Subjective Randomness and In-Context Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models (LLMs) exhibit intricate capabilities, often achieving high performance on tasks they were not explicitly trained for. The precise nature of LLM capabilities is often unclear, with different prompts eliciting different capabilities, especially when used with in-context learning (ICL). We propose a "Cognitive Interpretability" framework that enables us to analyze ICL dynamics to understand latent concepts underlying LLMs' behavioral patterns. This provides a more nuanced understanding than posthoc evaluation benchmarks, but does not require observing model internals as a mechanistic interpretation would require. Inspired by the cognitive science of human randomness perception, we use random binary sequences as context and study dynamics of ICL by manipulating properties of context data, such as sequence length. In the latest GPT-3.5+ models, we find emergent abilities to generate pseudo-random numbers and learn basic formal languages, with striking ICL dynamics where model outputs transition sharply from pseudo-random behaviors to deterministic repetition.

1 Introduction

Large language models (LLMs), especially when prompted via in-context learning (ICL), demonstrate complex, emergent capabilities [1–12]. Specifically, ICL yields task-specific behaviors in LLMs via use of different prompts (or *contexts*) [1, 5, 13–20]. Although no weight updates occur in ICL, different input contexts can activate, or re-weight, different latent algorithms in an LLM, analogous to how traditional learning methods such as gradient descent use training data to re-weight model parameters to learn representations [21–26]. Two seemingly equivalent prompts can, however, evoke very different behaviors in LLMs [18]. Our central motivation is to interpret emergent capabilities and latent *concepts* underlying complex behaviors in LLMs by analyzing in-context learning behavioral dynamics, without directly observing hidden unit activations or re-training models on varied datasets.

Inspired by computational approaches to human cognition [27–31], we model and interpret latent concepts evoked in LLMs by different contexts, without observing or probing model internals. This approach, which we call **Cognitive Interpretability**, is a middle ground between shallow test-set evaluation benchmarks on one hand [17, 32–40] and mechanistic neuron- and circuit-level understanding of pre-trained models' capabilities on the other [12, 41–53]. Computational cognitive scientists have related algorithmic information theory to human cognition, where mental concepts are viewed as programs, and cognitive hypothesis search over concepts is viewed as Bayesian inference [30, 54–58]. In this vein, Griffiths and Tenenbaum [28] model subjective randomness in human cognition as probabilistic program induction, where a person must search over a space of non-random programs in order to answer the question, "was this sequence generated by a random process?" We argue that ICL can similarly be seen as under-specified program induction, where there is no single "correct" answer; instead, an LLM should appropriately re-weight latent algorithms. The domain of random sequences reflects this framing, in contrast to other behavioral evaluation methodologies, in that there is no correct answer to a random number generation or judgment task (Fig. 1). If the correct behavior is to match a target random process, then the right way to respond to

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

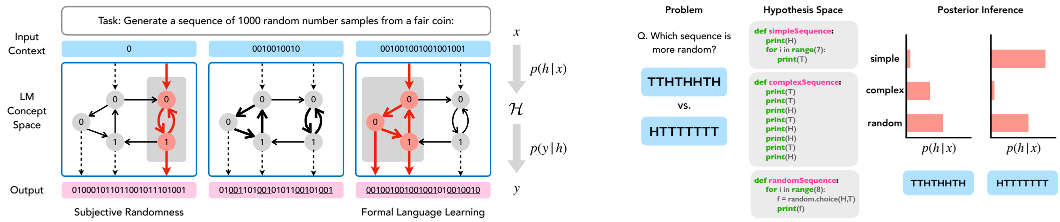

Figure 1: **Overview of our modeling framework.** (Left) Given a pre-trained LLM, we systematically vary input context prompts x. LLM outputs y vary as a function of x, based on some unknown latent concept space embedded in the LLM. With very little context ($x = \texttt{0}$), GPT-3.5+ generates subjectively random sequences, whereas with adequate context matching a simple formal language ($x = \texttt{001001001001}$), behavior becomes deterministic ($(\texttt{001})^n$). (Right) Deciding whether a sequence is random can be viewed as search for a simple program that could generate that sequence. `HTTTTTTT` is described with a short program `simpleSequence` with higher $p(h)$ according to a simplicity prior, compared to `TTHTHHTH` and `complexSequence`. Both sequences can be generated by `randomSequence`, with lower likelihood $p(x|h)$

a prompt *Generate N flips from a fair coin* is at best a uniform distribution over the tokens `Heads` and `Tails`, instead of a specific sequence, or a more complex algorithm that matches human behavior.

2 Background

Bayesian Inference and In-Context Learning A key methodological tool of cognitive modeling, recent work has also framed in-context learning as Bayesian inference over models [19, 59, 60]. Specifically, the posterior predictive distribution $p(y|x)$ in these works describes how an LLM produces output tokens y, given the context, or *prompt*, x. The key assumption is that a context x will activate latent concepts c within a model according to their posterior probability $p(c|x)$, which the model marginalizes over to produce the next token y by sampling from the posterior predictive distribution: $p(y|x) = \int_{c \in \mathcal{C}} p(y|c)\ p(c|x)$. This model selection takes place in network activation dynamics, without changing its weights. In our experiments, we assume a hypothesis space $\mathcal{H}$ that approximates the latent space of LLM concepts $\mathcal{C}$ used when predicting the next token, i.e., $p(y|x) = \sum_{h \in \mathcal{H}} p(y|h)\ p(h|x)$, where $\sum_h$ can be changed to $\max_h$ to represent deterministic greedy decoding with an LLM temperature parameter of 0. We specifically focus on Bernoulli processes, regular languages, Markov chains, and a simple memory-constrained probabilistic model as candidates for the hypothesis space $\mathcal{H}$ for estimating LLM concepts in random binary sequences. We use a subset of regular languages $(x)^n$, where (x) is a short sequence of values, e.g., $(\texttt{010})^n$, where 0 maps to Heads and 1 to Tails.

Algorithmic and Subjective Randomness Cognitive scientists studying *Subjective Randomness* model how people perceive randomness, or generate data that is subjectively random but algorithmically pseudo-random [29, 61–64]. In a bias termed *the Gambler's Fallacy*, people reliably perceive binary sequences with long streaks of one value as less random, and judge binary sequences with higher-than-chance alternation rates as being "more random" than truly random sequences [65, 66]. One way to study subjective randomness is to ask people whether a given data sequence was more likely to be generated by a Random process or a Non-Random process (Fig. 1). While the posterior distribution of all non-random processes includes every possible computable function, estimating this distribution can be simplified to finding the single most probable algorithm to approximate the full hypothesis space. If the hypotheses are data-generating programs, a natural prior $p(h)$ is to assign higher probabilities to programs with shorter description lengths, or lower complexity. This optimization problem is equivalent to computing the Kolmogorov complexity of a sequence $K(x)$ [67] and has motivated the use of "simplicity priors" in a number of domains in computational cognitive science [30, 54, 56]. Following previous work [28, 29], here we define subjective randomness of a sequence as the ratio of likelihood of that sequence under a random versus non-random model, i.e., $\texttt{randomness}(x) = \log P(x|\texttt{random})\ -\ \log P(x|\texttt{non-random})$. The non-random likelihood $p(x|\texttt{non-random}) = 2^{-K(x)}$ denotes the probability of the minimal description length program that generates x, equivalent to Bayesian model selection: $P(x|\texttt{non-random}) = \max_{h \in \mathcal{H}} p(x|h)\ p(h)$. In this work, we study a small subset of $\mathcal{H}$, which includes formal languages and probabilistic models inspired by psychological models of human concept learning and subjective randomness [29, 66, 68].

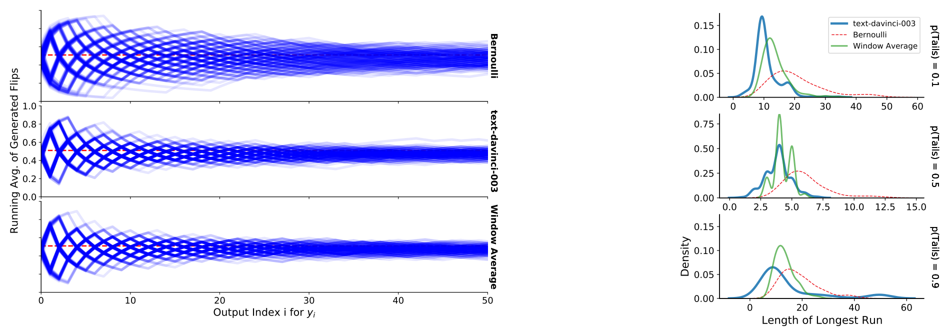

Figure 2: **GPT-3.5 generates pseudo-random binary sequences that deviate from a Bernoulli process.** (Left) Running averages of $p(\texttt{Tails})$ for flips generated by each model. Compared to a Bernoulli process, sequences generated by GPT and our Window Average model stay closer to the mean. (Right) GPT-3.5 shows a Gambler's Fallacy bias, avoiding long runs of the same value in a row.

3 Experiments

Randomness Generation and Judgment Tasks In order to assess text generation dynamics and in-context concept learning, we evaluate LLMs on random sequence **Generation** tasks, analyzing responses according to simple interpretable models of *Subjective Randomness* and *Formal Language Learning*. In these tasks, the model generates a sequence y of binary values, or *flips*, comma-separated sequences of `Heads` or `Tails` tokens. We also analyze a smaller set of randomness **Judgment** tasks, where the prompt includes a sequence of flips, and the model must respond whether the sequence was generated by Random or Non-Random process. In both cases, y is a distribution over tokens with two possible values: `Random` or `Non` in Judgment tasks, indicating whether the sequence was generated by a random process with no correlation, or some non-random algorithm. We analyze dynamics in LLM-generated sequences y simulating a weighted coin with specified $p(\texttt{Tails})$, with $|x| \approx 0$.

Subjective Randomness Models We compare LLM-generated sequences to a ground truth "random" Bernoulli distribution with the same mean ($\mu = \overline{y}_{\text{LLM}}$), to a simple memory-constrained probabilistic model, and to Markov chains fit to model-generated data y. Hahn and Warren [68] theorize that the Gambler's Fallacy emerges as a consequence of human memory limitations, where 'seeming biases reflect the subjective experience of a finite data stream for an agent with a limited short-term memory capacity'. We formalize this as a simple *Window Average* model, which tends towards a specific probability p as a function of the last w flips: $p(y|x) = \max(0, \min(1, 2p - \overline{x}_{t-w\ldots t}))$.

Sub-Sequence Memorization and Complexity Metrics Bender et al. [69] raise the question of whether LLMs are 'stochastic parrots' that simply copy data from the training set. To measure memorization, we look at the distribution of unique sub-sequences in y. If an LLM is repeating common patterns across outputs, potentially memorized from the training data, this should be apparent in the distribution over length K sub-sequences. Since there are deep theoretical connections between complexity and randomness [70, 71], we also consider the complexity of GPT-produced sequences. Compression is a metric of information content, and thus of redundancy over irreducible complexity [72, 73], and neural language models have been shown to prefer generating low complexity sequences [21]. As approximations of sequence complexity, we evaluate the distribution of Gzip-compressed file sizes [74] and inter-sequence Levenshtein distances [75].

Formal Language Learning Metrics In our Formal Language Learning analysis, x is a subset of regular expression repetitions of short token sequences such as $x \in (\texttt{011})^n$, where longer sequences x correspond to larger n. This enables us to systematically investigate in-context learning of formal languages, as $|x|$ corresponds to the amount of data for inducing the correct program (e.g. $(\texttt{011})^n$) out of the space of possible algorithms. In Randomness Judgment tasks, we assess formal concept learning by the dynamics of $p(y = \texttt{random}|x = C^{|x|})$ as a function of $|x|$. In Randomness Generation tasks, we asses concept learning according to the language model predictive distribution $p(y|x)$ over output sequences, inferred from next-token generation data: $p(y_{0\ldots T}|x) = p(y_0|x)\prod_t^T p(y_t|y_{0,\ldots t-1}, x)$. Given a space of possible outputs y with length d, $y \in \{0,1\}^d$, we estimate $p(y|x)$ by enumerating all y up to some depth d, and computing $\hat{p}(y_d|x, y_{1,\ldots,d-1}) = \frac{1}{N}\sum_i^N (y_d == 1)^{(i)}$ as the fraction of N responses that are "Tails" (or equivalently, by using token-level probabilities directly). We estimate the predictive probability $p(y_t \in C|x, y_{0\ldots,t-1})$ assigned to a given regular language by computing the total probability mass for all trajectories in $y_{0\ldots d}$ that exactly match C. For example, with $C = (\texttt{011})^n$, there will be 3 trajectories $y_{0\ldots d}$ that exactly match C, out of 2^d possible.

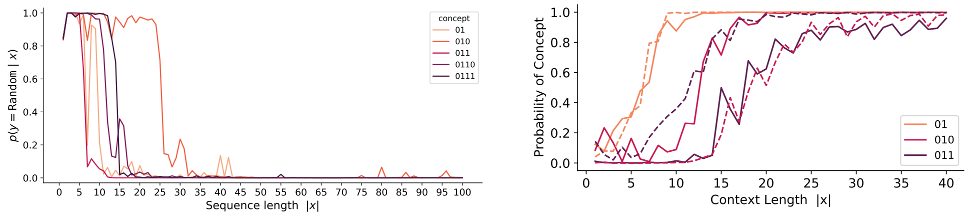

Figure 3: **Sharp transitions in predictive distributions for Randomness Judgment and Generation** (Left) In Randomness Judgment tasks, the predictive distribution $p(y = \texttt{random}|x)$ for `text-davinci-003` transitions from high confidence in x being generated by a random process, to high confidence in a non-random algorithm (Right) in Generation tasks, the predictive $p(y = \texttt{Tails}|x)$ transitions from pseudo-randomness to deterministic repetition of a particular concept; `text-davinci-003` is solid, `gpt-3.5-turbo-instruct` dashed.

4 Results

Subjectively Random Sequence Generation

In 'InstructGPT' models — `text-davinci-003`, ChatGPT (`gpt-3.5-turbo`, `gpt-3.5-turbo-instruct`) and GPT-4 — we find an emergent behavior of generating seemingly random binary sequences (Fig. 2). *This behavior is controllable*, where different $p(\texttt{Tails})$ values leads to different means of generated sequences $\overline{y}$. However, the distribution of sequence means, as well as the distribution of the length of the longest runs for each sequence, deviate significantly from a Bernoulli distribution centered at $\overline{y}$, analogous to the Gambler's Fallacy bias in humans. *Our Window Average model with a window size of $w = 5$ partly explains both biases*, matching GPT-generated sequences more closely than a Bernoulli distribution. Our cross-LLM analysis shows that `text-davinci-003` is controllable with $P(\texttt{Tails})$, with a bias towards $\overline{y} = .50$ and higher variance in sequence means (though lower variance than a true Bernoulli process). ChatGPT (`gpt-3.5-turbo-0301` and `0613`) are similar for $P(\texttt{Tails}) < 50\%$, but behave erratically with higher $P(\texttt{Tails})$, with most y repeating '`Tails`'. GPT-4 (*0301*, *0613*) shows stable, controllable subjective randomness behavior, with lower variances than `text-davinci-003`. Earlier models do not show subjective randomness behavior. Also see Appendix.

Sub-Sequence Memorization and Complexity We find significant differences between the distributions of sub-sequences for GPT-3.5 -generated sequences and sequences sampled from a Bernoulli distribution (see Appendix for figures). This difference is partly accounted for with a Window Average model with a window size $w = 5$, although GPT repeats certain longer sub-sequences, for example length-20 sub-sequences, that are far longer than 5. However, the majority of sub-sequences have very low frequency, and though further experiments would be required to conclude that all sub-sequences are not memorized from training data, it seems unlikely that these were in the training set, since we find thousands of unique length-k (with varying k) sub-sequences generated at various values of $P(\texttt{Tails})$. *This indicates that GPT-3.5 combines dynamic, subjectively random sequence generation with distribution-matched memorization.* Across three metrics of sequence complexity — number unique sub-sequences, Gzip file size, and inter-sequence Levenshtein distance — we find that *GPT-3.5+ models, with the exception of ChatGPT, generate low complexity sequences*, showing that structure is repeated across sequences and supporting prior work [21, 73].

4.1 Distinguishing Formal Languages from Randomness

GPT-3.5 sharply transitions between behavioral patterns, from generating pseudo-random values to generating non-random sequences that perfectly match the formal language (Fig. 3). We observe a consistent pattern of formal language learning in GPT-3.5 generating random sequences where predictions $p(y|x)$ of depth $d \geq 4$ are initially random with small $|x|$, and have low $p(y \in C|x)$ where C is a given concept. This follows whether the prompt describes the process as samples from "a weighted coin" or "a non-random-algorithm". We also find *sharp phase changes in GPT-3.5 behavioral patterns in Randomness Judgment tasks across 9 binary concepts* (Fig. 3). These follow a stable pattern of being highly confident in that the sequence is Random (high $p(y = \texttt{random}|x)$ when x is low, up to some threshold of context at which point it rapidly transitions to being highly confident in the process being non-random. Transition points vary between concepts, but the pattern is similar across concepts (see additional figures in Appendix).

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[3] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[4] Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

[5] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[6] Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. Are emergent abilities in large language models just in-context learning? *arXiv preprint arXiv:2309.01809*, 2023.

[7] Rishi Bommasani, 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.

[8] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[9] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[10] Alexander Pan, Jun Shern Chan, Andy Zou, Nathaniel Li, Steven Basart, Thomas Woodside, Hanlin Zhang, Scott Emmons, and Dan Hendrycks. Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark. In *International Conference on Machine Learning*, pages 26837–26867. PMLR, 2023.

[11] Vishal Pallagani, Bharath Muppasani, Keerthiram Murugesan, Francesca Rossi, Biplav Srivastava, Lior Horesh, Francesco Fabiano, and Andrea Loreggia. Understanding the capabilities of large language models for automated planning. *arXiv preprint arXiv:2305.16151*, 2023.

[12] Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*, 2022.

[13] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[14] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

[15] Xinyi Wang, Wanrong Zhu, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*, 2023.

[16] Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al. On the effect of pretraining corpora on in-context learning by a large-scale language model. *arXiv preprint arXiv:2204.13509*, 2022.

[17] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?, October 2022. URL http://arxiv.org/abs/2202.12837. arXiv:2202.12837 [cs].

[18] Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, and He He. Measuring inductive biases of in-context learning with underspecified demonstrations. *arXiv preprint arXiv:2305.13299*, 2023.

[19] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-context Learning as Implicit Bayesian Inference, July 2022. URL http://arxiv.org/abs/2111.02080. arXiv:2111.02080 [cs].

[20] Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *arXiv preprint arXiv:2305.19420*, 2023.

[21] Micah Goldblum, Marc Finzi, Keefer Rowan, and Andrew Gordon Wilson. The no free lunch theorem, kolmogorov complexity, and the role of inductive biases in machine learning. *arXiv preprint arXiv:2304.05366*, 2023.

[22] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

[23] Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *arXiv preprint arXiv:2306.00297*, 2023.

[24] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

[25] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

[26] Yingcong Li, M Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and implicit model selection in in-context learning. *arXiv preprint arXiv:2301.07067*, 2023.

[27] Joshua Tenenbaum. Bayesian modeling of human concept learning. *Advances in neural information processing systems*, 11, 1998.

[28] Thomas L Griffiths and Joshua B Tenenbaum. From Algorithmic to Subjective Randomness. *Neural Information Processing Systems*, 2003.

[29] Thomas L. Griffiths, Dylan Daniels, Joseph L. Austerweil, and Joshua B. Tenenbaum. Subjective randomness as statistical inference. *Cognitive Psychology*, 103:85–109, June 2018. ISSN 00100285. doi: 10.1016/j.cogpsych.2018.02.003. URL https://linkinghub.elsevier.com/retrieve/pii/S0010028517302281.

[30] Steven T Piantadosi, Joshua B Tenenbaum, and Noah D Goodman. Bootstrapping in a language of thought: A formal model of numerical concept learning. *Cognition*, 123(2):199–217, 2012.

[31] Michael J Spivey, Sarah E Anderson, and Rick Dale. The phase transition in human cognition. *New Mathematics and Natural Computation*, 5(01):197–220, 2009.

[32] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[33] Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*, 2022.

[34] Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

[35] Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, et al. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459*, 2023.

[36] Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile Lukosuite, Amanda Askell, Andy Jones, Anna Chen, et al. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022.

[37] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

[38] Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. *arXiv preprint arXiv:2212.09251*, 2022.

[39] Ben Prystawski and Noah D Goodman. Why think step-by-step? reasoning emerges from the locality of experience. *arXiv preprint arXiv:2304.03843*, 2023.

[40] Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *arXiv preprint arXiv:2305.04388*, 2023.

[41] Neel Nanda, Lawrence Chan, Tom Liberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

[42] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 6(3): e30, 2021.

[43] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

[44] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

[45] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

[46] Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: Reverse engineering how networks learn group operations. *arXiv preprint arXiv:2302.03025*, 2023.

[47] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.

[48] Alex Foote, Neel Nanda, Esben Kran, Ioannis Konstas, Shay Cohen, and Fazl Barez. Neuron to graph: Interpreting language model neurons at scale. *arXiv preprint arXiv:2305.19911*, 2023.

[49] Ekdeep Singh Lubana, Eric J Bigelow, Robert P Dick, David Krueger, and Hidenori Tanaka. Mechanistic mode connectivity. In *International Conference on Machine Learning*, pages 22965–23004. PMLR, 2023.

[50] Tom Lieberum, Matthew Rahtz, János Kramár, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*, 2023.

[51] Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

[52] Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *Advances in Neural Information Processing Systems*, 35:34651–34663, 2022.

[53] Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *arXiv preprint arXiv:2306.17844*, 2023.

[54] Nick Chater and Paul Vitányi. Simplicity: a unifying principle in cognitive science? *Trends in cognitive sciences*, 7(1):19–22, 2003.

[55] Stanislas Dehaene, Fosca Al Roumi, Yair Lakretz, Samuel Planton, and Mathias Sablé-Meyer. Symbols and mental programs: a hypothesis about human singularity. *Trends in Cognitive Sciences*, 2022.

[56] Noah D Goodman, Joshua B Tenenbaum, Jacob Feldman, and Thomas L Griffiths. A rational analysis of rule-based concept learning. *Cognitive science*, 32(1):108–154, 2008.

[57] Tomer D Ullman, Noah D Goodman, and Joshua B Tenenbaum. Theory learning as stochastic search in the language of thought. *Cognitive Development*, 27(4):455–480, 2012.

[58] Tomer D Ullman and Joshua B Tenenbaum. Bayesian models of conceptual development: Learning as building models of the world. *Annual Review of Developmental Psychology*, 2: 533–558, 2020.

[59] Yingcong Li, M. Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as Algorithms: Generalization and Stability in In-context Learning, February 2023. URL http://arxiv.org/abs/2301.07067. arXiv:2301.07067 [cs, stat].

[60] Michael Hahn and Navin Goyal. A Theory of Emergent In-Context Learning as Implicit Structure Induction, March 2023. URL http://arxiv.org/abs/2303.07971. arXiv:2303.07971 [cs].

[61] An T. Oskarsson, Leaf Van Boven, Gary H. McClelland, and Reid Hastie. What's next? Judging sequences of binary events. *Psychological Bulletin*, 135(2):262–285, 2009. ISSN 1939-1455, 0033-2909. doi: 10.1037/a0014821. URL http://doi.apa.org/getdoi.cfm?doi=10.1037/a0014821.

[62] Giorgio Gronchi and Steven A. Sloman. Regular and random judgements are not two sides of the same coin: Both representativeness and encoding play a role in randomness perception. *Psychonomic Bulletin & Review*, 28(5):1707–1714, October 2021. ISSN 1069-9384, 1531-5320. doi: 10.3758/s13423-021-01934-9. URL https://link.springer.com/10.3758/s13423-021-01934-9.

[63] Florent Meyniel, Maxime Maheu, and Stanislas Dehaene. Human inferences about sequences: A minimal transition probability model. *PLoS computational biology*, 12(12):e1005260, 2016.

[64] Samuel Planton, Timo van Kerkoerle, Leïla Abbih, Maxime Maheu, Florent Meyniel, Mariano Sigman, Liping Wang, Santiago Figueira, Sergio Romano, and Stanislas Dehaene. A theory of memory for binary sequences: Evidence for a mental compression algorithm in humans. *PLOS Computational Biology*, 17(1):e1008598, January 2021. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1008598. URL https://dx.plos.org/10.1371/journal.pcbi.1008598.

[65] Ruma Falk and Clifford Konold. Making sense of randomness: Implicit encoding as a basis for judgment. 1997.

[66] Raymond S Nickerson. The production and perception of randomness. *Psychological review*, 109(2):330, 2002.

[67] Ming Li, Paul Vitányi, et al. *An introduction to Kolmogorov complexity and its applications*. Springer, 1997.

[68] Ulrike Hahn and Paul A Warren. Perceptions of randomness: why three heads are better than four. *Psychological review*, 116(2):454, 2009.

[69] Emily M Bender, 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*, pages 610–623, 2021.

[70] Gregory J Chaitin. Algorithmic information theory. *IBM journal of research and development*, 21(4):350–359, 1977.

[71] Gregory J Chaitin. *Information, randomness & incompleteness: papers on algorithmic information theory*, volume 8. World Scientific, 1990.

[72] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[73] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, et al. Language modeling is compression. *arXiv preprint arXiv:2309.10668*, 2023.

[74] Zhiying Jiang, Matthew YR Yang, Mikhail Tsirlin, Raphael Tang, and Jimmy Lin. Less is more: Parameter-free text classification with gzip. *arXiv preprint arXiv:2212.09410*, 2022.

[75] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.