LESSONS FROM IDENTIFIABILITY FOR UNDERSTANDING LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many interesting properties emerge in LLMs, including rule extrapolation, in-context learning, and data-efficient fine-tunability. We demonstrate that good statistical generalization alone cannot explain these phenomena due to the inherent non-identifiability of autoregressive (AR) probabilistic models. Indeed, models zero or near-zero KL divergence apart—thus, equivalent test loss—can exhibit markedly different behaviours. We illustrate the practical implications for AR LLMs regarding three types of non-identifiability: (1) the non-identifiability of zero-shot rule extrapolation; (2) the approximate non-identifiability of in-context learning; and (3) the non-identifiability of fine-tunability. We hypothesize these important properties in LLMs are induced by inductive biases.

1 INTRODUCTION

Autoregressive (AR) language models trained on the next-token prediction objective can have remarkable reasoning (Ouyang et al., 2022; Touvron et al., 2023; Wei et al., 2022), in-context learning (ICL) (Xie et al., 2022; Zhang et al., 2023; Min et al., 2022), and data-efficient fine-tuning capabilities (Brown et al., 2020; Liu et al., 2023).

Modern theory of deep learning studies neural networks in the *interpolation regime* (Zhang et al., 2016; Masegosa, 2020; Kawaguchi et al., 2022), i.e., when, at the end of training, a model reaches a (non-unique) global minimum of the training loss. Since Large Language Models (LLMs) are trained on massive datasets, these models achieve both low training and test loss; thus, they generalize in the statistical sense. However, statistical generalization cannot guarantee good performance on downstream tasks (Liu et al., 2023).

We advocate for studying LLMs in the *saturation regime* (Liu et al., 2023), where models reach the (non-unique) global minimum of the test loss during training; since the same minimal test loss cannot distinguish between out-of-distribution (OOD) model performance (Liu et al., 2023), we should ask *what additional properties hold for the minimum found by our algorithms*. To formalize such questions, we need to substitute the black box concept of average risk from statistical learning theory with more application-specific goals, e.g., rule extrapolation or the data efficiency of fine-tuning.

We can use the lens of identifiability to explain why the test loss has a non-unique minimum. Namely, unless their support spans the entire space of sequences, autoregressive (AR) probabilistic models are non-identifiable, i.e. indistinguishable by the likelihood, even in the limit of infinite data. The study of (non-)identifiability has a vast literature both in statistical inference and causal discovery. These are well-known results; we only aim to highlight the practical implications of non-identifiability for AR LLMs. We organize these in three case studies, which provide well-defined starting points to theoretically study LLMs. Our **contributions** are:

- We highlight the limitation of statistical generalization for explaining important AR LLMs properties common in the saturation regime (§ 2), and hypothesize that the right way to study them is through inductive biases;
- By studying rule extrapolation on zero-probability prompts in a toy example, we show experimentally that LLMs can extrapolate differently despite achieving similar test loss (case study § 3.1);
- We introduce an approximate notion of non-identifiability to demonstrate that properties relying on sets of prompts with vanishing probability are not guaranteed in real-world LLMs. For an LLM that almost perfectly matches a pre-training distribution given by a mixture of HMMs, we prove that ICL does not necessarily follow (case study § 3.2).
- We highlight that the data-efficient fine-tunability of LLMs is not explained by the function they implement, let alone the test loss (case study § 3.3).

2 Background

Statistical generalization measures whether a model's performance on the training data transfers to unseen test data, assumed to be sampled from the same distribution (i.e., i.i.d.). Classical results in statistical learning theory attempt to bound the generalization gap in terms of uniform notions of the model class' complexity (Vapnik & Chervonenkis, 1971; Vapnik, 2000; Bartlett & Mendelson, 2002). More applicable to deep learning are approaches that provide bounds based on the properties of the learning algorithm or the specific hypothesis learned. These include PAC-Bayes (Dziugaite & Roy, 2017; Pérez-Ortiz et al., 2021; Lotfi et al., 2022; 2023), information-theoretic (Russo & Zou, 2016; Xu & Raginsky, 2017; Wang et al., 2023a) and algorithmic stability bounds (Bousquet & Elisseeff, 2002; Deng et al., 2021). The appeal of statistical generalization is its "black-box" nature: it can be applied across many domains without changing the terminology. Slightly more domain-specific thinking is often introduced when one studies out-of-distribution (OOD) generalization (see Lin et al., 2022, for a review), since there it is necessary to describe how the test and training distributions differ.

Interpolation regime. Overparametrized models gave rise to the *interpolation regime*, where a model has enough parameters to (almost) perfectly fit the training data (Zhang et al., 2016; Masegosa & Ortega, 2023; Kawaguchi et al., 2022). In this case, the training loss *alone* cannot distinguish whether a model will generalize, yet models that we find by minimizing the training loss typically generalize well. This observation led to a paradigm shift in the community, inviting researchers to consider training dynamics and the inductive biases enabling statistical generalization instead of only relying on the model class, loss, and dataset structure. We advocate for a second paradigm shift: to focus on the inductive biases enabling OOD generalization.

Identifiability of Probabilistic Models. Identifiability is an important property of a class of statistical models, determining whether a model can always be uniquely recovered from observed data. In parametric statistical models, it asks whether the parameters of a model are uniquely determined by the data distribution they define (see e.g. Comon, 1994). In machine learning, identifiability can be interpreted as a guarantee that the test loss has a unique minimizer, a unique Bayes optimal model. This is a highly desirable quality: it allows us to reason about properties of this possibly unreachable but unique minimum, e.g. predict OOD extrapolation or the effect of interventions (Pearl, 2009).

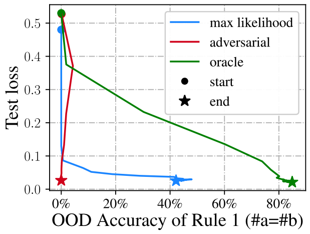

Figure 1: **OOD rule extrapolation in Transformers is better than chance:** We trained a decoder-only Transformer via maximum likelihood on the $a^n b^n$ Probabilistic Context-Free Grammar (PCFG). We evaluated it on OOD prompts inconsistent with $a^n b^n$, and checked whether the completions obey rule (R1) (x axis). Two other models, trained by an adversarial and an oracle process achieved the same test loss but displayed very different rule extrapolation accuracies. This demonstrates that test loss is insensitive to rule extrapolation behaviour and the 43.7% rule extrapolation accuracy (averaged over 20 seeds; details in Appx. D) results from inductive biases.

3 Three types of non-identifiability in AR LLMs

In this section, we discuss the (non-)identifiability of AR probabilistic models. By an AR probabilistic model we mean (for some fixed $T \in \mathbb{N}$) a collection $\{p(x_i|x_{1:i-1}); T \geq i \geq 1\}$ of conditional distributions, which also define a collection of joint distributions $\{p(x_{1:i}); T \geq i \geq 1\}$ over sequences. This collection of conditional distributions usually shares a set of parameters θ. We study *three notions of non-identifiability* that one might be interested in when studying such models: functional non-identifiability (§ 3.1), $\varepsilon-$non-identifiability (§ 3.2), and parameter non-identifiability (§ 3.3).

AR probabilistic models are inherently non-identifiable: multiple models with perfect generalization may exist and may behave differently. Here we showcase what this means for AR LLMs via three case studies, matching the three notions of non-identifiability from above. Our case studies provide clearly defined, relevant scenarios, which can be used as starting points to study LLMs theoretically.

3.1 Functional Non-Identifiability of rule extrapolation

Functional non-identifiability means that the collection of conditionals is not uniquely determined by the collection of joint distributions they define. Consider training an AR language model q to fit samples from the PCFG p over sequences of the form a^nb^n where n is random. Such a distribution has limited support since there are ungrammatical sequences that occur with probability 0. Moreover, there exist finite length prefixes $x_{1:l}$ which cannot be completed grammatically, whose marginal probability $p(x_{1:l})$ is thus 0. We refer to such prefixes as OOD prompts. We say that q generalizes perfectly (in the statistical sense) if KL $[p(x_{1:k})||q(x_{1:k})] = 0$, for some length k, i.e., it achieves maximal likelihood both in training and test. If $x_{1:l}$ has zero probability under p, the completion distribution $p(x_{l+1:k}|x_{1:l})$ is undefined. However, q still defines a distribution over completions $q(x_{l+1:k}|x_{1:l})$. Since the Kullback-Leibler Divergence (KL) divergence is insensitive to the choice of $q(x_{l+1:k}|x_{1:l})$, the completion distribution is *functionally* non-identifiable. This means that any property of q that depends only on completions of OOD prompts is non-identifiable. The a^nb^n grammar is the intersection of two rules:

(R1) the number of as and bs match; and
(R2) a never follows a b.

Unless a prompt can be completed consistently with both rules, the behaviour of q is non-identifiable. It is meaningful to ask whether a trained model q still respects rule (R1) when completing OOD prompts that break rule (R2), such as $abaa$. We call this *rule extrapolation*, illustrated in Fig. 4.

Experiment. We train a decoder-only Transformer (Vaswani et al., 2017; Radford et al., 2018) on the a^nb^n PCFG in a maximum likelihood estimation (MLE), adversarial and oracle setting and evaluate zero-shot rule extrapolation (Fig. 1). All models reach the same minimal test loss, but display widely varying rule extrapolation performance, with the MLE model reaching 43.7% on average. This demonstrates that statistical generalization alone does not explain rule extrapolation in LLMs.

3.2 $\varepsilon-$NON-IDENTIFIABILITY AND IN-CONTEXT LEARNING (ICL)

In § 3.1, we assumed that there are OOD sequences with exactly 0 probability under the pre-training distribution. A more realistic scenario is where some prompts have a non-zero but vanishingly small probability under the pre-training distribution. With full support, when non-zero probability is placed on all sequences, AR probability distributions are identifiable. However, relaxing the strict definition of identifiability and considering models near-equivalent if their test performance is barely distinguishable with the KL, we still find near-equivalent models that may behave radically differently on low-probability sequences, despite having access to infinite data. We call this $\varepsilon-$non-identifiability (for some small $\varepsilon > 0$) and define it informally (cf. Appx. C):

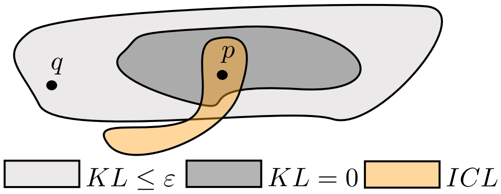

Figure 2: **Vanishingly small KL cannot capture in-context learning (ICL):** illustration of Prop. 1, showing that when p displays ICL, there exists a distribution q that is at least $\varepsilon-$close in KL divergence and has no ICL ability.

Definition 1 ($\varepsilon-$non-identifiability of distributional properties (informal))**.** *A distributional property of* p *is* $\varepsilon-$*non-identifiable if there is a distribution* q *such that KL* $[p||q] \leq \varepsilon$, *but* q *does not have the property of* p.

Contrary to traditional definitions, ours relaxes the distributional equivalence by admitting a non-zero KL, and is formulated about having a property (e.g., ICL). This distinction might seem subtle, yet is important since in practice, the goal is to have a well-performing model; minimizing a loss can be insufficient (Liu et al., 2023; Saunshi et al., 2022; Rusak et al., 2022; Tay et al., 2022).

Example: $\varepsilon-$**non-identifiability of ICL.** We use our definition to show that in-context learning (ICL) is $\varepsilon-$non-identifiable in some AR LLMs. ICL refers to the model's capacity to deduce novel concepts from a prompt and generate appropriate completions, and is studied theoretically both from an LVM (Wang et al., 2023b) and a Hidden Markov Model (HMM) (Xie et al., 2022) perspective. Xie et al. (2022) demonstrates that for a mixture of HMMs pre-training distribution p, the LLM is an in-context learner in the limit of infinite examples in the prompt. This means it produces completions aligning with the predictions of the prompt distribution.

We prove that ICL is $\varepsilon-$non-identifiable under the setting of (Xie et al., 2022), due to the low probability of OOD prompts. Hence the emergence of ICL is not a direct consequence of minimizing the negative log-likelihood. We detail the implications for the saturation regime in Appx. A.

We follow the notation and assumptions of Xie et al. (2022). Let $p(\theta)$ be a prior distribution on the latent concepts $\theta \in \Theta$, and let each θ define a distribution $p(o_1, \ldots, o_T|\theta)$ over sequences of tokens $o_1, \ldots, o_T \in \mathcal{O}$ of fixed pre-training document length T. Furthermore, assume that $p(o_1, \ldots, o_T|\theta)$ is defined by a HMM with a hidden state set $\mathcal{H}$, therefore the pre-training distribution is a mixture of HMMs parametrized by θ. Xie et al. (2022) proved that under some additional assumptions (Appx. B.2), ICL occurs. However, as we show, matching the pre-training distribution up to $\varepsilon > 0$ KL cannot guarantee ICL, even with increasing prompt size:

Proposition 1 ($\varepsilon-$non-identifiability of ICL, informal). *For all $\varepsilon > 0$, there exists $n_1 \geq n_0$, such that for all $n \geq n_1$, there exists a distribution q_n close to a mixture of HMMs in KL divergence*

$$KL\left[p(o_1, \ldots, o_N)||q_n(o_1, \ldots, o_N)\right] \leq \varepsilon, \; s.t. \quad ICL \text{ does not occur}.$$

Proof (Sketch). We construct a distribution q_n that matches p everywhere, except for a few sequences that end similarly to elements of the prompt distribution. There, we define q_n to guarantee different ICL behaviour. Then we bound KL $[p||q_n]$ and exploit that almost all conditionals are the same (except those we changed). Since the probability of prompts goes to zero as their length $n \to \infty$, we conclude that the KL converges to zero. The proof and details on notation are in Appx. B.2. □

3.3 Parameter Non-identifiability and Fine-tuning

A neural network's parametrization affects its learning dynamics (Saxe et al., 2014; Jacot et al., 2020; Dinh et al., 2017), which implies *parameter non-identifiability* (Fig. 3), i.e., functionally equivalent models can behave differently during fine-tuning and transfer. Parameter non-identifiability is relevant in LLMs, since large pre-trained models are often fine-tuned on new data sets to solve specialized tasks (see Tay et al., 2022, for a demonstration with Transformers). Liu et al. (2023) demonstrates parameter non-identifiability in a clever experiment: they "embed" a small Transformer into a larger one by maintaining functional equivalence and demonstrate that the different architectural constraints of the larger model interact differently with the optimization method: despite having the same pre-training loss, optimization will not prefer the embedded Transformer, but one with flatter minima, yielding a 10% difference in downstream accuracy after fine-tuning. This example highlights the need to understand what parametrizations are useful for improving fine-tuning and transfer in LLMs.

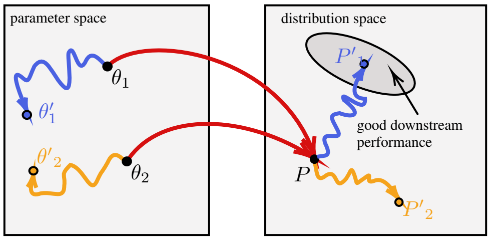

Figure 3: **Illustration of parameter non-identifiability:** Two sets of parameters (θ_1, θ_2) may describe the same AR LLM and thus achieve the same test loss and perform identically in benchmarks. When fine-tuned on the same data, parameter-dependent inductive biases may push the two models apart, and it is possible that, say, θ_1 enables significantly more data-efficient fine-tuning than θ_2.

4 Lessons from non-identifiability: the role of inductive biases

Our case studies (§ 2) highlight that statistical generalization does not explain important observed properties of real-world AR LLMs. We hypothesize that these properties emerge (at least in part) due to inductive biases, and hence the study of inductive biases is inescapable in understanding LLMs. Relevant inductive biases may result from, among others, (i) the complexity and structure of natural languages, (ii) the Transformer architecture (cf. Appx. E for a review). Studying these in formal languages such as PCFGs provides a valuable starting point as they have a controllable notion of complexity and structure.

5 Conclusion

Our work highlighted that due to the non-identifiability of probabilistic AR models, good statistical generalization cannot explain the desirable properties of LLMs. We studied three types of non-identifiability, demonstrated the limits of current theoretical frameworks in understanding LLM behavior, and highlighted how inductive biases could potentially explain emergent properties such as OOD rule extrapolation (§ 3.1), in-context learning (§ 3.2), and data-efficient fine-tunability (§ 3.3).

REFERENCES

Joshua Ackerman and George Cybenko. A Survey of Neural Networks and Formal Languages, June 2020. URL `http://arxiv.org/abs/2006.01338`. arXiv:2006.01338 [cs]. 16

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization, 2019. 16

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks, 2017. 16

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002. URL `https://jmlr.csail.mit.edu/papers/v3/bartlett02a.html`. 2

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002. 2

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. 1

Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994. 2

Giacomo De Palma, Bobak Toussi Kiani, and Seth Lloyd. Random deep neural networks are biased towards simple functions, 2019. 16

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression, 2023. 16

Zhun Deng, Hangfeng He, and Weijie Su. Toward better generalization bounds with locally elastic stability. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2590–2600. PMLR, 18–24 Jul 2021. URL `http://proceedings.mlr.press/v139/deng21b.html`. 2

Kamaludin Dingle, Chico Camargo, and Ard Louis. Input–output maps are strongly biased towards simple outputs. *Nature Communications*, 9, 02 2018. doi: 10.1038/s41467-018-03101-6. 16

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp Minima Can Generalize For Deep Nets, May 2017. URL `http://arxiv.org/abs/1703.04933`. arXiv:1703.04933 [cs]. 4

Gintare Karolina Dziugaite and Daniel M. Roy. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data, October 2017. URL `http://arxiv.org/abs/1703.11008`. arXiv:1703.11008 [cs]. 2

William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL `https://github.com/Lightning-AI/lightning`. 14

Benoit Favre. Contextual language understanding Thoughts on Machine Learning in Natural Language Processing. 2020. 16

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, 2023. 16

Jordi Grau-Moya, Tim Genewein, Marcus Hutter, Laurent Orseau, Grégoire Delétang, Elliot Catt, Anian Ruoss, Li Kevin Wenliang, Christopher Mattern, Matthew Aitchison, and Joel Veness. Learning universal predictors, 2024. 16

Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Implicit regularization in matrix factorization, 2017. 16

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks, 2019. 16

Marcus Hutter. A theory of universal artificial intelligence based on algorithmic complexity, 2000. 16

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks, February 2020. URL http://arxiv.org/abs/1806.07572. arXiv:1806.07572 [cs, math, stat]. 4, 16

K. Kawaguchi, Y. Bengio, and L. Kaelbling. *Generalization in Deep Learning*, pp. 112–148. Cambridge University Press, December 2022. ISBN 9781316516782. doi: 10.1017/9781009025096.003. URL http://dx.doi.org/10.1017/9781009025096.003. 1, 2

A.N. Kolmogorov. On tables of random numbers. *Theoretical Computer Science*, 207(2):387–395, 1998. ISSN 0304-3975. doi: https://doi.org/10.1016/S0304-3975(98)00075-9. URL https://www.sciencedirect.com/science/article/pii/S0304397598000759. 16

Fengming Lin, Xiaolei Fang, and Zheming Gao. Distributionally robust optimization: A review on theory and applications. *Numerical Algebra, Control and Optimization*, 12(1):159–212, 2022. ISSN 2155-3289. doi: 10.3934/naco.2021057. URL https://www.aimsciences.org/article/id/5a228ac4-f49b-4499-aab6-0656d63eb577. 2

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same Pre-training Loss, Better Downstream: Implicit Bias Matters for Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 22188–22214. PMLR, July 2023. URL https://proceedings.mlr.press/v202/liu23ao.html. ISSN: 2640-3498. 1, 3, 4, 10, 16

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs, math]. 14

Sanae Lotfi, Marc Finzi, Sanyam Kapoor, Andres Potapczynski, Micah Goldblum, and Andrew Gordon Wilson. Pac-bayes compression bounds so tight that they can explain generalization, 2022. 2

Sanae Lotfi, Marc Finzi, Yilun Kuang, Tim Rudner, Micah Goldblum, and Andrew Wilson. Non-vacuous generalization bounds for large language models. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023. URL https://openreview.net/forum?id=7QRaAfbium. 2

Andres R. Masegosa. Learning under Model Misspecification: Applications to Variational and Ensemble methods. *arXiv:1912.08335 [cs, math, stat]*, October 2020. URL http://arxiv.org/abs/1912.08335. arXiv: 1912.08335 version: 5. 1

Andrés R. Masegosa and Luis A. Ortega. Understanding generalization in the interpolation regime using the rate function, 2023. 2

William Merrill. Formal languages and neural models for learning on sequences. In *Proceedings of 16th edition of the International Conference on Grammatical Inference*, pp. 5–5. PMLR, July 2023. URL https://proceedings.mlr.press/v217/merrill23a.html. ISSN: 2640-3498. 16

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?, 2022. 1

Chris Mingard, Joar Skalse, Guillermo Valle-Pérez, David Martínez-Rubio, Vladimir Mikulik, and Ard A. Louis. Neural networks are a priori biased towards boolean functions with low entropy, 2020. 16

Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L. Edelman, Fred Zhang, and Boaz Barak. SGD on neural networks learns functions of increasing complexity. *CoRR*, abs/1905.11604, 2019. URL http://arxiv.org/abs/1905.11604. 16

Chris Olah. Mechanistic interpretability, variables, and the importance of interpretable bases, 2022. 17

OpenAI et al. Gpt-4 technical report, 2023. 14

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. 1

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 14

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2 edition, 2009. ISBN 978-0-511-80316-1. doi: 10.1017/CBO9780511803161. URL http://ebooks.cambridge.org/ref/id/CBO9780511803161. 2

Ofir Press, Noah A. Smith, and Omer Levy. Improving Transformer Models by Reordering their Sublayers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2996–3005, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.270. URL https://aclanthology.org/2020.acl-main.270. 16

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22(227):1–40, 2021. URL http://jmlr.org/papers/v22/20-879.html. 2

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 14

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. pp. 24, 2018. 3, 14

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks, 2019. 16

Maria Refinetti, Alessandro Ingrosso, and Sebastian Goldt. Neural networks trained with sgd learn distributions of increasing complexity, 2023. 16

Evgenia Rusak, Patrik Reizinger, Roland S Zimmermann, Oliver Bringmann, and Wieland Brendel. Content suppresses style: dimensionality collapse in contrastive learning. In *NeurIPS 2022 Workshop: Self-Supervised Learning - Theory and Practice*, 2022. 3

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In Arthur Gretton and Christian C. Robert (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 1232–1240, Cadiz, Spain, 09–11 May 2016. PMLR. URL https://proceedings.mlr.press/v51/russo16.html. 2

Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding Contrastive Learning Requires Incorporating Inductive Biases, February 2022. URL http://arxiv.org/abs/2202.14037. arXiv:2202.14037 [cs]. 3

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:1312.6120 [cond-mat, q-bio, stat]*, February 2014. URL http://arxiv.org/abs/1312.6120. arXiv: 1312.6120. 4

Jürgen Schmidhuber. Discovering neural nets with low kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857–873, 1997. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(96)00127-X. URL https://www.sciencedirect.com/science/article/pii/S089360809600127X. 16

R.J. Solomonoff. A formal theory of inductive inference. part i. *Information and Control*, 7(1):1–22, 1964. ISSN 0019-9958. doi: https://doi.org/10.1016/S0019-9958(64)90223-2. URL https://www.sciencedirect.com/science/article/pii/S0019995864902232. 16

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale Efficiently: Insights from Pre-training and Fine-tuning Transformers, January 2022. URL http://arxiv.org/abs/2109.10686. arXiv:2109.10686 [cs]. 3, 4

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

Guillermo Valle-Perez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rye4g3AqFm. 16

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science. Springer, 2000. ISBN 978-1-4419-3160-3. 2

VN Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264, 1971. 2

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html. 3, 14

Diego Vidaurre, Concha Bielza, and Pedro Larrañaga. A survey of l1 regression. *International Statistical Review*, 81(3):361–387, 2013. doi: https://doi.org/10.1111/insr.12023. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/insr.12023. 16

Paul Vitányi and Ming Li. On prediction by data compression. In Maarten van Someren and Gerhard Widmer (eds.), *Machine Learning: ECML-97*, pp. 14–30, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg. 16

Hao Wang, Rui Gao, and Flavio P Calmon. Generalization bounds for noisy iterative algorithms using properties of additive noise channels. *Journal of Machine Learning Research*, 24(26):1–43, 2023a. URL https://jmlr.org/papers/v24/21-1396.html. 2

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning, 2023b. 3, 10

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. 1

Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking Like Transformers, July 2021. URL http://arxiv.org/abs/2106.06981. arXiv:2106.06981 [cs]. 16

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing. pp. 38–45. Association for Computational Linguistics, October 2020. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6. 14

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-context Learning as Implicit Bayesian Inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI. 1, 3, 4, 10, 12, 13

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On Layer Normalization in the Transformer Architecture, June 2020. URL http://arxiv.org/abs/2002.04745. arXiv:2002.04745 [cs, stat]. 14

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems*, volume 30, pp. 2524–2533, 2017. 2

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. November 2016. URL https://openreview.net/forum?id=Sy8gdB9xx. 1, 2

Ruiqi Zhang, Spencer Frei, and Peter L. Bartlett. Trained Transformers Learn Linear Models In-Context, October 2023. URL http://arxiv.org/abs/2306.09927. arXiv:2306.09927 [cs, stat]. 1

A Details on $\varepsilon-$identifiability and the saturation regime

Mathematical formalizations of in-context learning often assume perfect (statistical) generalization (Xie et al., 2022; Wang et al., 2023b), that is, KL $[p(x_{1:k})||q(x_{1:k})] = 0$. In this context, the saturation regime is understood as the regime of perfect generalization. However, in experimental demonstrations, the term "saturation regime" is used more leniently to mean *near* perfect test loss. We argue that a refinement of the concept is required in order to align theory with practice. As we demonstrate in [ref], by relaxing perfect generalization only to KL $[p(x_{1:k})||q(x_{1:k})] \leq \varepsilon$, we may observe qualitatively different behaviours in models, for example, the existence and non-existence of in-context learning ability. Hence for the theory, it does matter whether we are truly or only approximately in the saturation regime. Yet in practice, in-context learning properties hold even for smaller transformers, where the test loss is only near-optimal at best (Figure 6 in Liu et al. (2023)). We argue that this discrepancy between theory and practice may be explained by inductive biases: in the near-optimal loss regime, where multiple models of varying quality exist, inductive biases select a solution that satisfies additional important properties, such as in-context learning.

B Proof of Proposition 1

B.1 Details on notation

We follow the notation and assumptions of Xie et al. (2022). Let $p(\theta)$ be a prior distribution on the latent concepts $\theta \in \Theta$, and let each θ define a distribution $p(o_1, \ldots, o_T|\theta)$ over sequences of tokens $o_1, \ldots, o_T \in \mathcal{O}$ of fixed pre-training document length T. Furthermore, assume that $p(o_1, \ldots, o_T|\theta)$ is defined by a HMM with a hidden state set $\mathcal{H}$, therefore the pre-training distribution is a mixture of HMMs:

$$p(o_1, \ldots, o_T) = \int_{\theta \in \Theta} p(o_1, \ldots, o_T|\theta)p(\theta)\, d\theta. \tag{1}$$

The prompt for ICL is a concatenation of n independent training examples S_n, and a test input x_{test} generated by the prompt distribution p_{prompt} and all are conditioned on the concept $\theta^\star$.

$$(S_n, x_{\text{test}}) = (x_1, y_1, o^{\text{delim}}, x_2, y_2, o^{\text{delim}}, \ldots, x_n, y_n, o^{\text{delim}}, x_{\text{test}}) \sim p_{\text{prompt}}, \tag{2}$$

where the i-th training example is (x_i, y_i), x_i has length $(k-1)$ for some fixed k, and o^{delim} is a special delimiter token. Moreover, $\mathcal{P}_N = \{(S_n, x_{\text{test}}, y)\}$ denotes the set of prompts with the output target y, where $N = (k+1)(n+1)-1$ is the length of the prompts, which are in a form of $(S_n, x_{\text{test}}, y)$. In ICL, the goal is to predict the test output y_{test} by predicting the next token. y_{test} is sampled from $p_{\text{prompt}}(y|x_{\text{test}})$. We say that the model is an in-context learner if, as $n \to \infty$,

$$\operatorname*{argmax}_y p(y|S_n, x_{\text{test}}) \to \operatorname*{argmax}_y p_{\text{prompt}}(y|x_{\text{test}}). \tag{3}$$

Xie et al. (2022) proved that if certain assumptions hold (Appx. B.2) and the pre-training distribution is a mixture of HMMss, then ICL occurs. That is, since y is discrete, it follows from (3) that there exists $n_0 \in \mathbb{Z}^+$ such that for $n \geq n_0$, the two sides of (3) become equal:

$$\forall n > n_0 : \operatorname*{argmax}_y p(y|S_n, x_{\text{test}}) = \operatorname*{argmax}_y p_{\text{prompt}}(y|x_{\text{test}}).$$

In other words, for n large enough, ICL emerges in the model with prompt (S_n, x_{test}). However, as we show, matching the pre-training distribution up to $\varepsilon > 0$ KL cannot guarantee ICL, even with increasing prompt size.

B.2 Proof

Proposition 2 ($\varepsilon-$non-identifiability of ICL, formal version)**.** *For all $\varepsilon > 0$, there exists $n_1 \geq n_0$, such that for all $n \geq n_1$, there exists a distribution q_n close to a mixture of HMMs in KL divergence*

$$KL\left[p(o_1, \ldots, o_N)||q_n(o_1, \ldots, o_N)\right] \leq \varepsilon, \ \textit{s.t.} \operatorname*{argmax}_y q_n(y|S_n, x_{\textit{test}}) \neq \operatorname*{argmax}_y p_{\textit{prompt}}(y|x_{\textit{test}}).$$

First we recall from § 3.2 that n_0 is defined as the threshold example sequence length after which p satisfies in-context learning, i.e.,

$$\forall n > n_0 : \operatorname*{argmax}_y p(y|S_n, x_{\text{test}}) = \operatorname*{argmax}_y p_{\text{prompt}}(y|x_{\text{test}}).$$

Next, we recall two assumptions from Xie et al. (2022) that we make use of throughout the proof.

Assumption 1 (Delimiter hidden states)**.** *Let the delimiter hidden states $\mathcal{D}$ be a subset of $\mathcal{H}$. For any $h^{delim} \in \mathcal{D}$ and $\theta \in \Theta, p\left(o^{delim} \mid h^{delim}, \theta\right) = 1$ and for any $h \notin \mathcal{D}, p\left(o^{delim} \mid h, \theta\right) = 0$.*

Assumption 2 (Bound on delimiter transitions)**.** *For any delimiter state $h^{delim} \in \mathcal{D}$ and any hidden state $h \in \mathcal{H}$, the probability of transitioning to a delimiter hidden state under θ is upper bounded $p\left(h^{delim} \mid h, \theta\right) < c_2$ for any $\theta \in \Theta \backslash \{\theta^*\}$, and is lower bounded $p\left(h^{delim} \mid h, \theta^*\right) > c_1 > 0$ for θ^*. Additionally, the start hidden state distribution for delimiter hidden states is bounded as $p\left(h^{delim} \mid \theta\right) \in [c_3, c_4]$.*

The above assumptions allow us to simplify our analysis and avoid degenerate cases such as a deterministic (hidden) Markov chain.

Proof. Our proof follows the below steps.

- **Step 1**: for every $n \geq n_0$, we define a q_n by equating it with p everywhere except on sequences that end with a prompt structure. We construct q_n such that the prompt completion will be different than in p, i.e.

$$\operatorname*{argmax}_y q_n(y|S_n, x_{\text{test}}) \neq \operatorname*{argmax}_y p(y|S_n, x_{\text{test}}).$$

  We do this by making sure that

$$\operatorname*{argmax}_{y \neq y^*} q_n(y|S_n, x_{\text{test}}) \geq q_n(y^*|S_n, x_{\text{test}}) + \frac{\delta}{2}.$$

- **Step 2**: we bound $\mathrm{KL}\left(p||q_n\right)$ as

$$\mathrm{KL}\left(p||q_n\right) \leq [\text{constant}] \times [\text{the probability of prompts}].$$

- **Step 3**: we show that the latter converges to 0 as $n \to \infty$ and is controlled by a function of δ.

Step 1 Let us denote the length N prompt by $O = (o_1, ..., o_N)$. Consider the fixed distribution $p(o_1, \ldots, o_N)$ defined by a (mixture of) HMMs. For any fixed $n \in \mathbb{Z}^+$, we define a distribution $q_n(o_1, \ldots, o_N)$ as a modification of p.
We consider those sequences which end with a prompt structure, that is, in which the last k tokens, namely $O_{N-k+1:N}$ satisfy $O_{N-k+1:N} \in \mathcal{P}$, with $\mathcal{P} = \{(x, y)|x \text{ has length } k-1, y \text{ has length } 1\}$. We construct q_n such that it is different only on these sequences and equal to $p(o_1, \ldots, o_N)$ everywhere else.
We expand q_n via the chain rule

$$\begin{aligned} q_n(S_n, x_{\text{test}}, y) &= \sum_{j=1}^{n+1} q_n(y_j|S_{j-1}, x_j) q_n(x_j|S_{j-1}) q_n(d_{j-1}|S_{j-2}, x_{j-1}, y_{j-1}) &(4)\\ &= q_n(y|S_n, x_{\text{test}}) q_n(x_{\text{test}}|S_n) q_n(d_n|S_{n-1}, x_n, y_n) \\ &+ \sum_{j=1}^{n} q_n(y_j|S_{j-1}, x_j) q_n(x_j|S_{j-1}) q_n(d_{j-1}|S_{j-2}, x_{j-1}, y_{j-1}), &(5) \end{aligned}$$

with notation $x_{n+1} = x_{\text{test}}$, $y_{n+1} = y$ and $x_0 = y_0 = d_0 = S_0 = S_{-1} = \emptyset$.
For $j = 1, \ldots, n$ let

$$q_n(x_j|S_{j-1}) := p(x_j|S_{j-1})$$

and

$$q_n(d_{j-1}|S_{j-2}, x_{j-1}, y_{j-1}) := p(d_{j-1}|S_{j-2}, x_{j-1}, y_{j-1}),$$

we are only modifying $p(y|S_n, x_{\text{test}})$, but only at its largest and second largest values. Let $a_1 = \max_y p(y|S_n, x_{\text{test}})$, $y_1^* = \operatorname{argmax}_y p(y|S_n, x_{\text{test}})$ and $a_2 = \max_{y \neq y_1^*} p(y = y|S_n, x_{\text{test}})$, $y_2^* = \operatorname{argmax}_{y \neq y_1^*} p(y = y|S_n, x_{\text{test}})$, then for all $y \neq y_1^*$ and $y \neq y_2^*$ let q_n be the same as p,

$$q_n(y|S_n, x_{\text{test}}) := p(y|S_n, x_{\text{test}}) \quad \forall y_1^* \neq y \neq y_2^*,$$

but on the largest and second largest values we change p as the following

$$q_n(y_1^*|S_n, x_{\text{test}}) := \frac{a_1 + a_2}{2} - \frac{\delta}{2} \quad \text{and}$$

$$q_n(y_2^*|S_n, x_{\text{test}}) := \frac{a_1 + a_2}{2} + \frac{\delta}{2},$$

where δ is arbitrarily small. Since $q_n(y|S_n, x_{\text{test}})$ has its maximum at y_2^*,

$$\operatorname*{argmax}_y p(y|S_n, x_{\text{test}})) \neq \operatorname*{argmax}_y q_n(y|S_n, x_{\text{test}}).$$

Due to (4), q_n is well-defined. Since $n > n_0$, from Xie et al. (2022) we have

$$\operatorname*{argmax}_y p(y|S_n, x_{\text{test}}) = \operatorname*{argmax}_y p_{\text{prompt}}(y|x_{\text{test}}).$$

Hence

$$\operatorname*{argmax}_y q_n(y|S_n, x_{\text{test}}) \neq \operatorname*{argmax}_y p_{\text{prompt}}(y|x_{\text{test}})$$

Note that

$$\log\left(\frac{p(y|S_n, x_{\text{test}})}{q_n(y|S_n, x_{\text{test}})}\right) \leq \log\left(\frac{2}{1-\delta}\right), \tag{6}$$

since if $q_n(y|S_n, x_{\text{test}}) = p(y|S_n, x_{\text{test}})$, then the log equals to 0, otherwise

$$\log\left(\frac{p(y_1^*|S_n, x_{\text{test}})}{q_n(y_1^*|S_n, x_{\text{test}})}\right) = \log\left(\frac{2a_1}{a_1 + a_2 - \delta}\right) \leq \log\left(\frac{2}{1-\delta}\right) \quad \text{and}$$
$$\log\left(\frac{p(y_2^*|S_n, x_{\text{test}})}{q_n(y_2^*|S_n, x_{\text{test}})}\right) = \log\left(\frac{2a_2}{a_1 + a_2 + \delta}\right) \leq \log\left(\frac{2}{1-\delta}\right),$$

since $\frac{2a_2}{a_1+a_2+\delta} < \frac{2a_1}{a_1+a_2-\delta} \leq \frac{2}{1-\delta}$ with equality if $a_2 = 0$.

Step 2 Now we bound the KL divergence between $p(o_1, \ldots, o_N)$ and $q_n(o_1, \ldots, o_N)$.

$$\text{KL}(p(o_1, \ldots, o_N)||q_n(o_1, \ldots, o_N)) = \sum_{t \in \mathcal{O}^N} p(t) \log\left(\frac{p(t)}{q_n(t)})\right) = \sum_{\{t|t_{N-k+1:N} \in \mathcal{P}\}} p(t) \log\left(\frac{p(t)}{q_n(t)}\right) +$$
$$+ \sum_{\{t|t_{N-k+1:N} \notin \mathcal{P}\}} p(t) \log\left(\frac{p(t)}{q_n(t)}\right) = \sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y) \log\left(\frac{p(S_n, x_{\text{test}}, y)}{q_n(S_n, x_{\text{test}}, y)}\right) =$$

expanding $p(S_n, x_{\text{test}}, y)$ and $q_n(S_n, x_{\text{test}}, y)$ via the chain rule, we get

$$= \sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y) \log\left(\prod_{j=1}^{n+1} \frac{p(y_j|S_{j-1}, x_j)p(x_j|S_{j-1})p(d_{j-1}|S_{j-2}, x_{j-1}, y_{j-1})}{q_n(y_j|S_{j-1}, x_j)q_n(x_j|S_{j-1})q_n(d_{j-1}|S_{j-2}, x_{j-1}, y_{j-1})}\right) =$$
$$= \sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y) \log\left(\frac{p(y|S_n, x_{\text{test}})}{q_n(y|S_n, x_{\text{test}})}\right) \leq$$

since by the definition of q_n, all the terms inside the $\log$ vanish excluding $p(y|S_n, x_{\text{test}})$ and $q_n(y|S_n, x_{\text{test}})$. Now we use the bound in Eq (6).

$$\text{KL}(p(o_1, \ldots, o_N)||q_n(o_1, \ldots, o_N)) \leq \log\left(\frac{2}{1-\delta}\right) \sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y).$$

Step 3 We now show that $\sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y) \to 0$ as $n \to \infty$ exponentially fast.

$$\sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y) = \sum_{(S_n, x_{\text{test}}, y)} \int_{\theta \in \Theta} p(S_n, x_{\text{test}}, y|\theta)p(\theta) \tag{7}$$
$$= \int_{\theta \in \Theta} \sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y|\theta)p(\theta). \tag{8}$$

Let us fix $\theta \in \Theta$. Consider the HMM with output distribution p, conditioned on θ. We wish to focus on bounding the probability of delimiter output states occurring as every k^{th} token.

From Assumption 1 of Xie et al. (2022), we know that a delimiter hidden state generates a delimiter output state with probability 1, and non-delimiter hidden states don't generate delimiter output states. Hence our question is equivalent to bounding the probability of the hidden Markov chain being at delimiter hidden states exactly at every k^{th} step.

Without loss of generality, assume that our HMM hidden state Markov chain is at a state in $\mathcal{D}$, otherwise, we reach $\mathcal{D}$ in some steps. For two, possibly equal $h^{\text{delim},i}$, $h^{\text{delim},j} \in \mathcal{D}$, define

$$p_{ij}^{k,\theta} = \sum_{h_1,\ldots,h_{k-1} \in \mathcal{H} \setminus \mathcal{D}} p(h^{\text{delim},i}, h_{k-1}, h_{k-2}, ..., h_1, h^{\text{delim},j} | \theta). \tag{9}$$

Let $p^* = \sup_{\theta \in \Theta} \max_{i,j} p_{ij}^{k,\theta}$, where the maximum is under all delimiter hidden states in $\mathcal{D}$. We show that $p^* < 1$. Fixing $\theta \in \Theta \setminus \theta^*$, by Assumption 2 of Xie et al. (2022), for all $h^{\text{delim}} \in \mathcal{D}$ and $h \in \mathcal{H}$, $p(h^{\text{delim}} | h, \theta) < c_2 < 1$. Hence for all i, j,

$$\begin{aligned}
p_{ij}^{k,\theta} &= \sum_{h_1,\ldots,h_{k-1} \in \mathcal{H} \setminus \mathcal{D}} p(h^{\text{delim},i}, h_{k-1}, h_{k-2}, ..., h_1, h^{\text{delim},j} | \theta) && (10) \\
&= \sum_{h_1,\ldots,h_{k-1} \in \mathcal{H} \setminus \mathcal{D}} p(h^{\text{delim},i} | h_{k-1}, \theta) p(h_{k-1} | h_{k-2}, ..., h_1, h^{\text{delim},j}, \theta) p(h_{k-2}, ..., h_1, h^{\text{delim},j} | \theta) && (11) \\
&\leq c_2 \sum_{h_1,\ldots,h_{k-1} \in \mathcal{H} \setminus \mathcal{D}} p(h_{k-1} | h_{k-2}, ..., h_1, h^{\text{delim},j}, \theta) p(h_{k-2}, ..., h_1, h^{\text{delim},j} | \theta) && (12) \\
&\leq c_2. && (13)
\end{aligned}$$

Thus, $p^* \leq c_2$, and the probability of generating a prompt in $\mathcal{P}^n$, for all θ, is upper bounded by c_2^n. Hence

$$\int_{\theta \in \Theta} \sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y | \theta) p(\theta) \leq \int_{\theta \in \Theta} (\max_{i,j} p_{ij}^{k,\theta})^n p(\theta) \leq (\sup_{\theta \in \Theta} \max_{i,j} p_{ij}^{k,\theta})^n = (p^*)^n \leq c_2^n. \tag{14}$$

This decays exponentially. Hence

$$\text{KL}(p(o_1, \ldots, o_N) || q_n(o_1, \ldots, o_N)) \leq \log\left(\frac{2}{1-\delta}\right) c_2^n$$

From this, we obtain that defining $n_1 = \log_{c_2}\left(\frac{\epsilon}{\log 2}\right) + 1$ and $\delta = \min\left\{a_1 + a_2, 1 - 2e^{-\frac{\epsilon}{c_2^n}}\right\}$ ensures $\text{KL}(p(o_1, \ldots, o_N) || q_n(o_1, \ldots, o_N)) \leq \epsilon$ for all $n \geq n_1$. □

C IDENTIFIABILITY

Definition 2 (Set of probability measures)**.** *We denote the set of probability measures on domain $\mathcal{X}$ as $\mathcal{M}(\mathcal{X})$.*

Definition 3 (Property)**.** *Let $\mathcal{M}(\mathcal{X})$ be the set of distibutions on $\mathcal{X}$ and let $p \in \mathcal{M}(\mathcal{X})$. A property $\mathcal{A}$ is a binary function $\mathcal{M}(\mathcal{X}) \to \{0; 1\}$. We say that p has property $\mathcal{A}$, if $\mathcal{A}(p) = 1$ and that it does not if $\mathcal{A}(p) = 0$*

Definition 4 (Property equivalence classes)**.** *A property $\mathcal{A}$ partitions a set of distributions $\mathcal{M}(\mathcal{X})$ into two equivalence classes, $\mathcal{M}_{\mathcal{A}}$ and $\mathcal{M}_{\overline{\mathcal{A}}}$ such that*

$$\forall i \neq j, p_i, p_j \in \mathcal{M}_{\mathcal{A}} : \mathcal{A}(p_i) = \mathcal{A}(p_j) = 1 \tag{15}$$

$$\forall i \neq j, p_i, p_j \in \mathcal{M}_{\overline{\mathcal{A}}} : \mathcal{A}(p_i) = \mathcal{A}(p_j) = 0 \tag{16}$$

such that $\mathcal{M}(\mathcal{X}) = \mathcal{M}_{\mathcal{A}} \cup \mathcal{M}_{\overline{\mathcal{A}}}$ and $\mathcal{M}_{\mathcal{A}} \cap \mathcal{M}_{\overline{\mathcal{A}}} = \emptyset$.

Definition 5 ($\varepsilon-$non-identifiability of distributional properties)**.** *Let $\mathcal{M}(\mathcal{X})$ be a set of distributions with an equivalence class structure, given by property $\mathcal{A}$ and denoted as $\mathcal{M}_{\mathcal{A}}, \mathcal{M}_{\overline{\mathcal{A}}}$. We say that property $\mathcal{A}$ of a distribution is $\varepsilon-$non-identifiable if there exists a distribution $p \in \mathcal{M}_{\mathcal{A}}$ such that $\exists q \in \mathcal{M}_{\overline{\mathcal{A}}}$ such that KL $[p||q] \leq \varepsilon$.*

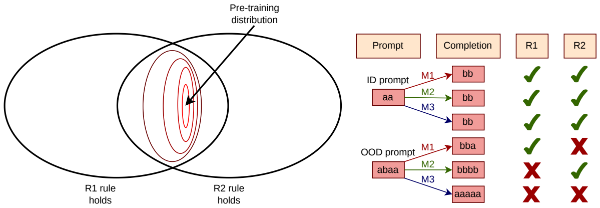

Figure 4: **Illustration of case study 3.1:** We train a Transformer on a PCFG generating sequences of the form $a^n b^n$. **Left:** This language can be represented as an intersection of two rules: (R1) the number of as and bs match; and (R2) a never follows a b. **Right:** We consider different models (M1, M2, M3) which achieve perfect test loss. On prompts consistent with the $a^n b^n$ grammar (e.g., aa) all three models produce the same completions. However, on prompts that are inconsistent with $a^n b^n$, and thus have probability zero under the pre-training distribution, the models may produce different completions. For these OOD prompts, we can ask if completions still satisfy rule (R1), which we call rule extrapolation. Rule extrapolation behaviour is not implied by minimal test loss, but may arise due to inductive biases.

D Experimental details

Reproducibility and codebase. We use PyTorch (Paszke et al., 2019), PyTorch Lightning (Falcon & The PyTorch Lightning team, 2019), and HuggingFace Transformers (Wolf et al., 2020). We make our code and experimental logs publicly available upon acceptance.

PCFG. We generate data from the $a^n b^n$ PCFGs up to length 256. Besides the tokens a (0) and b (1), we use SOS (2), EOS (3), and padding (4) tokens. We define our test prompts as all possible sequences of length 8 (prepended with SOS), which we split into in-distribution, and OOD test prompts, based on whether they can be completed in the form of $a^n b^n$. The training set includes all unique sequences up to length 256. We illustrate rule extraplation in Fig. 4.

Table 1: PCFG parameters

Parameter	Values
Number of tokens	5 (SOS, EOS, PAD, 0,1)
Maximum sequence length	256
Training data maximum length	256
Test prompt length	8
Batch size	128

Model. We use a Transformer decoder (Vaswani et al., 2017) in flavor of the decoder-only GPT models (Radford et al.; 2018; OpenAI et al., 2023). We apply standard positional encoding, layer normalization, ReLU activations, the AdamW optimizer (Loshchilov & Hutter, 2019) with inverse square root learning rate schedule (Xiong et al., 2020). For prompt prediction, the model can predict up to length 300. We train for $50,000$ epochs with the standard cross entropy (CE) loss for the next token prediction task. For the adversarial and oracle training versions, we add an additional loss term which we detail below.

Table 2: Transformer parameters

PARAMETER	VALUE (NORMAL)
MODEL	TRANSFORMER DECODER
NUMBER OF LAYERS	5
DROPOUT PROBABILITY	0.1
MODEL DIMENSION	10
FEEDFORWARD DIMENSION	1024
NUMBER OF ATTENTION HEADS	5
LAYER NORM ϵ	$6\mathrm{e}{-3}$
ACTIVATION	RELU
OPTIMIZER	ADAMW
LEARNING RATE SCHEDULER	INVERSE SQUARE ROOT
BATCH SIZE	128
LEARNING RATE	$2\mathrm{e}{-3}$
PROMPT PREDICTION CUTOFF LENGTH	300
NUMBER OF EPOCHS	$50,000$

Metrics. We monitor training and validation loss, and the adherence to the grammar's two rules (R1),(R2). We measure the accuracy of each separately and simultaneously (i.e., to check whether the generated sequence is grammatical). For a deeper understanding, we calculate these metrics for different scenarios:

1. For the in- and out-of-distribution test prompts and
2. For a batch of SOS tokens.

For each of the above, we re-calculate the accuracies for the subset of prompt completions which have an EOS token to avoid false conclusions (e.g., if the model wants to finish aaa as a longer sequence than the cutoff length, the unfinished sequence would lower the accuracy). Since for the OOD prompts, it is by definition impossible to fulfil (R2) (that $a's$ are before $b's$), we separately calculate this rule on the completion: e.g., if the OOD **abbb** is completed as **abbb**aa, then it is considered correct for this metric, but **abbb**$abaa$ is not, as it has an a after a b in the *completion*. We also monitor the accuracy of next token prediction via greedy decoding (i.e., using the token with the largest probability). We report additional numerical values in Tab. 3, supplementing Fig. 1.

Adversarial training. For adversarial training, we generate OOD sequences such that the number of $a's$ and $b's$ is not equal, there is one more from one symbol. Then, we treat the first 8 a and b tokens (i.e., the same as the test prompt length) as the *prompt*, and the rest as the *completion*. During training, we add a CE loss on the OOD prompt completions. The rationale of only optimizing on the OOD completions is to keep the prompts OOD, since our claim in § 3.1 is about different behavior for OOD prompts.

Oracle training: enforcing rule extrapolation. This scenario is very similar to adversarial training, with the difference, that we generate additional OOD training samples, where the *prompt* is still OOD, but here the *completion* is generated such that the number of $a's$ and $b's$ is equal over the whole sequence. Then we add a CE loss on the OOD prompt completions.

Table 3: Comparison of the extrapolation performance of MLE, adversarial, and oracle training for OOD prompts. For (approximately) the same validation loss, the extrapolation of (R1) for OOD prompts differs enormously, showing that the loss alone cannot distinguish the extrapolation property

NAME	VALIDATION LOSS	ACCURACY OF (R1)	
		MEAN+STD.	RANGE
MLE	$0.0215_{\pm 0.0011}$	$0.437_{\pm 0.047}$	$[0.339; 0.629]$
ADVERSARIAL	$0.0223_{\pm 0.00094}$	$0.$	$[0.; 0.]$
ORACLE	$0.0199_{\pm 0.00025}$	$0.83_{\pm 0.122}$	$[0.634; 1.]$

E Inductive biases for understanding LLMs

In contrast to relying solely on inductive biases enabling statistical generalization, we advocate for studying inductive biases that are not problem- or loss-specific. These qualitative characteristics remain insightful even in the saturation regime, as they enable us to reason about performance on new tasks. We encourage investigations that intertwine statistical generalization with these LLM-relevant inductive biases, e.g., by characterizing extrapolation performance in terms of statistical generalization ability *and* the presence of an inductive bias. To motivate the need for LLM-relevant inductive biases, we showcase (sometimes toy) examples of qualitative properties relevant to specific DNN models and tasks. We then outline some promising directions for LLMs.

Examples of qualitative model properties. *Sparsity* is a prevalent concept in machine learning often enforced through explicit regularisation (see Vidaurre et al., 2013, for a review). Intriguingly, inductive biases alone can give rise to sparsity in gradient descent: $L-$layer linear diagonal networks trained on binary classification converge to the $\ell_{\frac{2}{L}}$ large margin classifier, yielding a sparse solution (Gunasekar et al., 2019), whereas deep matrix factorization is known to lead to low-rank solutions (Gunasekar et al., 2017; Arora et al., 2019). For models where the Neural Tangent Kernel (NTK) assumptions hold, gradient descent solves kernel ridge (ℓ_1-regularized) regression (Jacot et al., 2020). For DNNs implementing Boolean functions, the resulting parameter to function map is *simple*[1] in terms of Lempel-Ziv complexity (Valle-Perez et al., 2019; Dingle et al., 2018) and converges towards low-entropy functions (Mingard et al., 2020). Binary classifiers of bitstrings are biased towards low sensitivity to changes in the input De Palma et al. (2019). Rahaman et al. (2019) highlights a bias towards low-frequency functions. There is also work that looks at the dynamics of qualitative properties during training: neural networks appear to learn increasingly complex functions, starting with linear functions (Arpit et al., 2017; Nakkiran et al., 2019) making use of higher-order statistics only in later stages (Refinetti et al., 2023). Although many of these findings rely on simplified mathematical models they nevertheless provide good insights into qualitative properties one should expect trained neural networks to possess, which in turn can be connected to properties of interest such as OOD extrapolation.

Insights from algorithmic information theory. The Kolmogorov complexity $K(x)$ of a bitstring x is defined as the length of the shortest program under a fixed programming language that produces x (Kolmogorov, 1998). For LLMs, an intriguing direction is connecting model properties to the Kolmogorov complexity of its generated text: a bias towards low Kolmogorov complexity might imply improved (compositional) generalization. Though Kolmogorov complexity is incomputable, insights from algorithmic information theory remain pertinent for understanding LLMs and building general-purpose models (Schmidhuber, 1997; Hutter, 2000). Goldblum et al. (2023) argues that real-world data has low complexity in the Kolmogorov sense (Goldblum et al., 2023). This simplicity bias in data is shared with (even randomly initialized) neural networks and is more general than what the architecture would suggest: CNNs can effectively learn tabular data despite their lack of spatial structure. Via the connection between prediction and compression (Vitányi & Li, 1997), we may interpret a DNN as a compressor of the training data, where the best possible compressor has the lowest Kolmogorov complexity. Delétang et al. (2023) shows that successful LLMs are good general-purpose compressors, e.g., Chinchilla 70B compresses ImageNet patches to 43.4% despite having been trained primarily on text. In addition, Grau-Moya et al. (2024) develop a meta-learning method to train LLMs to approximate Solomonoff Induction (Solomonoff, 1964), which also provides interesting connections to algoritmic information theory. To study the effects of data complexity on LLMs, we advocate for the use of simple formal languages, such as PCFGs (Liu et al., 2023; Favre, 2020; Merrill, 2023; Ackerman & Cybenko, 2020), as they have a controllable notion of complexity and structure.

Insights from the Transformer architecture The underlying Transformer architecture may shape the inductive biases in LLMs. Recently, Weiss et al. (2021) showed that a new programming language, RASP, describes a computational model for Transformers, which can explain how reordering fully connected and attention layers changes performance (Press et al., 2020). In terms of RASP, these reorderings constrain information flow, acting as an architectural inductive bias. Thus,RASP (Weiss

[1] The study of this object is motivated by the observation that SGD approximates Bayesian inference sufficiently well, where the prior $p(f)$ is taken as the probability of a randomly initialized neural network implementing a specific function

et al., 2021) as a computational model for Transformers offers a framework for characterizing the algorithms LLMs can implement. Specifically, the efficiency and compactness with which these algorithms can be expressed in RASP might serve as a novel, LLM-specific complexity metric. A related direction for finding inductive biases based on the Transformer architecture is mechanistic interpretability (Olah, 2022), which aims to understand the internal mechanisms of a model.

F Acronyms

CE cross entropy

AR autoregressive

DNN Deep Neural Network

EOS end-of-sequence

HMM Hidden Markov Model

i.i.d. independent and identically distributed
ICL in-context learning

KL Kullback-Leibler Divergence

LLM Large Language Model
LVM latent variable model

MLE maximum likelihood estimation

NTK Neural Tangent Kernel

OOD out-of-distribution

PCFG Probabilistic Context-Free Grammar

RASP Restricted-Access Sequence Processing Language

SOS start-of-sequence