Position: Understanding LLMs Requires More Than Statistical Generalization

Patrik Reizinger [*1] **Szilvia Ujváry** [*2,3] **Anna Mészáros** [*2] **Anna Kerekes** [*4,5]
Wieland Brendel [1,6,7] **Ferenc Huszár** [†2]

Abstract

The last decade has seen blossoming research in deep learning theory attempting to answer, "Why does deep learning generalize?" A powerful shift in perspective precipitated this progress: the study of overparametrized models in the interpolation regime. In this paper, we argue that another perspective shift is due, since some of the desirable qualities of LLMs are not a consequence of good statistical generalization and require a separate theoretical explanation. Our core argument relies on the observation that AR probabilistic models are inherently non-identifiable: models zero or near-zero KL divergence apart—thus, equivalent test loss—can exhibit markedly different behaviors. We support our position with mathematical examples and empirical observations, illustrating why non-identifiability has practical relevance through three case studies: (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 review promising research directions focusing on LLM-relevant generalization measures, transferability, and 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., 2023a).

	Interpolation regime	Saturation regime
$\min \mathcal{L}_{\text{train}}$	non-unique global min. reached	non-unique global min. reached
$\min \mathcal{L}_{\text{test}}$	no assumption	non-unique global min. reached
Questions	Is $\mathcal{L}_{\text{test}}$ small enough?	zero-shot extrapolation in-context learning transfer, finetunability

Table 1. **Comparison of Interpolation and Saturation regimes:** In the Interpolation regime we assume a global minimum of the training loss is found, but is not unique, so we ask whether the minimum we find generalises well. In the Saturation regime, we further assume that a global minimum of the test loss is found, but even that is not unique. We study additional properties of the minimum found, which are not implied by good generalization.

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. Here, the question is whether this model will also perform well on the *in-distribution* test set. 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., 2023a).

This position paper argues that we ought to study LLMs in the *saturation regime* [1] (coined by Liu et al., 2023a) instead (Tab. 1). In the saturation regime, 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., 2023a), 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

[*]Equal contribution [†] Senior author [1]Max Planck Institute for Intelligent Systems, Tübingen, Germany [2]University of Cambridge, UK [3]AI Center, UCL, London, UK [4]Department of Computer Science, ETH Zurich [5]Max Planck ETH Center for Learning Systems [6]ELLIS Institute Tübingen, Germany [7]Tübingen AI Center, Germany. Correspondence to: Patrik Reizinger <patrik.reizinger@tuebingen.mpg.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]The term "saturation regime" has also been used to refer to saturation of hidden units (Glorot & Bengio, 2010). Here we mean exclusively the saturation of the test loss.

more application-specific goals, e.g., rule extrapolation or the data efficiency of fine-tuning.

Why is the minimum of the test loss non-unique? We can use the lens of identifiability to understand this non-trivial question. Namely, unless their support spans the entire space of sequences, autoregressive (AR) probabilistic models are non-identifiable: different models are indistinguishable by 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. Our **contributions** are:

- highlighting the limitation of statistical generalization for understanding AR LLMs in the saturation regime (§ 2);
- demonstrating the relevance of our position through three case studies which provide well-defined starting points to theoretically study LLMs (§ 3);
- summarizing three topics that should be studied in LLMs in the saturation regime (§ 4), and proposing potential research directions (§ 5).

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). These, however, fail to account for the success of deep learning as argued by Zhang et al. (2016) and others. 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 focusing on statistical generalization stems from its "black-box" nature: it can be applied across many domains without changing terminology, be it machine translation or image processing. Stating that an algorithm generalises well requires no domain-specific understanding or qualitative description of any form of generalization behaviour, beyond the specification of a loss function. 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 one needs to describe how the test and training distributions differ.

Interpolation regime. Overparametrized models gave rise to the *interpolation regime* (Tab. 1), 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 the interpolation regime, the training loss *alone* cannot distinguish whether a model will generalize, yet models that we find by minimizing the training loss typically generalize well. In rare cases, optimization can switch between generalizing and non-generalizing solutions, as observed in grokking (Power et al., 2022). 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. In § 4, we argue that a second paradigm shift is needed: *we should move beyond the interpolation regime and study LLMs in the saturation regime, uncovering the inductive biases responsible for 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, identifiability asks whether the parameters of a model are uniquely determined by the data distribution they define (see e.g. Comon, 1994). In latent variable modeling, we may not care about parameters and ask instead whether the statistical relationship between latent and observed variables can be uniquely determined from the marginal distribution of observables, perhaps up to some invariances (see, e.g., Hyvärinen, 2013). In causal discovery, one asks whether the causal structure of a distribution (described, e.g., by a specific Markov factorisation) can be uniquely determined from the joint distribution (Hyvärinen et al., 2023).

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 as it allows us to reason about properties of this possibly unreachable but unique minimum, for example, predict OOD extrapolation or the effect of interventions (Pearl, 2009).

3. Identifiability in AR LLMs

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

(i) Functional non-identifiability happens when the collection of conditionals is not uniquely determined by the collection of joint distributions they define. As we will see, functional identifiability is only guaranteed when all finite prefixes occur with non-zero probability. If this is not the case, two models that differ only in how they com-

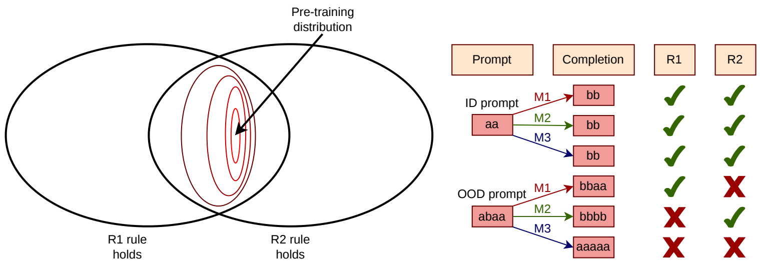

Figure 1. **Illustration of case study 3.1:** We train a Transformer on a PCFG generating sequences of the form a^nb^n. **Left:** This language can be represented as an intersection of two rules: (R1) the number of as and bs match; and (R2) a never follows a b. **Right:** We consider different models (M1, M2, M3) which achieve perfect test loss. On prompts consistent with the a^nb^n grammar (e.g., aa) all three models produce the same completions. However, on prompts that are inconsistent with a^nb^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.

plete zero-probability prefixes are indistinguishable by the Kullback-Leibler Divergence (KL). We illustrate implications of this on extrapolation behaviour in § 3.1;
(ii) $\varepsilon-$**non-identifiablity** is a new term we introduce, which relaxes the notion of functional non-identifiability. It posits that there exist properties of models p that other models q very close to p (at most ε in KL-sense) do not possess. This means that even though a unique global minimum exists, there are near-minima that differ from the global minimum in some important ways. We illustrate that the emergent ICL ability in some AR models is an example of this in § 3.2;
(iii) Parameter non-identifiability means that *functionally equivalent* models can be described by different sets of parameters. While this does not affect the model's zero-shot test performance, it can have significant implications for transfer learning and fine-tuning (§ 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 LLM behavior theoretically.

3.1. Case Study: Non-Identifiability of rule extrapolation

Consider training an AR language model q to fit samples from the probabilistic context-free grammar (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 $\mathrm{KL}\left[p(x_{1:k})||q(x_{1:k})\right] = 0$, 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 KL divergence is insensitive to the choice of $q(x_{l+1:k}|x_{1:l})$, the completion distribution is 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 can be described as 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. 1.

OOD rule extrapolation is non-identifiable, LLMs still extrapolate due to inductive biases

Trained Transformers have better-than-chance ability to extrapolate rules on OOD prompts in a zero-shot manner (chance is zero, calculated at initialization, cf. Figs. 1 and 2 and Tab. 4). Since the KL is insensitive to rule extrapolation, we attribute rule extrapolation to inductive biases.

Empirical demonstration. We train a decoder-only Transformer (Vaswani et al., 2017; Radford et al., 2018) on the a^nb^n PCFG and evaluate zero-shot rule extrapolation (Fig. 2), measured as the proportion of times OOD prompts of length 8 are completed consistently with rule R1.

As hypothesized, we find that the model develops an ability to extrapolate the rule in 43.7% of the cases, although the loss function is agnostic to this. To illustrate this last point, we train two additional models in an adversarial, and an oracle setting, whereby an extra supervised loss term is added

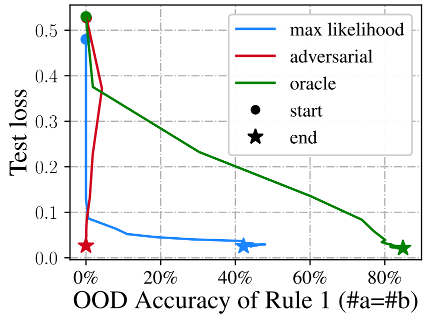

Figure 2. **OOD rule extrapolation in Transformers is better than chance:** We trained a Transformer via maximum likelihood on the $a^n b^n$ PCFG. We evaluated the model on OOD prompts which are inconsistent with $a^n b^n$, and checked whether the completions obey rule (R1) (x axis). Two other models, 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 that the 43.7% rule extrapolation accuracy (averaged over 20 seeds; details in Appx. E) results from inductive biases.

with either poisoned or helpful data. Consistently with our expectations, all three models reach the same, minimal test loss, but display widely varying rule extrapolation performance (Fig. 2; cf. Appx. E for training details).

Our findings demonstrate that AR probabilistic language models extrapolate OOD in meaningful ways. However, this behaviour is not merely a consequence of good generalisation, it arises as a result of additional inductive biases.

3.2. Case Study: $\varepsilon-$non-identifiability and in-context learning (ICL)

In § 3.1, we assumed that the pre-training and test distributions have limited support, i.e., 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. D):

Definition 3.1 ($\varepsilon-$non-identifiability of distributional properties (informal)). A distributional property of p is $\varepsilon-$non-identifiable if there is a distribution q such that $\mathrm{KL}\,[p||q] \leq \varepsilon$, 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., 2023a; Saunshi et al., 2022; Rusak et al., 2022; Tay et al., 2022). Indeed, this observation is widely accepted in evaluating generative models (Murphy, 2022, 20.4), where the trade-off between the likelihood, representation, and sample quality is well known (Huszár, 2017).

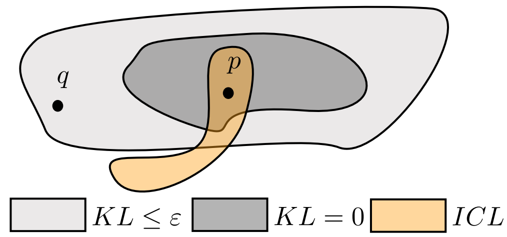

Figure 3. **Vanishingly small KL cannot capture in-context learning (ICL):** illustration of Prop. 3.1, showing that when p displays ICL property, there exists a distribution q that is $\varepsilon-$close in KL divergence, which has no ICL ability.

In-context learning (ICL) can be non-identifiable, LLMs still can do ICL

When the pre-training distribution is a mixture of Hidden Markov Models (HMMs) as in Xie et al. (2022), we demonstrate that ICL is $\varepsilon-$non-identifiable, even with infinite data, due to the insensitivity of the KL. However, LLMs can be still in-context learners; we need to understand why.

Example: $\varepsilon-$non-identifiability of ICL. ICL refers to the ability of a model to learn a downstream task based on a prompt consisting of input-output examples. That is, if the prompt includes sufficient input-output pairs (x_i, y_i) and a test input x_{test} then the model can predict the correct output y_{test} without fine-tuning, only relying on the pairs (x_i, y_i) in inferring the task.

Numerous theories exist in the literature, which prove the emergence of ICL under simplified settings. For a Transformer consisting of linear self-attention layers trained on regression data, ICL is shown to emerge as implicit gradient descent (Akyürek et al., 2023; von Oswald et al., 2023; Mahankali et al., 2024). This view interprets Transformers as meta-models that "learn to learn" in-context tasks within their forward pass. The framework requires specific model parameters. Other approaches prove ICL using assumptions on the structure of the training data (Xie et al., 2022; Wang et al., 2023b). We highlight an unmentioned type of fragility of these theories for real-world LLMs, which occurs in practice as soon as we slightly deviate from these assumptions. In the setting of (Xie et al., 2022), we prove that ICL can be $\varepsilon-$non-identifiable, i.e., for a distribution p, with ICL ability by construction, we construct another distribution q within ε KL divergence that provably does not exhibit ICL.

Our result demonstrates that the emergence of ICL is not a direct consequence of minimizing the negative log-likelihood. Indeed, an LLM that fits the pre-training distribution within ε KL divergence, can easily not exhibit ICL. We detail the implications for the saturation regime in Appx. A.

Xie et al. (2022) demonstrate 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. That is, it produces completions aligning with the predictions of the prompt distribution. In accordance with their notation, let $p_0(\theta)$ be a prior distribution on the latent concepts $\theta \in \Theta$, and for each θ let the distribution $p(o_1, \ldots, o_T|\theta)$ be a HMM for a token sequence $o_1, \ldots, o_T$ of a fixed pre-training document length T. First, a concept θ is sampled from $p_0(\theta)$, then a document is generated according to $p(o_1, \ldots, o_T|\theta)$. Therefore, the pre-training distribution (on token sequences) is a mixture of HMMs induced by these conditionals.

$$p(o_1, \ldots, o_T) = \int_{\theta \in \Theta} p(o_1, \ldots, o_T|\theta) p_0(\theta)\, d\theta. \quad (1)$$

The prompt for in-context learning is a concatenation of n independent input-output pairs (x_i, y_i) and a test example x_{test} each separated by a special delimiter token o^{delim}. Each example sampled from $p(\cdot|\theta^*)$, where θ^* is the ground-truth concept. The distribution of the concatenated examples are called the prompt distribution p_{prompt}

$$\begin{aligned}(S_n, x_{\text{test}}) = (&x_1, y_1, o^{\text{delim}}, x_2, y_2, o^{\text{delim}}, \\ &\ldots, x_n, y_n, o^{\text{delim}}, x_{\text{test}}) \sim p_{\text{prompt}}. \quad (2)\end{aligned}$$

In ICL, the goal is to predict the test output y_{test} by predicting the next token. y_{test} is defined by $\operatorname{argmax}_y p_{\text{prompt}}(y|x_{\text{test}})$. Details on the notation and the model are in Appx. B.
We say that the model is an in-context learner if, as $n \to \infty$,

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

Xie et al. (2022) proved that if certain assumptions hold (Appx. B) and the pre-training distribution is a mixture of HMMs, then ICL occurs when the number of examples n is large enough. Thus, the two sides of (3) become equal. However, as we show, matching the pre-training distribution up to $\varepsilon > 0$ KL cannot guarantee ICL, even with increasing prompt size:

Proposition 3.1. *[$\varepsilon-$non-identifiability of ICL] Let N be the length of a prompt (S_n, x_{test}). 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 p in KL divergence*

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

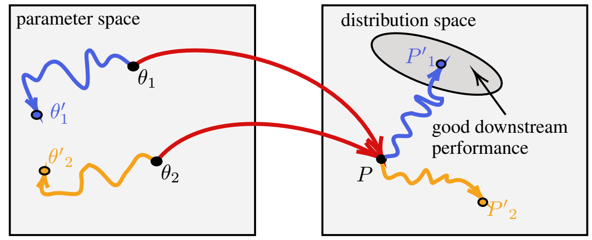

Figure 4. **Illustration of parameter non-identifiability:** Two sets of parameters (θ_1, θ_2) may describe the same AR LLM and thus achieve the same test loss and perform identically in benchmarks. When fine-tuned on the same data, parameter-dependent inductive biases may push the two models apart, and it is possible that, say, θ_1 enables significantly more data-efficient fine-tuning than θ_2.

Proof (Sketch). We construct a distribution q_n that matches p everywhere, except for a few sequences ending with a prompt structure. Changing the highest and second-highest probabilities changes the output of the argmax. 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 is in Appx. C. ☐

3.3. Case Study: 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. 4), i.e., functionally equivalent models can behave differently during fine-tuning and transfer[2]. Parameter non-identifiability is relevant in LLMs, since large pre-trained models are often fine-tuned on new data sets to solve specialized tasks. For example, given the same pre-training loss, a narrow and wide Transformer can perform better downstream than a wide and shallow one (Tay et al., 2022). Liu et al. (2023a) 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, the 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. A prominent method for LLM fine-tuning is Reinforcement Learning from Human Feedback (RLHF); recent results suggest that its success on OOD data might be due to an inductive

[2] *Fine-tuning* is training a pre-trained model with a new loss, or with a new data set; *transfer* is when the model can also change after pre-training, e.g., by applying a readout layer.

bias. The concurrent study by Kirk et al. (2024) empirically demonstrated better OOD performance than supervised fine-tuning—at the cost of reduced diveristy of the generated completions. The authors reason, based on arguments from (Goldberg, 2023; Xu et al., 2022) that the RLHF objective has an inductive bias: namely, it forces the LLM to optimize its policy for OOD data.

We need to understand how inductive biases in LLMs affect fine-tuning performance

Functionally equivalent models can have different parametrizations. These are indistinguishable by the test loss but may differ after fine-tuning in downstream performance, as inductive biases in different parametrizations affect gradient dynamics (Fig. 4).

4. The saturation regime

In § 3, we presented case studies to support our position that statistical generalization cannot explain the most interesting phenomena in LLMs. We believe that we need a paradigm shift akin to that brought about by the discovery of the interpolation regime. Our attention should be directed towards the regime where we can assume our model achieves both near-zero train and test loss. We should study inductive biases that give rise to beneficial qualities, not implied by statistical generalization.

The term saturation regime was coined by Liu et al. (2023a) in the context of transfer learning to argue that a minimal test pre-training loss does not guarantee the transfer performance observed in practice (§ 3.3). We adopt this name to study a broader set of phenomena, e.g. zero-shot OOD extrapolation (§ 3.1) and ICL (§ 3.2).

We focus on AR LLMs, which are in the intersection of probabilistic generative models and Self-Supervised Learning (SSL)/transfer learning methods. As our case studies show, all the questions raised by non-identifiability, are relevant and interesting for AR LLMs—e.g., ICL or zero-shot prompting cannot be interpreted for non-AR LLMs such as BERT (Devlin et al., 2019), though those can also operate in the saturation regime. Some of our case studies, e.g. transfer (§ 3.3) remain relevant in a much wider context, including vision models in SSL, where theoretical studies of relevant inductive biases already exist (HaoChen & Ma, 2022).

Position: we need to study LLMs in the saturation regime

In this regime, the relevant questions are
(i) assessing LLMs with better generalization measures;
(ii) understanding transferability; and
(iii) studying the inductive biases enabling them.

We highlight our arguments for these research directions, then review relevant research in § 5. We also suggest extensions for identifiability to capture the desired properties (generalization, transfer) and inductive biases.

(i) Generalization measures. In the saturation regime, the loss cannot capture the structures and properties of interest (such as the structure of natural languages); thus, we advocate for studying other (OOD) generalization measures: compositional, systematic, and symbolic generalization (§ 5.1). We argue that due to their well-defined structure, formal languages such as PCFGs are ideal for studying these generalization measures in LLMs.

(ii) Transferability. The size of state-of-the-art LLMs prohibits training them from scratch, making transferability crucial. Understanding and controlling transferability in LLMs is key to building strong general-purpose models and preventing them from being fine-tuned for harmful tasks (Qi et al., 2023). Since current metrics cannot reliably capture transfer performance (cf. § 3.3), we suggest developing suitable metrics for fine-tuning abilities in LLMs.

(iii) Inductive biases are widely researched in machine learning, e.g., flatness (Foret et al., 2021), gradient noise (Jiang et al., 2019), information geometry (Ju et al., 2022; Jang et al., 2022) and simplicity bias (Nakkiran et al., 2019). As opposed to computer vision (Klindt et al., 2021; Geirhos et al., 2018; 2020; 2022; Török et al., 2022; Offert & Bell, 2021; Goyal & Bengio, 2022; Papa et al., 2022), it is unclear what kind of inductive biases are useful for natural languages, especially for more complex tasks such as reasoning. Since the training and test losses do not indicate the desired properties, inductive biases offer alternative means to ensure good OOD performance (§ 3.1) and transfer (§ 3.3). For LLMs, we advocate focusing on inductive biases inducing specific model qualities either in weight or function space (§ 5.3), as these enable us to infer OOD and transfer performance, irrespective of the loss value.

Lessons for identifiability research. Identifiability theory mostly focuses on how functional, (Shimizu et al., 2006; Hoyer et al., 2008; Lachapelle et al., 2020; Gresele et al., 2021) and distributional (Hyvärinen & Pajunen, 1999; Hyvarinen & Morioka, 2016; Guo et al., 2022; Ke et al., 2020) assumptions can guarantee identifiability, almost exclusively assuming i.i.d. data, with a few pioneering works entering the non-i.i.d. realm by studying compositionality (Brady et al., 2023; Wiedemer et al., 2023b;a) or exchangeability (Guo et al., 2022). We must go beyond the i.i.d. assumption to understand LLMs and, we argue, there are useful lessons for identifiability theory to be drawn from studying the saturation regime, e.g., by considering that:

(i) studying the identifiability of task-specific properties, e.g., ICL (cf. § 3.2), as opposed to focusing on identifying all aspects of an underlying distribution;
(ii) inductive biases play a role in training non-identifiable

models, i.e., some non-identifiable properties might become identifiable once we consider which solutions are reachable by the optimization algorithm;

(iii) models in the equivalence class of global minima are reached with different probabilities by training; inductive biases are akin to a prior distribution over models;

(iv) non-identifiability may be a feature, not a bug when the MLE objective is misspecified and does not fully describe how the models are later used, but useful inductive biases are present that nudge the model towards a useful solution (Fig. 5).

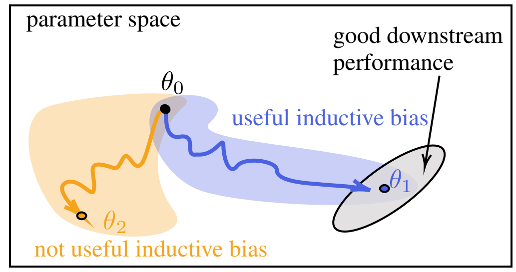

Figure 5. **Illustration of how inductive biases can affect identifiability:** In the saturation regime, training can result in different parameters θ_1, θ_2 with the same training and test loss, but different downstream performance. Even if the loss is insensitive to a model property that is required for good downstream performance, choosing a useful inductive bias can help capture said property, overcoming its non-identifiability.

5. Discussion: where next?

Our main message is highlighting the need to study LLMs in the saturation regime, where they achieve perfect (statistical) generalization, but this does not guarantee the presence of practical model properties we seek. These properties are OOD generalization and transferability, and we attribute them to qualitative, non problem-specific inductive biases (§ 4). Here we detail promising research directions and formulate concrete research questions for studying LLMs in the saturation regime, focusing on generalization measures (§ 5.1), computational language modeling (§ 5.2), and inductive biases (§ 5.3).

5.1. Better generalization measures

While statistical generalization offers a robust framework for generalization theory, bounding the test loss only does not explain important properties of LLMs, as illustrated in § 3. In this section, we give examples of alternative generalization metrics that better align with the properties of natural languages. We advocate for adapting the tools of statistical generalization to develop formalizations for these metrics in the saturation regime. Some works have already started extending the classical PAC-Bayes framework to generative models (Chérief-Abdellatif et al., 2022; Mbacke et al., 2023), meta-learning (Rezazadeh, 2022; Liu et al., 2021), and non-i.i.d. data (Alquier et al., 2012; Germain et al., 2016; Alquier & Guedj, 2017); however, further efforts are required. Results may also be established for quantities other than generalization error, perhaps bounding LLM-relevant properties, e.g., ICL.

Compositional generalization describes a model's capability to comprehend the properties of combined features by understanding multiple individual features: e.g., that by understanding words like "one," "cat," and "black", the model understands phrases like "one cat" or "black cat". Compositional generalization enables models to perform better on OOD data. Despite compositionality being a core property of natural languages, its theory is in its infancy for LLMs, in contrast to, e.g., computer vision and object-centric representation learning (Locatello et al., 2020; Brady et al., 2023; Wiedemer et al., 2023b;a; Löwe et al., 2023; Li et al., 2019).

Systematic generalization composes rules (Bahdanau et al., 2019; Gao et al., 2020; Ruis & Lake, 2022), and not syntactical compositions as compositional generalization does. Systematic generalization is relevant in reasoning by ensuring that LLMs can infer "Flo is the granddaughter of Betty", from the samples "Nat is the granddaughter of Betty", "Greg is the brother of Nat", "Flo is the sister of Greg" (Gontier et al., 2020). Despite its relevance, systematic generalization research for LLMs is still limited (Gontier et al., 2020).

Symbolic generalization is the model's ability to transfer the responses learned from the training data to another situation which is symbolically related (Hoon et al., 2020). Relevant examples come from mathematics, e.g., knowing that adding "$2+5x$" and "$3+4x$" yields "$(2+3)+(5+4)x$", a model that generalizes symbolically should correctly infer that adding "$A+Bx$" and "$C+Dx$" yields "$(A+C)+(B+D)x$". Here the ability to move from concrete examples to general examples with symbols is symbolic generalization. The active interest in LLMs's use for solving complex mathematical problems (Romera-Paredes et al., 2023; Liu et al., 2023b; Azerbayev et al., 2023; Imani et al., 2023) makes symbolic generalization a relevant concept to study.

Research questions.

Question 5.1.1. von Oswald et al. (2023) showed that for linear attention only Transformers, in-context learning is implemented by the model in the form of implicit gradient descent updates on the in-context examples in the prompt. This meta learning view of ICL could be leveraged to provide bounds on in-context learning accuracy via meta learning generalization bounds (Rezazadeh, 2022). Could we obtain non-vacuous bounds beyond these simple cases?

Question 5.1.2. How to formalize the out-of-distribution nature of NLP tasks? For example, what is a useful way to describe the part of the pre-training distribution's support with vanishingly small probabilities (e.g., ICL prompts) Is it possible to connect the algorithmic (e.g., Kolmogorov) complexity to qualitative model properties and better (compositional) generalization?

Question 5.1.3. How can we relate symbolic generalization or reasoning to the probability distributions? How do we accomplish this, for example, for symbolic ODEs? Context: Becker et al. (2023) proposed a Transformer-based method for learning a symbolic representation of ODEs. Since the data-generating process of such ODEs yields a stochastically generated tree, learning a probabilistic model of symbolic ODEs connects symbolic generalization and probabilistic modeling. Indeed, as the authors show in their Fig. 3., the Transformer learns a symbolic ODE that can extrapolate.

Question 5.1.4. Can the findings from the literature that demonstrate the linearity of LLMs in certain cases (Hernandez et al., 2024; Merullo et al., 2024) be used to explain OOD extrapolation in LLMs?

5.2. Computational language modeling to study transferability

In § 5.1, we advocated for more appropriate generalization measures that reflect the properties of natural languages. For studying phenomena such as compositionality, we can rely on the advancements in computational linguistics and theoretical computer science. We believe that using well-defined formal languages as our testbed and developing computational models for LLMs is a promising direction to understand (near) OOD behavior in language models.

Formal languages. Natural languages' expressive power hinges on their (approximate) discrete compositionality (Pinker, 1994), i.e., combining a discrete set of symbols according to the language rules. Since formal languages such as PCFGs can be constructed to be discrete compositional, they are an excellent testbed to study compositionality in LLMs (cf. our experiments in § 3.1). Namely, we can define the union or intersection of multiple grammars and study extrapolation and transfer performance. Indeed, many researchers rely on PCFGs for understanding aspects of natural languages (Liu et al., 2023a; Favre, 2020; Merrill, 2023; Ackerman & Cybenko, 2020).

A computational perspective on language modeling. Computational models of (formal) languages can provide new insights for understanding LLMs. Regular languages can be modelled as Finite State Machines (FSMs), and Recurrent Neural Networks (RNNs) have the same computational model (Cleeremans et al., 1989); whereas PCFGs need an additional stack (Allott, 2021). 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. As we detail in § 5.3, computational models of LLMs and algorithmic information theory can help characterize the inductive biases in LLMs.

Research questions.

Question 5.2.1. Is it possible with computational models (such as the RASP language) to define the minimal set of requirements for models (Transformers, state-space models such as Mamba) to perform well on transfer tasks? Are all components in these architectures necessary? Is there a component beyond the architecture (such as the optimization algorithm or weight initialization) without which good transfer is not guaranteed?

Question 5.2.2. Are the models with good fine-tuning performance related in weight space or function space, e.g., is there a (linear) mode connectivity result that describes the geometry of LLMs that transfer well to other tasks?

5.3. 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 and alternative generalization measures (§ 5.1). 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*[3] 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

[3]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

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 of trained neural networks, which 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 uncomputable, 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. 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., 2023a; 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. RASP (Weiss et al., 2021) as a computational model for Transformers (§ 5.2) 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.

Research questions.

Question 5.3.1. How do we go beyond the current study of inductive bias? We believe that fruitful potential directions include studying the computational models and information flow of SOTA models using an established toolbox from other fields such as the theory of finite-state automata, as well as modeling Transformers in the RASP language.

Question 5.3.2. What would happen by iteratively training GPT models to imitate each other? What are the "eigenfeatures" these models amplify, i.e., what are they sensitive to (akin to the periodic features CNNs learn)?

Question 5.3.3. Does model architecture and data contain an implicit bias towards ICL? What would happen if we trained a Transformer adversarially (such that it is not an in-context learner), how many (compared to the size of the original training set) adversarial samples do we need to suppress ICL? If we then train a second Transformer to imitate the first one, does ICL emerge in the second? If we add negligible adversarial perturbation to the data, can ICL still emerge?

6. Conclusion

In this paper, we articulated the position that deep learning theory needs yet another shift to understand emergent LLM behaviours. LLMs often operate in the saturation regime—i.e., they reach (near-)optimal training and test loss—, making statistical generalization insufficient to explain desired model properties because of its "black-box" nature and i.i.d. assumption. Therefore, we should focus on inductive biases that can induce qualitative properties relevant for natural language tasks. We support our position by demonstrating multiple ways in which AR LMMs are non-identifiable, but due to inductive biases, they can exhibit OOD rule extrapolation (§ 3.1), in-context learning (§ 3.2), and data-efficient fine-tunability (§ 3.3). Our case studies formulate well-defined scenarios that can serve as a basis to study LLMs theoretically. We encourage the community to (1) study more expressive generalization measures; (2) use formal languages and computational models to study LLM behaviour in well-controlled settings; and (3) analyze and propose helpful inductive biases that ensure good OOD performance and transferability.

Acknowledgements

The authors would like to thank the anonymous reviewers for theior comments improving the manuscript and inspiring some of the research questions, Siyuan Guo for fruitful comments on OOV generalization, Bence Nyéki for insights on practical aspects of natural language processing, Felix Leeb for discussions on LLM reasoning, Vince Velkey for constructive comments on the manuscript, and Gail Weiss for her insights on PCFGs. This work was supported by a Turing AI World-Leading Researcher Fellowship G111021. Patrik Reizinger acknowledges his membership in the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program and thanks the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for its support. This work was supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A. Wieland Brendel acknowledges financial support via an Emmy Noether Grant funded by the German Research Foundation (DFG) under grant no. BR 6382/1-1 and via the Open Philantropy Foundation funded by the Good Ventures Foundation. Wieland Brendel is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645. This research utilized compute resources at the Tübingen Machine Learning Cloud, DFG FKZ INST 37/1057-1 FUGG. Anna Kerekes would like to thank the Max Planck ETH Center for Learning Systems (CLS).

Impact Statement

A deeper understanding of contributing factors to the success LLMs has potentially positive impacts: it may lead to increased data-, cost- and energy-efficiency, and thus broader access to the benefits, and it may lead to better guarantees and predictions of model behaviour that increase safety and interpretability. Our work advocates for a more application-driven theory of deep learning (whereby researchers take into account ways in which these models are used) but falls short of making recommendations that these application-driven theories should incorporate sociotechnical aspects or account for models' effect on humans.

References

Ackerman, J. and Cybenko, G. A Survey of Neural Networks and Formal Languages, June 2020. URL http://arxiv.org/abs/2006.01338. arXiv:2006.01338 [cs]. 8, 9

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I. 4

Allott, N. *A companion to Chomsky*. Blackwell Companions to Philosophy. Wiley-Blackwell, Hoboken, NJ, May 2021. 8

Alquier, P. and Guedj, B. Simpler pac-bayesian bounds for hostile data. *Machine Learning*, 107(5):887–902, December 2017. ISSN 1573-0565. doi: 10.1007/s10994-017-5690-0. URL http://dx.doi.org/10.1007/s10994-017-5690-0. 7

Alquier, P., Li, X., and Wintenberger, O. Prediction of time series by statistical learning: general losses and fast rates, 2012. 7

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization, 2019. 8

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. A closer look at memorization in deep networks, 2017. 9

Azerbayev, Z., Schoelkopf, H., Paster, K., Santos, M. D., McAleer, S., Jiang, A. Q., Deng, J., Biderman, S., and Welleck, S. Llemma: An open language model for mathematics, 2023. 7

Bahdanau, D., Murty, S., Noukhovitch, M., Nguyen, T. H., de Vries, H., and Courville, A. Systematic generalization: What is required and can it be learned? In *Proceedings of the 2019 International Conference on Learning Representations*, 2019. 7

Bartlett, P. L. and Mendelson, S. 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

Becker, S., Klein, M., Neitz, A., Parascandolo, G., and Kilbertus, N. Predicting Ordinary Differential Equations with Transformers. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 1978–2002. PMLR, July 2023. URL https://proceedings.mlr.press/v202/becker23a.html. ISSN: 2640-3498. 8

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

Brady, J., Zimmermann, R. S., Sharma, Y., Schölkopf, B., von Kügelgen, J., and Brendel, W. Provably Learning Object-Centric Representations, May 2023. URL http://arxiv.org/abs/2305.14229. arXiv:2305.14229 [cs]. 6, 7

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020. 1

Chérief-Abdellatif, B.-E., Shi, Y., Doucet, A., and Guedj, B. On PAC-Bayesian reconstruction guarantees for VAEs. In

25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia / Virtual, Spain, March 2022. URL https://inria.hal.science/hal-03587178. 14 pages. 7

Cleeremans, A., Servan-Schreiber, D., and McClelland, J. L. Finite State Automata and Simple Recurrent Networks. *Neural Computation*, 1(3): 372–381, September 1989. ISSN 0899-7667, 1530-888X. doi: 10.1162/neco.1989.1.3.372. URL https://direct.mit.edu/neco/article/1/3/372-381/5483. 8

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

De Palma, G., Kiani, B. T., and Lloyd, S. Random deep neural networks are biased towards simple functions, 2019. 9

Delétang, G., Ruoss, A., Duquenne, P.-A., Catt, E., Genewein, T., Mattern, C., Grau-Moya, J., Wenliang, L. K., Aitchison, M., Orseau, L., Hutter, M., and Veness, J. Language modeling is compression, 2023. 9

Deng, Z., He, H., and Su, W. Toward better generalization bounds with locally elastic stability. In Meila, M. and Zhang, T. (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

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 6

Dingle, K., Camargo, C., and Louis, A. Input–output maps are strongly biased towards simple outputs. *Nature Communications*, 9, 02 2018. doi: 10.1038/s41467-018-03101-6. 8

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp Minima Can Generalize For Deep Nets, May 2017. URL http://arxiv.org/abs/1703.04933. arXiv:1703.04933 [cs]. 5

Dziugaite, G. K. and Roy, D. M. 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

Falcon, W. and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL https://github.com/Lightning-AI/lightning. 23

Favre, B. Contextual language understanding Thoughts on Machine Learning in Natural Language Processing. 2020. 8, 9

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM. 6

Gao, T., Huang, Q., and Mooney, R. J. Systematic generalization on gscan with language conditioned embedding. *CoRR*, abs/2009.05552, 2020. URL https://arxiv.org/abs/2009.05552. 7

Geirhos, R., Temme, C. R. M., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/0937fb5864ed06ffb59ae5f9b5ed67a9-Abstract.html. 6

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL https://www.nature.com/articles/s42256-020-00257-z. Number: 11 Publisher: Nature Publishing Group. 6

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, November 2022. URL http://arxiv.org/abs/1811.12231. arXiv:1811.12231 [cs, q-bio, stat]. 6

Germain, P., Habrard, A., Laviolette, F., and Morvant, E. A new pac-bayesian perspective on domain adaptation. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 859–868, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/germain16.html. 7

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia

Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html. 1

Goldberg, Y. Reinforcement Learning for Language Models, 2023. URL https://gist.github.com/yoavg/6bff0fecd65950898eba1bb321cfbd81. 6

Goldblum, M., Finzi, M., Rowan, K., and Wilson, A. G. The no free lunch theorem, kolmogorov complexity, and the role of inductive biases in machine learning, 2023. 9

Gontier, N., Sinha, K., Reddy, S., and Pal, C. Measuring systematic generalization in neural proof generation with transformers. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22231–22242. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/fc84ad56f9f547eb89c72b9bac209312-Paper.pdf. 7

Goyal, A. and Bengio, Y. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2266):20210068, October 2022. doi: 10.1098/rspa.2021.0068. URL https://royalsocietypublishing.org/doi/10.1098/rspa.2021.0068. Publisher: Royal Society. 6

Grau-Moya, J., Genewein, T., Hutter, M., Orseau, L., Delétang, G., Catt, E., Ruoss, A., Wenliang, L. K., Mattern, C., Aitchison, M., and Veness, J. Learning universal predictors, 2024. 9

Gresele, L., von Kügelgen, J., Stimper, V., Schölkopf, B., and Besserve, M. Independent mechanism analysis, a new concept? *arXiv:2106.05200 [cs, stat]*, June 2021. URL http://arxiv.org/abs/2106.05200. arXiv: 2106.05200. 6

Gunasekar, S., Woodworth, B., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization, 2017. 8

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks, 2019. 8

Guo, S., Tóth, V., Schölkopf, B., and Huszár, F. Causal de Finetti: On the Identification of Invariant Causal Structure in Exchangeable Data. *arXiv:2203.15756 [cs, math, stat]*, March 2022. URL http://arxiv.org/abs/2203.15756. arXiv: 2203.15756. 6

HaoChen, J. Z. and Ma, T. A Theoretical Study of Inductive Biases in Contrastive Learning, November 2022. URL http://arxiv.org/abs/2211.14699. arXiv:2211.14699 [cs, stat]. 6

Hernandez, E., Sharma, A. S., Haklay, T., Meng, K., Wattenberg, M., Andreas, J., Belinkov, Y., and Bau, D. Linearity of Relation Decoding in Transformer Language Models, February 2024. URL http://arxiv.org/abs/2308.09124. arXiv:2308.09124 [cs]. 8

Hoon, A., Freegard, G., and Dymond, S. Symbolic generalization of the near-miss in simulated slot-machine gambling. *Behavior Analysis: Research and Practice*, 2020. doi: 10.1037/bar0000198. URL http://dx.doi.org/10.1037/bar0000198. Advance online publication. 7

Hoyer, P., Janzing, D., Mooij, J. M., Peters, J., and Schölkopf, B. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/hash/f7664060cc52bc6f3d620bcedc94a4b6-Abstract.html. 6

Huszár, F. Is Maximum Likelihood Useful for Representation Learning? https://www.inference.vc/maximum-likelihood-for-representation-learning-2/, 2017. 4

Hutter, M. A theory of universal artificial intelligence based on algorithmic complexity, 2000. 9

Hyvarinen, A. and Morioka, H. Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA. *arXiv:1605.06336 [cs, stat]*, May 2016. URL http://arxiv.org/abs/1605.06336. arXiv: 1605.06336. 6

Hyvärinen, A. Independent component analysis: recent advances. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110534, February 2013. ISSN 1364-503X, 1471-2962. doi: 10.1098/rsta.2011.0534. URL https://royalsocietypublishing.org/doi/10.1098/rsta.2011.0534. 2

Hyvärinen, A. and Pajunen, P. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, April 1999. ISSN 0893-6080. doi: 10.1016/S0893-6080(98)00140-3. URL https://www.sciencedirect.com/science/article/pii/S0893608098001403. 6

Hyvärinen, A., Khemakhem, I., and Monti, R. Identifiability of latent-variable and structural-equation models: from linear to nonlinear, February 2023. URL http://arxiv.org/abs/2302.02672. arXiv:2302.02672 [cs, stat]. 2

Imani, S., Du, L., and Shrivastava, H. Mathprompter: Mathematical reasoning using large language models, 2023. 7

Jacot, A., Gabriel, F., and Hongler, C. Neural Tangent Kernel: Convergence and Generalization in Neural Networks, February 2020. URL http://arxiv.org/abs/1806.07572. arXiv:1806.07572 [cs, math, stat]. 5, 8

Jang, C., Lee, S., Park, F., and Noh, Y.-K. A reparametrization-invariant sharpness measure based on information geometry, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b2ba568effcc3ab221912db2fb095ea9-Paper-Conference.pdf. 6

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them, 2019. 6

Ju, H., Li, D., and Zhang, H. R. Robust fine-tuning of deep neural networks with hessian-based generalization guarantees. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10431–10461. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ju22a.html. 6

Kawaguchi, K., Bengio, Y., and Kaelbling, L. *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

Ke, N. R., Bilaniuk, O., Goyal, A., Bauer, S., Larochelle, H., Schölkopf, B., Mozer, M. C., Pal, C., and Bengio, Y. Learning Neural Causal Models from Unknown Interventions. *arXiv:1910.01075 [cs, stat]*, August 2020. URL http://arxiv.org/abs/1910.01075. arXiv: 1910.01075. 6

Kirk, R., Mediratta, I., Nalmpantis, C., Luketina, J., Hambro, E., Grefenstette, E., and Raileanu, R. Understanding the Effects of RLHF on LLM Generalisation and Diversity, February 2024. URL http://arxiv.org/abs/2310.06452. arXiv:2310.06452 [cs]. 6

Klindt, D., Schott, L., Sharma, Y., Ustyuzhaninov, I., Brendel, W., Bethge, M., and Paiton, D. Towards Nonlinear Disentanglement in Natural Data with Temporal Sparse Coding. *arXiv:2007.10930 [cs, stat]*, March 2021. URL http://arxiv.org/abs/2007.10930. arXiv: 2007.10930. 6

Kolmogorov, A. 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. 9

Lachapelle, S., Brouillard, P., Deleu, T., and Lacoste-Julien, S. Gradient-Based Neural DAG Learning. *arXiv:1906.02226 [cs, stat]*, February 2020. URL http://arxiv.org/abs/1906.02226. arXiv: 1906.02226. 6

Li, Y., Zhao, L., Wang, J., and Hestness, J. Compositional generalization for primitive substitutions. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4293–4302, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1438. URL https://aclanthology.org/D19-1438. 7

Lin, F., Fang, X., and Gao, Z. 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

Liu, H., Xie, S. M., Li, Z., and Ma, T. 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 2023a. URL https://proceedings.mlr.press/v202/liu23ao.html. ISSN: 2640-3498. 1, 4, 5, 6, 8, 9, 18

Liu, T., Lu, J., Yan, Z., and Zhang, G. Statistical generalization performance guarantee for meta-learning with data dependent prior. *Neurocomput.*, 465(C): 391–405, nov 2021. ISSN 0925-2312. doi: 10.1016/j.neucom.2021.09.018. URL https://doi.org/10.1016/j.neucom.2021.09.018. 7

Liu, Y., Singh, A., Freeman, C. D., Co-Reyes, J. D., and Liu, P. J. Improving large language model fine-tuning for solving math problems, 2023b. 7

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-Centric Learning with Slot Attention. *arXiv:2006.15055 [cs, stat]*, October 2020. URL http://arxiv.org/abs/2006.15055. arXiv: 2006.15055. 7

Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs, math]. 24

Lotfi, S., Finzi, M., Kapoor, S., Potapczynski, A., Goldblum, M., and Wilson, A. G. Pac-bayes compression bounds so tight that they can explain generalization, 2022. 2

Lotfi, S., Finzi, M., Kuang, Y., Rudner, T., Goldblum, M., and Wilson, A. 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

Löwe, S., Lippe, P., Locatello, F., and Welling, M. Rotating Features for Object Discovery, June 2023. URL http://arxiv.org/abs/2306.00600. arXiv:2306.00600 [cs]. 7

Mahankali, A. V., Hashimoto, T., and Ma, T. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8p3fu56lKc. 4

Masegosa, A. R. 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

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

Mbacke, S. D., Clerc, F., and Germain, P. Pac-bayesian generalization bounds for adversarial generative models. *CoRR*, abs/2302.08942, 2023. doi: 10.48550/arXiv.2302.08942. URL https://doi.org/10.48550/arXiv.2302.08942. 7

Merrill, W. 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. 8, 9

Merullo, J., Eickhoff, C., and Pavlick, E. Language Models Implement Simple Word2Vec-style Vector Arithmetic, April 2024. URL http://arxiv.org/abs/2305.16130. arXiv:2305.16130 [cs]. 8

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work?, 2022. 1

Mingard, C., Skalse, J., Valle-Pérez, G., Martínez-Rubio, D., Mikulik, V., and Louis, A. A. Neural networks are a priori biased towards boolean functions with low entropy, 2020. 8

Murphy, K. P. *Probabilistic machine learning: an introduction*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts London, England, 2022. ISBN 978-0-262-04682-4. 4

Nakkiran, P., Kaplun, G., Kalimeris, D., Yang, T., Edelman, B. L., Zhang, F., and Barak, B. SGD on neural networks learns functions of increasing complexity. *CoRR*, abs/1905.11604, 2019. URL http://arxiv.org/abs/1905.11604. 6, 9

Offert, F. and Bell, P. Understanding Perceptual Bias in Machine Vision Systems. 2021. 6

Olah, C. Mechanistic interpretability, variables, and the importance of interpretable bases, 2022. 9

OpenAI et al. Gpt-4 technical report, 2023. 24

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022. 1

Papa, S., Winther, O., and Dittadi, A. Inductive Biases for Object-Centric Representations in the Presence of Complex Textures, August 2022. URL http://arxiv.org/abs/2204.08479. arXiv:2204.08479 [cs]. 6

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 23

Pearl, J. *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

Pinker, S. *The language instinct*. William Morrow, New York, NY, December 1994. 8

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets, January 2022. URL http://arxiv.org/abs/2201.02177. arXiv:2201.02177 [cs]. 2

Press, O., Smith, N. A., and Levy, O. Improving Transformer Models by Reordering their Sublayers. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (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. 8

Pérez-Ortiz, M., Rivasplata, O., Shawe-Taylor, J., and Szepesvári, C. 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

Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., and Henderson, P. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023. 6

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 24

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language Models are Unsupervised Multitask Learners. pp. 24, 2018. 3, 24

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F. A., Bengio, Y., and Courville, A. On the spectral bias of neural networks, 2019. 9

Refinetti, M., Ingrosso, A., and Goldt, S. Neural networks trained with sgd learn distributions of increasing complexity, 2023. 9

Rezazadeh, A. A general framework for pac-bayes bounds for meta-learning, 06 2022. 7

Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M. P., Dupont, E., Ruiz, F. J. R., Ellenberg, J. S., Wang, P., Fawzi, O., Kohli, P., and Fawzi, A. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, December 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06924-6. URL http://dx.doi.org/10.1038/s41586-023-06924-6. 7

Ruis, L. and Lake, B. Improving systematic generalization through modularity and augmentation, 2022. 7

Rusak, E., Reizinger, P., Zimmermann, R. S., Bringmann, O., and Brendel, W. Content suppresses style: dimensionality collapse in contrastive learning. In *NeurIPS 2022 Workshop: Self-Supervised Learning - Theory and Practice*, 2022. 4

Russo, D. and Zou, J. Controlling bias in adaptive data analysis using information theory. In Gretton, A. and Robert, C. C. (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

Saunshi, N., Ash, J., Goel, S., Misra, D., Zhang, C., Arora, S., Kakade, S., and Krishnamurthy, A. Understanding Contrastive Learning Requires Incorporating Inductive Biases, February 2022. URL http://arxiv.org/abs/2202.14037. arXiv:2202.14037 [cs]. 4

Saxe, A. M., McClelland, J. L., and Ganguli, S. 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. 5

Schmidhuber, J. 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. 9

Shimizu, S., Hoyer, P. O., Hyvarinen, A., and Kerminen, A. A Linear Non-Gaussian Acyclic Model for Causal Discovery. pp. 28, 2006. 6

Solomonoff, R. 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. 9

Tay, Y., Dehghani, M., Rao, J., Fedus, W., Abnar, S., Chung, H. W., Narang, S., Yogatama, D., Vaswani, A., and Metzler, D. Scale Efficiently: Insights from Pre-training and

Fine-tuning Transformers, January 2022. URL http://arxiv.org/abs/2109.10686. arXiv:2109.10686 [cs]. 4, 5

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

Török, B., Nagy, D. G., Kiss, M., Janacsek, K., Németh, D., and Orbán, G. Tracking the contribution of inductive bias to individualised internal models. *PLOS Computational Biology*, 18(6):e1010182, June 2022. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1010182. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1010182. Publisher: Public Library of Science. 6

Valle-Perez, G., Camargo, C. Q., and Louis, A. A. 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. 8

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

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

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. 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, 24

Vidaurre, D., Bielza, C., and Larrañaga, P. 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. 8

Vitányi, P. and Li, M. On prediction by data compression. In van Someren, M. and Widmer, G. (eds.), *Machine Learning: ECML-97*, pp. 14–30, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg. 9

von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent, May 2023. URL http://arxiv.org/abs/2212.07677. arXiv:2212.07677 [cs]. 4, 7

Wang, H., Gao, R., and Calmon, F. P. 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

Wang, X., Zhu, W., Saxon, M., Steyvers, M., and Wang, W. Y. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning, 2023b. 4, 18

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022. 1

Weiss, G., Goldberg, Y., and Yahav, E. Thinking Like Transformers, July 2021. URL http://arxiv.org/abs/2106.06981. arXiv:2106.06981 [cs]. 8, 9

Wiedemer, T., Brady, J., Panfilov, A., Juhos, A., Bethge, M., and Brendel, W. Provable Compositional Generalization for Object-Centric Learning, October 2023a. URL http://arxiv.org/abs/2310.05327. arXiv:2310.05327 [cs]. 6, 7

Wiedemer, T., Mayilvahanan, P., Bethge, M., and Brendel, W. Compositional Generalization from First Principles, July 2023b. URL http://arxiv.org/abs/2307.05596. arXiv:2307.05596 [cs, stat]. 6, 7

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. 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. 23

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. 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, 4, 5, 18, 19, 21, 22

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T.-Y. On Layer Normalization in the Transformer Architecture, June 2020. URL http://arxiv.org/abs/2002.04745. arXiv:2002.04745 [cs, stat]. 24

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

Xu, T., Li, Z., Yu, Y., and Luo, Z.-Q. On Generalization of Adversarial Imitation Learning and Beyond, February 2022. URL http://arxiv.org/abs/2106.10424. arXiv:2106.10424 [cs]. 6

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. November 2016. URL https://openreview.net/forum?id=Sy8gdB9xx. 1, 2

Zhang, R., Frei, S., and Bartlett, P. L. 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. (2023a)). 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. Problem framework and notations

Our framework and notations are based on those in (Xie et al., 2022), but we provide some additional details.

Nomenclature

$\theta \in \Theta$ space of latent concepts, the concept θ determines the transition probability matrix of the HMM hidden states

p_0 a prior distribution on Θ

$o \in \mathcal{O}$ a single token and an output state of the HMM. And $\mathcal{O}$ is the set of all tokens

$h \in \mathcal{H}$ a hidden state of the HMM

$o^{\text{delim}} \in \mathcal{O}$ a unique delimiter token and an output state of the HMM

$h^{\text{delim}} \in \mathcal{D} \subset \mathcal{H}$ a hidden delimiter state of the HMM

T the (fixed) length of each pre-training document $(o_1, o_2, ..., o_T)$

$(x_i, y_i) \in \mathcal{O}^k$ an in-context learning example

k the length of each in-context learning example, i.e. $x_i \in \mathcal{O}^{k-1}$, $y_i \in \mathcal{O}$

S_n the concatenation of n in-context learning examples separated by delimiter tokens, $S_n = \{(x_i, y_i, o^{\text{delim}})|1 \leq i \leq n\}$

x_{test} a test example for in-context learning, with length $k-1$

$(S_n, x_{\text{test}}) \in \mathcal{P}_N$ the full in-context learning prompt with n examples and a test example

n the number of in-context learning examples in S_n

N the (fixed) length of the in-context learning prompts, $N = (k+1)(n+1) - 1$

$\mathcal{P}_N \subset \mathcal{O}^N$ the set of sequences of length N having prompt structure (Def C.1)

y the in-context learning output

p_{prompt} the distribution used to generate in-context prompts (S_n, x_{test}), defined on $\mathcal{O}^N \times \mathcal{H} \times \mathcal{D}^n$ (see below for a full definition). Note that p_{prompt} different from p due to the distribution shift induced by the concatenation with the delimiter tokens.

$p(\cdot|\theta)$ the distribution of token sequences (of some fixed maximal length) given the latent concept, defined by the HMM

q_n a distribution on token sequences, depends on n, the number of examples in the in-context learning prompt

Let $p_0(\theta)$ be a prior distribution on the latent concepts $\theta \in \Theta$. For each θ, let the distribution $p(o_1, \ldots, o_T|\theta)$ be a HMM, where the concept θ determines the transition probability matrix of the HMM. From this HMM, we generate token sequences $o_1, \ldots, o_T$ of a fixed pre-training document length T. First, a concept θ is sampled from $p_0(\theta)$, then a document is generated according to $p(o_1, \ldots, o_T|\theta)$. Therefore, the pre-training distribution (on token sequences) is a mixture of HMMs induced by these conditionals.

$$p(o_1, \ldots, o_T) = \int_{\theta \in \Theta} p(o_1, \ldots, o_T|\theta) p_0(\theta)\, d\theta. \tag{4}$$

Now we detail how the in-context prompts are generated. This iterative process defines p_{prompt}. The in-context prompt consists of example input-output pairs (x_i, y_i) and a test input x_{test}, and the goal is to predict the test output y_{test} by predicting the next token. All examples in the ICL prompt are connected via the same underlying concept θ^*—e.g., in the input-output pairs ("Gandhi was", "Indian"), ("Jefferson was", "American"), the underlying concept is the nationality of a person. The i-th example pair is independently generated as follows:

1. Generate a start hidden state h_i^{start} from a prompt start distribution $p_{\text{prompt}}(h)$. *This defines p_{prompt} on $\mathcal{H}$ as an arbitrary discrete distribution.*
2. Given h_i^{start}, generate the example sequence (x_i, y_i) from $p\left(o_1, o_2, \ldots, o_k \mid h_i^{\text{start}}, \theta^*\right)$, the pretraining distribution conditioned on a prompt concept θ^* (given by a HMM). *This defines $p_{prompt}(o_1, o_2, \ldots, o_k \mid h_i^{start}) := p(o_1, o_2, \ldots o_k \mid h_i^{start}, \theta^*)$.*
3. The delimiter state at the end of each example (except the test example) is sampled from $p_{\text{prompt}}(h^{\text{delim}})$. *This defines p_{prompt} on $\mathcal{D}$ as an arbitrary discrete distribution.*

The test input $x_{\text{test}} = x_{n+1}$ is sampled in the same way, i.e. $x_{\text{test}} \sim p\left(o_1, o_2, \ldots o_{k-1} \mid h_{n+1}^{\text{start}}, \theta^*\right)$. The prompt consists of a sequence of in-context examples S_n followed by the test example x_{test}, with a unique delimiter token o^{delim} between each element.

$$(S_n, x_{\text{test}}) = (x_1, y_1, o^{\text{delim}}, x_2, y_2, o^{\text{delim}}, \ldots, x_n, y_n, o^{\text{delim}}, x_{\text{test}}) \tag{5}$$

The above determines the probability of (S_n, x_{test}) under p_{prompt} as

$$\begin{aligned} p_{\text{prompt}}(S_n, x_{\text{test}}) = \int_{\mathcal{H}^{n+1}} &\left(\prod_{j=1}^{n} p_{\text{prompt}}(h_j^{\text{start}}) p(o_{1:k}^j \mid h_j^{\text{start}}, \theta^*) p_{\text{prompt}}(h_j^{\text{delim}}) \right) \cdot \\ &\left(p_{\text{prompt}}(h_{n+1}^{\text{start}}) p(o_{1:k-1}^j \mid h_{n+1}^{\text{start}}, \theta^*) \right) \mathrm{d}h_1^{\text{start}} \mathrm{d}h_2^{\text{start}} \ldots \mathrm{d}h_{n+1}^{\text{start}}. \end{aligned} \tag{6}$$

Note that it is not necessary to integrate with respect to the h_j^{delim} due to Assumption B.1 below. After sampling $(S_n, x_{\text{test}}) \sim p_{\text{prompt}}$, we treat them as fixed values without loss of generality.

For in-context learning, the ground-truth output y_{test} for the example x_{test} is defined by $y_{\text{test}} = \operatorname{argmax}_y p_{\text{prompt}}(y|x_{\text{test}})$, where $p_{\text{prompt}}(y|x_{\text{test}})$ can be calculated as

$$\begin{aligned} p_{\text{prompt}}(y|x_{\text{test}}) &= \int_{\mathcal{H}} p_{\text{prompt}}(y, h_{\text{test}}^{\text{start}}|x_{\text{test}})\, dh_{\text{test}}^{\text{start}} = \int_{\mathcal{H}} p_{\text{prompt}}(h_{\text{test}}^{\text{start}}|x_{\text{test}}) p_{\text{prompt}}(y|h_{\text{test}}^{\text{start}}, x_{\text{test}})\, dh_{\text{test}}^{\text{start}} = \\ &= \int_{\mathcal{H}} p_{\text{prompt}}(h_{\text{test}}^{\text{start}}|x_{\text{test}}) p(y|h_{\text{test}}^{\text{start}}, x_{\text{test}}, \theta^*)\, dh_{\text{test}}^{\text{start}} = \mathbb{E}_{h_{\text{test}}^{\text{start}} \sim p_{\text{prompt}}(h_{\text{test}}^{\text{start}}|x_{\text{test}})}[p(y|h_{\text{test}}^{\text{start}}, x_{\text{test}}, \theta^*)], \end{aligned} \tag{7}$$

where $h_{\text{test}}^{\text{start}}$ is the hidden state corresponding to the first token of x_{test}.

Then we focus on the in-context learning predictor $\operatorname{argmax}_y p(y|S_n, x_{\text{test}})$, which predicts the output with the highest probability over the pre-training distribution given the prompt. We say that the model is an in-context learner if, as $n \to \infty$,

$$\underset{y}{\operatorname{argmax}}\, p(y|S_n, x_{\text{test}}) \to y_{\text{test}}. \tag{8}$$

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

Assumption B.2 (Bound on delimiter transitions)**.** For any delimiter state $h^{\text{delim}} \in \mathcal{D}$ and any hidden state $h \in \mathcal{H}$, the probability of transitioning to a delimiter hidden state under θ is upper bounded $p\left(h^{\text{delim}} \mid h, \theta\right) < c_2 < 1$ for any $\theta \in \Theta \setminus \{\theta^*\}$, and is lower bounded $p\left(h^{\text{delim}} \mid h, \theta^*\right) > c_1 > 0$ for θ^*. Additionally, the start hidden state distribution for delimiter hidden states is bounded as $p\left(h^{\text{delim}} \mid \theta\right) \in [c_3, c_4]$.

The above two assumptions allow us to simplify our analysis and avoid degenerate cases such as a deterministic (hidden) Markov chain. The result of (Xie et al., 2022) (which we rely on) used 3 additional assumptions (Assumption 3,4,5), but those are omitted here.

C. Proof of Prop. 3.1

Proposition 3.1. *[ε−non-identifiability of ICL] Let N be the length of a prompt (S_n, x_{test}). 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 p in KL divergence*

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

We denote the threshold example sequence length as n_0, after which p satisfies in-context learning, i.e.,

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

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 have a prompt structure (to be defined). 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 KL $(p||q_n)$ as

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

- **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. To define q_n, we make use of the following definition.

Definition C.1. (Prompt structure) We say that a sequence of tokens $(o_1, ..., o_N)$ has prompt structure if it can be written in form $(S_n, x_{\text{test}}, y) = (x_1, y_1, o^{\text{delim}}, x_2, y_2, o^{\text{delim}}, \ldots, x_n, y_n, o^{\text{delim}}, x_{\text{test}}, y)$, with each x_i having length $k-1$ and y_i having length 1.

Let us denote the set of prompt structures in $\mathcal{O}^N$ as $\mathcal{P}_N$. We consider those sequences that have a prompt structure. 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 can expand q_n on prompt structures via the chain rule

$$\begin{aligned} q_n(S_n, x_{\text{test}}, y) &= \sum_{j=1}^{n+1} q_n(y_j|S_{j-1}, x_j) q_n(x_j|S_{j-1}) q_n(o^{\text{delim}}|S_{j-2}, x_{j-1}, y_{j-1}) && (9) \\ &= q_n(y|S_n, x_{\text{test}}) q_n(x_{\text{test}}|S_n) q_n(o^{\text{delim}}|S_{n-1}, x_n, y_n) \\ &+ \sum_{j=1}^{n} q_n(y_j|S_{j-1}, x_j) q_n(x_j|S_{j-1}) q_n(o^{\text{delim}}|S_{j-2}, x_{j-1}, y_{j-1}), && (10) \end{aligned}$$

with notation $x_{n+1} = x_{\text{test}}$, $y_{n+1} = y$ and $x_0 = y_0 = S_0 = S_{-1} = \emptyset$ in order to define the endpoints appropriately. For $j = 1, \ldots, n$ let

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

and

$$q_n(o^{\text{delim}}|S_{j-2}, x_{j-1}, y_{j-1}) := p(o^{\text{delim}}|S_{j-2}, x_{j-1}, y_{j-1}),$$

i.e., the same as p. We are only modifying $p(y|S_n, x_{\text{test}})$, and only at its largest and second largest values—as we show, that is sufficient to change the output of the argmax, thus, changing whether the model has ICL. Denote

$$\begin{aligned} a_1 &= \max_y p(y|S_n, x_{\text{test}}) \\ y_1^* &= \operatorname*{argmax}_y p(y|S_n, x_{\text{test}}) \\ a_2 &= \max_{y \neq y_1^*} p(y|S_n, x_{\text{test}}) \\ y_2^* &= \operatorname*{argmax}_{y \neq y_1^*} p(y|S_n, x_{\text{test}}). \end{aligned}$$

Then, for all $y \neq y_1^*$ and $y \neq y_2^*$ let q_n be the same as p, i.e.,

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

However, on the largest and second largest values we change p as the following

$$\begin{aligned} q_n(y_1^*|S_n, x_{\text{test}}) &:= \frac{a_1 + a_2}{2} - \frac{\delta}{2} \\ q_n(y_2^*|S_n, x_{\text{test}}) &:= \frac{a_1 + a_2}{2} + \frac{\delta}{2}, \end{aligned}$$

where δ is arbitrarily small. Due to (9), q_n is well-defined. Since $q_n(y|S_n, x_{\text{test}})$ has its maximum at y_2^*,

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

that is, the model q_n has a different output when the argmax operator is applied. Since $n > n_0$, from Xie et al. (2022) we know that p has ICL:

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

However, since p and q_n do not match, the following holds and q_n **cannot be an in-context learner**:

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

To bound KL $(p||q_n)$ in Step 2, we need the following inequality

$$\log\left(\frac{p(y|S_n, x_{\text{test}})}{q_n(y|S_n, x_{\text{test}})}\right) \leq \log\left(\frac{2}{1-\delta}\right). \tag{11}$$

It holds since if $q_n(y|S_n, x_{\text{test}}) = p(y|S_n, x_{\text{test}})$, then the log equals to 0, otherwise by (11):

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

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

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 $q_n(o_1, \ldots, o_N)$.

$$\mathrm{KL}(p(o_1, \ldots, o_N)||q_n(o_1, \ldots, o_N)) = \sum_{t \in \mathcal{O}^N} p(t) \log\left(\frac{p(t)}{q_n(t)})\right) = \sum_{t \in \mathcal{P}_N} p(t) \log\left(\frac{p(t)}{q_n(t)}\right) +$$

$$+ \sum_{t \notin \mathcal{P}_N} p(t) \log\left(\frac{p(t)}{q_n(t)}\right) = \sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y) \log\left(\frac{p(S_n, x_{\text{test}}, y)}{q_n(S_n, x_{\text{test}}, y)}\right) =$$

where the last equation is due to the fact that q_n is defined to equal p on non-prompt structures ($\mathcal{O}^N \setminus \mathcal{P}_N$). Expanding $p(S_n, x_{\text{test}}, y)$ and $q_n(S_n, x_{\text{test}}, y)$ via the chain rule, we get

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

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 (11).

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

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.

$$\sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y) = \sum_{(S_n, x_{\text{test}}, y)} \int_{\theta \in \Theta} p(S_n, x_{\text{test}}, y|\theta) p_0(\theta) \tag{12}$$
$$= \int_{\theta \in \Theta} \sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y|\theta) p_0(\theta). \tag{13}$$

Let us fix $\theta \in \Theta$. Consider the HMM with output distribution p, conditioned on θ. We wish to focus on the property of prompt structures that delimiter output states occur at every $(k+1)^{\text{th}}$ token. Bounding the probability of delimiter output states occurring at every $(k+1)^{\text{th}}$ token allows us to bound the probability of prompt structures.

From Assumption B.1 of Xie et al. (2022), we know that a delimiter hidden state generates the delimiter output state with probability 1, and non-delimiter hidden states do not generate the delimiter output state. Hence it is equivalent to bound the probability of the hidden Markov chain being at delimiter hidden states exactly at every $(k+1)^{\text{th}}$ step.

Let us consider those latent concepts θ, where it is possible to reach a delimiter hidden state. Without loss of generality, assume that our HMM hidden state Markov chain is at a delimiter hidden state (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

$$p_{ij}^{k,\theta} = \sum_{h_1,\dots,h_k \in \mathcal{H}} p(h^{\text{delim},i}, h_k, h_{k-1}, ..., h_1, h^{\text{delim},j}|\theta) \tag{14}$$

as the probability of the hidden Markov chain going from state $h^{\text{delim},j}$ to $h^{\text{delim},i}$ in $k+1$ steps (it may pass through delimiter hidden states in the meantime). For those latent concepts θ, where it is not possible to reach a delimiter hidden state, it is possible to define $p_{ij}^{k,\theta}$ accordingly, but its value is zero. As this does not affect our upper bound, we may neglect these cases. Let $p^* = \sup_{\theta \in \Theta \setminus \{\theta^*\}} \max_{i,j} p_{ij}^{k,\theta}$, where the maximum is under all pairs of delimiter hidden states in $\mathcal{D}$. We show that $p^* < 1$. For all $\theta \in \Theta \setminus \theta^*$, by Assumption B.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,

$$p_{ij}^{k,\theta} = \sum_{h_1,\dots,h_k \in \mathcal{H}} p(h^{\text{delim},i}, h_k, h_{k-1}, ..., h_1, h^{\text{delim},j}|\theta) \tag{15}$$
$$= \sum_{h_1,\dots,h_k \in \mathcal{H}} p(h^{\text{delim},i}|h_k, \theta) p(h_k|h_{k-1}, ..., h_1, h^{\text{delim},j}, \theta) p(h_{k-1}, ..., h_1, h^{\text{delim},j}|\theta) \tag{16}$$
$$< c_2 \sum_{h_1,\dots,h_k \in \mathcal{H}} p(h_k|h_{k-1}, ..., h_1, h^{\text{delim},j}, \theta) p(h_{k-1}, ..., h_1, h^{\text{delim},j}|\theta) \tag{17}$$
$$\leq c_2. \tag{18}$$

Thus, $p^* \leq c_2$, and the probability of generating a prompt structure, which has n instances of this pattern of a delimiter hidden state appearing after k non-delimiter hidden states, is upper bounded by c_2^n.

Hence

$$\int_{\theta\in\Theta} \sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y|\theta)p_0(\theta) \leq \int_{\theta\in\Theta} (\max_{i,j} p_{ij}^{k,\theta})^n p_0(\theta) =$$
$$= \int_{\theta\in\{\theta^*\}} (\max_{i,j} p_{ij}^{k,\theta})^n p_0(\theta) + \int_{\theta\in\Theta\setminus\{\theta^*\}} (\max_{i,j} p_{ij}^{k,\theta})^n p_0(\theta) \leq (\sup_{\theta\in\Theta\setminus\{\theta^*\}} \max_{i,j} p_{ij}^{k,\theta})^n = (p^*)^n \leq c_2^n, \tag{19}$$

where the second inequality is because the integral over $\theta \in \{\theta^*\}$ is zero, since $\{\theta^*\}$ has 0 Lebesgue measure. Therefore $\sum_{(S_n, x_{\text{test}}, y)} p(S_n, x_{\text{test}}, y)$ 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 $n \geq n_1$. □

D. Identifiability

Definition D.1 (Set of probability measures)**.** We denote the set of probability measures on domain $\mathcal{X}$ as $\mathcal{M}(\mathcal{X})$.

Definition D.2 (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 D.3 (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

$$\forall i \neq j, p_i, p_j \in \mathcal{M}_\mathcal{A} : \mathcal{A}(p_i) = \mathcal{A}(p_j) = 1 \tag{20}$$
$$\forall i \neq j, p_i, p_j \in \mathcal{M}_{\overline{\mathcal{A}}} : \mathcal{A}(p_i) = \mathcal{A}(p_j) = 0 \tag{21}$$

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 D.4 ($\varepsilon-$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 $\text{KL}\left[p||q\right] \leq \varepsilon$.

E. 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). Our code and experimental logs are publicly available at `https://github.com/rpatrik96/llm-non-identifiability`.

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.

Table 2. 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 3. 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	$6e{-3}$
ACTIVATION	RELU
OPTIMIZER	ADAMW
LEARNING RATE SCHEDULER	INVERSE SQUARE ROOT
BATCH SIZE	128
LEARNING RATE	$2e{-3}$
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. 4, supplementing Fig. 2.

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 4. 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	$0.0199_{\pm 0.00025}$	$0.83_{\pm 0.122}$	$[0.634; 1.]$

F. Acronyms

CE cross entropy

AR autoregressive

DNN Deep Neural Network

EOS end-of-sequence

FSM Finite State Machine

HMM Hidden Markov Model

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

KL Kullback-Leibler Divergence

LLM Large Language Model

MLE maximum likelihood estimation

NTK Neural Tangent Kernel

OOD out-of-distribution

PCFG Probabilistic Context-Free Grammar

RASP Restricted-Access Sequence Processing Language
RLHF Reinforcement Learning from Human Feedback
RNN Recurrent Neural Network

SOS start-of-sequence
SSL Self-Supervised Learning