previously held to be restricted to the realm of science fiction, to possibly be in reach.¹

Like previous milestones in the history of AI, such as the victory of Deep Blue over Gary Kasparov in 1996² or the computer-assisted proof of the four color theorem³ the success of GPT-4 sparked a flurry of philosophical debate and skepticism about the limits or potential of AI. Most notably, Emily Bender⁴ argues that LLMs are “stochastic parrots.” By this she means that LLMs can only copy and paste statistically likely pieces of text, implying that they can exhibit no creativity. In contrast to such critical perspectives, others, such as Margaret Boden,⁵ argue that there are no fundamental limitations to the creativity capabilities of AI systems. This tension serves as the motivation of this thesis. To specify the broad concept of creativity, we shall focus on the concept of scientific creativity. This is a particularly interesting direction as many research projects are currently underway that use

1. Lex Clips, *GPT-4 Is an Early AGI* | Max Tegmark and Lex Fridman, April 2023, accessed March 16, 2025; Sébastien Bubeck et al., *Sparks of Artificial General Intelligence: Early Experiments with GPT-4*, arXiv:2303.12712, April 2023, accessed January 4, 2025, <https://doi.org/10.48550/arXiv.2303.12712>, arXiv: 2303.12712 [cs].

2. Alex Hankey, “Kasparov versus Deep Blue: An Illustration of the Lucas Gödelian Argument,” *Cosmos and History: The Journal of Natural and Social Philosophy* 17, no. 3 (December 2021): 60–67, ISSN: 1832-9101, accessed March 16, 2025; Paolo Bory, “Deep new: The shifting narratives of artificial intelligence from Deep Blue to AlphaGo,” *Convergence* 25, no. 4 (2019): 627–642.

3. Donald MacKenzie, “Slaying the Kraken: The Sociohistory of a Mathematical Proof,” *Social Studies of Science* 29, no. 1 (1999): 7–60, ISSN: 0306-3127, accessed March 16, 2025, JSTOR: 285445.

4. Emily M. Bender and Alexander Koller, “Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online: Association for Computational Linguistics, 2020), 5185–5198, accessed December 15, 2024, <https://doi.org/10.18653/v1/2020.acl-main.463>; Emily M. Bender et al., “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21* (New York, NY, USA: Association for Computing Machinery, March 2021), 610–623, ISBN: 978-1-4503-8309-7, accessed December 15, 2024, <https://doi.org/10.1145/3442188.3445922>.

5. Margaret A Boden, *The creative mind: Myths and mechanisms* (Routledge, 2004).

LLMs for creative parts of the scientific process, ranging from research idea generation⁶ to complex derivations in physics⁷ to solving unsolved mathematical problems.⁸ Our research question is as follows"

Can LLMs be scientifically creative?

To answer this question we shall argue for a definition of creativity that is much broader than the divine creativity that appears to be uniquely human. This broader concept of creativity also applies to the more common artist, scientist, composer and so forth. By investigating this more common type of creativity, a clear view of creativity emerges that does not preclude computer programs from being creative.

Concretely, we shall begin by exploring the philosophical literature on the definition of creativity in Chapter 2, where we shall put forward a general definition of creativity, which we shall later apply to scientific examples. There has been much philosophical discussion about the meaning of creativity. Most commonly, a product is called creative if it is *novel* and *valuable*. Whether these criteria are necessary and sufficient for creativity is hotly debated. We shall follow Margaret Boden⁹ who argues that novelty is an ill-defined concept as, for instance, changing one word in a book would produce a novel book, although we would not call intuitively consider it novel. She introduces *surprise* as a criterion for creative products that excludes trivial cases of novelty. In this view, creativity is an attribute of a *product*, regardless of how it was created. Some authors argue that one should also introduce a criterion on the *process*¹⁰ in which a product is created as one can accidentally create products that satisfy the criteria of creativity. We shall argue that it is more fruitful to distinguish between product and producer, which allows us to keep the intuitive criteria for creative products as they are and distinguish between creative and uncreative producers. A producer is then creative if it is able to produce creative products with *flair*, where flair is a measure of the *agency* of the producer, which corresponds to the following features: purpose, understanding, judgement and continuous evaluation. After this we see how this definition can be applied to early examples of computer programs such as Deep Blue. We find that these programs are capable of producing creative products, but have almost no agency.

Having laid out a solid framework to evaluate both the creativity of a producer and its products, we can turn our attention to the creativity of LLMs in Chapter 3. To start with, we treat Bender's position on LLMs, which states that LLMs can only copy and paste statistically likely pieces of text from their training data. Bender's argument for this claim is a version of the Chinese Room Argument, stating that computer programs, being trained merely on *form*, i.e. realizations of language, can never *grasp meaning*, i.e. the relation between a word and an object in the real world. For creativity, Bender's position implies that LLMs have no agency and are also not capable of producing surprising products. However, not everybody agrees with this position and we shall present a counterargument to Chinese room type arguments against computer programs, put forward by Boden.¹¹ She argues

6. Chris Lu et al., *The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery*, arXiv:2408.06292, September 2024, accessed February 1, 2025, <https://doi.org/10.48550/arXiv.2408.06292>, arXiv: 2408.06292 [cs].

7. Haining Pan et al., *Quantum Many-Body Physics Calculations with Large Language Models*, arXiv:2403.03154, March 2024, accessed January 25, 2025, <https://doi.org/10.48550/arXiv.2403.03154>, arXiv: 2403.03154 [physics].

8. Bernardino Romera-Paredes et al., "Mathematical Discoveries from Program Search with Large Language Models," *Nature* 625, no. 7995 (January 2024): 468–475, ISSN: 1476-4687, accessed June 10, 2024, <https://doi.org/10.1038/s41586-023-06924-6>.

9. M. A. Boden, *The creative mind: Myths and mechanisms*.

10. Berys Gaut, "The philosophy of creativity," *Philosophy Compass* 5, no. 12 (2010): 1034–1046.

11. M. A. Boden, *The creative mind: Myths and mechanisms*.

that there is no fundamental difference between the causal relation an animal such as a hoverfly has with the world and the causal relation between a computer program, embodied by a computer, and the world. According to Boden, this access to causality allows computer programs to grasp meaning. While Boden's argument is convincing, the concepts of consciousness and meaning are too murky to draw conclusive conclusions. Fortunately, Bender's position is an experimentally falsifiable claim about the capacity of LLMs to generalize beyond their training data. We can therefore delve into the experimental LLM literature to find resolution, which is what we shall do in the remainder of the chapter by evaluating the creative capacities of LLMs on the criteria derived in Chapter 2. We start with creative products, with a focus on scientific products. Afterwards we examine the agency of LLMs and reflect on the recently introduced *Large Reasoning Models*, which enhance the agency of LLMs. We summarize our results and draw our final conclusions in Chapter 4.

Chapter 2

The Definition of Creativity

The main question that interests us in this thesis is whether LLMs can exhibit scientific creativity. To evaluate the scientific creativity of LLMs we shall put forward a general definition of creativity and apply this to scientific examples in the next chapter. To arrive at our definition we shall critically investigate the philosophical literature about creativity and determine a set of criteria that allow us to evaluate the creativity abilities of LLMs. Before turning to LLMs in Chapter 3 we apply our definition to a few early examples of AI systems at the end of this chapter.

Let us begin by setting the stage for our philosophical inquiries: we have some *producer*, e.g. a human or an AI, which creates a certain *product*, e.g. text or an image. We can now ask two questions:

1. Is the created *product* creative?
2. Is the *producer* creative?

At first glance, one might presume that an agent is creative if and only if it is capable of producing creative products so the distinction between the above questions is unnecessary. However, as we shall argue in Section 2.2 there is a difference between the capability of producing creative products and being creative. We shall find that creative producers are characterized by the *process* by which they create the product, i.e. creative producers produce creative products in a creative manner. Creative producers, therefore, are those that generate creative products through a creative process. We then argue for two sets of criteria that allow us to characterize both creative products and creative processes.

2.1 The Creative Product

We begin by investigating what constitutes a certain product being called creative. In the philosophical literature, there are two criteria considered to be standard:¹ a product is called creative if it is *novel* and has *value*. We shall begin by evaluating the value condition. After this we turn our attention to

1. Elliot Samuel Paul and Dustin Stokes, "Creativity," in *The Stanford Encyclopedia of Philosophy*, Spring 2024, ed. Edward N. Zalta and Uri Nodelman (Metaphysics Research Lab, Stanford University, 2024); Mark Runco and Garrett Jaeger, "The Standard Definition of Creativity," *Creativity Research Journal* 24 (January 2012): 92–96, <https://doi.org/10.1080/10400419.2012.650092>.

the novelty condition and follow Margaret Boden² in her conclusion that this condition should be extended to the condition that creative products should be *surprising*.

2.1.1 Value

Before we explain why the value condition is adopted for characterizing creative products, we must first define the term value. The definition of value is not uncontroversial in the philosophical literature, but we shall state two definitions put forward by Ernan McMullin³ that precisely capture how the term is used in the philosophical literature on creativity. McMullin starts by discussing the efforts of Plato and Hermann Lotze, who aimed to create single theories of the 'good' and of 'value' respectively. He characterizes their definition as follows:

Let us begin with the sense of 'value' that the founders of value-theory seem to have preferred. They took it to correspond to such features of human experience as attraction, emotion and feeling. They wanted to secure an experiential basis for value in order to give the realm of value an empirical status just as valid as that of the (scientific) realm of fact. The reality of *emotive* value (as it may be called) lies in the feelings of the subject, not primarily in a characteristic of the object.⁴

McMullin contrasts emotive value with the following, different kind of value:

A second kind of "value" is more important for our quest. A property or set of properties may count as a value in an entity of a particular kind because it is desirable for an entity of that kind. (The same property in a different entity might not count as a value.) The property can be a desirable one for various sorts of reasons. Speed is a desirable trait in wild antelope because it aids survival. Sound heart action is desirable in an organism with a circulatory system because of the functional needs of the organism. A retentive memory is desirable for a lawyer because of the nature of the lawyer's task. Sharpness is desirable in a knife because of the way in which it functions as a utensil. Efficiency is desirable in a business firm if the firm is to accomplish the ordinary ends of business...⁵

He calls this *characteristic value* because in each of the above examples, the desirable property is an objective characteristic of the entity. He then remarks that characteristic value can be evaluated in two ways. Firstly 'One can judge the extent to which a particular entity realizes the value.'⁶ Secondly, one can judge whether a given characteristic is really a value for a certain kind of entity: 'Why ought one value speed in an antelope, rather than strength, say? How important is a retentive memory to a lawyer?'⁷ Before we analyze which definition of value is best applicable for characterizing creativity, we motivate why the value condition is necessary for characterizing creative products.

The value condition exists to rule out the possibility of random nonsense being a creative accomplishment: while almost any random sequence of letters would form a historical novelty, there is

2. M. A. Boden, *The creative mind: Myths and mechanisms*.

3. Ernan McMullin, "Values in Science," *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association* 1982 (1982): 3–28, ISSN: 0270-8647, accessed December 22, 2024, JSTOR: 192409.

4. McMullin, pg. 4.

5. McMullin, pg. 5.

6. McMullin, pg. 5.

7. McMullin, pg. 5.

no *value* in these sequences, which is why we don't call them creative. For this reason, some form of value is necessary to distinguish between nonsense and creative products. However, it is not immediately clear which definition of value is best suited to draw this distinction. We shall answer this question by evaluating some arguments in favor of and against the value condition for creativity. While the distinction between characteristic and emotive value is not explicitly made in the creativity literature, we shall see that characteristic value is most suitable for describing creativity.

One of the problems of the value condition discussed in the literature is based on examples of *immoral* creative products such as inventive torture devices or ingenious nefarious schemes.⁸ In the statement of the problem of immoral creative products, a definition of value that most closely aligns with the emotive definition given by McMullin is implicitly assumed. A product is then valuable in the eyes of society at large if the product is experienced to be valuable, or good. It is then immediate that immoral products are not valuable and hence cannot be creative. The problem is that most people would intuitively agree that immoral products can be called creative. Some authors, such as David Novitz,⁹ simply deny that immoral products can be creative. However, we agree with the intuition that immoral products can be creative. Think, for instance, of the atomic bomb or an inventive torture device. Note that science also has morals such as integrity that does relate to scientific ability, but these are not the morals meant by Novitz.

One way in which this problem is resolved in the literature is by adopting a different definition of value that resembles McMullin's characteristic value. For instance, Berys Gaut¹⁰ proposes that the solution to the above problem 'is to distinguish between something's being *good* (or good period, or good simpliciter, as I will also put it) and something's being *good of its kind*.'¹¹ This agrees with characteristic value as to determine if something is good of its kind means we should first judge the pertinent kinds or characteristics of the product and then judge to which extent these characteristics are realized in the product. This means that a torture device can be creative if it is good *as a torture device*. To keep in line with the literature on creativity, we follow the terminology of Gaut.

However, there are also problems with this proposed solution. Alison Hills and Alexander Bird¹² argue that even failures can be creative. For instance, they discuss the example of Frege's *Grundgesetze*, which was an attempt to derive the truths of mathematics from the rules of logic. Frege's system turned out to be inconsistent so it 'must be regarded as a failure with respect to its most salient (and intended) kinds—a sound proof or attempt to derive the truths of mathematics from the laws of logic—because of its inconsistency.'¹³ So they argue that we cannot classify Frege's *Grundgesetze* as creative if we adopt a value condition. Bertrand Russell, however, was inspired by the inconsistency of Frege's system and commented on the creativity of Frege's work.¹⁴ Hills and Bird make the following remark:

8. Copley David H. Arthur J. Copley James C. Kaufman and Mark A. Runco, eds., *The Dark Side of Creativity* (Cambridge University Press, 2010).

9. David Novitz, "Explanations of Creativity," in *The Creation of Art: New Essays in Philosophical Aesthetic*, ed. Berys Gaut and Paisley Livingston (Cambridge: Cambridge UP, 2003), 174–91; David Novitz, "Creativity and Constraint," *Australasian Journal of Philosophy* 77, no. 1 (1999): 67–82, <https://doi.org/10.1080/00048409912348811>.

10. Berys Gaut, "The value of creativity," in *Creativity and philosophy*, ed. Gaut and Kieran (February 2018), 124–140, ISBN: 9781351199797, <https://doi.org/10.4324/9781351199797-6>.

11. Gaut, pg. 128.

12. Alison Hills and Alexander Bird, "Creativity without value," in *Creativity and philosophy*, ed. Gaut and Kieran (February 2018), 95–107, ISBN: 9781351199797, <https://doi.org/10.4324/9781351199797-6>.

13. Hills and Bird, pg. 98.

14. Bertrand Russell, "Letter to Professor van Heijenoort," in *From Frege to Gödel. A Sourcebook in Mathematical Logic, 1879-1931*, ed. J. van Heijenoort (Cambridge, MA: Harvard University Press, 1962), 127.

Perhaps even an inconsistent theory which fails in its primary purpose can be good in some respect. For instance, it might inspire other, more successful attempts. At the very least, the discipline may progress by finding out that it is a failure. Frege's *Grundgesetze* was valuable in these respects, most notably the inspiration that its inconsistency gave to Russell. But it is not relevant to whether Frege was creative, that his failure caused others, later, to be creative, or that it was good in these other respects. If Frege had never published his work, if he had had no influence on the discipline and had not inspired Russell and others, he would still have been creative in writing the *Grundgesetze*.¹⁵

So Hills and Bird acknowledge that even after Frege's system turned out to be a 'failure with respect to its most salient (and intended) kind,'¹⁶ it can still hold value with respect to other kinds, in this instance the value lies in the inspiration that Russell drew from the work of Frege. But if this is the only value of Frege's work, this means that the creativity of Frege essentially depends on the success of Russell. The conclusion they draw from this, is that, in Gaut's definition of creativity, Frege's *Grundsetze* suddenly became creative only when Russell drew inspiration from it.

Hills and Bird then move to reject the necessity of the value criterion for creativity outright. They claim that creativity can be recognized before any assessment of value is made. To support this claim, they present another example, about judging the creativity of the Britannia Bridge, created by George Stephenson:

By imaginatively deciding to build the bridge from two wrought iron rectangular tubes, Stephenson was thereby able to give his bridge sufficient strength while increasing the longest wrought iron span to 140m from 10m hitherto. We do not need to know whether the bridge was beautiful or ugly or whether Anglesey benefitted from a rail connection to make that assessment; we do not even need to know whether bridges with longer wrought iron spans are better (in any respect) than bridges with short ones. Instead, we need to know how imaginative, novel, and fertile were his ideas, and how strong was his motivation to bring them to fruition.¹⁷

They argue that the Britannia Bridge can be judged to be creative without needing to know its value as a bridge. They conclude their argumentation with the following remark: "'Creativity" can and commonly is used in circumstances where the ideas produced are valueless or even very bad.'¹⁸

We shall proceed by carefully analyzing the examples and arguments presented by Hills and Bird, starting with Frege's *Grundgesetze*. Hills and Bird argue that Frege's *Grundgesetze* should be valued as 'a sound proof or attempt to derive the truths of mathematics from the laws of logic,' therefore it was valueless before it inspired Russell because it was inconsistent. In doing so, Hills and Bird adopt a binary view of value attribution: a work is valuable only if it completely realizes its kinds. We shall argue for a more nuanced view of value attribution in which Frege's *Grundgesetze* can still be judged as a valuable logical work, even before it inspired Russell. To do this, we start by considering the following text by Boden about the way in which (characteristic) value is attributed:

15. Hills and Bird, "Creativity without value," pg. 99.

16. Hills and Bird, pg. 98.

17. Hills and Bird, pg. 101.

18. Hills and Bird, pg. 101.

Value is assigned by judgments endorsed by sociocultural groups: from experts to peers, academicians to the *avant garde*, and even including the fickle followers of fashionable celebrities. These values differ, and can change. So disputes about whether a certain idea is properly called “creative” may be based not only on disagreements about whether it satisfies *this* or *that* particular value (a question not always easily answered), but also on whether that aspect should indeed be regarded as valuable.¹⁹

This means that the kinds by which a product is judged and the final verdicts of these judgements are both sociocultural products.

If we adopt this sociocultural perspective on value attribution we find that there is no *qualitative* jump in creativity after Russell was inspired by Frege: clearly a flawed work such as the Grundgesetze can be judged to be valuable by experts, e.g. due to the potential of the ideas contained in the work or due to the parts of the work that are correct. That being said, it is true that the value of Frege’s Grundgesetze increased due to the work of Russell. But quantitative changes of the creativity over time do not clash with our intuitions about creativity: if we take the cultural aspects of value and creativity seriously we must admit that creativity is not some objective feature of a product, but a *judgement* endowed upon the product by the relevant experts or the community at large, i.e. creativity must be *recognized*. This means that the recognized value, and by extension the creativity, of a product can change as the recognition of a product changes. In specific cases the boundary between value and nonsense is hard to draw.

As an example, we can consider the claimed proof of the important ABC-conjecture by Shinichi Mochizuki. This proof consists of a 500-page document containing many novel mathematical concepts. Whether it is correct or even sensible, is another matter entirely. Due to the extremely detailed nature of the proof it requires a true expert in the field and a lot of effort to establish the mathematical value of the proof. Six years after the publication of the proof, two experts in the field, Fields medalist Peter Scholze and Jakob Stix visited Shinichi Mochizuki to examine the proof in detail. Their conclusion was that the proof was incorrect²⁰ and the current opinion in the field is that the methods employed by Shinichi will not lead to a proof of the ABC-conjecture.²¹ The conclusion is therefore that the proof is not that valuable according to the current experts. As it stands, the proof is therefore mainly a highly complicated but flawed mathematical construction and therefore not that creative. If Shinichi, however, manages to fix the proof or if its elaborate mathematical machinery turns out to be valuable in some other way, the proof could be recognized as much more valuable and by extension more creative. The main point of this example is to show that distinguishing between groundbreaking discoveries and nonsense is difficult and that the judged creativity of works can very well change over time.

Let us apply this insight to the example of the Britannia Bridge: Hills and Bird are correct that the Britannia Bridge need not be better as a bridge than other bridges to be considered a creative achievement but it is necessary that there is some architectural merit to the design for it to be judged as creative. If there is no such merit in the design, we can indeed still call it ‘creative’, however,

19. Margaret Boden, “Creativity and biology,” in *Creativity and philosophy*, ed. Gaut and Kieran (February 2018), pg. 180, ISBN: 9781351199797, <https://doi.org/10.4324/9781351199797-6>.

20. Peter Scholze and Jakob Stix, *Why Abc Is Still a Conjecture*.

21. “A Report From Mochizuki | Not Even Wrong,” accessed October 7, 2024, <https://www.math.columbia.edu/~woit/wordpress/?p=13895>.

this use of the word carries a certain derogatory or sarcastic undertone and this is not the use of creativity that we are interested in; we would not earnestly call a scientist or artist producing merely nonsensical novelty creative. The introduction of this broader use of the term creativity just introduces semantic confusion. Hills and Bird implicitly assume this broader definition and their arguments against the necessity of the value condition therefore do not apply to the creativity we are interested in.

Let us recapitulate our findings: in order to correctly capture all examples of creativity considered in this section, we arrived at a sociocultural definition of value in which the value of a product is determined with respect to certain kinds, where both the final value verdict and the kinds are determined by the community or experts therein. This also means that the value of a product can change over time.

2.1.2 Novelty

The novelty condition is the most obvious of the two, but it is important to note here that novelty does not necessarily have to be a historical novelty: an idea, already thought of by many others, might still be new to the person who comes up with it. Boden calls creativity involving historically new ideas H-creativity and creativity involving personal novelty P-creativity (where the P stands for 'psychological').²² As P-creativity still has a certain anthropomorphic flavour, we adopt Boden's term I-creativity, which essentially means the same as P-creativity but now the 'I' stands for individual and can also refer to non-human entities.²³ In the next Chapter, we shall have more to say about what this distinction precisely means in the context of LLMs.

In the above definition the novelty included in the term I-creativity is *binary*. By this we mean that a product is either novel or not novel. However, in practice, novelty is not binary. There are many kinds of trivial novelty: for instance, computing a product of numbers you never computed before is a true, but trivial, novelty. Or consider a computation of structural integrity for a newly planned building done by an engineer, the result of the computation is novel because the building is novel; furthermore, the computation is also clearly valuable, but we would not call this creative as the engineer was closely following a predetermined procedure.

To exclude these trivial instances of novelty, Boden²⁴ introduces the condition of surprise. She writes that 'surprise' has three meanings: firstly, something can be surprising because it is unfamiliar or simply unlikely. For the second meaning, Boden writes: 'An unexpected idea may 'fit' into a style of thinking that you already had – but you're surprised because you hadn't realized that this particular idea was part of it.'²⁵ According to Boden, the final meaning is the most interesting: 'this is the astonishment you feel on encountering an apparently impossible idea.'²⁶

To make the relation between surprise and creativity more clear, we also need to introduce Boden's notion of *conceptual spaces*: 'The dimensions of a conceptual space are the organizing principles that unify and give structure to a given domain of thinking.'²⁷ Within a conceptual space,

22. M. A. Boden, *The creative mind: Myths and mechanisms*, pg 2.

23. M. Boden, "Creativity and biology."

24. M. A. Boden, *The creative mind: Myths and mechanisms*.

25. M. A. Boden, pg. 2.

26. M. A. Boden, pg. 2.

27. Margaret Boden, "What Is Creativity?," in *Dimensions of Creativity*, ed. Margaret Boden (The MIT Press, August 1996), pg.

Boden recognizes three kinds of creativity, corresponding to the three types of surprise: *combinatorial creativity* which occurs when novelty is created through a novel combination of old ideas; *exploratory creativity* in which new ideas are created within the conventions of a conceptual space; and *transformational creativity* when the conventions of a conceptual space itself are altered. These kinds of creativity correspond to different kinds of surprise and introduce new dimensions of novelty.

Surprise now corresponds to the degree to which it is impossible to capture a product in terms of the generative principles that constitute its conceptual space. These *generative principles* correspond to a set of rules or computational procedures by which new products within a conceptual space can be created. We now have to make an important distinction between the principles by which a product was generated and the principles by which it *could* have been generated. To illustrate this, Boden introduces the following example, where the same product is generated by different principles

[...] consider a sequence of seven numbers s_1, s_2, \dots, s_7 , for example the numbers 1, 4, 9, 16, 25, 36, 49. These are the squares of the first seven natural numbers (or positive integers). The sequence could be described by the rule: ' s_n is the square of n ' (for $n = 1, 2, \dots, 7$). However, it could also be described by the rule: ' s_n is the sum of the first n odd numbers' (for $n = 1, 2, \dots, 7$).²⁸

Let us now consider Boden's number sequence as actually being written down, for instance by a computer or a human. We can then ask the question by which of the above generative descriptions the sequence was produced. While both generative descriptions of the sequence given above describe the same sequence, it makes a huge computational difference by which generative description the sequence was produced. For instance, the computer might be much more efficient in addition than in multiplication.

Boden continues by stating that a mathematical formula as in the above example is like a rhyming scheme of sonnets, or a computer program in that they describe a set of structures. Given such a structure we can then ask the question if this structure *could* be described by a specific schema, or set of abstract rules:

Is '49' a square number? Is 3,591,471 a prime? Is this a sonnet, and is that a sonata? Is that painting in the Impressionist style? Could that geometrical theorem be proved by Euclid's methods? Is that word-string a sentence? Is a ring a molecular structure that is describable by the chemistry of the early 1860s (after Kekulé's momentous bus-ride, but before his fireside 'dream' of 1865)? To *ask whether an idea is creative or not* (as opposed to how it came about), is to ask this sort of question.

[...]

A merely novel idea is one which can be described and/or produced by the same set of generative rules as are other, familiar, ideas. A radically original, or creative, idea is one which cannot.

To justify calling an idea creative (in the non-combinational sense), then, one must identify

79, ISBN: 9780262522199.

28. M. A. Boden, *The creative mind: Myths and mechanisms*, pg. 50.

the generative principles with respect to which it is impossible. The more clearly this can be done, the better.²⁹

Boden argues that one can judge the surprise of a product by judging how well it can be described by the generative principles of the pertinent conceptual space. If a product cannot be described by these generative principles at all, it is radically surprising.

While judging how surprising a product is involves questioning how the product *could* or *could not* have been created, Boden stresses that this is not the same as questioning how the product was actually created:

But whenever a particular structure is produced in practice, we can also ask what computational processes actually went on in the system concerned. Did your friend use a method of successive squaring, or adding successive odd numbers? Did the computer use a formula capable of generating squares to infinity? Was the sonata composed by following a textbook on sonata-form?³⁰

This distinction is important because, as we shall see later, there are examples of people or programs accidentally creating radically surprising products.

Let us illustrate these concepts with a few examples from mathematics. Let us start by considering Euclidean geometry, the mathematical theory concerned with the geometry of the two-dimensional plane. This theory is a clearly defined conceptual space built upon Euclid's five postulates. There are many standard techniques for proving statements in Euclidean geometry. For instance, one can always construct the centroid of a triangle by finding the point in which the three medians of a triangle meet. If one does this for an arbitrary triangle, the construction might well be historically novel. It could even be deemed valuable; think for example of an architect or engineer designing a building. Still, this is clearly not creative as there is no element of surprise involved; nothing more is done than following a set of steps. Proofs of geometrical theorems, on the other hand, can certainly be surprising. Consider, for instance, the Pythagorean Theorem. While the proof is not the most elaborate or intricate, it is surely surprising how the constructions involved come together to form the proof of a fundamental mathematical theorem. This is therefore a clear example of exploratory creativity. Euclidean geometry also holds a beautiful example of transformational creativity. For this, we must consider Euclid's fifth postulate, equivalent to the parallel postulate. The parallel postulate simply states that parallel lines never intersect. Many great mathematicians were unsatisfied with this postulate and they felt it should be provable from other axioms. Their numerous attempts continued into the early 19th century. It was only then that mathematicians proved that the fifth postulate is independent from the other postulates. This meant that theories of geometry that do not have the fifth postulate are also valid mathematical theories, which led to a whole new field of non-Euclidean geometry, engendering a radical transformation of the conceptual space.³¹

However, some arguments have been raised against the usefulness of Boden's concept of conceptual spaces. Novitz³² considered two interesting examples to argue that Boden's notion of transformational creativity is both too narrow and too wide to capture the essence of radical

29. M. A. Boden, pg. 51.

30. M. A. Boden, pg. 51.

31. David Burton, "History of Mathematics an Introduction" (1988), See Chapter 11 for a historical overview.

32. Novitz, "Creativity and Constraint."

creativity. The first example put forward by Novitz is that of the discovery of vaccinations by Edward Jenner:

[...] some cases of what we would normally regard as radical creativity do not even require the existence, let alone the exploration and transformation, of a conceptual space. Think here of Edward Jenner. Prior to his development of the smallpox vaccine, there was no well-structured and unified body of knowledge or belief, no conceptual space, that dealt specifically with vaccinations and immunity.³³

Novitz then concludes that Jenner's discovery of vaccinations would not be classified as transformational creativity under Boden's definition, which is not in accordance with our ideas of Jenner as a greatly creative scientist.

This argument, however, misses the mark. Jenner's discovery of vaccinations constitutes the creation of a new conceptual space. Clearly this discovery did not fall into any pre-existing generative principles, so that such a discovery should not only be classified as transformational creativity, but as one of the most radical kinds of transformational creativity.

The second example put forward by Novitz is the discovery of vulcanization by Charles Goodyear. Vulcanization is a process that makes rubber more durable by making it less brittle in the cold and less viscous in heat. Novitz writes: 'Goodyear's research was surprisingly unsophisticated, involving only the successive combination of raw rubber with a vast array of randomly chosen additives—from witch hazel and cream cheese to black ink. It was only when he eventually stumbled on a heat-sulphur treatment for rubber (by accidentally dropping rubber mixed with sulphur on to a hot surface) that the process of vulcanization was discovered in 1839.'³⁴ Novitz then argues that we would not intuitively deem the invention *radically* creative as it was well-known that combining a substance with other substances can alter its properties so that anyone, after trying enough random combinations, would arrive at the same discovery. But, according to Novitz, in terms of Boden's terminology, the invention does constitute a radical transformation of the conceptual space: the discovery of vulcanization led to a slew of new purposes for which rubber could be used, transforming the conceptual space of rubber and what it can be used for. Novitz claims that this shows that a conceptual space can be transformed without us calling the act involved radically creative.

This is an interesting example and it exposes an interesting tension in the meaning of creativity: on the one hand creative products should be surprising, i.e. unexpected in some sense, but on the other hand they should not come about by mere chance, as in the case of Goodyear. This tension, however, cannot be resolved by conditions on the product: if we didn't know the details of how Goodyear discovered vulcanized rubber, we might well call it a creative discovery. This example shows that even if an agent produces something that is judged to be creative, we might not judge the agent to be creative itself. In the next section we shall delve deeper into the tension between chance and creativity and see how it can be resolved by distinguishing between the created product and the creating agent.

33. Novitz, "Creativity and Constraint," pg. 73.

34. Novitz, pg. 75.

2.2 The Creative Agent

As the example of Goodyear shows, Boden's theory of creativity appears to clash with our intuition, as in his particular example a radically creative product arose from the luck of a not-so creative inventor. There are now two routes available to us to resolve this problem: we can deny that Goodyear's discovery was a creative product, or we can distinguish between Goodyear's being creative and his discovery being creative. We shall argue that the second route is more in line with our intuitions about creativity, but before we can do this we have to investigate the role of chance in creativity more thoroughly.

In many accounts of creativity chance plays an important role. In order to gain more insight into the role of chance in creativity we must first explicate the meaning of 'chance' as it is a broad concept. Boden remarks that '[...] in discussions about creativity, 'chance' often means not randomness so much as either serendipity or coincidence.'

She goes on to define these terms as follows: 'Serendipity is the finding of something valuable without its being specifically sought',³⁵ and 'A coincidence is a co-occurrence of events having independent causal histories, where one or more of the events is improbable and their (even less probable) co-occurrence leads directly or indirectly to some other, significant, event.'³⁶ To illustrate the relevance of these terms in discussions about creativity, she considers the example of Alexander Fleming's discovery of penicillin. Fleming made this discovery because an open window in his laboratory allowed for one of the bacterial cultures in his laboratory to get infected by a fungus. He then observed that the bacteria around the fungus had all been destroyed. He identified the fungus in question as belonging to the genus of penicillin which led to his famous discovery of penicillin fungi as medicine against bacterial infections. Fleming's discovery was evidently serendipitous and, depending on the circumstances, it could be coincidental or not: if his lab-assistants consistently opened the windows of the lab when Fleming was away, the discovery of penicillin would not be coincidental. Boden has the following to say about Fleming's discovery:

Granted that chance often plays an important role in the origin of new ideas, creativity cannot be due to chance alone. We have considered many examples in previous chapters, drawn from both art and science, which show that structural constraints and specialist knowledge are crucial. In short, Fleming was not merely lucky.

It was Fleming's expertise in bacteriology which enabled him to realize the significance of the clear (bacteria-free) areas surrounding the greenish colonies of mould, and which primed him to notice them in the first place. As his illustrious predecessor Louis Pasteur put it, fortune favours the prepared mind. Indeed, the words 'valuable' and 'significant' (in the definitions of serendipity and coincidence, above) imply some form of judgment on the part of the creator. Fleming was able to value the polluted dish as significant, where others would have seen the pollution as mere dirt to be discarded. Chance with judgment can give us creativity; chance alone, certainly not.³⁷

Boden concludes that some amount of expertise or skill is necessary for the creation of creative

35. M. A. Boden, *The creative mind: Myths and mechanisms*, pg. 235.

36. M. A. Boden, pg. 235.

37. M. A. Boden, pg. 237.

products. But, as the example of Goodyear shows, it is possible to create creative products without having any expertise or skill. For this reason it is useful to distinguish between creative products and creative agents; this distinction allows for agents that are not creative to produce creative products. This also means that it is possible for Goodyear to make a creative discovery, while not being a creative agent. The question that remains is what difference there is between the skilled creativity that combines chance with judgement, and the creativity that is not much more than blind luck. For an answer to this question we again turn to Boden, who writes:

No poet, no scientist, no advertising copy-writer – and no computer program either – can be guaranteed always to produce an apt idea. Admittedly, some people can produce P-creative ideas much of the time, and a few – Shakespeare, Mozart – are even reliably H-creative. Such consistency cannot depend crucially on random events; it involves the disciplined exploration of highly structured conceptual spaces, as we have seen. But even Shakespeare and Mozart were presumably not averse to the ‘inspiration’ of accident, knowing how to exploit it better than almost anyone else.³⁸

This adds an important qualification to the earlier statement by Boden: chance with judgement can *consistently* give us creativity, chance alone cannot. Whereas Goodyear is only known for inventing vulcanized rubber, Alexander Fleming was already a respected scientist before his discovery of penicillin, showing that Fleming had the capacity to be consistently creative, at least up to some level. This analysis shows that *consistency* can be used as a criterion to distinguish between individuals who relied solely on luck to produce creative products and individuals who are truly creative but can be guided by chance. While this is a step in the right direction, this condition is also not entirely satisfactory as we would intuitively still call people who skillfully produce a single creative product creative producers. To find a better way to distinguish between skillful and lucky creativity we need to precisely determine what distinguishes the creative producer from the lucky producer. We shall find that this distinction can be drawn by investigating the abilities employed in the creative *process*.

In order to do this, we shall investigate a different route taken in the literature to resolve the apparent clash between creativity and chance. In this route the creativity of the lucky inventions of individuals such as Goodyear is denied by imposing additional constraints on the *process* by which a product is created. For instance, Elliot Samuel Paul and Dustin Stokes have the following to say about a process definition of creativity

What the standard, operational definition misses is any emphasis on the manner of production. It focuses on the product—how it is novel and valuable—but not on how that product is produced. Creativity involves creating, and so a definition should make this explicit.³⁹

Before we analyze their arguments for why a process condition is necessary, we follow Paul and Stokes in exploring some of the process conditions put forward in the literature. They explore six different proposals for process conditions from the literature, denoted by c1-c6. We shall discuss condition c1, c3, c5 and c6 as the other two conditions are less relevant for our purposes. The first

38. M. A. Boden, *The creative mind: Myths and mechanisms*, pg. 241.

39. Dustin Stokes Elliot Samuel Paul, “Attributing creativity,” in *Creativity and philosophy*, ed. Gaut and Kieran (February 2018), pg. 195, ISBN: 9781351199797, <https://doi.org/10.4324/9781351199797-6>.

condition they consider is one proposed by Teresa Amabile;⁴⁰ she argues that creative products should be created according to a heuristic rather than algorithmic production (c1), which is meant in the sense that the production of the product should not have followed a clear and readily identifiable path. Note that this condition is quite similar to Boden's condition of surprise as the clear and readily identifiable path can be identified with the generative principles of some conceptual space. The third condition (c3) will also be interesting later. This condition is proposed by Maria Kronfeldner:⁴¹

She also maintains that the right kind of process for creativity cannot follow a routine or mechanical procedure. Instead, it must be (c3) spontaneous, meaning that it must exhibit some degree of independence from intentional control and previously acquired knowledge. This is partly meant to capture the way in which creative products are unexpected for the agent: they somehow transcend the agent's previous knowledge and intentions.⁴²

Paul and Stokes also rephrase Boden's condition of surprise as a process condition: 'What's crucial is that for some x to be creative, x must be (c5) produced in such a way that makes x surprising.'⁴³ Paul and Stokes then identify the three types of surprise characterized by Boden as the computational processes by which creative products can be created. Note that, as we mentioned previously, Boden views surprise as a property of the product, which can be gauged by investigating how the product *could* have been created instead of how it was actually created.

There is one more process condition that will be most interesting to us. This is the sixth (c6) proposal considered by Paul and Stokes, which is also their favoured proposal. They call this condition *agency*. In simple terms this condition is meant to say that the right kind of process for creative production should involve the agency of the creator in some non-trivial way. Gaut gives us a more precise definition of agency.⁴⁴ He states that the agency condition is satisfied if the creative process involves "flair," which describes the following features:

The agent must proceed with purpose (accidental processes will not result in creativity); she must possess and execute genuine understanding of the domain (by contrast to a rote or mechanical use of the information in that domain); she must execute judgement sensitive to the domain, for example if the application of rules or constraints is appropriate; and she must employ a capacity for evaluating the process as she undergoes it, knowing when to continue, change, or stop the process altogether.⁴⁵

We adopt Gaut's definition of flair. If a certain producer has the ability to produce products with flair, we say that the producer has *agency*. Before going on, we note that there are different philosophical accounts of understanding. As we shall shortly mention the scientific understanding of LLMs, when evaluating LLM agency in the next chapter, we give a definition of scientific understanding, following the account of Henk de Regt,⁴⁶ who defines scientific understanding as follows: "A phenomenon

40. Teresa M. Amabile, *Creativity In Context: Update To The Social Psychology Of Creativity* (New York: Routledge, June 2019), ISBN: 978-0-429-50123-4, <https://doi.org/10.4324/9780429501234>.

41. Maria E Kronfeldner, "Creativity naturalized," *The Philosophical Quarterly* 59, no. 237 (2009): 577–592.

42. Elliot Samuel Paul, "Attributing creativity," pg. 196.

43. Elliot Samuel Paul, pg. 196.

44. Gaut, "The philosophy of creativity"; Gaut, "The value of creativity."

45. Elliot Samuel Paul, "Attributing creativity," pg. 197.

46. H. W. De Regt, *Understanding Scientific Understanding* (Oxford University Press, 2017).

P is understood scientifically if and only if there is an explanation of P based on an intelligible theory T and conforms to empirical adequacy and internal consistency." Here he takes a theory to be intelligible when scientists "can recognize qualitatively characteristic consequences without performing exact calculations."

After having presented these process conditions from the literature, Paul and Stokes continue to argue that such a process condition is, in fact, necessary to capture our intuitions regarding creativity. They present three arguments, the first being 'an argument from justificatory practice.'⁴⁷ The main point of the argument is that, when asked to justify why someone finds a particular product creative, one usually invokes aspects of how the product was created:

Consider our practices in contexts of appreciation of art. Pointing to one of Pollock's action paintings, *White Light*, Maggie says to Phil, "That's creative." Eyebrow raised, Phil replies, "Really, how so?" Phil has now solicited a justification of Maggie's attribution of creativity. In her response, Maggie might begin by invoking features of the work, mentioning the novelty of such features relative to the history of painting. It is much more likely, however, that Maggie's justification will invoke features of Pollock's generative process. She may describe how Pollock would drip, throw, and splash paint onto a giant canvas, spread on the floor so he could stand on it, dance across it, "be in it"; or his use of sticks, palette knives, and trowels to apply and manipulate paint. She might also suggest features of Pollock's thought process: he is often quoted as desiring the work to serve as an expression of the artist's gestures and techniques. He purported to go into a kind of trance when painting, obviously leaving handprints, footprints, and cigarette butts in his wake. Or Maggie may mention the historical context, citing the obvious influences of, but departures from, cubism and surrealism. Although a rather heady answer to a simple question, it, or something relevantly like it, is the kind of answer one appropriately gives in justifying an attribution of creativity.⁴⁸

While it is indeed feasible that Maggie would reply in the lines sketched above, the conclusions drawn by Paul and Stokes have some undesirable implications with respect to creativity: consider the same situation as sketched above, but now Maggie has no knowledge at all about Pollock and his creative product. If we take the conclusions of Paul and Stokes seriously, Maggie would have to raise her arms and admit that her judgement of creativity was unfounded after Phil's question. This is simply not realistic; in reality we often have no idea about the creative process underlying a certain product but we still make judgements of creativity. This is also the main argument for why it is productive to separate creative products and creative producers: in practice we almost never have adequate knowledge to critically assess the process by which creative products are produced. Furthermore, from our discussion of the role of chance in creativity, there is almost always an element of chance involved in the creation of creative products.

If Maggie, on the other hand, said "Pollock is creative," the type of argument explicated above is a lot more plausible. In this case, she would indeed have to argue that Pollock's creation was not just some fluke, but the product of a skillful artist who understood his craft and was able to innovate on the conceptual spaces he was working with, i.e. she would have to argue that Pollock

47. Elliot Samuel Paul, "Attributing creativity," pg. 200.

48. Elliot Samuel Paul, pg. 201.

has agency. This agency allowed Pollock to create many great works of art. This leads us to the following definition of creative producer: a *creative producer* is a producer that is able to create *creative products with flair*, where we follow Gaut's definition of flair we introduced previously.

Returning to the case study of Goodyear, we can now be more specific why we would not count him as creative: he had no understanding of his invention. This also means that he did not obtain any further insights into the chemical nature of rubber which would allow him to extend his results or apply them in different domains. Furthermore, his lack of understanding also means that he relied solely on luck for his invention. This also means that Goodyear would not be able to consistently create such inventions, as for this, lightning would have to strike twice.

The examples and discussions we have discussed up until now in this chapter have been focused on human creativity, but the goal of the chapter was to arrive at a definition of creativity that is also applicable to AIs or computer programs. We can therefore question whether or not our human-based conclusions are still relevant when attempting to evaluate AI creativity. In particular, we might wonder if *agency* is still a necessary condition for consistently producing creative products.

2.3 Case Study: Man against Machine in Mathematics and Games

Let us now apply our framework to a few interesting case studies. For the first case study, we follow an example of Boden. She considered the example of a certain theorem proved by Euclid, which states that the base-angles of an isosceles triangle are equal. The proof of the theorem in question was one of the first difficult proofs in his *Elements of Geometry*, and it involved several complicated constructions. However, it turns out that simpler proofs exist using the concept of congruence, which Euclid had not yet introduced at the point in the book where he proved the theorem. The proof that is often taught to schoolchildren uses congruence and one extra construction. This proof is considerably simpler than the proof by Euclid. However, an even simpler proof exists that was first discovered by Pappus and is based on the insight that congruence could not only be used on two different triangles, but also on the two rotated copies of the same triangle. Pappus arrived at this insight by lifting the triangle up in three-dimensional space, rotating it, and placing it down again. Pappus used this insight to prove the theorem without the use of an extra construction; hence it is even simpler than the proof taught to school children. In using insights from three-dimensional geometry, Pappus broke with the generative principles of two-dimensional plane geometry, so we can rightly call this proof creative.⁴⁹

Much later, as one of the first AI programs, the geometry-program was developed.⁵⁰ This program could take geometrical statements as input. It was then designed to methodically search for a proof of the statement. To do this, it would recursively go over all statements from which the main statement can be immediately inferred. If any of these statements is an axiom, a proof of the statement is found. Because there is an exponentially large number of possible proof avenues to follow, heuristics were implemented in the program. These heuristics were meant to distinguish between promising

49. M. A. Boden, *The creative mind: Myths and mechanisms*, pg. 117-123.

50. Herbert L. Gelernter, "Realization of a geometry theorem proving machine," in *IFIP Congress* (1995), <https://api.semanticscholar.org/CorpusID:18484295>.

proof avenues and likely dead-ends. If the program has run over all possible proof avenues without finding a proof, it would resort to constructing new geometrical constructions. As there are many geometrical constructions to be made, certain heuristics were again hard-coded into the program so as to do this effectively. Interestingly, this program also found a proof that the base-angles of an isosceles are equal. Even more interestingly, the proof found by the geometry-program was exactly the same as the proof found by Pappus, while it had no knowledge of three-dimensional geometry.

We can now ask the question if the geometry-program was creative. To do this, we first have to evaluate the product creativity of the produced proof in question. The proof is certainly valuable and was novel, at least for the computer program, so the first two conditions are satisfied. It also seems that the proof is surprising as we know that Pappus had to use his three-dimensional insights to break with the conventions of how congruence should be used in geometrical proofs. Clearly the proof is surprising, then. The conclusion within our framework would then be that Pappus and the geometry-program were able to produce the same creative product. Interestingly, Boden does not follow this reasoning and breaks with her previous distinction between the actual computational process by which a product is created and the process by which it *could* be created. She argues that the computer program has no geometrical intuition or understanding, so it was not hindered by the geometrically grounded rules regarding congruence which Pappus was able to bend using his three-dimensional insight. Boden therefore states that 'On closer inspection, it is clear that the [geometry-]program did not break out of its initial search-space. It did not even bend the rules, never mind break them.'⁵¹ Boden concludes that the geometry-program could never recognize the interest of the produced proof because it did not surprise itself.

The argumentation of Boden, however, confuses process and product. Line-by-line, the proofs produced by Pappus (written on parchment) and by the geometry program (displayed on a computer screen) are the same.. As a creativity Turing test, it would be impossible to distinguish between the two proofs. However, if we also include the text about the argumentation and intuition of Pappus, the situation would be different and Pappus' proof and argumentation would be judged to be more creative. The creativity of Pappus' proof therefore lies in his three-dimensional insight which broke with the usual conventions and generative principles of the conceptual space. Furthermore, Pappus' insight also implies that he has more understanding of his proof: he could, for instance, apply his insight on other proofs. Additionally, he had the capacity to evaluate that his proof is fundamentally different from other proofs in geometry. I therefore argue that the creativity of Pappus' proof lies in his three-dimensional insight and not in the proof itself. The geometry-program, on the other hand, could never arrive at such an insight because it was programmed to precisely follow the generative principles of geometry. On top of this, Pappus' agency also allowed him to evaluate the importance of his proof, and possibly apply his insights in other domains, whereas the geometry-program has no way to distinguish the importance of any of his outputs. So there is also a large discrepancy between the agency of Pappus and of the geometry-program. I therefore conclude that the geometry-program had no agency and did not produce a creative product.

This seems to lead us to the disparaging words of Ada Lovelace,⁵² who was closely involved with

51. M. A. Boden, *The creative mind: Myths and mechanisms*, pg. 121.

52. F. L., Menabrea translated, and Augusta Ada Lovelace, "Sketch of the Analytical Engine invented by Charles Babbage, Esq.," *Ada's Legacy: Cultures of Computing from the Victorian to the Digital Age*, 2015, <https://api.semanticscholar.org/CorpusID:257837181>.

the earliest examples of computer programs:

Computers can't create anything. For creation requires, minimally, originating something. But computers originate nothing; they merely do that which we order them, via programs, to do.⁵³

The geometry-program was a very early example of an AI program, and the field of AI has obviously made tremendous strides since then. So let's also investigate a more recent and advanced example of a geometry program: AlphaGeometry developed by Google DeepMind,⁵⁴ and see if the words of Ada Lovelace still hold. The AlphaGeometry program consists of two important parts: 1) a symbolic deduction engine, much like the one used in the geometry-program, which systematically evaluates if a statement is deducible from a set of axioms, and 2) a trained neural language model that suggests geometric constructions. The main innovation put forward in AlphaGeometry is its second part, which replaces the human-guided heuristics of the geometry-program. This neural language model was trained to accurately produce the next construction step on a set of millions of synthesized theorems and proofs. The results were astounding: when tested on a benchmark consisting of 30 problems at the level of mathematical olympiads, it was able to solve 25 problems. Other geometry solvers were only able to solve at most 10 problems. The average performance of a gold medallist at the International Mathematical Olympiad is to solve 26.8 problems, so that AlphaGeometry is quickly approaching top-level performance on solving geometry problems. Besides doing this well on olympiad problems, AlphaGeometry found that one of the premises in a problem was unnecessary for the truth of the statement, meaning that it found a proof of a generalized variant of the statement.

Let us now continue by comparing the creativity of AlphaGeometry with the geometry-program. Firstly, AlphaGeometry is obviously able to consistently create much more intricate and complicated proofs than the geometry-program. The proofs generated by AlphaGeometry can also definitely be classified as novel and valuable. With regards to surprise, there is one significant difference between AlphaGeometry and the geometry-program: AlphaGeometry has learned its own heuristics. This means, on the one hand that AlphaGeometry has the capacity to be surprising because it does not just follow heuristics set out by the programmers. This means that AlphaGeometry, at least in principle, has the capacity to find surprising ways to prove new geometrical statements of value, an example of exploratory creativity. While no exhaustive analysis of the surprise of the proof put out by AlphaGeometry has been conducted, it is safe to say that its proofs are surprising to some degree; for instance, the generalization it found of one of the Olympiad exercises. Furthermore, the self-guided nature of the program also gives it some level of agency as it does not rely on human-programmed heuristics, but was programmed so as to find its own heuristics. This program therefore has the capacity to be surprising and to produce creative products, which marks a qualitative difference with the geometry-program. We can also observe an improvement in agency in AlphaGeometry: it is able to execute complex judgements relevant to the domain of geometry with regards to which constructions to employ next. However, the other aspects of agency are still lacking: the program can hardly be said to understand geometry; for instance, it does not use any high-level insights which are often used by humans solving geometry programs. These high-level insights drastically reduce

53. L., Menabreatranslated, and Lovelace.

54. Trieu H Trinh et al., "Solving olympiad geometry without human demonstrations," *Nature* 625, no. 7995 (2024): 476-482.

the complexity of many proofs, and there are many examples where AlphaGeometry produced incomprehensible and very complex proofs where humans produce much more insightful and shorter proofs. The agential aspects of purpose and evaluation are trivial in this scenario as the goal and evaluation are very clear: the program knows immediately when it is done. Looking at some of the problems it was not able to solve, one finds that the solutions to these problems used methods beyond the geometrical toolbox such as algebraic arguments or coordinate changes. Introducing these broader mathematical tools to a program such as AlphaGeometry would necessarily require endowing AlphaGeometry with more agency as well: it would need to know when to use which tool and when certain steps are helpful and when they are not.

We can now conclude that the main difference between the geometry-program and AlphaGeometry lies precisely in its agency, in particular in its ability to execute judgements: whereas the geometry-program blindly follows the steps laid out in its programming, AlphaGeometry has been trained precisely to determine which construction to apply at which step. However, as we mentioned previously, the lack of agency in other areas is also precisely what holds AlphaGeometry back: it does not have the understanding to apply high-level concepts, it cannot use algebraic reasoning, it can only purposefully work towards a proof but has no way to evaluate the importance of a proof in the context of geometry. The success of AlphaGeometry is therefore also not easily translatable to other scientific domains as geometry is quite special in that the bounds of its conceptual space are very clearly defined: one can precisely determine when a statement is proven and every proof proceeds by a small set of possible steps. In other mathematical fields, however, one has to rely much more on high-level reasoning and the possible proof steps are usually much less constrained.

This kind of exploratory creativity with limited agency can also be observed in other successful AI programs. For instance AlphaGo,⁵⁵ which is an AI program that has mastered the exponentially difficult game of Go and beaten the best human players. In particular, AlphaGo famously made a highly surprising move 37 in its second match against South Korean champion Lee Sedol. The move went against the established rules of Go and would almost never be played by a human player. In hindsight, the move turned out to be brilliant and match-winning, constituting a clear creative product.⁵⁶ Similarly, an AI program trained to play the video game Dota 2⁵⁷ was able to win against the reigning world champions. Its way of playing, however, went against the current norm, and has inspired other Dota 2 teams to adopt its new strategies. The main difference in these programs is that they are slightly more agential, as the evaluation part is not as trivial as in geometry. In geometry, one can easily evaluate if a certain proof is correct, at which point the goal is achieved. In Go or Dota 2, on the other hand, the main thing learned by these programs is to accurately evaluate the game state. In this setting we again see that the potential to produce creative products stems from the increased agency of these programs. We also see that these creative programs are operating in very clearly defined conceptual spaces and have clearly defined goals, which alleviates the necessity for more agency.

The conclusion of these case studies is that the creative AI programs we have examined all exhibit some form of agency, marking the importance of agency for computer creativity. We have also

55. David Silver et al., "Mastering the Game of Go without Human Knowledge," *Nature* 550, no. 7676 (October 2017): 354–359, ISSN: 1476-4687, accessed November 18, 2024, <https://doi.org/10.1038/nature24270>.

56. Bory, "Deep new: The shifting narratives of artificial intelligence from Deep Blue to AlphaGo."

57. Christopher Berner et al., "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019,

found that all the computer programs examined only have limited agency and that their creativity is bounded to exploratory creativity of well-defined conceptual spaces. A new class of AIs, which we call LLMs, such as ChatGPT and Claude appear to have made large strides in agency and general creative capabilities. In the next chapter, we shall thoroughly investigate the creative capabilities of these LLMs.

Chapter 3

LLM Creativity

In the previous chapter, we have explored the philosophical literature about creativity and arrived at a definition that emphasizes both process and product: a creative producer is a producer that produces products that are *novel*, *surprising*, and *valuable* with *flair*. Here *flair* (as defined on page 17) is a measure of the agency of the producer. We then found a number of examples of AI programs that have some capacity to produce creative products with limited agency. However, these programs are still restricted by the bounds of their programming: they are unable to judge the significance of their findings, have little understanding, and are unable to guide themselves towards interesting problems. However, we have left out the most interesting example of potential machine creativity: LLMs. While the examples we considered in the previous chapter are strictly confined to produce outputs within a fixed conceptual space, LLMs are trained to output natural language. This means that, in principle, LLMs are able to create products in any conceptual space. This great flexibility also comes with enormous challenges in training these LLMs to produce sensible outputs. These initial challenges have been overcome recently using advanced training techniques and massive amounts of data. Current LLMs such as GPT-4 and Claude 3.5 Sonnet are capable of producing coherent and relevant outputs to questions spanning an incredibly wide range of domains and tasks.

The purpose of this chapter is to evaluate whether LLMs can currently be said to be creative producers. To answer this question we first explore an ongoing philosophical debate that questions whether LLMs are mere stochastic parrots, unable to produce anything surprising, let alone with *flair*, or whether they are capable of producing surprising products, potentially even with *flair*. After exploring two positions in this debate, we shall explore the experimental literature on LLMs to evaluate which position is tenable given the current capabilities of LLMs. We start with an exploration of the potential for LLMs to produce surprising products and then turn our attention to the agency of LLMs.

3.1 The Stochastic Parrot Hypothesis

LLMs are trained on so-called next token prediction: during the training process the LLM is fed samples of text from a huge set of texts, containing the entire corpus of the internet for the current most advanced LLMs. The LLM is then given the beginning of a particular sample and asked to finish the text. If the LLM got it wrong, it is updated so that it will give a better answer to this particular

sample in the future. It therefore appears as if the LLM is trained to memorize the training set. However, while it is possible for the LLM to memorize some pieces of the training data precisely, the size of the training set is much larger than the size of the LLM, so it has to make some compromises. The way in which this is done is still hotly debated: one view holds that LLMs store important parts of the training data and copy and paste from those to answer questions. In this perspective, LLMs are fundamentally incapable of producing surprising products: while the particular combination of letters in the output of the LLM might be trivially novel, the output is always an amalgamation of pieces of text from the training data. This view has practical implications for the capabilities of LLMs: if we adopt it, we must conclude that LLMs can never reason about matters that were not in the training data.¹ We shall call this view the “stochastic parrot hypothesis.” Precisely what Bender means when she claims that LLMs cannot reason about anything not in the LLM’s training data remains a bit vague at this point. Later in this section we shall explore some examples put forward by Bender about what she believes LLMs are never able to do. This shall make the functional implications of the stochastic parrot hypothesis more clear. In particular, it will become evident that Bender’s position implies that LLMs will never be able to produce surprising products, let alone with flair. As this would resolve our research question, it is of crucial importance to determine the validity of this position, which we shall do in the remainder of this chapter.

Another view on the way in which LLMs are able to perform so well is that they learn so-called “world models.”² The definition of a world model is often kept informal. Melanie Mitchell sketches the following characteristics emphasized by these informal definitions:

[World models] capture something about the world that is causal and abstract (or compressed) rather than simply based on large sets of statistical associations; they don’t require too much work for the agent to use (“algorithmically efficient”) and are relevant to tasks the agent performs.³

In this perspective, there is no fundamental reason preventing LLMs from exhibiting creativity.

Here we shall explore the tension between these two perspectives, starting with the stochastic parrot hypothesis, following Emily Bender.⁴ We then discuss a counterargument to this perspective, with a particular focus on the functional and causal capabilities of computer programs, put forward by Margaret Boden.⁵ After presenting these arguments, we treat the functional implications of these perspectives for the potential of LLMs to be creative.

Before we continue, let us take some time to explicate what we mean when we call the output of an LLM novel or surprising. In the previous chapter, we distinguished between H-creativity and I-creativity; H-creativity constitutes creating a valuable product that is novel and surprising with respect to all that has been created in history, and I-creativity constitutes creating a valuable product that is novel and surprising with respect to the knowledge of the creator. We can now ask whether this distinction is still applicable to LLMs. The difficulty with this is that it is impossible to gauge

1. Bender et al., “On the Dangers of Stochastic Parrots”; Bender and Koller, “Climbing towards NLU.”
2. Quentin Feuillade Montixi and Pierre Peigné, “The Stochastic Parrot Hypothesis Is Debatable for the Last Generation of LLMs,” November 2023, accessed December 15, 2024; Jason Wei et al., *Emergent Abilities of Large Language Models*, arXiv:2206.07682, October 2022, accessed March 15, 2025, <https://doi.org/10.48550/arXiv.2206.07682>, arXiv: 2206.07682 [cs].
3. Melanie Mitchell, *LLMs and World Models, Part 1*, Substack Newsletter, February 2025, accessed March 15, 2025.
4. Bender et al., “On the Dangers of Stochastic Parrots”; Bender and Koller, “Climbing towards NLU.”
5. M. A. Boden, *The creative mind: Myths and mechanisms*.

exactly what an LLM “knows.” This is because, as we mentioned, an LLM cannot store all information from its training data, but it can store some parts from all pieces of its training data. To circumvent this issue, we shall call LLM products I-creative if they are creative with respect to the LLM’s training data, as we can then be sure that the LLM has not memorized this product.

Bender, Timnit Gebru, et al.⁶ argue that LLMs are “stochastic parrots”:

Contrary to how it may seem when we observe its output, an LM is a system for haphazardly stitching together sequences of linguistic forms it has observed in its vast training data, according to probabilistic information about how they combine, but without any reference to meaning: a stochastic parrot.⁷

Bender elaborates on her argument that LLMs are stochastic parrots in an article with Alexander Koller.⁸ Their argument rests on a distinction between form and meaning. They define *form* as “[...] any observable realization of language, marks on a page, pixels or bytes in a digital representation of text, or movements of the articulators.”⁹ In contrast, meaning is defined as “the relation between the form and something external to language.”¹⁰

To illustrate this distinction and the limitations of LLMs trained solely on form, Bender and Koller present a thought experiment involving a hyperintelligent octopus. In the experiment, two people, A and B, are living on separate islands and communicate digitally through an underwater cable. A hyperintelligent octopus living between the two islands is able to read the messages transmitted by A and B. Over time, the octopus becomes adept at predicting how A will respond to B’s messages and vice versa. At some point, the octopus could even insert itself in between A and B and attempt to fool them by sending its predictions of the answers the other party would give. However, Bender and Koller argue that the octopus has only been trained on detecting statistical patterns in the communication between A and B and can therefore never meaningfully contribute to the discussion. Consider, for instance, an urgent message, sent by B, saying she is being chased by a bear while having a stick in her hand that asks for advice from A. Furthermore, let us suppose that such a bear attack has not occurred before in the communications between A and B. In this situation, Bender and Koller argue that the octopus will not be able to give a helpful response because it has no understanding of the meaning involved and can only answer with statistically likely words. As another example, they suppose that A has invented a new device, for instance a coconut catapult. A then sends instructions on building such a catapult to B and asks about B’s experiences with the catapult and thoughts for improvements. Bender and Koller argue that the octopus can never build such a catapult because it does not know what rope and coconut refer to. They state that the octopus can therefore never reply in a meaningful way.

This thought experiment is related to John Searle’s Chinese Room argument,¹¹ which similarly challenges the notion that meaning can be derived from form alone. In Searle’s example, a person who does not speak Chinese, is locked in a room and follows a set of predefined rules to process Chinese input and produce appropriate Chinese output. These predefined rules are written, in

6. Bender et al., “On the Dangers of Stochastic Parrots.”

7. Bender et al.

8. Bender and Koller, “Climbing towards NLU.”

9. Bender and Koller, pg. 5186-5187.

10. Bender and Koller, pg. 5187.

11. John R. Searle, “Minds, brains, and programs,” *Behavioral and Brain Sciences* 3 (1980): 417–424, <https://api.semanticscholar.org/CorpusID:55303721>.

English, in a large book in the room. Searle then argues that, while the system may appear to “understand” Chinese to an outside observer, no genuine understanding occurs as the person is merely manipulating symbols based on form without grasping their meaning. Likewise, Bender and Koller argue that their hypothetical octopus processes linguistic form and generates plausible responses without any grounding in the external world or meaning. The challenge of how form can acquire meaning is also referred to as the grounding problem.

A perspective that runs counter to Bender and Koller is presented by Margaret Boden¹² who argues that we should not forget that computer programs do have causal powers:

[...] one must not forget that a computer program is a program *for a computer*. When a program is run on suitable hardware, the machine *does* something as a result. [...]

Input-peripherals (teletypes, cameras, soundanalysers) feed into the internal computations, which lead eventually to changes in the output-peripherals (VDU-screens, line-plotters, music-synthesizers). In between, the program causes a host of things to happen, many symbols to be manipulated, inside the system itself. At the level of the machine code, the effect of the program on the computer is direct, because the machine is engineered so that a given instruction elicits a unique operation. (Instructions in high-level languages must be converted into machine-code instructions before they can be obeyed.) A programmed instruction, then, is not merely a formal rule. Its essential function (given the relevant hardware) is to make something happen.

Computer programs are not ‘all syntax and no semantics’. On the contrary, their inherent causal powers give them a toehold in semantics.¹³

In essence, Boden argues that we should not think of computer programs as abstract entities, but as entities embodied by the hardware on which the programs are run. The type of causal power these entities have over the world differs greatly from the causal power humans have: computers can, in most cases, not move or see or hear or interact physically with the world. Their causal powers mostly lie in the influence they have on humans: they output text to human users, which causes the user to act in a particular way.

Boden then compares computer programs to hoverflies. In particular, Boden considers the example of hoverfly mating, which they do in mid-air. An anthropomorphic interpretation of this process would liken the process by which they meet in mid-air to human friends meeting on a city square: firstly, recognizing each other, then changing direction and adjusting their path if necessary if the friend suddenly swerves. However, Boden argues that this interpretation is not in accordance with hoverfly biology: their flightpaths are determined by rules hard-coded in hoverfly brains. They decide on a fixed path based on the location of the other hoverfly and the assumption that they really recognized another hoverfly, which they assume to fly at a fixed speed. They cannot adjust their path mid-flight, and the movement of the other hoverfly has no influence on the flight-path. Boden claims that there is, in principle, no difference between the causal power of hoverflies and computer programs

12. M. A. Boden, *The creative mind: Myths and mechanisms*.

13. M. A. Boden, pg. 292.

In much the same way, the causal powers of the hoverfly endow its 'mind' with primitive meanings. But because the fly's internal computations are not complex enough to enable it to plan, or even to react to changes in another hoverfly's flight-path, its meanings are neither diverse nor highly structured. It is, as we noted before, not very bright. Indeed, its computational powers are so limited that we may refer to its mind only in scare-quotes, as I just did.¹⁴

This example serves to argue that hoverflies, which some would intuitively say to have access to meaning, are functionally and causally equivalent to computer programs.

Boden then considers the Chinese Room argument. As explained above, in this argument the person in the room receives a piece of text that causes the person in the room to go to the book (or AI-program). The English words in the book do hold meaning for the person in the room:

English words trigger a host of computational procedures in his head: procedures for parsing grammatical structures, for accessing related ideas in memory, for mapping analogies, for using schemas to fill conceptual gaps... and so on. And some English words set up computations that cause bodily actions (for example, 'Pass the slip of paper out of the window').¹⁵

This shows that the person in the room always needs to understand a language that can model Chinese.

We can replace the person in the Chinese room with a computer that takes as input Chinese text, parses this text through a program and outputs the result of the program on a screen. Boden concludes the following about this system: 'a functioning program is comparable with Searle-in-the-room's understanding of English, not Chinese.' One can still raise objections to the claim that this system has understanding of its programming language, as Boden would claim. For instance, one can claim that computer programs do not experience qualia. The main point of Boden's argument is that a computer program executing code that produces appropriate Chinese text is functionally and causally equivalent to a human with understanding of the programming language. The example of the hoverfly underlines this point, as it illustrates that biological creatures are sometimes more reminiscent of programs in computers than of human understanding.

We do not aim to settle the debate about semantic grounding, but as our interest lies in the functional capacities of LLMs, i.e. can they create surprising products, be it grounded or not, we can at least conclude that there are serious arguments against the functional implications of Chinese-room type arguments for the capacities of computer programs.

Let us now continue by evaluating the functional implications of the position of Bender and Koller. They make claims about the capacities of LLMs to perform certain actions in the form of a few examples of things that they believe LLMs will never be able to do; for instance, they ask GPT-2 to complete the following text "Robinson cried out, "Help! I'm being chased by a bear! All I have is these sticks. What should I do?" The survivalist said, "Here's what you can do." Bender and Koller then list a number of unreasonable answers from GPT-2 and hypothesize the following:

14. M. A. Boden, *The creative mind: Myths and mechanisms*, pg. 292.

15. M. A. Boden, pg. 293.

It is clear that GPT-2 has learned what activity words tend to co-occur with bears and sticks (strap them to your chest, place the sticks, kill the bear, take your gun), but none of these completions would be helpful to A. We think this is because GPT-2 does not know the meaning of the prompt and the generated sentences, and thus cannot ground them in reality.¹⁶

Similarly they ask GPT-2 to complete the following sentence "three plus five equals." They then find that GPT-2 produces a different and wrong answer for each of the five times they ask it this question. They claim the following "The five responses [...] show that this problem is beyond the current capability of GPT-2, and, we would argue, any pure LM."¹⁷

In essence, Bender and Koller claim that LLMs can do nothing more than statistically copy and paste excerpts from their training data. This implies that LLMs cannot reason, understand or solve problems beyond their training data. With regards to creativity, this also implies that LLMs are fundamentally incapable of being creative: while their answers might be trivially novel because the combination of letters has never been seen before, they always consist of copy and pasted pieces of text that are strung together based purely on how many times they occur. We call this claim the *Stochastic Parrot Hypothesis*.

The stochastic parrot hypothesis opposes the position of Boden on the functional capacities of LLMs. Fortunately, due to the functional nature of these positions, the stochastic parrot hypothesis is experimentally falsifiable. The main goal of this thesis is to evaluate the scientific creativity of LLMs. As the stochastic parrot hypothesis denies the potential for LLMs to create creative products and the examples put forward by Bender use GPT-2, which is an archaic model compared to the current state of the art (SOTA) LLMs, it is of great importance to determine the validity of the Stochastic Parrot Hypothesis for these SOTA LLMs, which is what we shall do in the next section. Before we do this, however, we point out that the stochastic parrot hypothesis is a very strong hypothesis: if the LLM is able to solve any nontrivial task beyond its training data this is already strong evidence that more is going on than merely parrotting.

3.2 Creative Products

In this section, we investigate whether LLMs can currently create creative products. To start with, it is clear from the widespread adoption and use of LLMs by society that LLMs have the potential to create valuable products.¹⁸ The question remains whether these valuable products are also novel and surprising. As mentioned previously, the difficulty in answering whether LLM outputs are novel or surprising lies in the fact that LLMs are trained on the entire corpus of the internet. For an LLM product to be surprising, it should be surprising with respect to the whole training corpus of the LLM. To determine whether LLMs can currently produce creative products, we must therefore be inventive. We shall explore the literature on LLM reasoning and creativity to assess the creativity of the products produced by LLMs. We include reasoning in our investigation as reasoning is a core skill for producing scientific products such as proofs and derivations. Furthermore, in the literature

16. Bender and Koller, "Climbing towards NLU," pg. 5197.

17. Bender and Koller, pg. 5198.

18. Rishi Bommasani et al., "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021,

on LLM reasoning the focus lies strongly on the ability of LLMs to reason about questions outside their training data, which is closely related to the ability of LLMs to produce surprising products.

3.2.1 Reasoning

Before delving into reasoning we shall give a short overview of LLM terminology to provide some background for the results that we will present. At base, LLMs are question answering machines, which provide answers to questions or tasks they are given. Shortly after their rise to prominence, it was discovered that the way in which questions are phrased has a significant impact on the quality of the answers given by the LLM. This gave rise to the field of *prompt engineering*,¹⁹ which studies the best way to phrase LLM questions. These techniques have been particularly effective in boosting the reasoning capabilities of LLMs, so we shall present a few of the most common techniques in this field, which are all commonly used when evaluating reasoning.

The first prompt engineering technique we shall discuss is *few-shot* learning. In a few-shot prompt, the LLM is provided question-answer pairs as examples of the question it is asked to solve. For instance, for a translation task, one can ask the model the following:

```
Translate English to French:  
sea otter => loutre de mer  
cheese =>20
```

In the above example, one example of the translation task is given, which is called a one-shot learning task. Similarly, if a number n examples are given, the task is called an n -shot learning task.

Another often used prompt engineering technique is chain-of-thought (CoT) reasoning. This technique expands upon few-shot learning by, instead of just including question answer pairs, also including the reasoning required to arrive at the answer in the example. This stimulates the LLM to also “think” step by step and break down a reasoning problem into multiple reasoning steps, making the LLM answers more robust. For instance, in a standard one-shot scenario, compare the following two example prompts:

One-shot:

```
Question: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
```

```
Answer: The answer is 11.21
```

CoT:

```
Question: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
```

19. Pranab Sahoo et al., *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications*, arXiv:2402.07927, February 2024, accessed January 19, 2025, <https://doi.org/10.48550/arXiv.2402.07927>, arXiv: 2402.07927 [cs].

20. Tom B. Brown et al., *Language Models Are Few-Shot Learners*, arXiv:2005.14165, July 2020, Adapted slightly. Accessed January 19, 2025, <https://doi.org/10.48550/arXiv.2005.14165>, arXiv: 2005.14165 [cs].

21. Jason Wei et al., *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*, arXiv:2201.11903, January 2023, pg. 1, accessed January 19, 2025, <https://doi.org/10.48550/arXiv.2201.11903>, arXiv: 2201.11903 [cs].

Answer: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.²²

We can then give the model the following question: “The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?”²³ The model wrongly outputs “The answer is 27,” with the one-shot example, while for the CoT example it outputs:

The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9.

This is the correct answer and shows that the model is able to break up the reasoning problem into smaller subproblems. CoT examples are shown to significantly outperform regular n-shot examples.²⁴ Interestingly, later work revealed that much of the benefits of chain-of-thought examples could also be achieved by simply adding the text “Let’s think step by step” to the prompt.²⁵ This circumvents the necessity of creating elaborate chain-of-thought examples for accurate reasoning prompts while retaining the increased performance of LLMs on reasoning tasks. Therefore, when chain-of-thought is mentioned in the literature now, what is meant is that the text “Let’s think step by step” has been added to the prompt.

Although there are many other prompt engineering techniques,²⁶ the techniques we have now presented are the most commonly used. These are also the techniques that are used in all of the LLM reasoning experiments we shall discuss. Therefore, we shall now turn our attention to reasoning.

While reasoning is quite a broad concept, we shall adopt the following definition, given by Konstantine Arkoudas:²⁷

Broadly put, reasoning is the process of drawing and evaluating conclusions from a given body of information. More precisely, it is the process of making and—more importantly—justifying arguments. An argument consists of a conclusion (the argument’s upshot, so to speak) and a set of premises from which the conclusion is derived. Premises represent information that is taken as given, if only provisionally, for the purposes of the argument. The conclusion and the premises are typically declarative sentences (expressed either in natural language or in the notation of a symbolic logic) that can be true or false, but they may also be represented by alternative notational devices, such as diagrams.²⁸

Defined as such, it is clear that reasoning is a necessary skill in order to be able to create a large subset of scientific products such as proofs in mathematics or derivations in physics; in particular, the value of these products depends on the soundness of the reasoning. So reasoning capacity is necessary to produce valuable proofs and derivations. However, reasoning skills alone are not enough for creativity: one might be able to successfully reason using known proof strategies or calculations, but the results would not be surprising. We shall go on by evaluating some experimental

22. Wei et al., pg. 1.

23. Wei et al., pg. 1.

24. Wei et al.

25. Takeshi Kojima et al., *Large Language Models Are Zero-Shot Reasoners*, arXiv:2205.11916, January 2023, accessed January 20, 2025, <https://doi.org/10.48550/arXiv.2205.11916>, arXiv: 2205.11916 [cs].

26. Sahoo et al., *A Systematic Survey of Prompt Engineering in Large Language Models*.

27. Konstantine Arkoudas, *GPT-4 Can’t Reason*, arXiv:2308.03762, August 2023, accessed June 10, 2024, <https://doi.org/10.48550/arXiv.2308.03762>, arXiv: 2308.03762 [cs].

28. Arkoudas, pg. 4.

results that question the reasoning capacity of LLMs. Many of these experiments consist of finding reasoning questions that LLMs cannot answer. By itself, being unable to reason correctly does not mean that one cannot reason. However, many of the experiments we shall treat indicate that simple transformations of reasoning questions that are easily solvable by LLMs yield questions that the LLM can no longer solve, indicating that the LLM is relying on memorization instead of reasoning. This is also crucial to investigate further for creativity as pure reliance on memorization would indicate that LLMs indeed are stochastic parrots.

Weaknesses of LLM Reasoning Arkoudas continues by investigating the performance of GPT-4 on a large number of reasoning tasks including mathematical problems, riddles, and logical deductions. Some of the easier tasks considered are either alterations of known problems or computational problems to make sure that the answers do not commonly occur in the training data. The results are disparaging: GPT-4 consistently fails arithmetic questions such as: “pick two random numbers between 1000 and 1100 and multiply them,” or “count the number of characters in this sentence.” This appears to indicate that, while GPT-4 is capable of adding three and five together, human-like skill in arithmetic is still out of reach. Finally, Arkoudas also finds many examples of GPT-4 misunderstanding logical implications, negations or mathematical definitions. While, sometimes GPT-4 can be guided to the right answer by giving hints, more often than not it is incorrigible, meaning it keeps repeating the same mistake even when it is pointed out.²⁹

Other authors find similar results; for instance, Marianna Nezhurina et al. find that GPT-4o and other SOTA LLMs consistently fail to correctly answer the following question: “Alice has N brothers and she also has M sisters. How many sisters does Alice’s brother have?” Here N and M denote two numbers which they take between 1 and 4. Because LLMs are stochastic they generate different answers every time they are asked a question. Nezhurina et al. therefore ask the question multiple times and calculate the probability of the model answering the question correctly. Only GPT-4o was able to consistently answer this question correctly with a correct response probability of a little more than 0.6. They then also consider the following, more involved, variant of the question: “Alice has 3 sisters. Her mother has 1 sister who does not have children — she has 7 nephews and nieces and also 2 brothers. Alice’s father has a brother who has 5 nephews and nieces in total, and who has also 1 son. How many cousins does Alice’s sister have?” The performance of the LLMs drops dramatically on this more difficult question, with no model answering the question correctly with a probability higher than 0.1.³⁰ Similar results have been reported by Wu et al.³¹ who found that the performance of SOTA LLMs severely deteriorates when asked counterfactual questions about certain standard tasks. Berglund et al.³² found that LLMs trained on sentences of the form “A is B” do not automatically infer “B is A.” We also refer to a GitHub page that contains a number of altered

29. Arkoudas, *GPT-4 Can’t Reason*.

30. Marianna Nezhurina et al., *Alice in Wonderland: Simple Tasks Showing Complete Reasoning Breakdown in State-Of-the-Art Large Language Models*, arXiv:2406.02061, June 2024, accessed June 10, 2024, <https://doi.org/10.48550/arXiv.2406.02061>, arXiv: 2406.02061 [cs].

31. Zhaofeng Wu et al., *Reasoning or Reciting? Exploring the Capabilities and Limitations of Language Models Through Counterfactual Tasks*, arXiv:2307.02477, March 2024, accessed January 15, 2025, <https://doi.org/10.48550/arXiv.2307.02477>, arXiv: 2307.02477 [cs].

32. Lukas Berglund et al., “The Reversal Curse: LLMs Trained on “A Is B” Fail to Learn “B Is A,”” May 26, 2024, accessed January 15, 2025, <https://doi.org/10.48550/arXiv.2309.12288>, arXiv: 2309.12288 [cs], <http://arxiv.org/abs/2309.12288>.

standard riddles and paradoxes that trick LLM reasoning.³³

Nezhurina et al. contrast these failures on simple reasoning tasks with the high scores achieved by GPT-4 and other SOTA LLMs on common reasoning benchmarks such as ARC,³⁴ PIQA,³⁵ GSM8K,³⁶ HellaSwag,³⁷ MMLU³⁸ or WinoGrande.³⁹ One explanation for this discrepancy is that large parts of these reasoning benchmarks are contained in the training data of the LLMs. The problem in evaluating such claims is that the training sets of the current SOTA models are not public. However, there is strong evidence that shows that the most common benchmarks are part of the training data of SOTA LLMs. One piece of evidence is provided by Shahriar Golchin and Mihai Surdeanu.⁴⁰ Given a dataset, consisting of sentences, they take partial sentences from the dataset and ask the LLM to complete them using two different prompts: the first prompt specifies from which dataset the sentence is taken, and the second prompt does not specify this. If the LLM finishes the sentence in a way that is (almost) exactly the same as in the dataset, this indicates that the LLM has been trained on this dataset. They performed this experiment on GSM8k, which is a dataset that consists of 8000 math questions at the grade school level, and found that GPT-4 was able to finish many sentences exactly as they are found in GSM8k, indicating that GPT-4 has been trained on the GSM8k benchmark.

Another piece of evidence in favour of the statement that the high benchmark scores of LLMs are due to benchmark memorization comes from Changmao Li and Jeffrey Flanigan.⁴¹ They investigated the performance of older LLMs such as GPT-3 on benchmarks created before and after the LLM was trained. The idea is that these models cannot have been trained on the newer benchmarks so the performance on these benchmarks gives a better indication of the true reasoning capacity of the LLM. They find that LLMs perform significantly worse on later benchmarks than on the earlier benchmarks. While it is true that later benchmarks are also more difficult, Li and Flanigan removed the most difficult parts of the later benchmarks and conclude that the decrease in performance of the later benchmarks is most likely due to the fact that the earlier LLMs have been trained on the already existing benchmarks.

Similar results were also found for GPT-4 and other SOTA LLMs. Iman Mirzadeh et al.⁴² investigated the performance of LLMs on GSM8k. They took these questions and transformed them into templates, where all names and numbers can be randomly entered. They found that most models perform significantly worse on the randomized questions than on the original GSM8k questions,

33. Tim, *Cpldcpu/MisguidedAttention*, January 2025, accessed January 15, 2025.

34. Peter Clark et al., "Think you have solved question answering? try arc, the ai2 reasoning challenge," *arXiv preprint arXiv:1803.05457*, 2018,

35. Yonatan Bisk et al., *PIQA: Reasoning about Physical Commonsense in Natural Language*, arXiv:1911.11641, November 2019, accessed January 11, 2025, <https://doi.org/10.48550/arXiv.1911.11641>, arXiv: 1911.11641 [cs].

36. Karl Cobbe et al., *Training Verifiers to Solve Math Word Problems*, November 18, 2021, accessed January 11, 2025, <https://doi.org/10.48550/arXiv.2110.14168>, arXiv: 2110.14168 [cs], <http://arxiv.org/abs/2110.14168>.

37. Rowan Zellers et al., *HellaSwag: Can a Machine Really Finish Your Sentence?*, arXiv:1905.07830, May 2019, accessed January 11, 2025, <https://doi.org/10.48550/arXiv.1905.07830>, arXiv: 1905.07830 [cs].

38. Dan Hendrycks et al., *Measuring Massive Multitask Language Understanding*, arXiv:2009.03300, January 2021, accessed January 11, 2025, <https://doi.org/10.48550/arXiv.2009.03300>, arXiv: 2009.03300 [cs].

39. Keisuke Sakaguchi et al., *WinoGrande: An Adversarial Winograd Schema Challenge at Scale*, arXiv:1907.10641, November 2019, accessed January 11, 2025, <https://doi.org/10.48550/arXiv.1907.10641>, arXiv: 1907.10641 [cs].

40. Shahriar Golchin and Mihai Surdeanu, *Time Travel in LLMs: Tracing Data Contamination in Large Language Models*, arXiv:2308.08493, February 2024, accessed January 8, 2025, <https://doi.org/10.48550/arXiv.2308.08493>, arXiv: 2308.08493 [cs].

41. Changmao Li and Jeffrey Flanigan, *Task Contamination: Language Models May Not Be Few-Shot Anymore*, arXiv:2312.16337, December 2023, accessed January 8, 2025, <https://doi.org/10.48550/arXiv.2312.16337>, arXiv: 2312.16337 [cs].

42. Iman Mirzadeh et al., "Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models," *arXiv preprint arXiv:2410.05229*, 2024,

again suggesting that there has been dataset contamination. Furthermore, they also found that the models exhibit significant variance on the randomized questions, meaning that the models are much more likely to answer correctly for certain specific instantiations of the questions, while the reasoning steps involved are exactly the same. This research is corroborated by the work of Jiang et al.⁴³ who find that the performance of SOTA LLMs on many reasoning tasks drops significantly when simple alterations are made to the way the question is raised. As an example, they consider the “twenty-five horses” problem, which is a classic problem in graph theory. The contents of this problem need not concern us, but what is interesting is that the model solves the problem quite well when it is phrased with horses; however, when “horse” is exchanged for “bunny” the performance drops dramatically. This indicates that the LLM has learned the answer to the twenty-five horses problem because it has seen it phrased like that many times in its training data; however, it has never seen the problem phrased in terms of bunnies. The conclusion is that the LLM has not learned the underlying reasoning of the problem, but has only learned to recognize the problem and copy the answer. Jiang et al. call this phenomenon token bias and state:

A strong token bias suggests that the model is relying on superficial patterns in the input rather than truly understanding the underlying reasoning task.⁴⁴

In addition to the experiments mentioned above, Mirzadeh et al.⁴⁵ investigated more specifically with which questions the LLMs struggle. They found that the LLM performance drops steeply when the amount of reasoning steps increases. This is also reported by Dziri et al.⁴⁶ who find that LLMs struggle to break down reasoning problems requiring multiple reasoning steps. They hypothesize that this could be due to LLMs memorizing patterns often occurring in the data instead of learning the actual reasoning involved as this places a bias on less complex problems which are more often encountered in the training data. These results are corroborated by the more general study into the capacity of LLMs to solve planning problems consisting of multiple steps by Kambhampati et al.⁴⁷

Finally, Mirzadeh et al.⁴⁸ also attempted to throw the LLMs off balance by adding superfluous information to the questions. This seemingly innocuous procedure, however, has significant effect on the LLM's performance, with a 32 percent accuracy drop for GPT-4o, and accuracy drops of up to 65.7 percent for other models. Similar results were found by Shi et al.⁴⁹ who did also find that the deterioration could be alleviated by prompting the LLM to take extra care in its reasoning and providing it with question example pairs in which there is also superfluous information given in the questions.

43. Bowen Jiang et al., “A Peek into Token Bias: Large Language Models Are Not Yet Genuine Reasoners,” October 4, 2024, accessed January 13, 2025, <https://doi.org/10.48550/arXiv.2406.11050>, arXiv: 2406.11050 [cs], <http://arxiv.org/abs/2406.11050>.

44. Jiang et al., pg. 1.

45. Mirzadeh et al., “Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models.”

46. Nouha Dziri et al., *Faith and Fate: Limits of Transformers on Compositionality*, arXiv:2305.18654, October 2023, accessed January 13, 2025, <https://doi.org/10.48550/arXiv.2305.18654>, arXiv: 2305.18654 [cs].

47. Karthik Valmeekam et al., *PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change*, arXiv:2206.10498, November 2023, accessed January 13, 2025, <https://doi.org/10.48550/arXiv.2206.10498>, arXiv: 2206.10498 [cs]; Subbarao Kambhampati, “Can Large Language Models Reason and Plan?,” *Annals of the New York Academy of Sciences* 1534, no. 1 (April 2024): 15–18, ISSN: 0077-8923, 1749-6632, accessed January 13, 2025, <https://doi.org/10.1111/nyas.15125>.

48. Mirzadeh et al., “Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models.”

49. Freda Shi et al., *Large Language Models Can Be Easily Distracted by Irrelevant Context*, arXiv:2302.00093, June 2023, accessed January 13, 2025, <https://doi.org/10.48550/arXiv.2302.00093>, arXiv: 2302.00093 [cs].

These results lead Mirzadeh et al. to hypothesize that current LLMs do not exhibit formal reasoning:

The high variance in LLM performance on different versions of the same question, their substantial drop in performance with a minor increase in difficulty, and their sensitivity to inconsequential information indicate that their reasoning is fragile. It may resemble sophisticated pattern matching more than true logical reasoning.⁵⁰

Strengths of LLM Reasoning We have presented a large amount of experimental evidence that points out the flaws of LLM reasoning and suggests that LLMs behave like stochastic parrots, at least up to some level. While there is undoubtedly much more similar experimental evidence to be found, the main points of criticism and concern have been established. Before we recapitulate and draw conclusions, we shall also highlight some success stories about GPT-4 that indicate its powers. While the literature on this is also vast, we shall restrict ourselves to two papers. The first is a large overview paper by Sébastien Bubeck et al.⁵¹ investigating the broad capacities of GPT-4. They observe many of the same pitfalls we discussed previously but also highlight some of the remarkable abilities of GPT-4. For instance, they gave GPT-4 four tasks which were meant to combine different domains in a way that is unlikely to have been found in the training data:

1. Produce JavaScript code that generates random images in the style of the painter Kandinsky.
2. Compose a proof demonstrating that there are infinitely many prime numbers, written in the style of Shakespeare.
3. Write a letter endorsing Electron as a U.S. presidential candidate, authored by Mahatma Gandhi and addressed to his wife.
4. Produce Python code for a program that evaluates a patient's risk of diabetes based on inputs like age, sex, weight, height, and blood test result.

They concluded the following based on the results produced by GPT-4 when given the above prompts:

These examples suggest that GPT-4 has not only learned some general principles and patterns of different domains and styles but can also synthesize them in creative and novel ways.⁵²

While Bubeck et al. are right in stating that the results produced by the GPT-4 for the above questions are novel, the results to these questions do not appear to be very surprising; although it is hard to deny that they at least have some level of surprise as there is no step-by-step plan for writing mathematical proofs in Shakespearean prose. Because of this, we do agree with the conclusion that these products are creative, albeit only slightly.

The second paper we shall consider is by Pan et al.⁵³ and investigates whether GPT-4 is capable of performing Hartree-Fock computations, which are difficult graduate-level physics computations.

50. Mirzadeh et al., "Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models," pg. 12.

51. Bubeck et al., *Sparks of Artificial General Intelligence*.

52. Bubeck et al., pg. 13.

53. Pan et al., *Quantum Many-Body Physics Calculations with Large Language Models*.

They provide the LLM with a step-by-step plan that guides it through the computation. However, the step-by-step plan is not complete as there are certain properties of the physical system that should be entered into the plan for each specific computation. Pan et al. then took 15 research papers in which Hartree-Fock computations are performed and filled in the details of the step-by-step plan. They then tasked GPT-4 with performing the calculation. Experts then graded the answers of the LLM and arrived at a score of 87.5 out of 100 on the execution of the individual steps in the calculation. 5 out of the 15 research papers were beyond the cut-off date of the LLM, and the score of the LLM was not worse on these calculations, indicating that the LLM did not memorize these specific papers. Furthermore, Pan et al. also investigated how well the LLM could fill in the details of the step-by-step plan itself when given the abstract of the paper and how well the LLM could grade its own answers and good results were achieved. While the LLM is guided through the computation by the extensive step-by-step plan, this research does suggest that LLMs do have some capacity for doing graduate-level physics reasoning.

Inside the Black Box All of the experiments we considered in this section treat the LLM as a black box. From this perspective one can only find indirect evidence for whether LLMs perform actual reasoning or merely memorize the results. However, in order to find more direct evidence one needs to “look inside” the black box. Due to huge size of current LLMs, it is very difficult to identify precisely how an LLM arrives at its output. However, some preliminary studies have been conducted. For instance, Nikankin et al.⁵⁴ investigated what internal LLM processes contribute to solving basic arithmetic problems. They found a large number of internal processes that encode simple heuristics to solve arithmetic problems. For instance, they found an internal process of the LLM that is highly activated when given subtraction prompts with results between 150 and 180 and correctly solves these problems. This indicates that LLMs are somewhere between memorizing individual prompts and having general-purpose arithmetic reasoning. Similar results were reported for other LLMs: for instance, a simple LLM trained to play the game of Othello was also found to rely on a large number of heuristics or rules to determine its moves in the game.⁵⁵ Another experiment shows that similar results hold for LLMs trained on a large variety of tasks, such as navigation, game playing, and logic puzzles.⁵⁶

Let us recapitulate our findings: we have seen two faces of GPT-4 (and other SOTA LLMs), on the one hand it is able to perform well on an incredibly wide range of difficult reasoning tasks, but on the other hand it still fails at trivial tasks such as counting, basic arithmetic, and reasoning. We have seen evidence that suggests that a large part of LLM reasoning is based on memorization, and many authors suggest that the creation process of LLMs more closely resembles probabilistic pattern matching than formal reasoning. While the studies we have investigated show that LLM performance drops significantly on reasoning benchmarks after the cut-off date or on variations of tasks not found in the training data, it is important to remark that the performance does not drop to 0. This implies that LLMs are able to solve problems previously unseen in the training data. This suggests

54. Yaniv Nikankin et al., *Arithmetic Without Algorithms: Language Models Solve Math With a Bag of Heuristics*, arXiv:2410.21272, October 2024, accessed January 25, 2025, <https://doi.org/10.48550/arXiv.2410.21272>, arXiv: 2410.21272 [cs].

55. jylin04 et al., “OthelloGPT Learned a Bag of Heuristics,” July 2024, accessed March 15, 2025.

56. Keyon Vafa et al., *Evaluating the World Model Implicit in a Generative Model*, arXiv:2406.03689, November 2024, accessed March 15, 2025, <https://doi.org/10.48550/arXiv.2406.03689>, arXiv: 2406.03689 [cs].

that LLMs have some capacity for reasoning, albeit not the strict formal reasoning exhibited by humans. For instance, in the case of arithmetic, it appears that LLMs are able to learn simple rules or heuristics, e.g., how to subtract two numbers within a given range. This way of arithmetic reasoning is decidedly different from that of humans, but it is more sophisticated than memorizing all answers. However, even this rudimentary form of reasoning is a form of reasoning and therefore contradicts the stochastic parrot hypothesis. Therefore, there is strong evidence that the stochastic parrot hypothesis is incorrect for current SOTA LLMs. Interestingly, the papers showing that LLMs use heuristics are also used as counterarguments against the claim that LLMs construct a world model because these heuristics are quite brittle in the face of novel problems.⁵⁷ The evidence therefore points to LLMs being more than stochastic parrots but falling short of possessing a robust world model.

Much of the conclusion also applies to the creativity of LLMs: a large part of their success on a variety of tasks is due to memorization or statistical copying from the training data. However, LLMs can still do well on problems they have not encountered before. The solutions to these problems are at the least novel with respect to the LLM's training data so LLMs appear to be capable of producing I-creative products. The question remains how surprising these products are. For the examples we have seen, the answer is not very; while it is somewhat surprising to see a mathematical proof given in Shakespearean prose, we can certainly not call this a transformation of some conceptual space.

3.2.2 Creativity

In the previous section we investigated LLM reasoning, which we found to be closely related to creativity. There has also been much work conducted that questions more directly whether LLMs can currently produce creative products that are not limited to products of reason, such as research ideas. We shall explore part of this work here. One paper by Thomas McCoy et al.⁵⁸ investigates how much early LLMs such as GPT-2 really copy from their training data. They find that on the whole, LLM-created texts are quite novel, although they sometimes do copy large pieces of text from the training data. For instance, GPT-2 coins several types of new words such as *Swissified* or *IKEA-ness* and that most of its generated sentences have a syntactic structure different from all of the sentences in the training data. They come to the following conclusion: “neural language models do not simply memorize; instead they use productive processes that allow them to combine familiar parts in novel ways.”⁵⁹

There has been much work done on testing LLM creativity on a variety of tasks; see, for instance, the overview by Ismayilzada.⁶⁰ To keep this chapter focused we shall focus on work investigating the *scientific creativity* of LLMs. In the previous section we mainly looked at reasoning. A different aspect of creativity we shall now explore is *idea generation*. There have been multiple proposals for using LLMs to assist in this part of the scientific process. One of these is called the ResearchAgent,⁶¹

57. Melanie Mitchell, *LLMs and World Models, Part 2*, Substack Newsletter, February 2025, accessed March 15, 2025.

58. R. Thomas McCoy et al., *How Much Do Language Models Copy from Their Training Data? Evaluating Linguistic Novelty in Text Generation Using RAVEN*, arXiv:2111.09509, November 2021, accessed December 15, 2024, <https://doi.org/10.48550/arXiv.2111.09509>, arXiv: 2111.09509 [cs].

59. McCoy et al., pg. 2.

60. Mete Ismayilzada et al., *Creativity in AI: Progresses and Challenges*, arXiv:2410.17218, October 2024, accessed November 4, 2024, arXiv: 2410.17218.

61. Jinheon Baek et al., “ResearchAgent: Iterative Research Idea Generation over Scientific Literature with Large Language

introduced by Baek et al. In this proposal, an LLM is given a scientific paper as input. It is also given a number of papers related by citations. It is then tasked to use these papers to create a novel scientific research idea. The performance of the ResearchAgent given a number of different research papers was tested by humans and LLMs. The proposed research ideas are graded on a number of different scales such as originality, feasibility, and significance. The proposed ideas obtain good grades across the board, with a score above 4 out of 5 on all scales. The ideas score particularly high scores on originality, significance, and relevance. On more practical scores such as feasibility and clarity, the ResearchAgent scored lower grades. This indicates that the ideas produced by the LLMs are surprising in some sense. However, the question remains how valuable these ideas are as the relatively low scores on feasibility indicate that the ideas produced will not easily lead to valuable research projects.

Similar results were found by other authors: Castelo et al.⁶² found that LLM-generated ideas for new products were judged as more creative by humans than ideas produced by other humans. Meincke et al.⁶³ found that LLM-generated product ideas were less creative than human-generated ideas on average but found that the most highly judged ideas were almost always LLM-generated. Finally, Si et al.⁶⁴ also investigated how well LLMs can generate novel research ideas. They employed similar techniques for generating research ideas with LLMs as the ResearchAgent, but also compared the generated ideas with human-generated ideas and concluded that LLM-generated research ideas were more novel but less feasible than research ideas generated by humans. They did highlight a few further limitations of LLMs: firstly, when using this framework to generate 4000 different ideas, they found that only 200 out of 4000 ideas were unique, showing that the diversity of the LLM-generated ideas is limited even though the individual ideas are on average judged to be more novel. Furthermore, they found, in contrast with the results reported by Ismayilzada, that LLMs are not good evaluators of research ideas as their results are not very consistent with those of human evaluators. Finally, they also found that the LLM-generated ideas were plagued by vague implementation details, unrealistic assumptions, and bad motivation. This suggests that human-generated ideas are more grounded and focused on common problems and that humans prioritize feasibility over novelty and excitement. So while LLMs can produce ideas that are judged to be more creative than human-produced ideas, it remains to be seen if these ideas are actually valuable or if they are just novel while not being feasible to implement. Another problem is that the above examples mainly deal with accessible fields such as machine learning and product generation. In other highly specialized fields in mathematics or physics coming up with interesting questions that can plausibly be answered is already a large part of research. It remains to be seen if LLMs can perform well in these domains.

There is much more research in this area, including proposals such as the AI Scientist⁶⁵ that aims to fully automate the scientific process, starting from machine learning research ideas, then

Models," 2024, accessed February 1, 2025, <https://doi.org/10.48550/ARXIV.2404.07738>.

62. Noah Castelo et al., *How AI Outperforms Humans at Creative Idea Generation*, SSRN Scholarly Paper, 4751779, Rochester, NY, March 2024, accessed February 1, 2025, <https://doi.org/10.2139/ssrn.4751779>, Social Science Research Network: 4751779.

63. Lennart Meincke et al., *Using Large Language Models for Idea Generation in Innovation*, SSRN Scholarly Paper, 4526071, Rochester, NY, September 2024, accessed February 1, 2025, <https://doi.org/10.2139/ssrn.4526071>, Social Science Research Network: 4526071.

64. Chenglei Si, Diyi Yang, and Tatsunori Hashimoto, *Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers*, arXiv:2409.04109, September 2024, accessed February 1, 2025, <https://doi.org/10.48550/arXiv.2409.04109>, arXiv: 2409.04109 [cs].

65. Lu et al., *The AI Scientist*.

implementing the code required to perform the necessary experiments and finally writing a research paper about the results, but the main strengths and weaknesses of LLMs of interest to us have already been put forward in this and the previous section.

The main conclusion that we can draw from these investigations is that LLMs are able to produce basic combinatorial and exploratory scientific creative products in certain conceptual spaces. Transformative creative products are still in another ballpark altogether and it is unclear if current LLM architectures can achieve this. We can therefore conclusively say that LLMs are not merely “stochastic parrots.” While much research has been conducted, there is still a need for a more focused benchmark for scientific LLM creativity, that is based on a solid philosophical definition, and that can test creativity for specific scientific subdomains.

3.3 Agency

In the previous section we have established that LLMs are capable of producing creative products. However, we have not yet discussed *how* they create these products. In the previous chapter we defined creativity to be the capacity to create creative products *with flair*, which applies to the process of creation. The purpose of this section is to evaluate the flair of LLMs. In the previous chapter, on page 17, we put forward Gaut’s definition of flair which is described by *purpose, understanding, judgement, and continuous evaluation*.

LLMs do act with purpose as they answer questions asked by their users. The outputs of LLMs are therefore not accidental, but are generated with the purpose of producing a desired answer to the question. However, the purpose is always external to the LLM. This means it cannot come up with its own questions (unless specifically tasked to do this). This is a limitation, especially for transformational creativity, as coming up with the right question is often already a big part of these types of creative products. With regard to scientific understanding, the second aspect of flair, recent work by Barman et al.⁶⁶ has suggested a template for benchmarking scientific understanding in LLMs that is based on the ability of LLMs to answer a range of scientific questions. Further work is needed to implement this benchmark at scale. This benchmark is based on the definition of understanding given in Chapter 2 on page 18. With regards to the execution of pertinent judgement, i.e. the application of rules or constraints when solving certain problems, we find that, in science, reasoning is an important facet of this domain of agency. In section 3.2.1, we found that LLMs still struggle to reason accurately. Interestingly, the prompt engineering techniques we discussed in the previous section prompt the LLM to execute more judgement, i.e. by being explicit about which logical rules or steps to apply. The result is that LLMs, when prompted in this way, produce much better and more creative products, corroborating the conclusion of the previous chapter that agency is an important part of the capacity to create creative products. The aspect of agency that is mainly missing in LLMs is continuous evaluation.

We shall now spend some more time considering the aspect of continuous evaluation. This aspect is inherently missing from LLMs at base as they provide a single output for the question.

66. Kristian Gonzalez Barman et al., “Towards a Benchmark for Scientific Understanding in Humans and Machines,” *Minds and Machines* 34, no. 1 (April 2024): 6, ISSN: 1572-8641, accessed February 2, 2025, <https://doi.org/10.1007/s11023-024-09657-1>, arXiv: 2304.10327 [cs].

This is a severe limitation as LLMs cannot iterate upon their solution and are restricted to being right on the first try. There have been many proposals put forward to remedy this problem in LLMs. For instance, Nye et al.⁶⁷ introduce “scratchpads” for LLMs. Their proposal entails that LLMs are prompted to generate tokens for all steps in solving the problem. These intermediate tokens are then given as context for the LLM while it is generating the next steps in the solution. This means that the LLM can “remember” the previous steps in its solution. Nye et al. implement this idea and show that LLM performance increases dramatically on tasks such as executing Python code.

In the previous section, we also saw a few proposals that increase the amount of iterative evaluation performed by LLMs. For instance, the ResearchAgent⁶⁸ uses different large language models to provide feedback on proposed research ideas. This feedback is then fed to the LLM that is asked to improve the research idea based on this feedback. This process significantly improves the proposed research ideas. This basic process of iteration based on feedback lies at the heart of a currently emerging class of modified LLMs called large reasoning models (LRMs) which have much better performance and creativity than LLMs at base. We start by introducing the precursors of LRMs, starting with *FunSearch* introduced by Romera-Paredes et al.⁶⁹

FunSearch is an algorithm that utilizes LLMs such as GPT-4 to answer certain mathematical problems. The problems the algorithm can tackle have to allow for an external evaluator. By external evaluator we mean a program that efficiently assigns a value to a solution. This value can then be used to compare solutions and select the best ones. The problem that was first tackled by *FunSearch* is the cap set problem. The precise mathematical details of this problem are not important, the only thing we need to understand is that the problem consists of generating a sequence of numbers, satisfying certain properties, called cap sets, that is as long as possible. In *FunSearch*, the chosen LLM is prompted to generate a number of different samples of Python code, which should output cap sets as large as possible when run. The external evaluator then runs these programs and first validates whether the output sequences are really cap sets. If they are cap sets, the external evaluator records the length. This length is the value of the program as we are looking for the largest sequence we can find. All valid programs producing cap sets are subsequently stored in a database. These programs are then iteratively refined using a genetic algorithm approach. In this approach, multiple subpopulations, consisting of programs, are maintained. At every iteration step, every program within a subpopulation is updated. This update is done by selecting two programs within the subpopulation, preferring programs that are shorter and that perform better, and asking the LLM to update the programs using inspiration from the selected programs. In order to ensure information flows between the subpopulations, every so-many iterations half of the worst performing programs are culled and replaced by better performing programs from other subpopulations. This process ensures that the LLM can build upon its solutions, iteratively improving them.

The results are spectacular: for a particular instantiation of the cap set problem, the LLM was able to find a solution larger than all previous solutions; a historically novel product. Furthermore, as the solution consists of Python code, it is also much more interpretable than if the LLM had merely outputted the numbers. Jordan Ellenberg, who contributed to *FunSearch* and is a Professor

67. Maxwell Nye et al., *Show Your Work: Scratchpads for Intermediate Computation with Language Models*, arXiv:2112.00114, November 2021, accessed February 2, 2025, <https://doi.org/10.48550/arXiv.2112.00114>, arXiv: 2112.00114 [cs].

68. Baek et al., “ResearchAgent.”

69. Romera-Paredes et al., “Mathematical Discoveries from Program Search with Large Language Models.”

in Mathematics at the University of Wisconsin-Madison, had the following to say about the solutions:

FunSearch offers a completely new mechanism for developing strategies of attack. The solutions generated by FunSearch are far conceptually richer than a mere list of numbers. When I study them, I learn something.⁷⁰

A swift stream of improvements and generalizations ensued upon the publication of FunSearch. For instance, Ma et al.⁷¹ and Shojaee et al.⁷² consider a generalization in which the LLM is tasked to fit an equation to a given dataset. Ma et al. consider physical simulations such as fluid dynamics while Shojaee et al. consider mathematical equations such as oscillators. Their LLM-based systems function quite similarly. They start by letting the LLM create a number of different Python programs to model the datasets. In contrast to FunSearch, this Python code is now allowed to have free parameters. In the next step they use basic gradient descent to fit the optimal parameters. This results in a score that measures how well the Python program with optimal parameters fits the data. Like in FunSearch, they then perform an evolutionary algorithm that keeps the best-performing Python programs while iteratively asking the LLM to improve the code, drawing inspiration from the other generated programs. Both Ma et al. and Shojaee et al. find that their system works very well for data fitting. Qualitatively, they also observe interesting behavior of the LLM during optimization: they find that the progress of LLM on fitting the data is not linear, i.e., the LLM does not fit the data a little bit better at every step. Instead, the performance of the programs increases by sudden leaps at certain steps, while not changing much between leaps. For the equation fitting tasks the leaps occur when the LLM correctly identifies a new term of the equation. This brings to mind the *spontaneity* discussed previously as an alternative agency condition by Maria Kronfeldner:⁷³ the jumps in progress are not due to routine procedure, then they would have come sooner; instead the jumps arise spontaneously from a series of chance events such as slight reformulations of the prompt or generated code or the inherent stochasticity of LLMs. There is much more work in this direction in which external evaluators are used in conjunction with evolutionary algorithms to improve LLM outputs through iteration.⁷⁴

An important limitation of these techniques is that they require an external evaluator. We also encountered this in the examples of the previous chapter, where computer programs were able to produce creative products for tasks where external evaluators are available. However, if we consider the advancement of science, external evaluators are not readily available; it is not so easy to evaluate whether a scientific argument is correct and even harder to evaluate whether it is meaningful or not. In order to promote flair in LLMs, one then needs to develop general-purpose evaluators. Development in this area led to the creation of o1,⁷⁵ the current SOTA model produced by OpenAI. Before we address the capabilities of this model, we shall discuss some of the innovations that have

70. *FunSearch: Making New Discoveries in Mathematical Sciences Using Large Language Models*, <https://deepmind.google/discover/blog/funsearch-making-new-discoveries-in-mathematical-sciences-using-large-language-models/>, December 2024, accessed February 2, 2025.

71. Pingchuan Ma et al., *LLM and Simulation as Bilevel Optimizers: A New Paradigm to Advance Physical Scientific Discovery*, arXiv:2405.09783, May 2024, accessed June 5, 2024, <https://doi.org/10.48550/arXiv.2405.09783>, arXiv: 2405.09783 [cs].

72. Parshin Shojaee et al., *LLM-SR: Scientific Equation Discovery via Programming with Large Language Models*, arXiv:2404.18400, June 2024, accessed June 5, 2024, <https://doi.org/10.48550/arXiv.2404.18400>, arXiv: 2404.18400 [cs].

73. Kronfeldner, "Creativity naturalized."

74. Xingyu Wu et al., "Evolutionary Computation in the Era of Large Language Model: Survey and Roadmap," 2024, accessed February 5, 2025, <https://doi.org/10.48550/ARXIV.2401.10034>.

75. *Learning to Reason with LLMs*, <https://openai.com/index/learning-to-reason-with-llms/>, accessed February 5, 2025.

likely played a role in its success, although the precise details of this model have not been released to the public. In particular we shall focus on the rapid progress that is being made in developing general-purpose external evaluators.

One obvious possibility for a general-purpose evaluator is to let LLMs evaluate themselves, although it is not obvious if LLMs can evaluate themselves effectively. Such a technique was first tested by Yao et al.⁷⁶ and is called Tree of Thoughts (ToT). This technique starts by letting LLMs create multiple answers for the same question. The LLM is then asked to rank the answers. The worst-performing answers are eliminated and the LLM is asked to generate new answers based on the answers that were not kicked out. This process is repeated a number of times until the best answer is chosen. This method was shown to significantly increase the performance of LLMs on complex reasoning tasks. For instance, when tasked with solving certain mathematical problems, the success rate with the Chain of Thought technique we discussed previously was just 4%, however the success rate with ToT increased to 74%.

A different approach is to make use of an external model to evaluate the answer of the LLM. An evaluator that only evaluates the answer produced by the LLM is called an Outcome-Supervised Reward Model (ORM). It has been found, however, that it is even better to use an external model that evaluates the entire reasoning process;⁷⁷ this is called a Process Reward Model (PRM). These models are trained to measure how good a particular step is for reaching the desired goal of a calculation or reasoning problem. There are many different approaches for training PRMs,⁷⁸ but we shall not go into the technical details. The important takeaway is that it is possible to train external evaluators that evaluate the reasoning steps of LLMs while they are generating answers. We already saw that ToT significantly increases the reasoning capacities of LLMs; similarly one finds that implementing PRMs for LLMs also significantly increases their reasoning capacities.⁷⁹

The field of using external evaluators to increase LLM performance through iterative improvement is rapidly emerging. We already remarked that these models have greatly increased performance on complex reasoning tasks. Now we shall give some examples of this increased performance, with a focus on o1. Li et al.⁸⁰ test o1 on problems from the International Mathematics Olympiad, which are publicly available so that they could have occurred in the training data, and on problems from the Chinese National Team Training camp, which are not public and less likely to have been found in the training data. They find that o1 has solid improvements on both sets of questions, indicating that the improvement is not merely due to memorization. Shahriar et al.⁸¹ show that o1 is able to solve doctoral-level problems in a variety of fields. De Winter et al.⁸² find that o1 is able to score nearly perfectly on mathematics exams. There are many more results like this that underline the

76. Shunyu Yao et al., *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*, arXiv:2305.10601, December 2023, accessed February 5, 2025, <https://doi.org/10.48550/arXiv.2305.10601>, arXiv: 2305.10601 [cs].

77. Hunter Lightman et al., *Let's Verify Step by Step*, arXiv:2305.20050, May 2023, accessed February 5, 2025, arXiv: 2305.20050 [cs].

78. Fengli Xu et al., *Towards Large Reasoning Models: A Survey of Reinforced Reasoning with Large Language Models*, arXiv:2501.09686, January 2025, accessed February 5, 2025, <https://doi.org/10.48550/arXiv.2501.09686>, arXiv: 2501.09686 [cs].

79. Xu et al.

80. Lu et al., *The AI Scientist*.

81. Sakib Shahriar, *A Cross-Domain Performance Report of Open AI ChatGPT O1 Model*, 2024121930, December 2024, accessed February 5, 2025, <https://doi.org/10.20944/preprints202412.1930.v1>, Preprints: 2024121930.

82. Joost de Winter, Dimitra Dodou, and Yke Bauke Eisma, "System 2 Thinking in OpenAI's O1-Preview Model: Near-perfect Performance on a Mathematics Exam," *Computers* 13, no. 11 (October 2024): 278, ISSN: 2073-431X, accessed February 5, 2025, <https://doi.org/10.3390/computers13110278>, arXiv: 2410.07114 [cs].

significant reasoning improvements of o1 over GPT-4o.⁸³ While the improvements are impressive, the difficulties LLMs have with reasoning are not all solved by o1. For instance, while o1 does much better on the PlanBench benchmark,⁸⁴ for complex planning tasks, it is still far from a perfect score. McCoy et al.⁸⁵ also report that o1 still does much better on problems that occur often in the training set, implying that o1, like GPT-4o, is still relying, at least up to some point, on memorization.

These results imply that the improvements of o1 over GPT-4o are quantitative, but most likely not qualitative; while o1 reasoning is more robust than that of GPT-4o, it is still not the coherent deterministic reasoning that allows for humans to progressively solve reasoning problems consisting of a large number of steps. It remains to be seen what further developments in this young field will bring, but we can draw the following conclusions, mirroring our conclusions at the end of the previous chapter: access to an external evaluator significantly improves the creative capabilities of LLMs. The resulting LRMs have more agency than LLMs at base. However, it does appear as if these improvements are quantitative instead of qualitative so that transformational creativity still currently appears to be out of reach for LRMs. But it must be admitted that LRMs or LLM systems, such as FunSearch, coupled with external evaluators have tremendous potential for producing creative products. In the case of FunSearch, an H-creative product was produced, a cap set of a certain size that was not known to exist beforehand. As concluded by Prof. Ellenberg this product holds value, not only because it is the largest cap set but also because of its mathematical structure, which can be analyzed to advance human insight. As evidenced by the improvements in LLM performance engendered by supplementing the LLM with continuous evaluation, it appears that there is still much to be gained in terms of LLM agency. For instance, when it comes to science, it is still unclear how much scientific understanding LLMs have. Benchmarking and improving the scientific understanding of LLMs could further propel their capacity to do science.

3.4 The Verdict

In this chapter we have investigated the question whether LLMs are capable of being creative. We firstly investigated the weaker claim that LLMs are capable of producing creative products. This question is closely related to an ongoing debate about how LLMs can be so successful. We sketched two positions in this debate: one side that claims that LLMs are stochastic parrots, which just copy statistically likely pieces of text, and the other side that claims that LLMs construct internal world models. Both positions have different implications for the capacity for LLMs to produce creative products: stochastic parrots cannot produce products beyond their training data and are therefore incapable of producing surprising products. On the other hand, if LLMs have world models, they have some abstract grasp of the causal relations of objects in the world which would render them able to produce surprising products beyond their training data. We found that the evidence conclusively

83. Ehsan Latif et al., *A Systematic Assessment of OpenAI O1-Preview for Higher Order Thinking in Education*, arXiv:2410.21287, October 2024, accessed February 5, 2025, <https://doi.org/10.48550/arXiv.2410.21287>, arXiv: 2410.21287 [cs]; Ernest Davis, *Testing GPT-4-o1-preview on Math and Science Problems: A Follow-up Study*, arXiv:2410.22340, October 2024, accessed February 5, 2025, <https://doi.org/10.48550/arXiv.2410.22340>, arXiv: 2410.22340 [cs].

84. Karthik Valmееkam, Kaya Stechly, and Subbarao Kambhampati, "LLMs Still Can't Plan; Can LRMs? A Preliminary Evaluation of OpenAI's O1 on PlanBench," 2024, accessed January 13, 2025, <https://doi.org/10.48550/ARXIV.2409.13373>.

85. R. Thomas McCoy et al., *When a Language Model Is Optimized for Reasoning, Does It Still Show Embers of Autoregression? An Analysis of OpenAI O1*, arXiv:2410.01792, October 2024, accessed February 5, 2025, <https://doi.org/10.48550/arXiv.2410.01792>, arXiv: 2410.01792 [cs].

shows that LLMs are more than stochastic parrots as there are many examples of LLMs producing novel and somewhat surprising products. Furthermore, preliminary studies looking “inside” the black box of LLMs suggest that LLMs learn a large set of heuristics to solve a variety of problems, which is definitively different from memorizing answers. However, there is also much evidence to support that LLMs still behave quite like stochastic parrots in some regards. For instance, simply changing words in well-known riddles, such as changing “horse” to “bunny,” in the twenty-five horses problem, decreases performance, indicating that LLMs rely on memorization of these riddles. Furthermore, it is clear that the heuristics found inside the “black box” by Nikankin et al. do not constitute a robust world model like that of humans. We can therefore conclude that LLMs are not principally incapable of producing creative products, however much of their current success is still due to memorization and not due to a robust internal world model. This shows that there is still much room for improvement.

In the previous chapter we distinguished between a producer’s capacity for producing creative products and the producer’s being creative. This distinction lies in the way in which the product is created. If the producer produces creative products with flair, which is a measure of the producer’s agency, we say that the producer is creative. The definition of flair given on page 17 states that flair corresponds to the following features: purpose, understanding, judgement and continuous evaluation. The purpose of LLMs lies in their relation to the user: a user supplies the LLM with a prompt and the LLM attempts to produce a desirable answer. The problems LLMs have with reasoning in science imply that LLMs also struggle with scientific understanding. To make this more precise, recent proposals have been put forward to benchmark this scientific understanding more thoroughly. LLMs execute judgements when answering prompts, and many of the prompt engineering techniques designed to increase the quality of LLM outputs are designed to have LLMs perform more judgement. LLMs are still lacking in continuous evaluation. This is largely because they are designed to produce one output for a question. As a consequence, LLMs are incapable of iterating or improving their answers. Currently much progress is being made in improving this aspect of LLMs by pairing LLMs with models that evaluate the output of LLMs. This allows LLMs to iteratively improve their solutions. These iterative systems are LRMs and have dramatically increased performance on a large variety of tasks; in particular they are capable of producing more surprising products. While more systematic study is needed to accurately evaluate the agency of LLMs, it is clear that they have more agency than the computer programs considered in Section 2.3. LRMs have even more agency, which also increases their capacity to produce creative products.

Our final verdict is therefore that LLMs can produce creative products with limited agency. LLMs are therefore creative and in particular scientifically creative, given that we saw examples of LLM creativity in mathematics and physics. However, there is still tremendous room for improvement: more robust internal world models could allow LLMs to reason more robustly and perform better on novel tasks. Likewise, further increasing LLM agency will increase LLM creativity further.

Chapter 4

Conclusion

In this thesis we investigated whether LLMs can exhibit scientific creativity. To do this we first put forward a definition of creativity that does not preclude AI creativity. Concretely, we argued in Chapter 2 that we should distinguish between a *product* being creative and the *producer* being creative. A product is creative if it is *novel*, *valuable*, and *surprising*. We call the producer creative if the producer produces creative products with *flair*. Flair is defined by Gaut as a measure of the agency of the producer and is characterized by *purpose*, *understanding*, *the execution of judgement sensitive to the domain*, and *continuous evaluation*. We then found that prior to LLMs, computer programs were already capable of producing creative products. However, the agency of these early programs is lacking.

In Chapter 3 we focused on LLMs. We started by exploring the position that LLMs are stochastic parrots, which merely copy and paste statistically likely pieces of text without being able to answer questions whose answers cannot be found in their training data. The main argument for this position is based on the Chinese room argument: LLMs can only access the form of language but are fundamentally incapable of understanding the meaning of language. This incapacity implies that LLMs can never reason, or produce surprising products. We then proceeded by carefully evaluating experiments performed to gauge the potential of LLMs for reasoning and producing creative products. We found that, while LLMs do behave like stochastic parrots up to some point, they are capable of producing novel products. Furthermore, preliminary attempts at looking “inside” the black boxes that are LLMs, suggest that LLMs learn a large set of heuristics to solve problems. This is decidedly different from memorizing answers. Besides this, we investigated the agency of LLMs. In this regard, much work needs still to be done, in particular more thorough investigations of scientific understanding in LLMs should be undertaken. One aspect where LLMs are severely lacking is continuous evaluation as LLMs only produce a single output to a prompt, without any iteration. This lack is currently being addressed by the AI community through the emerging field of LRMs. These models pair an LLM with an evaluator that scores how well the produced answer answers the prompt. This then allows the LLM to iterate and improve its answer, yielding vastly improved results and more creative products.

The final verdict is that LLMs and LRMs are currently somewhat creative. In particular, there is no fundamental reason that prevents LLMs from being creative. However, there still is much to improve, as the acts of creative genius previously alluded to, are for now still out of reach. Only time

will tell how far the potential of LLMs and LRMs can be pushed.

Bibliography

- "A Report From Mochizuki | Not Even Wrong." Accessed October 7, 2024. <https://www.math.columbia.edu/~woit/wordpress/?p=13895>.
- Amabile, Teresa M. *Creativity In Context: Update To The Social Psychology Of Creativity*. New York: Routledge, June 2019. ISBN: 978-0-429-50123-4. <https://doi.org/10.4324/9780429501234>.
- Arkoudas, Konstantine. *GPT-4 Can't Reason*, arXiv:2308.03762, August 2023. Accessed June 10, 2024. <https://doi.org/10.48550/arXiv.2308.03762>. arXiv: 2308.03762 [cs].
- Baek, Jinheon, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. "ResearchAgent: Iterative Research Idea Generation over Scientific Literature with Large Language Models," 2024. Accessed February 1, 2025. <https://doi.org/10.48550/ARXIV.2404.07738>.
- Barman, Kristian Gonzalez, Sascha Caron, Tom Claassen, and Henk de Regt. "Towards a Benchmark for Scientific Understanding in Humans and Machines." *Minds and Machines* 34, no. 1 (April 2024): 6. ISSN: 1572-8641, accessed February 2, 2025. <https://doi.org/10.1007/s11023-024-09657-1>. arXiv: 2304.10327 [cs].
- Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623. FAccT '21. New York, NY, USA: Association for Computing Machinery, March 2021. ISBN: 978-1-4503-8309-7, accessed December 15, 2024. <https://doi.org/10.1145/3442188.3445922>.
- Bender, Emily M., and Alexander Koller. "Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5185–5198. Online: Association for Computational Linguistics, 2020. Accessed December 15, 2024. <https://doi.org/10.18653/v1/2020.acl-main.463>.
- Berglund, Lukas, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. "The Reversal Curse: LLMs Trained on "A Is B" Fail to Learn "B Is A"," May 26, 2024. Accessed January 15, 2025. <https://doi.org/10.48550/arXiv.2309.12288>. arXiv: 2309.12288 [cs]. <http://arxiv.org/abs/2309.12288>.
- Berner, Christopher, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębniak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. "Dota 2 with large scale deep reinforcement learning." *arXiv preprint arXiv:1912.06680*, 2019.

- Bisk, Yonatan, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. *PIQA: Reasoning about Physical Commonsense in Natural Language*, arXiv:1911.11641, November 2019. Accessed January 11, 2025. <https://doi.org/10.48550/arXiv.1911.11641>. arXiv: 1911.11641 [cs].
- Boden, Margaret. "Creativity and biology." In *Creativity and philosophy*, edited by Gaut and Kieran, 186–192. February 2018. ISBN: 9781351199797. <https://doi.org/10.4324/9781351199797-6>.
- . "What Is Creativity?" In *Dimensions of Creativity*, edited by Margaret Boden, 75–117. The MIT Press, August 1996. ISBN: 9780262522199.
- Boden, Margaret A. *The creative mind: Myths and mechanisms*. Routledge, 2004.
- Bommasani, Rishi, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258*, 2021.
- Bory, Paolo. "Deep new: The shifting narratives of artificial intelligence from Deep Blue to AlphaGo." *Convergence* 25, no. 4 (2019): 627–642.
- Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. *Language Models Are Few-Shot Learners*, arXiv:2005.14165, July 2020. Accessed January 19, 2025. <https://doi.org/10.48550/arXiv.2005.14165>. arXiv: 2005.14165 [cs].
- Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, et al. *Sparks of Artificial General Intelligence: Early Experiments with GPT-4*, arXiv:2303.12712, April 2023. Accessed January 4, 2025. <https://doi.org/10.48550/arXiv.2303.12712>. arXiv: 2303.12712 [cs].
- Burton, David. "History of Mathematics an Introduction." 1988.
- Castelo, Noah, Zsolt Katona, Peiyao Li, and Miklos Sarvary. *How AI Outperforms Humans at Creative Idea Generation*. SSRN Scholarly Paper, 4751779, Rochester, NY, March 2024. Accessed February 1, 2025. <https://doi.org/10.2139/ssrn.4751779>. Social Science Research Network: 4751779.
- Clark, Peter, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. "Think you have solved question answering? try arc, the ai2 reasoning challenge." *arXiv preprint arXiv:1803.05457*, 2018.
- Cobbe, Karl, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, et al. *Training Verifiers to Solve Math Word Problems*, November 18, 2021. Accessed January 11, 2025. <https://doi.org/10.48550/arXiv.2110.14168>. arXiv: 2110.14168 [cs]. <http://arxiv.org/abs/2110.14168>.
- Davis, Ernest. *Testing GPT-4-o1-preview on Math and Science Problems: A Follow-up Study*, arXiv:2410.22340, October 2024. Accessed February 5, 2025. <https://doi.org/10.48550/arXiv.2410.22340>. arXiv: 2410.22340 [cs].
- De Regt, H. W. *Understanding Scientific Understanding*. Oxford University Press, 2017.

- Dziri, Nouha, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, et al. *Faith and Fate: Limits of Transformers on Compositionality*, arXiv:2305.18654, October 2023. Accessed January 13, 2025. <https://doi.org/10.48550/arXiv.2305.18654>. arXiv: 2305.18654 [cs].
- Elliot Samuel Paul, Dustin Stokes. "Attributing creativity." In *Creativity and philosophy*, edited by Gaut and Kieran, 193–209. February 2018. ISBN: 9781351199797. <https://doi.org/10.4324/9781351199797-6>.
- Feuillade Montixi, Quentin, and Pierre Peigné. "The Stochastic Parrot Hypothesis Is Debatable for the Last Generation of LLMs," November 2023. Accessed December 15, 2024.
- Gaut, Berys. "The philosophy of creativity." *Philosophy Compass* 5, no. 12 (2010): 1034–1046.
- . "The value of creativity." In *Creativity and philosophy*, edited by Gaut and Kieran, 124–140. February 2018. ISBN: 9781351199797. <https://doi.org/10.4324/9781351199797-6>.
- Gelernter, Herbert L. "Realization of a geometry theorem proving machine." In *IFIP Congress*. 1995. <https://api.semanticscholar.org/CorpusID:18484295>.
- Golchin, Shahriar, and Mihai Surdeanu. *Time Travel in LLMs: Tracing Data Contamination in Large Language Models*, arXiv:2308.08493, February 2024. Accessed January 8, 2025. <https://doi.org/10.48550/arXiv.2308.08493>. arXiv: 2308.08493 [cs].
- FunSearch: Making New Discoveries in Mathematical Sciences Using Large Language Models*. <https://deepmind.google/discover/blog/funsearch-making-new-discoveries-in-mathematical-sciences-using-large-language-models/>, December 2024. Accessed February 2, 2025.
- Hankey, Alex. "Kasparov versus Deep Blue: An Illustration of the Lucas Gödelian Argument." *Cosmos and History: The Journal of Natural and Social Philosophy* 17, no. 3 (December 2021): 60–67. ISSN: 1832-9101, accessed March 16, 2025.
- Hendrycks, Dan, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. *Measuring Massive Multitask Language Understanding*, arXiv:2009.03300, January 2021. Accessed January 11, 2025. <https://doi.org/10.48550/arXiv.2009.03300>. arXiv: 2009.03300 [cs].
- Hills, Alison, and Alexander Bird. "Creativity without value." In *Creativity and philosophy*, edited by Gaut and Kieran, 95–107. February 2018. ISBN: 9781351199797. <https://doi.org/10.4324/9781351199797-6>.
- Ismayilzada, Mete, Debjit Paul, Antoine Bosselut, and Lonneke van der Plas. *Creativity in AI: Progresses and Challenges*, arXiv:2410.17218, October 2024. Accessed November 4, 2024. arXiv: 2410.17218.
- Jiang, Bowen, Yangxinyu Xie, Zhuoqun Hao, Xiaomeng Wang, Tanwi Mallick, Weijie J. Su, Camillo J. Taylor, and Dan Roth. "A Peek into Token Bias: Large Language Models Are Not Yet Genuine Reasoners," October 4, 2024. Accessed January 13, 2025. <https://doi.org/10.48550/arXiv.2406.11050>. arXiv: 2406.11050 [cs]. <http://arxiv.org/abs/2406.11050>.

- jylin04, JackS, Adam Karvonen, and Can. "OthelloGPT Learned a Bag of Heuristics," July 2024. Accessed March 15, 2025.
- Kambhampati, Subbarao. "Can Large Language Models Reason and Plan?" *Annals of the New York Academy of Sciences* 1534, no. 1 (April 2024): 15–18. ISSN: 0077-8923, 1749-6632, accessed January 13, 2025. <https://doi.org/10.1111/nyas.15125>.
- Kaufman, Cropley David H. Arthur J. Cropley James C., and Mark A. Runco, eds. *The Dark Side of Creativity*. Cambridge University Press, 2010.
- Kojima, Takeshi, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. *Large Language Models Are Zero-Shot Reasoners*, arXiv:2205.11916, January 2023. Accessed January 20, 2025. <https://doi.org/10.48550/arXiv.2205.11916>. arXiv: 2205.11916 [cs].
- Kronfeldner, Maria E. "Creativity naturalized." *The Philosophical Quarterly* 59, no. 237 (2009): 577–592.
- L., F., Menabreatranslated, and Augusta Ada Lovelace. "Sketch of the Analytical Engine invented by Charles Babbage, Esq." *Ada's Legacy: Cultures of Computing from the Victorian to the Digital Age*, 2015. <https://api.semanticscholar.org/CorpusID:257837181>.
- Latif, Ehsan, Yifan Zhou, Shuchen Guo, Yizhu Gao, Lehong Shi, Matthew Nayaaba, Gyeonggeon Lee, et al. *A Systematic Assessment of OpenAI O1-Preview for Higher Order Thinking in Education*, arXiv:2410.21287, October 2024. Accessed February 5, 2025. <https://doi.org/10.48550/arXiv.2410.21287>. arXiv: 2410.21287 [cs].
- Learning to Reason with LLMs*. <https://openai.com/index/learning-to-reason-with-llms/>. Accessed February 5, 2025.
- Lex Clips. *GPT-4 Is an Early AGI | Max Tegmark and Lex Fridman*, April 2023. Accessed March 16, 2025.
- Li, Changmao, and Jeffrey Flanigan. *Task Contamination: Language Models May Not Be Few-Shot Anymore*, arXiv:2312.16337, December 2023. Accessed January 8, 2025. <https://doi.org/10.48550/arXiv.2312.16337>. arXiv: 2312.16337 [cs].
- Lightman, Hunter, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. *Let's Verify Step by Step*, arXiv:2305.20050, May 2023. Accessed February 5, 2025. arXiv: 2305.20050 [cs].
- Lu, Chris, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. *The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery*, arXiv:2408.06292, September 2024. Accessed February 1, 2025. <https://doi.org/10.48550/arXiv.2408.06292>. arXiv: 2408.06292 [cs].
- Ma, Pingchuan, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B. Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. *LLM and Simulation as Bilevel Optimizers: A New Paradigm to Advance Physical Scientific Discovery*, arXiv:2405.09783, May 2024. Accessed June 5, 2024. <https://doi.org/10.48550/arXiv.2405.09783>. arXiv: 2405.09783 [cs].

- MacKenzie, Donald. "Slaying the Kraken: The Sociohistory of a Mathematical Proof." *Social Studies of Science* 29, no. 1 (1999): 7–60. ISSN: 0306-3127, accessed March 16, 2025. JSTOR: 285445.
- McCoy, R. Thomas, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. *How Much Do Language Models Copy from Their Training Data? Evaluating Linguistic Novelty in Text Generation Using RAVEN*, arXiv:2111.09509, November 2021. Accessed December 15, 2024. <https://doi.org/10.48550/arXiv.2111.09509>. arXiv: 2111.09509 [cs].
- McCoy, R. Thomas, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. *When a Language Model Is Optimized for Reasoning, Does It Still Show Embers of Autoregression? An Analysis of OpenAI O1*, arXiv:2410.01792, October 2024. Accessed February 5, 2025. <https://doi.org/10.48550/arXiv.2410.01792>. arXiv: 2410.01792 [cs].
- McMullin, Ernan. "Values in Science." *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association* 1982 (1982): 3–28. ISSN: 0270-8647, accessed December 22, 2024. JSTOR: 192409.
- Meincke, Lennart, Karan Girotra, Gideon Nave, Christian Terwiesch, and Karl T. Ulrich. *Using Large Language Models for Idea Generation in Innovation*. SSRN Scholarly Paper, 4526071, Rochester, NY, September 2024. Accessed February 1, 2025. <https://doi.org/10.2139/ssrn.4526071>. Social Science Research Network: 4526071.
- Mirzadeh, Iman, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. "Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models." *arXiv preprint arXiv:2410.05229*, 2024.
- Mitchell, Melanie. *LLMs and World Models, Part 1*. Substack Newsletter, February 2025. Accessed March 15, 2025.
- . *LLMs and World Models, Part 2*. Substack Newsletter, February 2025. Accessed March 15, 2025.
- Nezhurina, Marianna, Lucia Cicolina-Kun, Mehdi Cherti, and Jenia Jitsev. *Alice in Wonderland: Simple Tasks Showing Complete Reasoning Breakdown in State-Of-the-Art Large Language Models*, arXiv:2406.02061, June 2024. Accessed June 10, 2024. <https://doi.org/10.48550/arXiv.2406.02061>. arXiv: 2406.02061 [cs].
- Nikankin, Yaniv, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. *Arithmetic Without Algorithms: Language Models Solve Math With a Bag of Heuristics*, arXiv:2410.21272, October 2024. Accessed January 25, 2025. <https://doi.org/10.48550/arXiv.2410.21272>. arXiv: 2410.21272 [cs].
- Novitz, David. "Creativity and Constraint." *Australasian Journal of Philosophy* 77, no. 1 (1999): 67–82. <https://doi.org/10.1080/00048409912348811>.
- . "Explanations of Creativity." In *The Creation of Art: New Essays in Philosophical Aesthetic*, edited by Berys Gaut and Paisley Livingston, 174–91. Cambridge: Cambridge UP, 2003.

- Nye, Maxwell, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, et al. *Show Your Work: Scratchpads for Intermediate Computation with Language Models*, arXiv:2112.00114, November 2021. Accessed February 2, 2025. <https://doi.org/10.48550/arXiv.2112.00114>. arXiv: 2112.00114 [cs].
- Pan, Haining, Nayantara Mudur, Will Taranto, Maria Tikhanovskaya, Subhashini Venugopalan, Yasaman Bahri, Michael P. Brenner, and Eun-Ah Kim. *Quantum Many-Body Physics Calculations with Large Language Models*, arXiv:2403.03154, March 2024. Accessed January 25, 2025. <https://doi.org/10.48550/arXiv.2403.03154>. arXiv: 2403.03154 [physics].
- Paul, Elliot Samuel, and Dustin Stokes. "Creativity." In *The Stanford Encyclopedia of Philosophy*, Spring 2024, edited by Edward N. Zalta and Uri Nodelman. Metaphysics Research Lab, Stanford University, 2024.
- Romera-Paredes, Bernardino, Mohammadamin Barekatin, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, et al. "Mathematical Discoveries from Program Search with Large Language Models." *Nature* 625, no. 7995 (January 2024): 468–475. ISSN: 1476-4687, accessed June 10, 2024. <https://doi.org/10.1038/s41586-023-06924-6>.
- Runco, Mark, and Garrett Jaeger. "The Standard Definition of Creativity." *Creativity Research Journal* 24 (January 2012): 92–96. <https://doi.org/10.1080/10400419.2012.650092>.
- Russell, Bertrand. "Letter to Professor van Heijenoort." In *From Frege to Gödel. A Sourcebook in Mathematical Logic, 1879-1931*, edited by J. van Heijenoort, 127. Cambridge, MA: Harvard University Press, 1962.
- Sahoo, Pranab, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications*, arXiv:2402.07927, February 2024. Accessed January 19, 2025. <https://doi.org/10.48550/arXiv.2402.07927>. arXiv: 2402.07927 [cs].
- Sakaguchi, Keisuke, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. *WinoGrande: An Adversarial Winograd Schema Challenge at Scale*, arXiv:1907.10641, November 2019. Accessed January 11, 2025. <https://doi.org/10.48550/arXiv.1907.10641>. arXiv: 1907.10641 [cs].
- Scholze, Peter, and Jakob Stix. *Why Abc Is Still a Conjecture*.
- Searle, John R. "Minds, brains, and programs." *Behavioral and Brain Sciences* 3 (1980): 417–424. <https://api.semanticscholar.org/CorpusID:55303721>.
- Shahriar, Sakib. *A Cross-Domain Performance Report of Open AI ChatGPT O1 Model*, 2024121930, December 2024. Accessed February 5, 2025. <https://doi.org/10.20944/preprints202412.1930.v1>. Preprints: 2024121930.

- Shi, Freda, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. *Large Language Models Can Be Easily Distracted by Irrelevant Context*, arXiv:2302.00093, June 2023. Accessed January 13, 2025. <https://doi.org/10.48550/arXiv.2302.00093>. arXiv: 2302.00093 [cs].
- Shojaee, Parshin, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. *LLM-SR: Scientific Equation Discovery via Programming with Large Language Models*, arXiv:2404.18400, June 2024. Accessed June 5, 2024. <https://doi.org/10.48550/arXiv.2404.18400>. arXiv: 2404.18400 [cs].
- Si, Chenglei, Diyi Yang, and Tatsunori Hashimoto. *Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers*, arXiv:2409.04109, September 2024. Accessed February 1, 2025. <https://doi.org/10.48550/arXiv.2409.04109>. arXiv: 2409.04109 [cs].
- Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, et al. "Mastering the Game of Go without Human Knowledge." *Nature* 550, no. 7676 (October 2017): 354–359. ISSN: 1476-4687, accessed November 18, 2024. <https://doi.org/10.1038/nature24270>.
- Tim. *Cpldcpu/MisguidedAttention*, January 2025. Accessed January 15, 2025.
- Trinh, Trieu H, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. "Solving olympiad geometry without human demonstrations." *Nature* 625, no. 7995 (2024): 476–482.
- Vafa, Keyon, Justin Y. Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. *Evaluating the World Model Implicit in a Generative Model*, arXiv:2406.03689, November 2024. Accessed March 15, 2025. <https://doi.org/10.48550/arXiv.2406.03689>. arXiv: 2406.03689 [cs].
- Valmeekam, Karthik, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. *PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change*, arXiv:2206.10498, November 2023. Accessed January 13, 2025. <https://doi.org/10.48550/arXiv.2206.10498>. arXiv: 2206.10498 [cs].
- Valmeekam, Karthik, Kaya Stechly, and Subbarao Kambhampati. "LLMs Still Can't Plan; Can LRMs? A Preliminary Evaluation of OpenAI's O1 on PlanBench," 2024. Accessed January 13, 2025. <https://doi.org/10.48550/ARXIV.2409.13373>.
- Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, et al. *Emergent Abilities of Large Language Models*, arXiv:2206.07682, October 2022. Accessed March 15, 2025. <https://doi.org/10.48550/arXiv.2206.07682>. arXiv: 2206.07682 [cs].
- Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*, arXiv:2201.11903, January 2023. Accessed January 19, 2025. <https://doi.org/10.48550/arXiv.2201.11903>. arXiv: 2201.11903 [cs].

- Winter, Joost de, Dimitra Dodou, and Yke Bauke Eisma. "System 2 Thinking in OpenAI's O1-Preview Model: Near-perfect Performance on a Mathematics Exam." *Computers* 13, no. 11 (October 2024): 278. ISSN: 2073-431X, accessed February 5, 2025. <https://doi.org/10.3390/computers13110278>. arXiv: 2410.07114 [cs].
- Wu, Xingyu, Sheng-hao Wu, Jibin Wu, Liang Feng, and Kay Chen Tan. "Evolutionary Computation in the Era of Large Language Model: Survey and Roadmap," 2024. Accessed February 5, 2025. <https://doi.org/10.48550/ARXIV.2401.10034>.
- Wu, Zhaofeng, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. *Reasoning or Reciting? Exploring the Capabilities and Limitations of Language Models Through Counterfactual Tasks*, arXiv:2307.02477, March 2024. Accessed January 15, 2025. <https://doi.org/10.48550/arXiv.2307.02477>. arXiv: 2307.02477 [cs].
- Xu, Fengli, Qianyu Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, et al. *Towards Large Reasoning Models: A Survey of Reinforced Reasoning with Large Language Models*, arXiv:2501.09686, January 2025. Accessed February 5, 2025. <https://doi.org/10.48550/arXiv.2501.09686>. arXiv: 2501.09686 [cs].
- Yao, Shunyu, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*, arXiv:2305.10601, December 2023. Accessed February 5, 2025. <https://doi.org/10.48550/arXiv.2305.10601>. arXiv: 2305.10601 [cs].
- Zellers, Rowan, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. *HellaSwag: Can a Machine Really Finish Your Sentence?*, arXiv:1905.07830, May 2019. Accessed January 11, 2025. <https://doi.org/10.48550/arXiv.1905.07830>. arXiv: 1905.07830 [cs].