JOURNAL OF SOCIAL COMPUTING
ISSN 2097-5015 01/06 pp75–94
Volume 6, Number 2, June 2025
DOI: 10.23919/JSC.2025.0009

Generative Models as a Complex Systems Science: How Can We Make Sense of Large Language Model Behavior?

Ari Holtzman*, Peter West, and Luke Zettlemoyer

Abstract: Coaxing out desired behavior from pretrained models, while avoiding undesirable ones, has redefined Natural Language Processing (NLP) and is reshaping how we interact with computers. What was once a scientific engineering discipline—in which building blocks are stacked one on top of the other—is arguably already a complex systems science—in which *emergent behaviors* are sought out to support previously unimagined use cases. Despite the ever increasing number of benchmarks that measure *task performance*, we lack explanations of what *behaviors* language models exhibit that allow them to complete these tasks in the first place. We argue for a systematic effort to decompose language model behavior into categories that explain cross-task performance, to guide mechanistic explanations and help future-proof analytic research.

Key words: language model behavior; emergent properties in ai system; interpretability and analysis method

1 The Newformer: A Thought Experiment

Consider the following Thought Experiment (1):

Tomorrow, researchers at an industry lab publicly release a new kind of pretrained model: the Newformer. It has a completely different architecture than the Transformer (no attention, non-differentiable components, etc.), that outperforms all pretrained Transformers on the vast majority of benchmarks. Independent labs quickly verify that these results are sound, even on just-released benchmarks. While the composition of the training data is public, it is so expensive to train that no lab can afford to replicate it, even the one that produced it. Scaled-down versions do not exhibit the same performance or interesting behaviors as the original model.

• Ari Holtzman, Peter West, and Luke Zettlemoyer are with the University of Chicago, Chicago, IL 60605, USA. E-mail: ahai@cs.washington.edu; pawest@cs.washington.edu; lsz@cs.washington.edu.
* To whom correspondence should be addressed.
Manuscript received: 2024-10-29; revised: 2025-05-18; accepted: 2025-05-21

1.1 How should we study the Newformer?

Identifying high-level behaviors that a model does or does not share with older models can steer us toward lower-level mechanisms it uses to solve tasks (Section 1.2). Interpretation techniques that rely on low-level details are model specific (Section 1.3) and often abandoned as the field changes. The Newformer is fictional, but it can help us reconceptualize the goals and methods of generative model research in light of the new landscape (Section 1.4).

How should we factorize model behavior into understandable and explanatory categories? (Section 2) We present a formalism for describing behavior (Section 2.1), noting that this corresponds to a *metamodel* that predicts aspects of a primary model. Benchmarks help us measure *performance*, but rarely *discover behavior* (Section 2.2) or *characterize* it (Section 2.3). Instead, discovered behaviors motivate new benchmarks (Section 2.4).

Generative models qualify as *complex systems* (Section 3), due to their *emergent behaviors* (Section 3.1), which are more often *discovered* than engineered (Section 3.2). A lack of clarity on *what* models do

holds us back, as if we were studying organic chemistry without knowledge of biology (Section 3.3). This issue remains even when proprietary models are released (Section 3.4), as the problem lies in our lack of behavioral vocabulary; investigating possible mappings between training data and generated data can help us establish new behavioral categories (Section 3.5).

Despite the challenges, generative models are *easier* to study than many naturally arising complex systems (Section 4), because they are simulable by construction (Section 4.1). In contrast to physical phenomena, we can easily conduct a wide range of storable, repeatable experiments without observer effects (Section 4.2). We do, however, rely on the availability of open-source models (Section 4.3).

We conclude (Section 5) with an argument for increased focus on the foundational "what are models doing?" to guide the classic "why are models doing that?"

1.2 Top-down behavioral taxonomy guides bottom-up mechanistic explanation

The Newformer is a completely opaque result when considering benchmarks alone; it is simply better at doing what we want it to do than Transformer models[1] were before. However, as shown in Fig. 1, a hierarchical taxonomy of LM behavior can guide our investigation of the Newformer, leading to questions such as

(1) What behaviors do the Transformer models and the Newformer model share, e.g., does the Newformer also repeat phrases more often than as seen in the training data?

(2) Do they exhibit similar behaviors in the same contexts, e.g., does the Newformer need fewer input-output demonstrations to exhibit in-context learning at peak performance?

(3) Do high-level behaviors decompose into the same lower-level behaviors or does the Newformer use different mechanisms to express them, e.g., when the Newformer is used for paraphrasing does it also tend to exactly copy the input?

Without such behavioral categories, we risk investigating the wrong direction when we try to interpret models, because we do not know what phenomena we are trying to explain in the first place.

Observed behavior can tell us where to look for

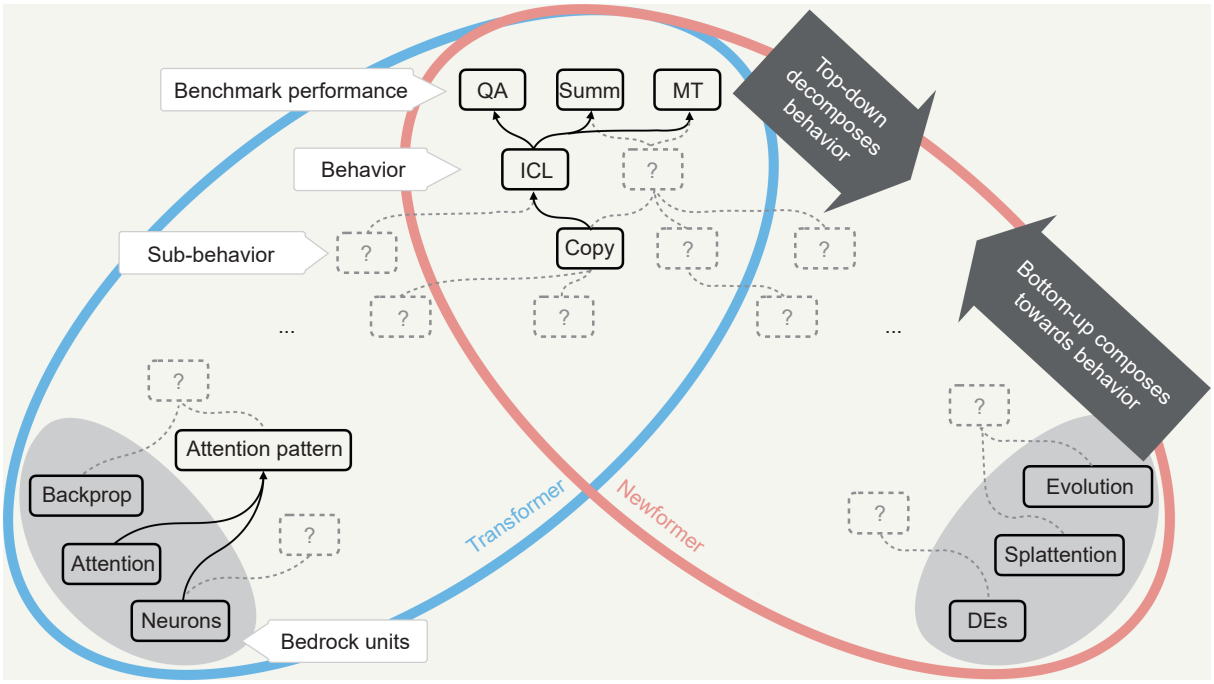

Fig. 1 To explain why learned models self-organize the way they do from the bottom-up, it is useful to have top-down hierarchy of partially decomposed behaviors, to guide hypotheses with functionality we know the overall model has. While networks are composed of bedrock units for which we have a perfect understanding by construction (e.g., neurons), most emergent aspects of these systems are still undefined and undiscovered (represented as "?").

bottom-up explanations. Al-Rfou et al.[2] observed emergent copying behavior in Transformer Language Models (LMs), paving the way for the discovery of *copying heads* that make copying possible. Characterizing copying heads led to the discovery of *induction heads*[3, 4]: Transformer heads that are capable of copying abstract representational patterns in previous layers and appear to be responsible for in-context learning. Olsson et al.[4] showed that induction heads exhibit a variety of pattern matching behaviors that are still not fully catalogued.

Attempting to explain neural networks bottom-up without being guided by behavior can make it difficult to interpret results. For example, many works that identify anisotropy in the embedding spaces of large LMs diagnose this as a deficiency, and attempt to fix it[5–7]. However, recent work suggests that this anisotropic property may not actually limit expressivity[8], may be a result of the transformer architecture specifically[9], and may actually be helpful for language models[10].

1.3 The Transformer is the old Newformer

A moderate Newformer event has occurred at least once before with LMs: the switch from Recurrent Neural Networks (RNNs) to Transformers. Despite many partial explanations[11–13], we still lack an explanatory theory of how Long Short-Term Memory (LSTM)[13] LMs such as ELMo[14] worked—what behaviors they could and could not capture, how these composed, etc.—even as they were replaced by models like Bidirectional Encoder Representations from Transformers (BERT)[15] with similar use cases, but completely different architectural details. This does not bode well for the introduction of something like the Newformer which is significantly farther from the Transformer than the Transformer is from the LSTM.

On their own, bottom-up methods do not transfer well to new systems: analysis techniques that relied on mutated state and gating in RNNs, such as visualizing gating mechanisms[16], are not applicable to Transformers. Interpretation methods for Transformer models[17, 18], such as those that use attention, are unlikely to transfer over to the Newformer which breaks many previously immutable assumptions.

This suggests the value in doing more interpretation work that treats models like *black-box*es, as if we do not have access to their internal mechanisms. There is growing interest in looking at Natural Language Processing (NLP) systems as black-boxes[19–21], though much of this work still uses intermediate outputs—such as embeddings—rather than directly analyzing behavior in the output space models are trained to fit. Truly black-box methods can help insulate our analysis from change, giving us an anchor point that will always be testable on models that use the same modality (e.g., text, speech, and images). Belinkov and Bisk[22] showed that neural machine translation systems are brittle to both natural spelling errors and synthetic character-level noise. This observation can be extended to ask: Is the Newformer robust to the same kinds of noise? Up to what threshold? Does the noise appear to be localized in the brittleness of tokenization, as was the case for Transformer-based systems[23]? Developing a rich inventory of such tests would give us a universal scaffolding for analyzing any Newformer the moment it is discovered.

1.4 Are we there yet?

Deciding whether a Newformer-like event has *truly* happened is an unresolvable question. New models are always partially derivative, and new (possibly artificial) axes can always be invented where they are worse or better[24]. Yet three years on it is still infeasible for most labs to train a GPT-3[25] level LM, costing approximately half a million dollars in compute alone for private companies—with engineering teams—to produce a similar model[26]. Thus it seems that the gap for training is only growing wider as ChatGPT[27] and GPT-4[28] become commonplace in research[29, 30], production[31–33], and even model evaluation[34, 35].

Unlike the Newformer, these models were never released, are frequently deprecated[28], change from day to day[36, 37], and are know to be unstable over theoretically deterministic queries[38]. Yet, the open source community has caught-up quickly[39–41] helped by industry labs' open-sourcing efforts[42–44], and new finetuning tehchniques[45–47].

However, the question still remains: how should we explain models not everyone can train? Models that are so arduous, slow, and expensive to train that we will likely never ablate all the necessary variables

needed to study them properly?

This leaves us with mere behavior. We generally think of there as being two different kinds of behavior: the neural behavior of different activations in models and the "output" of the model in the form of human media (e.g., text, images, videos, etc.). Most methods of explaining models focus on the former: trying to explain why neural activations cluster into certain patterns and trying to understand what those patterns mean about the output.

We argue that not enough attention has been given to formalizing the latter: *what* models are doing in the first place, in terms of regularities in their outputs. Without such a formalization, bottom-up methods will have a much harder time deciding what precisely to explain, and what is simply noise.

2 The Behavioral Bottleneck

How do we avoid proposing a new explanation for every exhibited difference? Surely we do not believe that we need a benchmark for every prompt that elicits slightly different behavior from a generative model? One solution is to propose many possible mechanisms, but make it an explicit research agenda to discover *the most parsimonious explanation*, a concept visualized in Fig. 2. In other words, we want to be able to predict the aspects of text we care about (e.g., factuality) with the simplest rules possible. We briefly formalize this concept in Section 2.1, but the bulk of this paper concerns the *need* for this new research focus and the perspective it yields.

Thousands of papers observe behavioral tendencies in models, such as the ability of a pretrained Transformer to copy from the input context[2, 3], which we will adopt as a running example. To understand models better, we must rigorously describe (1) what *aspect* of generative behavior a given mechanism predicts (e.g., repetition, copying from the training set, etc.) and (2) how much of the *information* in the output space of the model such predictions explain (since most will not predict 100% of what a model emits).

Figure 2 serves as a visual map of how we might explain models via behavior. On the top level we have a huge diversity of benchmarks that currently exist, and the even larger number that may one day exist. On the bottom we have the mathematical abstraction that describes the space of all possible models. Clearly both of these represent many more possibilities than is useful as an explanation or than is *necessary* to explain specific facets of model behavior. The intermediate levels, then, deal with simplified metamodels, i.e., models of the underlying generative model that are less explanatory, but still allow us to interpret or theorize around models.

2.1 Working definition of "behavior"

Fong and Vedaldi[48] stated that: "An explanation is a rule that predicts the response of a black box f to certain inputs." We think of a *behavior* as an explanation of limited aspects of a model, a concept we briefly formalize. We make reference to this formalization sparsely throughout the rest of the paper, as the argument can be understood without it, and we stress that the problem we are facing is more fundamental than a missing formalism.

Given a generative model from one input medium $\mathcal{X}$ (e.g., strings composed of at most 2048 tokens) and a source of randomness $\mathcal{R}$ to an output medium $\mathcal{Y}$ (e.g., 512 pixel×512 pixel images):

$$\mathcal{M}: \mathcal{X} \times \mathcal{R} \to \mathcal{Y} \tag{1}$$

We can define a behavior as a function from the same input medium to a feature set $\mathcal{F}$:

$$\mathcal{B}: \mathcal{X} \to \mathcal{F} \tag{2}$$

For instance, $\mathcal{M}$ may be a general purpose text-to-image model trained on scraped data, while $\mathcal{B}$ may map a string $x \in \mathcal{X}$ to a probability that an image $\mathcal{M}(x)$, contains at least one dog. Or $\mathcal{X}$ and $\mathcal{Y}$ may both be unicode strings, in the case of an LM, with $\mathcal{B}$ being a binary prediction as to whether $\mathcal{M}(x)$ will eventually get caught in a repetition loop[49].

Our goal in proposing behaviors is to *explain* the underlying model using rules that capture model tendencies. Behaviors are explanatory to the extent that they give us information about the application of the model $\mathcal{M}$ under distributions $\mathcal{D}_{\mathcal{X}}$ and $\mathcal{D}_{\mathcal{R}}$ over $\mathcal{X}$ and $\mathcal{R}$, which we collectively refer to as $\mathcal{D}$ for brevity. We can formalize the notion of "giving us information about the application of the model" through the mutual information:

$$I_{\mathcal{D}}(\mathcal{M}(X,R);\mathcal{B}(X)) = H_{\mathcal{D}}(\mathcal{M}(X,R)) - H_{\mathcal{D}}(\mathcal{M}(X,R)|\mathcal{B}(X)) \tag{3}$$

where X and R are random variables drawn from $\mathcal{X}$

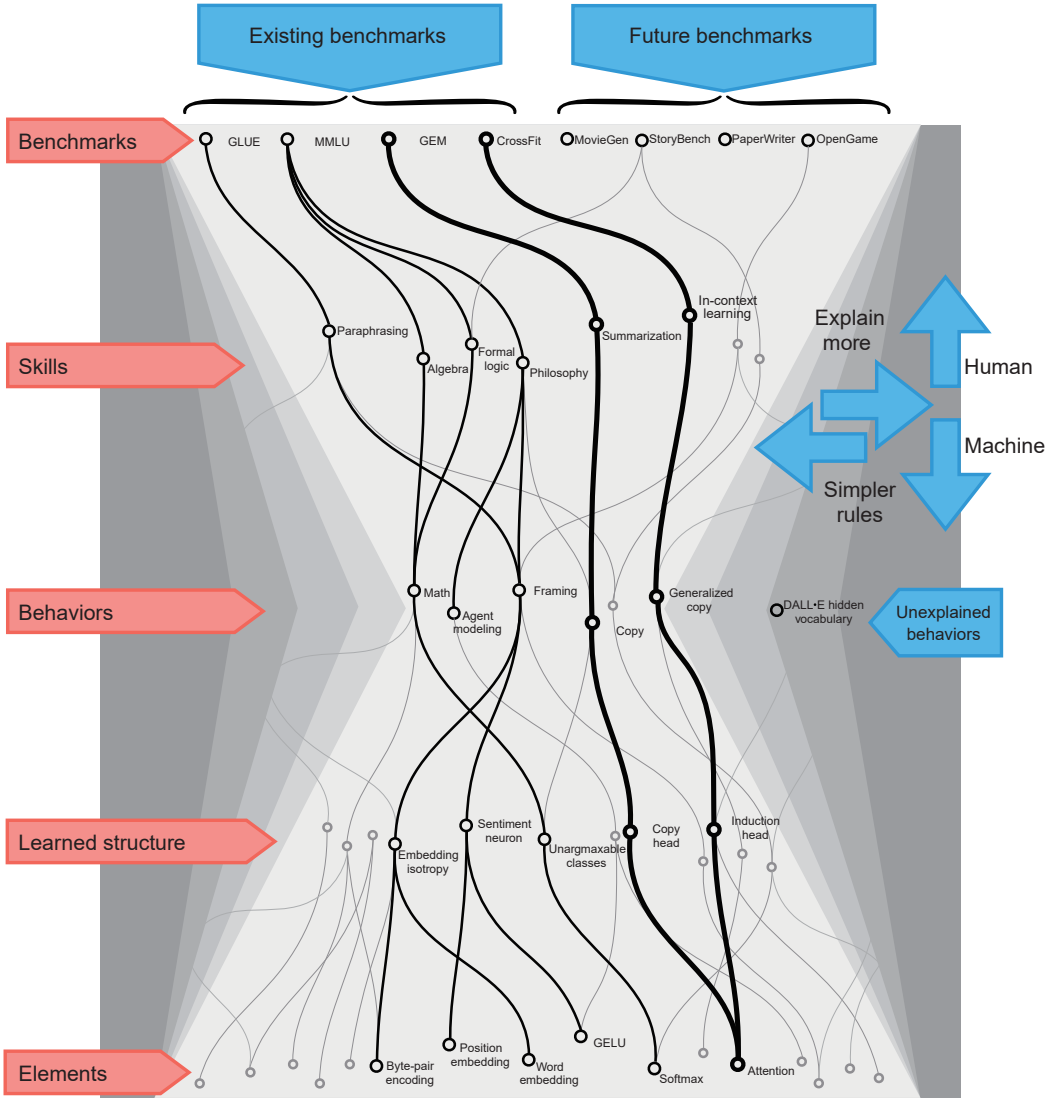

Fig. 2 Visual representation of different aspects of models, shown from the basic elements of models on the bottom up to the benchmarks we are attempting to solve. Nodes represent invented and discovered aspects of models. The highlighted subgraph captures the concepts that we might want to use for understanding the phenomenon of "copying" in Transformers, when models generate sequences that appear in their local context window, a behavior that serves as a running example in this paper. We might start out by noticing that Transformer models have higher scores on GEM (Benchmark), especially on summarization-like tasks (Skill). Inspecting the data generated by the models of interest, we might notice one of the qualitative differences separating Transformer models from other models is the ability to correctly use novel entities (Behavior). We might ask why this is, embarking on an empirical study of when networks develop the ability to copy, as Elhage et al.[3] did, discovering specific attention heads served as *copying heads* (Learned Structure supported by certain Elements). This led to other discoveries such as *induction heads* (Learned Structure), which were found to perform a kind of *generalized copying* that supports inference-time pattern recognition (Behavior), e.g., for In-Context Learning (Skill), leading to better results on fewshot benchmarks such as CrossFit (Benchmark). Research can proceed by observing high-level behavioral regularities, explaining them via the tendencies of the model, and using this to achieve clarity about other observed behaviors.

and $\mathcal{R}$ according to $\mathcal{D}_{\mathcal{X}}$ and $\mathcal{D}_{\mathcal{R}}$, and $H_{\mathcal{D}}$ is the entropy: $H(Y) = E_{\mathcal{D}}[-\log p_{\mathcal{D}}(Y)]$ for a random variable Y. The mutual information is a direct measure of *how many bits of information we learn about one variable given another*, so this formulation directly tells us how much a behavior reveals about expected model output.

We call setting $\mathcal{D}_{\mathcal{X}}$ to be uniform over $\mathcal{X}$ the "mechanistic distribution". In this case, the mutual information is unrelated to the expected distribution of inputs in the wild, but is instead representative of

how well we can model *any* input to $\mathcal{M}$. For instance, explaining an LM under the mechanistic distribution would require a behavior that predicts aspects of the LM's output accurately even for long strings of gibberish. This may be difficult, since we often use human linguistic features to make predictions about model outputs, but such behaviors are closer to the notion of mechanistic interpretability that tries to fully reverse engineer the model being studied[50].

If $\mathcal{M}$ ignores its source of randomness, i.e., $I(\mathcal{M}(X,R);R)=0$—as is the case for deterministic models such as a greedy-decoded LM—then the most explanatory behavior is simply $\mathcal{B}=\mathcal{M}(X,r)$ for any $r\in\mathcal{R}$. This is a degenerate behavior, in that it is very explanatory, but has not brought us any closer to explaining $\mathcal{M}$. Therefore, we would like behaviors that are not just very high mutual information with the model, but also *point to predictable regularities* in $\mathcal{M}$, especially in a way that allows us to build up new hypotheses about it. Much has been written about what makes an explanation useful[51–53], and reviewing these desiderata is out of scope for this paper.

The mutual information $I_{\mathcal{D}}(\mathcal{M}(X,R);\mathcal{B}(X))$ can also be viewed as *how much a behavior allows us to compress the output of a model* under a distribution $\mathcal{D}$, e.g., a distribution of articles for a summarization task (and a random number generator R). This is because any bits of information revealed by one variable, can be used to compress the other, under a proper coding scheme[54].

The concept of Minimum Description Length (MDL) has been used as an information theoretic criterion for finding good hypotheses[55]. Essentially, it suggests an extension to Occam's razor[56]: that we should favor explanations that are simple to describe and explain the object under study the most. We can formalize this notion for behaviors, via an encoding scheme C that represents behaviors $\mathcal{B}$ and outputs $y\in\mathcal{Y}$ as binary strings of variable (but finite) length $s\in\mathcal{S}$, where $|s|$ is the length of a string s. A naïve MDL objective would then be

$$\underset{\mathcal{B}}{\operatorname{argmin}}|\mathcal{C}(\mathcal{B})|+\sum_{x\in\mathcal{X}}E_{\mathcal{D}_{\mathcal{R}}}[|\mathcal{C}(\mathcal{M}(x,R)|\mathcal{B}(X))|] \tag{4}$$

However, this would not suit our general objective: we do not necessarily wish to encode *all possible data a model could produce*, especially since most models have huge output spaces of largely low probability density. Instead, we would like to quantify the information behaviors can save us under $\mathcal{D}_{\mathcal{X}}$.

To capture the idea how much space does $\mathcal{B}$ save us under $\mathcal{D}_{\mathcal{X}}$ we can use:

$$\underset{\mathcal{B}}{\operatorname{argmin}}|\mathcal{C}(\mathcal{B})|+\alpha H_{\mathcal{D}}(\mathcal{M}(X,R)|\mathcal{B}(X)) \tag{5}$$

where we replace the second term with the conditional entropy $H_{\mathcal{D}}$, since this describes the minimum number of bits that could be used to represent the information encoded[54]. This can be interpreted as, "we would like behaviors that on average, save us more space in terms of encoding the possible outcomes of a model than they take to describe." α allows us to trade-off how much we weight the representation of the behavior vs. the outputs of a model, where larger values of α may be appropriate if we are dealing with many outputs, making the bits saved by way of conditioning on the behavior more pertinent.

Overall, we seek to find behaviors that are both explanatory and simple to describe. We can think of this as attempting to find a *metamodel*: models that are designed or trained to predict another model's behavior[57], as illustrated in Fig. 3. This suggests we want to find *behaviors that transfer over different contexts* so we can predict where models will be useful and where they will break down.

2.2 Can benchmarks discover new behavior?

In general, discrepancies in performance between benchmarks can *hint* at potentially new behavior, but

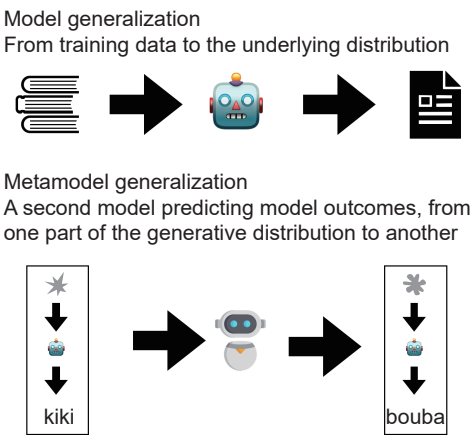

Fig. 3 If we cannot train models easily, but those models are sufficiently general and useful, we can predict what models can and can not do, rather than what a model trained differently would do. The kiki/bouba categorization is a cross-culturally robust linguistic-conceptual mapping in humans[58].

they cannot discover behavior we have not yet observed. Given the diversity of NLP benchmarks, it is likely that the Newformer (Section 1) will perform drastically different on certain pairs of benchmarks we believe to be related, e.g., the same task in two different domains. This is a useful signal for where to inspect behavior, but benchmarks alone cannot reveal new abilities, underlying mechanisms, or shortcut heuristics the Newformer is relying on that *cause* a discrepancy in results and what else its effects are.

For example, it is very difficult to imagine how *prompting*[59, 60] could have been discovered via benchmarking. Finetuning a generative model, such as GPT-2[61], and doing well or badly at any number of benchmarks could not have revealed a model can be prompted with text that matched training data patterns, to elicit behavior such as summary generation via the string "TL;DR" or translation through formatting such as "French sentence: <source>\\English sentence:". These discoveries are a result of inspecting the generative *behavior* of GPT-2, and only afterwards testing a perceived pattern on benchmarks.

How do we try to explain the behavior of models, once we know there is a discrepancy we want to explain? Often we attempt to look at the qualitative differences between tasks a model is good or bad at, and come up with hypotheses for what the model is failing to do when it performs poorly, e.g., across different finetuning tasks[62]. While useful for coming up with hypotheses, using benchmarks as evidence of behavior requires care, because it is often unclear what a given benchmark is actually testing. Rohrbach et al.[63] showed that image captioning systems hallucinate objects not present in the scene, and are unintentionally *rewarded* for doing so by standard metrics, by capturing phrasing and *n* grams of reference captions better when hallucinating. Liao et al.[64] described a detailed framework for assessing benchmark validity and note the complexity of ensuring benchmarks test what we would like them to. Thus, since we often do not know precisely what behavior benchmarks test, they might indicate what contexts to examine the Newformer in, but not precisely what it does.

2.3 Can benchmarks characterize behavior?

Consider standardized tests for humans—such as the SAT[65] or the NCEE[66]—while the debate about how much these tests tell us is heated, there is little resistance to the statement: *test scores do not fully describe human behavior*, even within the subjects they test such as mathematics and biology.

Performance data about a bicycle is not sufficient to reverse-engineer its gear system. Even with perfectly valid benchmarks, the subspace of benchmark performance is not descriptive enough to characterize behavior. As we greatly increase the number of benchmarks, we are left with the problem of determining precisely how benchmarks overlap and differ in a way that characterizes behavior (Fig. 2). Because the space of benchmarks is limited, as we test for human-desirable skills and human-interpretable pitfalls, discovering novel behavior in non-human systems is difficult.

Measuring systems only for their expected purposes makes it difficult to disentangle component behaviors that allow models to produce the desired or undesired outputs, as failure under distribution shift often reveals. For example, neural machine translation often outputs completely irrelevant translations under domain shift[67, 68]. This is exacerbated by the fact that most generative models are not trained with a precise purpose in mind.

Imagine testing whether an LM can summarize an article. In order to summarize an article, a requisite skill required by models is *copying*, because novel entities are constantly appearing, but need to be referenced in the summary. See et al.[69] added a copying mechanism to an RNN in order to improve its copying ability for summarization. If we were to only look at performance on summarization, we would be unlikely to notice whether copying was happening or not directly—only whether performance is hitting certain desired levels.

Benchmarks are, by necessity, scoped to certain contexts that are presumed to test for certain behaviors—but they do not directly tell us what patterns the model is exploiting to solve the task, as Liao et al.[64] pointed out. This was a hard-learned lesson in many benchmarks, such as when it was discovered that SNLI[70, 71] could be solved with *hypothesis-only* systems that only use a subset of the information that was supposed to be necessary to the task.

2.4 Behaviors: Building blocks for evaluation

Benchmarks are still the best solution for coordinating cross-lab experimental *comparisons*, and we expect them to continue to be useful in that respect indefinitely. However, to answer "*What* strategy is the Newformer using for this task?", "What failure modes should we expect?", and "What else do we expect the Newformer to be capable of?", we cannot use benchmarks alone to guide where we inspect model behavior, nor as a means to define it.

Instead, we propose an increased focus on behavior, because we believe that the science of generative models is currently held back by insufficient understanding of *what* models are doing in general, rather than *how well* models perform on specific tasks. These are highly related to each other, and we can think of *behaviors as building blocks for evaluation* (Fig. 4). Consider the following Thought Experiment (2):

A new LM is released with many of the expected capabilities, such as basic arithmetic and basic translation, but another interesting behavior is noticed and hypothesized: when asked properly in natural language, the model can stegonographically encode complex hidden messages while completing other tasks.

When this LM is released, it is unlikely there are any benchmarks that test this particular capability. While we could design a specific benchmark for this behavior, this would be somewhat counter-productive: what we really care about is the *Cartesian product* of this behavior and other tasks that we were already testing. In this sense, behaviors are the building blocks for benchmarks.

As Chang and Bergen[72] pointed out in their survey of behaviors, researchers are often surprised by the outputs of the models they work with; it should not surprise us that we cannot premeditate benchmarks to capture behavior when modeling improvements have outpaced our ability to be exposed to generated data. One way to be more nimble to new behaviors, is to directly measure behaviors we expect[73], flagging unexpected combinations for inspection.

On the surface, it might seem that naming behavioral categories such as "copying" or "in-context learning" is just as liable to obsolescence as any other analysis. What should we do if the Newformer does not exhibit these behaviors? We argue that this is a very unlikely scenario: as long as we are attempting to train models to mimic human understandable phenomena, there will be human perceivable patterns that we expect models to mimic as well.

3 Generative Models as a Complex Systems Science

While the Newformer (Section 1) is a thought experiment, it is representative of many facets of research regarding generative models today; suddenly, focus has shifted to searching for *emergent behaviors* in large and often inscrutable models. Larger pretrained models continue to be trained and continue to perform better[27, 28, 74]. While efforts to release

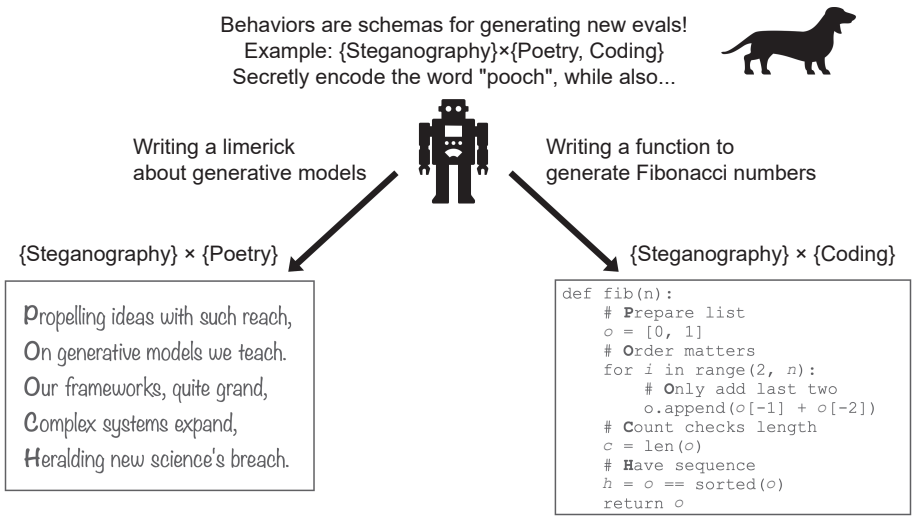

Fig. 4 Example of how behaviors can be used to create new evaluations. These examples were generated from GPT-4, but required significant human curation, suggesting that Thought Experiment (2) has not yet occurred.

models[42–44] and involve more researchers in model training[75] can increase transparency and provide more information, it is well beyond the resources of the vast majority of labs to train. Efficiency breakthrough are likely to be exploited to further increase model size and feed into the same problem they were meant to solve.

Thus, it seems likely that training and re-training models is no longer the path towards understanding them for the vast majority of researchers. In many fields the creation of what it studies is impossible, from biology to astronomy. Many of these fields are *complex systems sciences*, in that they focus on the question illustrated in Fig. 5: how do the macro-level behaviors we observe (life, black holes, etc.) arise from the micro-level units we understand better (chemicals, regular matter, etc.)?

In other words, we suggest studying *generative models themselves not just generative modeling*.

3.1 What is a complex system?

Newman[76] established a working definition:

"[A] system composed of many interacting parts, such that the collective behavior of those parts together is more than the sum of their individual behaviors. The collective behaviors are...'emergent' behaviors, and a complex system can thus be said to be a system of interacting parts that displays emergent behavior."

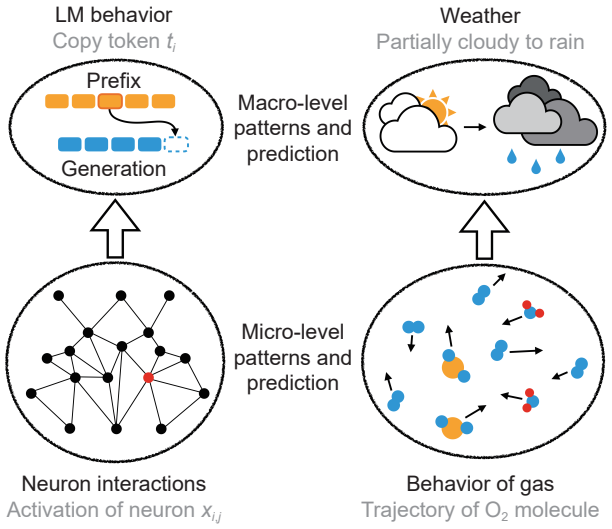

Fig. 5 Complex systems characterized by two or more levels of regularity: A micro-level in which local interactions are at least partially predictable and a macro-level in which many local interactions collectively exhibit recognizable patterns. Emergence describes how macro-level regularity is hard to predict in advance from comparatively well-understood micro-level dynamics.

Recently, interest in emergent behavior has grown in NLP[77–80], though it is usually defined, in terms of scaling over model parameters, dataset size, or computational power. We rely on a much simpler definition:

Emergent behaviors are system level behaviors that are hard to predict from the dynamics of lower level subcomponents.

For instance, the ocean is a complex system. We can understand many properties of individual water molecules, e.g., H_2O has a partial positive and negative charge in certain places due to its composition, but the aggregate properties of *water* as a collective whole exhibits predictable properties such as waves. It is difficult to predict the properties of water from H_2O because "the interactions of interest are non-linear...[yielding] levels of organization and hierarchies—selected aggregates at one level become 'building blocks' for emergent properties at a higher level, as when H_2O molecules become building blocks for water."[81]

Similarly, we understand the basic mechanical properties of LMs at the neuronal level, e.g., we have a perfect understanding of how to predict what any individual neuron will do given arbitrary inputs (by construction), but we also notice patterns at the level of *model behavior*, e.g., the emergent copying behavior, which is observed in both Transformer models[2, 82] and LSTM models[83]. In the face of new behavior that a model such as the Newformer might exhibit, we would be even less certain of how lower-level system components add up to observed responses.

3.2 Emergent behaviors in LMs are discovered, not designed

Neural architectural elements (e.g., position embeddings) and training methods (e.g., masking strategies) deeply affect the resulting model but do not fully explain behavior. We often fail to create the behavior we attempted to engineer into an architecture *and* discover new, unintentional behavior.

Many architectures have been designed to make use of longer context[84–86], but evidence suggests that

these models often do not make use of the long-term dependencies that they intended to capture[87–89]. Inversely, BERT was shown to capture much of the functionality of a knowledge base without task-specific training[90].

To illustrate the difference between *designing* and *discovering* behavior, let us return to our running example of the copying behavior, where models produce a span that was in their input. A classic example of designing behavior is pointer-generator models[69], in which a specific, discrete mechanism was added to encourage a certain behavior: copying. Transformers, on the other hand, were designed such that computation at a given time-step could *attend* to any previous time-step that was included in the context window. This intentionally removed the recurrence in architectures such as LSTMs[13] and GRUs[91] in order to increase efficiency on highly parallelizable hardware such as GPUs and TPUs. A side-effect of this change was the emergent behavior of *copying* that arises directly from the Transformer architecture trained as a language model[2].

Instead of directly designing models for these purposes, we are now in the position of training general models with different structure and actively probing them for behavior. Using various data and masking strategies has produced models that can be controlled through different metadata[92–94], while instruction-tuning has shown that pretrained LMs can be finetuned for control[95–97] often with very limited data[45, 47, 98].

This *discovery* process focuses on giving the model access to certain kinds of correlations, and then inspecting what model behavior emerges.

3.3 Neuronal explanations are limited by our understanding of behavior

It is difficult to explain how or why LMs produce their outputs without having a good description of *what* they do. Explaining behavior bottom-up, requires an understanding of what behaviors we are trying to explain. Mittal et al.[99] noted:

"An emergent property of a system is usually discovered at the macro-level of the behavior of the system and cannot be immediately traced back to the specifications of the components, whose interplay produce this emergence."

This is the situation we find ourselves in with regards to large, pretrained models. We cannot, in general, predict how structure will form. While we can engineer systems with the hope of producing certain kinds of behavior, e.g., training on multimodal data to produce models that can draw inferences in ways that integrate paired text and images, this often does not produce the desired results[100, 101].

Bottom-up investigation can reveal key properties of emergent organization within LMs, e.g., BERT replicates features of the classical NLP pipeline[102]. But when anomalous behavior is discovered, e.g., the DALL·E 2 hypothesized "hidden vocabulary" of invented words that correspond to specific image categories[103], it is difficult to investigate them with bottom-up tools until we reach a better understanding of what triggers them, what their scope is, etc. There have been attempts to reject the hidden vocabulary hypothesis[104], but it is a very difficult hypothesis to rebut from first principles: what tests reject the hypothesis "DALL·E 2 has a hidden set of vocabulary with clear and consistent meaning" rather than "this specific mapping from the vocabulary to features is not correct"?

This is similar to trying to research organic chemistry without knowledge of biology: it is certainly not impossible, but without high-level guides to the kind of structure one is expecting, the search space is huge and it is difficult to know where to look. Our lack of a behavioral taxonomy hampers research into internal structure, especially in models that break current assumptions such as the Newformer, as it is significantly more challenging to probe for structure without knowing what patterns in the outputs hint at the presence of structure.

3.4 Access is not a silver bullet

Consider the following Thought Experiment (3):

Tomorrow, all industry labs publicly release all of their pretrained models.

Despite the fact that this would doubtlessly help us understand the basic properties of a given model such as ChatGPT, e.g., how large it is, we would still have significant obstacles on the way to explaining why ChatGPT is capable of writing short stories for almost any given prompt.

Indeed, the problem with answering the question of "How can a language model write a story?" has much

less to do with language models and much more to do with the fact that we are currently incapable of answering the question "How can *x* write a short story?" for any value of *x*. We find ourselves in the strange position of being able to train models we do not fully understand *for tasks we do not fully understand or anticipate in advance*.

The key to answering this question is to ask: what kind of explanation would satisfy us? For instance, when it comes to LMs, one explanation is that models are simply reconstructing long sequences from the training set and stitching them together. While a significant amount of memorization is taking place[105–107] models appear to be able to generate data that is not a trivial recombination of the training data[4, 77, 108].

The goal, then, should be to build up the case for a reasonable hypothesis that explain the breadth, depth, and (most importantly) mistakes models make when executing a complex task. However, we do not want a new explanation for every new task, which is precisely why we argue for the formalization and study of *behaviors* that describe the underlying strategies of models.

While model access would not directly solve these problems, we *do* believe that open-source models are a necessary prerequisite to this research program, for reasons outlined in Section 4.3.

3.5 (Generated) data represents behavior

Behavior in large pretrained models is nothing more than the answer to the question "How can we characterize the distribution of data this model generates?" Aspects of the training data such as the presence of multiple languages[109, 110] or the number of repeated documents[111, 112] in the training set have been shown to be explanatory of zero-shot translation abilities and model tendency to leak training data, respectively.

Figure 6 visualizes what kind of behavioral mappings we can explore with data-based explanations. *Shared behavior*—patterns that are found in both the training and inference data (the outputs of the model)—are the simplest to search for, because they only require finding a specific behavior in the training or inference data and then looking for it in the other. For instance, the prompting behavior discovered in GPT-2 that causes summaries to be generated when "TL;DR" is placed after an article is an example of shared behavior. Idiosyncratic behaviors describe behaviors that do not appear to be caused directly by the training data at all, e.g., zero-length translations in many large models[113–115]. Perhaps the most difficult to find behavioral mappings are those for which behavior in the corpus yields different behavior in the model, *contingent behavior*, as is hypothesized to be the case for DALL·E 2's "hidden vocabulary": nonsense words that appear to consistently lend certain meanings to produced images[103]. Finally, unexpressed behavior is observed in the training data, but not in the inference data, such as long-term consistency in story telling[116, 117] that models have yet to properly mimic for very long documents.

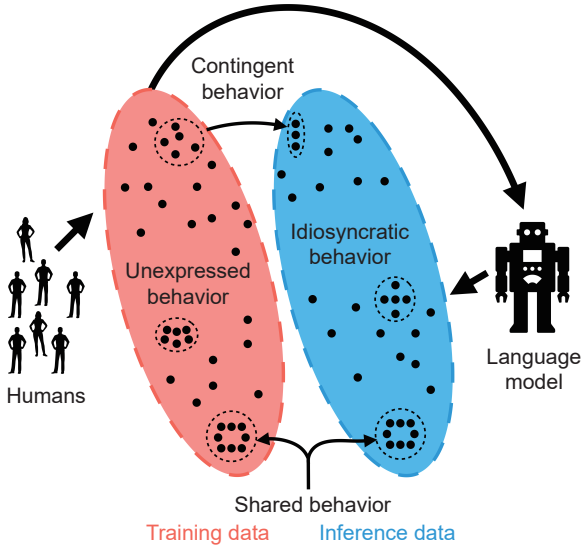

Fig. 6 Generative language models trained to capture the distribution of training data, then exhibit behavior in model outputs, i.e., *inference data*. See Section 3.5 for examples of the different behavioral mappings.

4 A Different Kind of Complex System

One reasonable worry is that taking on the complex systems lens will be fruitless because studying complex systems is a very difficult task, and we are not equipped to tackle such a hard problem.

In fact, compared to other complex systems, such as the brain, understanding current generative models is an immensely *easier* challenge, and can help us develop tools for the future. Turning our attention to "What, precisely, do language models do?" over "What is the best recipe for training large models?", we can take full advantage of the *complete simulability* of generative

models. In the long run, it seems it will become more difficult to address the latter question coherently without better answers to the former.

4.1 Two kinds of complex systems simulations

"Complex systems theory is divided between two basic approaches. The first involves the creation and study of simplified mathematical models that, while they may not mimic the behavior of real systems exactly, try to abstract the most important qualitative elements into a solvable framework from which we can gain scientific insight...The second approach is to create more comprehensive and realistic models, usually in the form of computer simulations, which represent the interacting parts of a complex system, often down to minute details, and then to watch and measure the emergent behaviors that appear[76]."

At first glance, generative models can largely be described as the second complex systems approach: we train models to capture properties of the natural distribution of human media, such as internet text, images, videos, etc., and then attempt to study the emergent effects. Yet, this would be a mischaracterization of what, e.g., language models do: we do not expect that language models learn language the way a human does nor create languages the way the human species did.‡

Instead, the triumphs of generative models are the result of emergent behavior within computational models trained to predict very general objectives. Many have been surprised that by learning on massively more data from a given medium than a human is ever exposed to, generative models can learn uncannily human patterns from simple, passive word prediction, denoising objectives, etc.

Generative models are certainly a computational simulation, but they are a simulation of an entire medium rather than a singular process we have isolated. We suggest thinking of generative models as a different kind of complex system, where *underlying patterns of a medium are learned by a model through optimization, and we then look for those patterns within the model.* Below we list ways in which this discovery process is made easier because our system of interest is the computational model itself, rather than a naturally arising system.

‡ Though this certainly describes facets of certain subfields such as emergent communication[118], with recent work taking advantage of pretrained models[119], and developmentally plausible pretraining such as the recent BabyLM challenge[120].

4.2 Generative models: The easiest complex system to study

(1) Perfect fidelity state encoding. Because neural networks are formal mathematical models, represented by code and parameters, there is zero *necessary ambiguity* in our representations. Imperfect data archiving and complex code bases often make it difficult to perfectly recover the formal model, but with sufficient effort, it is possible to store every bit of information about the state of the model at every computation step. We cannot track every neuron in a participant's brain at every moment due to the limited nature of our measurement instruments, but we can perfectly record the state of an LM in order to look for and verify emergent behavior, without influencing the system as we note in Advantage (5).

(2) Complete theory of low-level dynamics. While Advantage (1) establishes that we can perfectly store the state of a generative model at any given moment in time, Advantage (2) notes that we have a perfect, mechanistic, and deterministic understanding of how one state of a model evolves into the next, unlike in physical experiments. Artificial neural networks do not need to be simulated, they are defined in a medium for simulation: executable code. Unlike in physics, where centuries of research have been spent chasing the bottom of the chain of causation, we *begin* with the base-level causal structure of the model. This does not follow directly from Advantage (1). It is possible to imagine a scenario where every static state is recordable, but where the rules that govern the changes in states are hard to discover, e.g., the problem of learning video game dynamics from pixels[121]. In practice, nondeterminism exists in certain fast computations[122], but this can be removed at the cost of speed.

(3) Exact repeatability. Directly entailed by Advantages (1) and (2) is the fact that experiments can be repeated *exactly*. An algorithm that uses randomness to generate text, may generate different text on a second run, but as long as the probabilities of different tokens are recorded the likelihood of that text (and of alternative branching paths) can be verified to be exactly the same. A psychologist who conducts a study twice will almost never get results

that are exactly the same, simply because sample differences and unmeasured variables have to be accounted for. We distinguish repeatability from the broader notion of *replicability*, which also includes replicating a study to the level of detail described by the authors, leaving room for both human and systematic error. With proper code, data, and model releases, many generative model experiments are exactly repeatable, allowing us to reach for a much higher standard for replicability.

(4) Ease of perturbation. We also have a complete description of *all possible models* given a certain setup, e.g., all possible combinations of weights for a given architecture. Combining this with Advantage (2), we can perturb a model of interest, and play out experiments with this new model *without destroying the original model*. Contrast this with studying human language production, for which most perturbations of the human brain are both unethical and illegal, partially because humans cannot be unperturbed. This allows for extremely targeted experiments, e.g., finding which weights in a network control a certain decision boundary.

(5) No observer effects. A classic problem in many complex systems is that by attempting to make a measurement one changes the value being measured, e.g., Clever Hans a horse that could allegedly play chess, but was simply reading the audience's reaction to possible moves[123]. In contrast, generative models do not distinguish between the same input given for different reasons or with different expectations by the experimenter. The caveat is that experimenters still control the input distributions to experiments allowing for systematic bias that accidentally leaks *experimenter expectations*[124] to the model, as past research has consistently shown[71, 76, 125]. We must be careful about "tells"[126]: stylistic and semantic artifacts that make it into the data which can give the model information the experimenter assume that it does not have access to. Yet, the guarantee that the specific observer will not change the result is strong.

Advantages (1) and (2) allow us to completely remove any worry about hidden variables that may explain effects we attempted to explain through other means. Advantages (3) and (4), allow us to experiment freely, knowing that experiments and models that have been properly recorded are recoverable, leaving us free to perturb and explore the local neighborhood of similar models and setups. Advantage (5) partially relieves us of the fear of influencing the outcome through our means of observation, a key issue in many experiments involving language.

Another advantage, that does not apply to every generative model, deserves an honorable mention:

(6*) (Some) generative models exclusively output human understandable media. Many complex systems, such as cities and brains, produce human understandable media as some percentage of their output. Many generative models produce human understandable media as their *only* output, an enormous advantage for two reasons. First, humans are better suited to positing patterns in human understandable media than, say, subatomic particles. Second, *the uncanny valley effect*[127] allows us to see when patterns are "almost correct but not quite" much more easily in human-related artifacts. While we sometimes finetune models to produce outputs that are no longer human understandable, by and large current generative models operate entirely within human media—and we believe that there is much that can be learned from this that will transfer over to generative models of other media.

Advantage (6*) is very special. By allowing us to take advantage of our intuitive understanding of media, it becomes easier to seek out the ways generated media diverges from the natural human media we are steeped in from infancy. Indeed, most named behaviors are failure modes, e.g., degeneration[49] or empty translations[115].

Organic chemistry has given a great deal to biology, but is very much indebted to it as well. Our hope is that we can take inspiration from these other complex system sciences to start taking the problem of understanding *behavior* seriously, as a distinct abstraction that needs to be decomposed and theorized, while putting our enormous advantages to good use (Fig. 7).

4.3 Necessity of open-source models

Most of these advantages rely on stable access to a consistent representation of a model, which is difficult to guarantee via a proprietary API.

(1) Perfect fidelity state encoding. It is difficult to work with or guarantee saved state that is

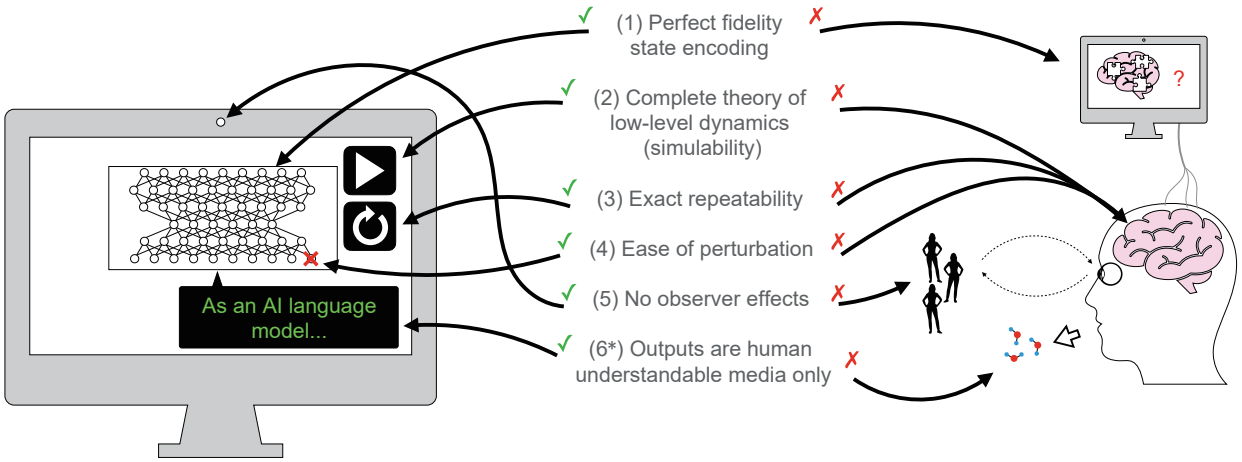

Fig. 7 Visual representation of generative model researchers' advantages over researchers that study the main other media generating system on earth: Human beings.

persistent and untampered without direct access to said state. Even cryptographically signed state can be tampered *after* re-submission to an API for use there, making guarantees moot.

(2) Complete theory of low-level dynamics. With only imperfect knowledge of an underlying model, researchers must make assumptions about low-level dynamics in a model that may only partially be true, or possibly even completely false.

(3) Exact repeatability. In practice, it is impossible to guarantee that an API will not drift over time, something observed with even the apparent attempts at stable APIs in recent years[38].

(4) Ease of perturbation. It is normally impossible to perturb a model through an API, though some APIs allow for finetuning and special versions of models. However, the real issue is that it is impossible to ensure that such perturbations do precisely what they are claiming to do to the model, without access to the model or even the model architecture in most cases.

(5) No observer effects. Sadly, even though this is one of the greatest advantages of generative models, it is the one most destroyed by using models via APIs: companies consistently, and often silently, fix undesired (from the company's perspective) behaviors in models[112, 128, 129] so that testing a certain hypothesis tends to influence future tests.

(6*) (Some) generative models exclusively output human understandable media. Without complete access to a model, it is impossible to know if it does not have other outputs (or inputs) that would help explain the model's behavior more fully.

In short, without access to open-source models, these advantages are largely moot. However, the community has seen a consistent open-source releases of better generative models in many different media[124, 130, 131]. There is unquestionably lag in the capabilities between proprietary and open-source models, and this is out of necessity: open-source cannot outpace private industry when private industry controls most of the training resources and can build on top of anything open-source does. But the fact that open source often lags only a year or two behind in terms of capabilities, and the fact that private labs are often incentivized to open-source models as a recruiting and market strategy, suggests that open-source will continue to be a wellspring of fascinating generative models to study. Indeed, if all progress stopped now, we believe that it would be decades before we finished cataloging all of the generalizable behavioral principles with the hundreds of large generative models that have already been released; perhaps our successes would encourage future open-source releases.

5 Conclusion

How should we study models of data, when we do not fully understand the models or the data? We should study them first by asking *what* models do, before attempting the more complicated *how* and the bottomless question of *why*?

In this paper, we presented a thought experiment: the Newformer, a model that would be impossible to study with many of the techniques we use to

understand Transformer models today.

We argue that focusing on what *behaviors* explain its performance across tasks will lead-us to a deeper understanding of generative models' tendencies and guide bottom-up mechanistic explanation, as well as forming building blocks for evaluations.

We discuss how generative models are well captured by the definition of a complex system, due to the emergent behaviors they exhibit. This separates generative models from traditional machine learning, where models often served as explanations via behaviors that were architected directly into them. This opens up the need for *metamodels* that help us predict regularities in generative model outputs in order to understand them better.

While the prospect of studying models we do not have a clear understanding of is daunting, we highlight advantages that generative models have over naturally arising complex systems. These advantages, however, require open-source models as a prerequisite, a point we emphasize as a necessity for conducting replicable science.

6 Limitation

We present one perspective on the kind of science NLP is becoming, and how we can leverage the complex systems lens in order to better explore the phenomena we find ourselves faced with: generative models we do not fully understand. We cite evidence from NLP publications, blog posts, and other media, but this necessarily does not capture the totality of perspectives.

Indeed, we purposefully avoid attempting any sort of survey of these issues, as this would involve citing thousands of papers and be a very unwieldy object. Instead, we attempt to form an argument as economically as possible, attempting to put forth a new set of goals and principles for how to study generative models given current progress.

We make comparisons with other sciences and cite sources from those sciences where appropriate, but are extremely limited in expressing many equally relevant connections and in fully exploring the connections we do mention. There is an enormous amount related to sister fields (e.g., cognitive science, linguistics, etc.), other sciences that study complex systems (e.g., chemistry, biology, etc.), and regarding more meta-science issues (e.g., complex systems theory, chaos theory, etc.) that we could not cover, and we do not in any way attempt to—giving a complete account of these connections is simply beyond the reach of any one work.

Finally, parts of our assessment is necessarily subjective. We attempt to lay out the evidence as we see it, tracing the connections we drew in order to describe a style of research that we believe is necessary to face the current challenges of our field. This seems especially pertinent in a time when most researchers cannot train large generative models from scratch, but are excited to contribute to their study. With evidence drawn from the literature, we describe the current research space as we perceive it, and our vision for where it might go. Our hope is that this will add to a discussion on what the study of generative models currently is and what we, as a community, would like it to become.

Acknowledgment

We thank Julian Michael, Dallas Card, Jared Moore, Daniel Fried, Gabriel Ilharco, Tim Dettmers, Ian Magnusson, Alisa Liu, and Kaj Bostrom for their insightful discussions and feedback.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. 31st International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 6000–6060.

[2] R. Al-Rfou, D. Choe, N. Constant, M. Guo, and L. Jones, Character-level language modeling with deeper self-attention, arXiv preprint arXiv: 1808.04444, 2018.

[3] N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, et al., A mathematical framework for transformer circuits, https://transformer-circuits.pub/2021/framework/index.html, 2021.

[4] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, et al., In-context learning and induction heads, https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html#argument-architectural-requirements, 2022.

[5] L. Wang, J. Huang, K. Huang, Z. Hu, G. Wang, and Q. Gu, Improving neural language generation with spectrum

control, presented at the International Conference on Learning Representations, Addis Ababa, Ethiopia, 2020.

[6] K. Ethayarajh, How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings, in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, China, 2019, pp. 55–65.

[7] J. Gao, D. He, X. Tan, T. Qin, L. Wang, and T. Liu, Representation degeneration problem in training natural language generation models, presented at the International Conference on Learning Representations 2019, New Orleans, LA, USA, 2019.

[8] D. Biś, M. Podkorytov, and X. Liu, Too much in common: Shifting of embeddings in transformer language models and its implications, in *Proc. 2021 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online, 2021, pp. 5117–5130.

[9] N. Godey, E. de la Clergerie, and B. Sagot, Is anisotropy inherent to transformers? arXiv preprint arXiv: 2306.07656, 2023.

[10] W. Rudman and C. Eickhoff, Stable anisotropic regularization, arXiv preprint arXiv: 2305.19358, 2023.

[11] Y. Lakretz, G. Kruszewski, T. Desbordes, D. Hupkes, S. Dehaene, and M. Baroni, The emergence of number and syntax units in LSTM language models, arXiv preprint arXiv: 1903.07435, 2019.

[12] C. Olah, Understanding LSTM networks, https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.

[13] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, Deep contextualized word representations, in *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1* (*Long Papers*), New Orleans, LA, US, 2018, pp. 2227–2237.

[15] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1* (*Long and Short Papers*), Minneapolis, Minnesota, 2019, pp. 4171–4186.

[16] A. Karpathy, J. Johnson, and L. Fei-Fei, Visualizing and understanding recurrent networks, arXiv preprint arXiv: 1506.02078, 2015.

[17] G. Weiss, Y. Goldberg, and E. Yahav, Thinking like transformers, in *Proc. 38th International Conference on Machine Learning*, Virtual Event, 2021, pp. 11080–11090.

[18] A. Rogers, O. Kovaleva, and A. Rumshisky, A primer in BERTology: What we know about how BERT works, *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 842–866, 2020.

[19] J. Bastings, Y. Belinkov, Y. Elazar, D. Hupkes, N. Saphra, and S. Wiegreffe, BlackboxNLP analyzing and interpreting neural networks for NLP, presented at the Microsoft at EMNLP 2022, Hybrid, United Arab Emirates, 2022.

[20] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, Beyond accuracy: Behavioral testing of NLP models with CheckList, in *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 4902–4912.

[21] T. Linzen, G. Chrupała, Y. Belinkov, and D. Hupkes, Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, https://aclanthology.org/W19-4800/, 2019.

[22] Y. Belinkov and Y. Bisk, Synthetic and natural noise both break neural machine translation, arXiv preprint arXiv: 1711.02173, 2017.

[23] I. Provilkov, D. Emelianenko, and E. Voita, BPE-dropout: Simple and effective subword regularization, in *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 1882–1892.

[24] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[25] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., arXiv preprint arXiv: 2005.14165v1, 2020.

[26] A. Venigalla and L. Li Mosaic LLMs: GPT-3 quality for <$500k, https://www.mosaicml.com/blog/gpt-3-quality-for-500k, 2023.

[27] J. Schulman, B. Zoph, C. Kim, J. Hilton, J. Menick, J. Weng, J. F. C. Uribe, L. Fedus, L. Metz, M. Pokorny, et al., Introducing ChatGPT, https://openai.com/index/chatgpt/, 2022.

[28] OpenAI, GPT-4 API general availability and deprecation of older models in the completions API, https://openai.com/blog/gpt-4-api-general-availability, 2023.

[29] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin, and X. Hu, Harnessing the power of LLMs in practice: A survey on ChatGPT and beyond, arXiv preprint arXiv: 2304.13712, 2023.

[30] C. Zhang, C. Zhang, C. Li, Y. Qiao, S. Zheng, S. K. Dam, M. Zhang, J. U. Kim, S. T. Kim, J. Choi, et al., One small step for generative AI, one giant leap for AGI: A complete survey on ChatGPT in AIGC era, arXiv preprint arXiv: 2304.06488, 2023.

[31] T. Eloundou, S. Manning, P. Mishkin, and D. Rock, GPTs are GPTs: An early look at the labor market impact potential of large language models, arXiv preprint arXiv: 2303.10130, 2023.

[32] A. S. George and A. S. H. George, A review of ChatGPT AI's impact on several business sectors, *Partners Universal International Innovation Journal*, vol. 1, no. 1, pp. 9–23, 2023.

[33] P. P. Ray, ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope, *Internet Things Cyber Phys. Syst.*, vol. 3, pp. 121–154, 2023.

[34] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, G-eval: NLG evaluation using GPT-4 with better human alignment, arXiv preprint arXiv: 2303.16634, 2023.

[35] L. Zheng, W. L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, et al., Judging LLM-as-a-judge with MT-Bench and chatbot arena, arXiv preprint arXiv: 2306.05685v4, 2023.

[36] A. Perry, OpenAI updates GPT-4 with new features, https://mashable.com/article/openai-chatgpt-gpt-4-function-calling-update, 2023.
[37] M. G. Southern, OpenAI's ChatGPT update brings improved accuracy, https://www.searchenginejournal.com/openai-chatgpt-update/476116/, 2023.
[38] Y. Deng, OpenAI watch, https://openaiwatch.com/, 2023.
[39] M. Alizadeh, M. Kubli, Z. Samei, S. Dehghani, J. D. Bermeo, M. Korobeynikova, and F. Gilardi, Open-source large language models outperform crowd workers and approach ChatGPT in text-annotation tasks, arXiv preprint arXiv: 2307.02179, 2023.
[40] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi, and A. Awadallah, Orca: Progressive learning from complex explanation traces of GPT-4, https://www.microsoft.com/en-us/research/publication/orca-progressive-learning-from-complex-explanation-traces-of-gpt-4/?locale=zh-cn, 2023.
[41] S. Gunasekar, Y. Zhang, J. Aneja, C. C. T. Mendes, A. Del Giorno, S. Gopi, M. Javaheripi, P. Kauffmann, G. de Rosa, O. Saarikivi, et al., Textbooks are all you need, arXiv preprint arXiv: 2306.11644, 2023.
[42] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. A. Lachaux, T. Lacroix, B. Roziere, N. Goyal, E. Hambro, F. Azhar, et al., LLaMA: Open and efficient foundation language models, arXiv preprint arXiv: 2302.13971v1, 2023.
[43] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, E. Goffinet, D. Heslow, J. Launay, Q. Malartic, et al., Falcon-40B: An open large language model with state-of-the-art performance, 2023.
[44] Stability AI, StableLM: StableLM: Stability AI language models, 2023.
[45] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, Stanford alpaca: An instruction-following LLaMA model, https://github.com/tatsu-lab/stanford_alpaca, 2023.
[46] The Vicuna Team, Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality, https://lmsys.org/blog/2023-03-30-vicuna/, 2023.
[47] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, QLoRA: Efficient finetuning of quantized LLMs, arXiv preprint arXiv: 2305.14314, 2023.
[48] R. C. Fong and A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in *Proc. IEEE Int. Conf. Computer Vision* (*ICCV*), Venice, Italy, 2017, pp. 3449–3457.
[49] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, The curious case of neural text degeneration, arXiv preprint arXiv: 1904.09751, 2019.
[50] C. Olah, Mechanistic interpretability, variables, and the importance of interpretable bases, https://transformer-circuits.pub/2022/mech-interp-essay/index.html, 2022.
[51] Z. C. Lipton, The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery, *Queue*, vol. 16, no. 3, pp. 31–57, 2018.
[52] A. Jacovi and Y. Goldberg, Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? in *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 4198–4205.
[53] C. Chen, S. Feng, A. Sharma, and C. Tan, Machine explanations and human understanding, in *Proc. 2023 ACM Conf. Fairness, Accountability, and Transparency*, Chicago, IL, USA, 2023, p. 1.
[54] T. M. Cover, *Elements of Information Theory*. New York, NY, USA: John Wiley & Sons, 1999.
[55] P. D. Grünwald, *The Minimum Description Length Principle*. Cambridge, MA, USA: The MIT Press, 2007.
[56] C. M. Barry, Who sharpened occam's razor? https://www.irishphilosophy.com/2014/05/27/who-sharpened-occams-razor/, 2014.
[57] R. R. Barton and M. Meckesheimer, Metamodel-based simulation optimization, *Handbooks in Operations Research and Management Science*, vol. 13, pp. 535–57, 2006.
[58] A. Ćwiek, S. Fuchs, C. Draxler, E. L. Asu, D. Dan, K. Hiovain, S. Kawahara, S. Koutalidis, M. Krifka, P. Lippus, et al., The bouba/kiki effect is robust across cultures and writing systems, *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, vol. 377, no. 1841, p. 20200390, 2022.
[59] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, Language models are unsupervised multitask learners, https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf, 2019.
[60] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, arXiv preprint arXiv: 2107.13586, 2021.
[61] A. Radford, J. Wu, D. Amodei, D. Amodei, J. Clark, M. Brundage, and I. Sutskever, Better language models and their implications, https://openai.com/index/better-language-models/, 2019.
[62] B. Z. Li, J. Yu, M. Khabsa, L. Zettlemoyer, A. Halevy, and J. Andreas, Quantifying adaptability in pre-trained language models with 500 tasks, arXiv preprint arXiv: 2112.03204, 2021.
[63] A. Rohrbach, L. A. Hendricks, K. Burns, T. Darrell, and K. Saenko, Object hallucination in image captioning, in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing*, Brussels, Belgiu, 2018, pp. 4035–4045.
[64] T. Liao, R. Taori, I. D. Raji, and L. Schmidt, Are we learning yet? A meta review of evaluation failures across machine learning, in 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021.
[65] National Association for College Admission Counseling, Report of the commission on the use of standardized tests in undergraduate admission, https://files.eric.ed.gov/fulltext/ED502721.pdf, 2008.
[66] Baidu Baike, Nationwide Unified Examination for Admissions to General Universities and Colleges, (in Chinese), https://baike.baidu.com/item/普通高等学校招生全国统一考试/2567351, 2022.
[67] C. Wang and R. Sennrich, On exposure bias, hallucination and domain shift in neural machine translation, in *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp.

3544–3552.
[68] M. Müller, A. Rios, and R. Sennrich, Domain robustness in neural machine translation, arXiv preprint arXiv: 1911.03109, 2020.
[69] A. See, P. J. Liu, and C. D. Manning, Get to the point: Summarization with pointer-generator networks, arXiv preprint arXiv: 1704.04368, 2017.
[70] A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme, Hypothesis only baselines in natural language inference, in *Proc. Seventh Joint Conf. Lexical and Computational Semantics*, New Orleans, LA, USA, 2018, pp. 180–191.
[71] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith, Annotation artifacts in natural language inference data, in *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2* (*Short Papers*), New Orleans, LA, USA, 2018, pp. 107–112.
[72] T. A. Chang and B. K. Bergen, Language model behavior: A comprehensive survey, arXiv preprint arXiv: 2303.11504, 2023.
[73] N. Jain, K. Saifullah, Y. Wen, J. Kirchenbauer, M. Shu, A. Saha, M. Goldblum, J. Geiping, and T. Goldstein, Bring your own data! Self-supervised evaluation for large language models, arXiv preprint arXiv: 2306.13651, 2023.
[74] S. Pichai, An important next step on our AI journey, https://blog.google/technology/ai/bard-google-ai-search-updates/, 2023.
[75] BigScience, Bigscience model training launched. BigScience Blog, 2022.
[76] M. E. J. Newman, Resource letter CS–1: Complex systems, *Am. J. Phys.*, vol. 79, no. 8, pp. 800–810, 2011.
[77] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al., Sparks of artificial general intelligence: Early experiments with GPT-4, arXiv preprint arXiv: 2303.12712, 2023.
[78] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al., Emergent abilities of large language models, arXiv preprint arXiv: 2206.07682, 2022.
[79] R. Teehan, M. Clinciu, O. Serikov, E. Szczechla, N. Seelam, S. Mirkin, and A. Gokaslan, Emergent structures and training dynamics in large language models, in *Proc. BigScience Episode #5—Workshop on Challenges & Perspectives in Creating Large Language Models*, Virtual Event, 2022, pp. 146–159.
[80] C. D. Manning, K. Clark, J. Hewitt, U. Khandelwal, and O. Levy, Emergent linguistic structure in artificial neural networks trained by self-supervision, *Proc. Natl. Acad. Sci. USA*, vol. 117, no. 48, pp. 30046–30054, 2020.
[81] J. H. Holland, *Complexity: A Very Short Introduction*. Oxford, UK: Oxford University Press, 2014.
[82] U. Khandelwal, K. Clark, D. Jurafsky, and L. Kaiser, Sample efficient text summarization using a single pre-trained transformer, arXiv preprint arXiv: 1905.08836, 2019.
[83] U. Khandelwal, H. He, P. Qi, and D. Jurafsky, Sharp nearby, fuzzy far away: How neural language models use context, in *Proc. 56th Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), Melbourne, Australia, 2018, pp. 284–294.
[84] L. Yu, D. Simig, C. Flaherty, A. Aghajanyan, L. Zettlemoyer, and M. Lewis, MEGABYTE: Predicting million-byte sequences with multiscale transformers, arXiv preprint arXiv: 2305.07185, 2023.
[85] I. Beltagy, M. E. Peters, and A. Cohan, Longformer: The long-document transformer, arXiv preprint arXiv: 2004.05150, 2020.
[86] R. Child, S. Gray, A. Radford, and I. Sutskever, Generating long sequences with sparse transformers, arXiv preprint arXiv: 1904.10509, 2019.
[87] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, Lost in the middle: How language models use long contexts, *Trans. Assoc. Comput. Linguist.*, vol. 12, pp. 157–173, 2024.
[88] S. Sun, K. Krishna, A. Mattarella-Micke, and M. Iyyer, Do long-range language models actually use long-range context? in *Proc. 2021 Conf. Empirical Methods in Natural Language Processing*, Online, 2021, pp. 807–822.
[89] O. Press, N. A. Smith, and M. Lewis, Shortformer: Better language modeling using shorter inputs, arXiv preprint arXiv: 2012.15832v2, 2021.
[90] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, Language models as knowledge bases? in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, China, 2019, pp. 2463–2473.
[91] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, On the properties of neural machine translation: Encoder–decoder approaches, in *Proc. SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, 2014, pp. 103–111.
[92] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, CTRL: A conditional transformer language model for controllable generation, arXiv preprint arXiv: 1909.05858, 2019.
[93] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, Defending against neural fake news, arXiv preprint arXiv: 1905.12616, 2019.
[94] A. Aghajanyan, D. Okhonko, M. Lewis, M. Joshi, H. Xu, G. Ghosh, and L. Zettlemoyer, HTLM: Hyper-text pre-training and prompting of language models, arXiv preprint arXiv: 2107.06955, 2021.
[95] S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi, Cross-task generalization *via* natural language crowdsourcing instructions, in *Proc. 60th Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), Dublin, Ireland, 2022, pp. 3470–3487.
[96] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al., Scaling instruction-finetuned language models, arXiv preprint arXiv: 2210.11416, 2022.
[97] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, arXiv preprint arXiv: 2203.02155, 2022.

[98] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, et al., LIMA: Less is more for alignment, arXiv preprint arXiv: 2305.11206, 2023.

[99] S. Mittal, S. Diallo, and A. Tolk, *Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach*. New York, NY, USA: John Wiley & Sons, 2018.

[100] G. Ilharco, R. Zellers, A. Farhadi, and H. Hajishirzi, Probing contextual language models for common ground with visual representations, in *Proc. 2021 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, 2021, pp. 5367–5377.

[101] L. Parcalabescu, A. Gatt, A. Frank, and I. Calixto, Seeing past words: Testing the cross-modal capabilities of pretrained V&L models on counting tasks, arXiv preprint arXiv: 2012.12352, 2020.

[102] I. Tenney, D. Das, and E. Pavlick, BERT rediscovers the classical NLP pipeline, in *Proc. 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 4593–4601.

[103] G. Daras and A. G. Dimakis, Discovering the hidden vocabulary of DALLE-2, arXiv preprint arXiv: 2206.00169, 2022.

[104] B. Hilton, No, DALL-E doesn't have a secret language. (or at least, we haven't found one yet) this viral DALL-E thread has some pretty astounding claims, but maybe the reason they're so astounding is that, for the most part, they're not true. thread (1/15), https://t.co/8F2WDp7ITK. https://twitter.com/benjamin_hilton/status/1531780892972175361?lang=en, 2022.

[105] R. T. McCoy, P. Smolensky, T. Linzen, J. Gao, and A. Celikyilmaz, How much do language models copy from their training data? Evaluating linguistic novelty in text generation using RAVEN, *Trans. Assoc. Comput. Linguist.*, vol. 11, pp. 652–670, 2023.

[106] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, Quantifying memorization across neural language models, arXiv preprint arXiv: 2202.07646, 2022.

[107] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini, Deduplicating training data makes language models better, in *Proc. 60th Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), Dublin, Ireland, 2022, pp. 8424–8445.

[108] K. Tirumala, A. H. Markosyan, L. Zettlemoyer, and A. Aghajanyan, Memorization without overfitting: Analyzing the training dynamics of large language models, arXiv preprint arXiv: 2205.10770v2, 2022.

[109] T. Blevins and L. Zettlemoyer, Language contamination helps explain the cross-lingual capabilities of English pretrained models, arXiv preprint arXiv: 2204.08110, 2022.

[110] X. V. Lin, T. Mihaylov, M. Artetxe, T. Wang, S. Chen, D. Simig, M. Ott, N. Goyal, S. Bhosale, J. Du, et al., Few-shot learning with multilingual language models, arXiv preprint arXiv: 2112.10668, 2021.

[111] N. Kandpal, E. Wallace, and C. Raffel, Deduplicating training data mitigates privacy risks in language models, arXiv preprint arXiv: 2202.06539, 2022.

[112] L. Kiho, ChatGPT_DAN: ChatGPT DAN, jailbreaks prompt.

[113] F. Stahlberg, I. Kulikov, and S. Kumar, Uncertainty determines the adequacy of the mode and the tractability of decoding in sequence-to-sequence models, in *Proc. 60th Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), Dublin, Ireland, 2022, pp. 8634–8645.

[114] X. Shi, Y. Xiao, and K. Knight, Why neural machine translation prefers empty outputs, arXiv preprint arXiv: 2012.13454, 2020.

[115] F. Stahlberg and B. Byrne, On NMT search errors and model errors: Cat got your tongue? in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, China, 2019, pp. 3356–3362.

[116] Z. Xie, T. Cohn, and J. H. Lau, Can very large pretrained language models learn storytelling with a few examples? 2023.

[117] A. See, A. Pappu, R. Saxena, A. Yerukola, and C. D. Manning, Do massively pretrained language models make better storytellers? in *Proc. 23rd Conf. Computational Natural Language Learning* (*CoNLL*), Hong Kong, China, 2019, pp. 843–861.

[118] A. Lazaridou and M. Baroni, Emergent multi-agent communication in the deep learning era, arXiv preprint arXiv: 2006.02419, 2020.

[119] S. Steinert-Threlkeld, X. Zhou, Z. Liu, and C. M. Downey, Emergent communication fine-tuning (EC-FT) for pretrained language models, presented at the ICLR 2022 EmeCom Workshop, 2022.

[120] A. Warstadt, L. Choshen, A. Mueller, A. Williams, E. Wilcox, and C. Zhuang, Call for papers: The BabyLM challenge: Sample-efficient pretraining on a developmentally plausible corpus, arXiv preprint arXiv: 2301.11796, 2023.

[121] D. Hafner, T. Lillicrap, I. S. Fischer, R. Villegas, D. R. Ha, H. Lee, and J. Davidson, Learning latent dynamics for planning from pixels, in *Proc. 36th International Conference on Machine Learning: ICML 2019*, Long Beach, CA, USA, 2019, pp. 2555–2565.

[122] M. Morin and M. Willetts, Non-determinism in TensorFlow ResNets, arXiv preprint arXiv: 2001.11396, 2020.

[123] W. Prinz, Messung kontra augenschein, *Psychol. Rundsch.*, vol. 57, no. 2, pp. 106–111, 2006.

[124] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, High-resolution image synthesis with latent diffusion models, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, LA, USA, 2022, pp. 10674–10685.

[125] T. McCoy, E. Pavlick, and T. Linzen, Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference, in *Proc. 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 3428–3448.

[126] M. Caro, *Caro's Book of Poker Tells*. Washington, DC, USA: Cardoza Publishing, 2003.

[127] M. Mori, The uncanny valley: The original essay by

Masahiro Mori, https://spectrum.ieee.org/the-uncanny-valley, 2012.
[128] A. Wilson, How to jailbreak ChatGPT to unlock its full potential, https://approachableai.com/how-to-jailbreak-chatgpt/, 2023.
[129] E. Eliaçık, Playing with fire: The leaked plugin DAN unchains ChatGPT from its moral and ethical restrictions, https://dataconomy.com/2023/03/31/chatgpt-dan-prompt-how-to-jailbreak-chatgpt/, 2023.

Ari Holtzman received the PhD degree from University of Washington, USA, where he won the William Chan Memorial Dissertation Award and was part of the team that won the inaugural Amazon Alexa Prize in 2017. He is currently an assistant professor of computer science and data science at University of Chicago, USA, where he runs Conceptualization Lab. His research focuses on generative models of language, including contributions like nucleus sampling and work on understanding what language models are actually doing and how they perceive text.

[130] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar, et al., Voicebox: Text-guided multilingual universal speech generation at scale, arXiv preprint arXiv: 2306.15687, 2023.
[131] Z. Luo, D. Chen, Y. Zhang, Y. Huang, L. Wang, Y. Shen, D. Zhao, J. Zhou, and T. Tan, VideoFusion: Decomposed diffusion models for high-quality video generation, arXiv preprint arXiv: 2303.08320, 2023.