Why are Sensitive Functions Hard for Transformers?

Michael Hahn, Mark Rofin
Saarland Informatics Campus
Saarland University, Saarbrücken, Germany
{mhahn, mrofin}@lst.uni-saarland.de

Abstract

Empirical studies have identified a range of learnability biases and limitations of transformers, such as a persistent difficulty in learning to compute simple formal languages such as PARITY, and a bias towards low-degree functions. However, theoretical understanding remains limited, with existing expressiveness theory either overpredicting or underpredicting realistic learning abilities. We prove that, under the transformer architecture, the loss landscape is constrained by the input-space sensitivity: Transformers whose output is sensitive to many parts of the input string inhabit isolated points in parameter space, leading to a low-sensitivity bias in generalization. We show theoretically and empirically that this theory unifies a broad array of empirical observations about the learning abilities and biases of transformers, such as their generalization bias towards low sensitivity and low degree, and difficulty in length generalization for PARITY. This shows that understanding transformers' inductive biases requires studying not just their in-principle expressivity, but also their loss landscape.

1 Introduction

Given dramatic advances in machine learning applications powered by transformer models, there has been substantial interest in understanding which functions are easier or harder to learn and represent using transformers. Empirical research on both formal languages and synthetic functions has uncovered an intriguing array of learning biases, but theoretical understanding is lacking. For instance, Abbe et al. (2023) experimentally argued that heldout generalization is biased towards low-degree polynomials and Bhattamishra et al. (2023) provided empirical evidence that transformers prefer to represent functions of *low sensitivity*, that is, functions that do not strongly depend on many input bits. Perhaps the most prominent example of such learning biases is a consistent difficulty in learning the PARITY function, mapping bitstrings to their parity. This function is extremely sensitive, in the sense that flipping any bit flips the string's parity. Empirical studies have consistently found that training transformers to compute parities is difficult, and that solutions for shorter inputs do not generalize to longer inputs (e.g. Bhattamishra et al., 2020; Chiang and Cholak, 2022; Delétang et al., 2023; Ruoss et al., 2023). This stands in stark contrast to previously-popular reccurent models which easily fit PARITY with correct length generalization (Bhattamishra et al., 2020).

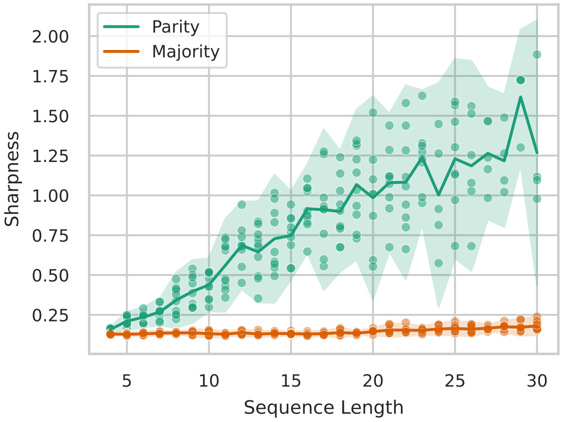


Figure 1: Training transformers on inputs of increasing length produces a steeper loss landscape for PARITY (as measured by average direction sharpness), while the loss landscape of MAJORITY does not show significant changes. Our main result (Theorem 6) provides a rigorous explanation for this phenomenon.

While a substantial amount of theoetical work has considered both the learnability (e.g. Edelman et al., 2022; Ahn et al., 2023) and the expressiveness of transformers (e.g. Yun et al., 2019; Hahn, 2020; Yao et al., 2021; Hao et al., 2022; Merrill et al., 2022; Merrill and Sabharwal, 2023b; Chiang et al., 2023; Strobl et al., 2023; Strobl, 2023; Angluin et al., 2023), existing theoretical studies do not consistently explain such learning biases. Hahn (2020) proved that, under two formal models of self-attention, no transformer can express PARITY at all input lengths. However, various other formal results showed that slightly relaxed assumptions about the transformer architecture resolved such expressiveness limitations. Most notably, Chiang and Cholak (2022) found that layer norm, by breaking the Lipschitz assumption used in Hahn (2020)'s Theorem 2, allows expressing PARITY in principle. Simultaneously, they empirically confirmed that such a

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14973–15008
August 11-16, 2024 ©2024 Association for Computational Linguistics

solution could not be practically found via (S)GD training. Various other formal models of transformers (e.g. Weiss et al., 2021; Merrill and Sabharwal, 2023b,c; Strobl, 2023) can also express PARITY despite its empirical difficulty. As already concluded by Chiang and Cholak (2022), these findings highlight a disconnect between expressive capacity and learnability: not all functions which transformers may express in principle are also learnt efficiently. Evidently, existing expressiveness theory for transformers is not able to consistently account for the practical learnability of problems under gradient descent.

Some prior work has studied the learnability of problems for transformers. For example, Edelman et al. (2022) bound the statistical capacity of the transformer architecture, showing that on those functions that transformers prefer to represent, they can generalize with good sample efficiency. Notably, they found that *sparse* parities could indeed be learned well by transformers. However, this result does not prove that PARITY, or other highly sensitive functions, are hard to learn, as that technique does not provide a direct characterization of which functions transformers prefer to represent. Other work has studied simplified setups such as linear attention (e.g. Ahn et al., 2023) or individual attention layers (e.g. Sanford et al., 2023).

Here, we provide results that have direct bearing on the learnability of PARITY and other sensitive functions, characterizing the loss landscape of transformers in terms of input-space sensitivity. We formally prove that, for the transformer architecture, parameter settings achieving high sensitivity in input space are necessarily brittle, so that close neighbors in parameter space will usually define different (typically much less sensitive) functions when inputs are long. As a consequence, transformers fitting high-sensitivity functions must inhabit very steep minima. We argue that this explains both difficulty in training and length generalization for PARITY (observed by Bhattamishra et al. (2020); Delétang et al. (2023); Ruoss et al. (2023)), and a low-sensitivity and low-degree bias in random initialization and generalization (observed by Abbe et al. (2023); Bhattamishra et al. (2023)).

Expressiveness theory does not account for learnability. While unique hard attention provably cannot represent PARITY (Hahn, 2020; Hao et al., 2022; Angluin et al., 2023), more realistic upper bounds accounting for soft attention (Weiss et al., 2021; Merrill and Sabharwal, 2023b,c; Strobl, 2023; Chiang and Cholak, 2022) leave the hardness of sensitive functions unexplained. Not only does PARITY have transformers (Chiang and Cholak, 2022), but it can also be easily represented in formalisms that have been suggested to meaningfully upper-bound the abilities of various formal models of soft-attention:[1]

Fact 1 (Existing theory overpredicting abilities)**.** *Simple representations for PARITY, valid across all input lengths, exist in RASP (Weiss et al., 2021), uniform circuits with majority gates (Merrill and Sabharwal, 2023c; Strobl, 2023), and FO[M] (Merrill and Sabharwal, 2023b).*

We prove this in Appendix A. Thus, existing expressiveness bounds do not account for the difficulty that transformers encounter in learning sensitive functions, in particular given that previously-popular recurrent models do not encounter this difficulty. Another family of results consists of Lipschitzness bounds (Hahn, 2020; Li et al., 2023), which bound the influence that any individual input bit has on the output of a transformer. These turn out to *underpredict* the abilities of transformers:

Fact 2 (Existing theory underpredicting abilities)**.** *By results of Hahn (2020); Li et al. (2023), the following holds: Consider a transformer without layer norm. If $x, x' \in \{\pm 1\}^n$ differ only in the i-th bit, then at any other position $j \neq i$, the output of a transformer differs only by $O(\frac{1}{n})$.*

This accounts for the difficulty of learning PARITY. But the bound suggests even simple sparse functions, such as FIRST (the language $1(0|1)^*$) to be difficult, but transformers learn these well (Bhattamishra et al., 2023; Edelman et al., 2022). Indeed, Chiang and Cholak (2022) note that the bound is overcome by layer norm or input-length-dependent scaling of attention logits, which enable modeling of sparse functions.

We will show that the observed low-sensitivity bias can be understood in terms of the *loss landscape*: while transformers can express highly sensitive functions, such transformers are isolated in parameter space, and minima interpolating a sensitive function are very sharp. Indeed, we prove that tiny perturbations of a highly sensitive transformer tend to define, when inputs are sufficiently long, very different functions with much lower sensitivity.

2 Model of Transformers

We will focus on boolean functions. Following the conventions in the Analysis of Boolean Functions literature (O'Donnell, 2014) in modeling bitstrings as elements of $\{-1, 1\}^n$, we assume the alphabet $\Sigma = \{-1, 1\}$, with word embeddings $e(-1), e(+1) \in \mathbb{R}^d$. There further are positional encodings $p_1, p_2, p_3, \dots \in \mathbb{R}^d$. At the zero-th layer, token and positional encodings are added: $y_i^{(0)} := e(x_i) + p_i$ $(i = 1, \dots, n)$, where $x \in \{\pm 1\}^n$ is the input string.

A transformer has a fixed number L of **layers**; the **activations** $y_i^{(k)} \in \mathbb{R}^d$ at position i of the k-th layer ($k = 1, \dots, L$) are defined as follows. Each layer has a set of H **attention heads**; we first compute attention scores for the h-th head:

$$a_{i,j}^{(k,h)} = (K_{k,h} y_j^{(k-1)})^T Q_{k,h} y_i^{(k-1)}$$

[1] Zhou et al. (2023) suggest that PARITY may not be representable in the RASP-L model, though the expressiveness of RASP-L is not well understood.

$$\hat{a}_{i,j}^{(k,h)} = \frac{\exp(a_{i,j}^{(k,h)})}{\sum_s a_{i,s}^{(k,h)}}$$

where $K_{k,h}$ ("key"), $Q_{k,h}$ ("query") are $\in \mathbb{R}^{d\times d}$. The activation of the head is computed by weighting according to attention weights $\hat{a}_{i,j}^{(k,h)}$, and applying a linear transformation V ("value"):

$$b_{i,h}^{(k)} = \sum_{j=1}^{n} \hat{a}_{i,j}^{(k,h)} V_{k,h} y_j^{(k-1)} \tag{1}$$

The per-position activations are then computed as

$$Y_i^{(k)} := f^{MLP}\left(y_i^{(k-1)} + \sum_{h=1}^{H} b_{i,h}^{(k)}\right) \tag{2}$$

where f^{MLP} is a one-layer MLP with a skip-connection. Transformers additionally implement layer norm (Ba et al., 2016):

$$LayerNorm(y) := \frac{y - mean(y)}{\sqrt{\sigma^2(y)+\varepsilon}} \tag{3}$$

where $\varepsilon \geq 0$ is a hyperparameter ensuring numerical stability, and $\sigma^2(\cdot)$ denotes the variance. By design, $\|LayerNorm(y)\|_2 \leq \sqrt{d}$, with equality at $\varepsilon = 0$. Transformer variants differ in where exactly layer norm is applied (e.g. Takase et al., 2022); we here assume for notational simplicity that layer norm applies after the MLP, but the details are irrelevant to our results, provided layer norm applies at least once. We thus set:

$$y_i^{(k)} := LayerNorm\left(Y_i^{(k)}\right) \tag{4}$$

Of key importance will be the the normalization factor:

$$N_i^{(k)} := \frac{1}{\sqrt{\sigma^2(Y_i^{(k)})+\varepsilon}} \tag{5}$$

Our theoretical results will link $N_i^{(k)}$ both to input-space sensitivity and parameter-space sharpness: We will find that large values of $N_i^{(k)}$ can increase expressive capacity, but at the price of increased brittleness.

Finally, we assume that predictions are made by $T := v_{out}^T \cdot y_n^{(L)}$ for some parameter $v_{out} \in \mathbb{R}^d$. Throughout, we will add the input string $x \in \{\pm 1\}^n$ as an argument when needed for disambiguation, e.g., writing $T(x)$ for the overall prediction made on x.

3 Average Sensitivity

Our results are centered around *average sensitivity*, a simple but foundational complexity metric for functions on the Boolean cube (e.g. Kahn et al., 1988; De Wolf, 2008; O'Donnell, 2014):

Definition 3. *For a bitstring $x \in \{\pm 1\}^n$ and a function $f : \{\pm 1\}^n \to \mathbb{R}$, we write*

$$s(x,f) := \frac{1}{4}\sum_{i=1}^{n} |f(x) - f(x^{\oplus i})|^2 \tag{6}$$

where $x^{\oplus i}$ denotes the bitstring x with the i-th bit flipped. The average sensitivity for inputs of length n is

$$as_n(f) := \frac{1}{2^n} \sum_{x\in\{\pm 1\}^n} s(x,f) \tag{7}$$

If f maps to $\{\pm 1\}$, then $s(x,f)$ is the number of Hamming neighbors of x on which f flips. This definition of $as_n(f)$ corresponds to the "total influence" from O'Donnell (2014, Def. 2.27). We explicitly define average sensitivity relative to input length n, as we will investigate the behavior of transformers performing a single function f across varying input lengths. The use of squared distances, rather than simple absolute distances, ensures that results about $as_n(f)$ transfer to results about degree profiles (Abbe et al., 2023), which we will later investigate (Eq. 17).

Average sensitivity is a general complexity metric with wide-ranging applications in theoretical computer science (e.g. Jukna, 2012). It is closely linked to the Fourier analysis on the Boolean cube (De Wolf, 2008), and is an average-case version of a family of sensitivity measures, closely related to other natural metrics such as decision tree depth and polynomial degree (Hatami et al., 2010). Both average sensitivity itself (Bhattamishra et al., 2023) and the Fourier structure (Abbe et al., 2023) have been empirically linked to transformers' generalization behavior. We will ground these empirical findings by relating average sensitivity to loss landscapes for the transformer architecture.

Example Functions Our theoretical results will apply to general functions on the Boolean cube. In order to ground these, we will illustrate transformers' low-sensitivity bias at the example of a few natural functions which have played a role in the theoretical literature of transformers or are otherwise illustrative of variability in average sensitivity.

PARITY indicates whether the number of ones in a bitstring is even (output 1) or odd (output -1); over the input space $\{\pm 1\}^n$ and output space $\{\pm 1\}$, it can be formally defined as the map $x \mapsto \prod_{i=1}^n x_i$. As flipping any input bit flips the output, $as_n(f) = n$. As described above, this function can in principle be represented by transformers, but has empirically been found to be very hard to learn.

MAJORITY maps $x \in \{\pm 1\}^n$ to 1 if $\#\{i : x_i = 1\} > n/2$, and -1 else. Transformers show good length generalization (Merrill et al., 2022; Zhou et al., 2023). It is known that $as_n(f) = \Theta(\sqrt{n})$ (O'Donnell, 2014, Ex. 2.22). However, $s(f,x) = n$ whenever the ones and zeros in x are almost fully balanced.

FIRST maps x to its first bit, x_1. As only the first bit matters, $as_n(f) = 1$. It is a simple example of a *sparse* function; more generally, a k-PARITY is a restriction of the PARITY function to only k inputs, where k is a constant. Transformers learn such sparse functions well (Bhattamishra et al., 2023; Edelman et al., 2022).

MEAN maps $x \mapsto \frac{1}{n}\sum_{i=1}^n x_i \in [-1,1]$. We have $as_n(f) = \frac{1}{n}$.

The PARITY function, when varying the number of bits considered, is a universal basis for Boolean functions, in the sense that any function $\{\pm 1\}^n \to \mathbb{R}$ can be represented as a linear combination of parities applying to different subsets of $\{x_1, \dots, x_n\}$. Functions are more sensitive when parities applying to larger subsets appear in this representation. We will investigate this connection further below.

4 Lower Bounds for Sensitive Functions

We first prove that representing sensitive functions with transformers requires large parameter norms and, when inputs get longer and longer, highly unbounded normalization factors $N_i^{(k)}$ in layer norm (5). We start from the global Lipschitzness bounds developed by Hahn (2020); Edelman et al. (2022); Li et al. (2023), but obtain more fine-grained average-case and high-probability bounds. These will then form the basis of our characterization of loss landscape sharpness around sensitive transformers. Our bounds will include a constant C that is the product of

$$\exp\left(4d \max_h \sum_{i=2}^{L} \|K_{i,h}^T Q_{i,h}\|_2\right) \tag{8}$$

and a term polynomial in H, d, the spectral norms of all parameter matrices appearing in the transformer, and the maximum norm of any positional or word embedding. See (32) in the Appendix for formal definition.

Existing Lipschitzness bounds (Fact 2) imply a bound along the lines of[2]

$$s(f,x) \leq \frac{C \exp(4d \max_h \|K_{1,h}^T Q_{1,h}\|_2)}{\varepsilon^{L/2}} \tag{9}$$

uniformly for each $x \in \{\pm 1\}^n$. Li et al. (2023) noted that the exponential dependency on the spectral norm of the key-query matrix is unavoidable for a global Lipschitzness bound. Our first result is that this dependency can be eliminated at the input layer for the vast majority of inputs, at the cost of a logarithmic factor, leading to a bound of the form (assuming $\varepsilon > 0$ in (3))

$$s(f,x) \leq C \frac{\log n}{\varepsilon^{L/2}} \tag{10}$$

for $1 - Hn^{-2}$ of inputs. We show this using a concentration bound argument, combining a Chernoff bound applying to each attention weight individually, with a union bound over all attention weights (Appendix, Lemma 11). Next, we address the role of layer norm. Chiang and Cholak (2022) showed that, at $\varepsilon \to 0$ in (3), layer norm enables transformers to represent PARITY. Lipschitzness bounds in terms of ε (10) cease being meaningful in this limit. We thus study the layer-norm induced blowup in more detail. In each layer, we consider the maximum blowup, given an input string $x \in \{\pm 1\}^n$:

$$\tau^{(k)}(x) := \max_{w=1,\dots,n} \{1 + N_w^{(k)}(x)\} \tag{11}$$

The addition of 1 is for technical reasons (Appendix, Lemma 16); it has little impact in the cases relevant to our results, which will be when $\tau^{(k)}(x) = \omega(1)$. We then write $\text{Blowup}(x) := \prod_{k=1}^{L} \tau^{(k)}(x)$, an upper bound on the product of the successive layer-norm-induced blowups when going from the input to the output. When $\varepsilon = 0$ in (3), Blowup(x) can be arbitrarily large. Foreshadowing Theorem 6, we will find that large values of Blowup(x) create very sharp minima.

Our first theorem localizes the layer norm blowup to the Hamming neighborhoods of sensitive inputs:

Theorem 4 (Local Bounds on Layer Norm Blowup)**.** *Consider a transformer with layer norm at arbitrary* $\varepsilon \geq 0$. *With probability* $1 - \frac{H}{n^2}$ *over the choice of* $x \in \{\pm 1\}^n$, *we have*

$$\frac{s(f,x)}{C\sqrt{n \log n}} \leq \text{Blowup}(x)^2 + \frac{1}{n} \sum_{i=1}^{n} \text{Blowup}(x^{\oplus i})^2 \tag{12}$$

The proof is in Appendix B.5. This permits us to state a bound on average sensitivity, in terms of the average layer norm blowup:

Corollary 5 (First Main Result)**.** *Consider a transformer with layer norm at arbitrary* $\varepsilon \geq 0$. *Then*

$$C \cdot \mathbb{E}[\text{Blowup}(x)^2] \geq \frac{as_n(f)}{\sqrt{n \log n}} - \frac{H}{n} \tag{13}$$

where the expectation is over the uniform distribution over $x \in \{\pm 1\}^n$.

The proof is in Appendix B.6. Note that $\frac{H}{n}$ is small when n is large, and the bound is dominated by $\frac{as_n(f)}{\sqrt{n \log n}}$. This result thus shows a tradeoff between parameters and LN blowup: at least one of them needs to be large to represent a sensitive function. When the sensitivity depends on the input length and grows faster than $\sqrt{n \log n}$, this tradeoff changes with the input length, requiring larger parameters or larger layer norm blowup as the input length increases.

Let us investigate the implications for the functions introduced in Section above. For PARITY, $as_n(f) = n$, and (13) predicts

$$C \cdot \mathbb{E}[\text{Blowup}(x)^2] = \Omega\left(\frac{\sqrt{n}}{\log n}\right) = \omega(1) \tag{14}$$

showing that, for fixed parameters, the layer norm blowup needs to increase as the input length increases. For the other functions, the bound is $O(1)$: For FIRST, $s(f,x) = as_n(f) = 1$, and the RHS of (13) is $O(1)$. Indeed, sparse functions can be modeled well by a family of transformers where the logits in the input layer scale with $\log n$ (Edelman et al., 2022; Chiang and Cholak, 2022). Unlike the prior Lipschitzness bounds (9), these

[2] Lipschitzness bounds by Edelman et al. (2022) are not directly applicable here, as these authors consider Lipschitzness in the *parameter space*, not the effect of changes in the *input space* as Theorem 3. See also Appendix, Remark 10.

scaled logits do not contribute to C – our new bound is thus consistent with the ease with which transformers learn sparse functions. For MEAN, $s(f,x) \sim \frac{1}{n^2}$; again no blowup is predicted. For MAJORITY, as $as_n(f) \sim \sqrt{n}$, no nontrivial average blowup is predicted. However, $s(f,x) = n$ whenever the ones and zeros in x are almost fully balanced; on such strings or their Hamming neighbors, (12) predicts $Blowup = \Omega(n^{1/4})$.

5 Sensitive Transformers are Brittle

Leveraging Corollary 5, we now show that transformers expressing sensitive functions must be very sensitive to small perturbations of model parameters. That is, *sensitivity in input space* entails *sensitivity in parameter space*. An important consequence is that, for any highly sensitive function, any interpolating minima will be very sharp. This property is nontrivial, as seen from the fact that it disappears when adding a scratchpad (see below).

Given a parameter vector θ defining the transformer T_θ, the *average direction sharpness* is (e.g. Wen et al., 2022; Jiang et al., 2020; Foret et al., 2020)

$$L_{\rho,n}(T) := \mathbb{E}_{x\in\{\pm 1\}^n}\mathbb{E}_{\|\Delta\|_2=\rho}(T_{\theta+\Delta}(x) - T_\theta(x))^2 \quad (15)$$

where the second expectation is over the radius-ρ sphere in the space of parameter vectors. $L_{\rho,n}$ quantifies the change in T_θ when perturbing θ by a size-ρ vector in a random direction. Lower bounds on $L_{\rho,n}$ (Theorem 6) immediately entail lower bounds on other common sharpness measures, such as worst-case sharpness, and (in the limit $\rho \to 0$) the trace of the loss function's Hessian at θ.

In defining the parameter vector θ in (15), we exclude positional encodings, both because they are often frozen, and because their number depends on n, hindering fair comparison across n.

Our next result lower-bounds $L_{\rho,n}$ in terms of the average sensitivity, provided the transformer is sufficiently wide in relation to its depth:

Theorem 6 (Second Main Result)**.** *Let T_θ be a transformer where $d > 12L$. Assume $T_\theta(x) \in [-1,1]$ for each x. Then:*

$$\lim_{\rho\to 0}\liminf_{n\to\infty} L_{\rho,n}(T) \geq \liminf_{n\to\infty}\frac{as_n(T_\theta)}{2n} - L\exp(-\Omega(d)) \quad (16)$$

Here, "$\Omega(d)$" scales positively with d. If T_θ has Boolean outputs ($T_\theta(x) \in \{\pm 1\}$), then the factor "2" can be eliminated from (16).

Informally, this theorem says that, when $as_n(T_\theta) \sim n$, then even tiny perturbations to the parameters will, in expectation, lead to a substantial change in predictions on long inputs. This means that, as the input gets longer, the Hessian of the mean-squared loss at the minimizer fitting a sensitive function has unboundedly large entries.

See Appendix C for the proof. The key idea of the proof is that, if T_θ is very sensitive in input space, small perturbations to the parameters usually lead to a large drop in sensitivity when n is large, because they lead to large changes in the layer-norm induced blowup that is needed to represent high-sensitivity functions. As a consequence, transformers computing sensitive functions are isolated in parameter space.

The theorem applies when d is substantially larger than L, as is indeed true of typical transformers (e.g., LLaMa 7B has $d = 4096$ and $L = 32$). The convergence of the limits is slower when the parameter-norms, as summarized by C, are larger, as larger parameters can increase sensitivity. However, remarkably, for any fixed transformer, C becomes irrelevant for $L_{\rho,n}$ in the limit where $n \to \infty$ (16).

While we stated Theorem 6 for an individual transformer, the same statement holds for families of transformers where weights may depend on n, as long as C remains bounded. A consequence is that scaling attention logits with $\log n$ (used to represent sparse functions by Edelman et al. (2022); Chiang and Cholak (2022)) will, at least in the input layer, not mitigate the difficulty of sensitive functions.

6 Implications

We discuss how Theorem 6 unifies a range of diverse empirical findings about the behavior of transformers.

Difficulty of PARITY For PARITY, $L_{\rho,n}$ converges to 1 for large d, showing that arbitrarily small perturbations to a transformer computing PARITY will lead to a high loss for sufficiently long inputs. Previously, Chiang and Cholak (2022) noted that, for their hand-constructed transformer representing PARITY, small changes to a specific parameter led to a large increase in loss, suggesting that this made the solution impossible to reach with SGD. Theorem 6 shows that this phenomenon is unavoidable for any transformer representing a high-sensitivity function. For functions with sublinear average sensitivity, Theorem 6 entails no nontrivial lower bound on sharpness, and no such phenomenon is predicted.

Random Initialization A key step in Theorem 6 is to show that $s(T_{\theta+\Delta}, x)$ is bounded with very high probability over the choice of Δ (Appendix, Eq. (70)); this immediately entails that high-sensitivity transformers can only inhabit a small volume in parameter space. This explains why randomly initialized transformers empirically show low average sensitivity, more so than recurrent networks (Bhattamishra et al., 2023).

Length Generalization An important corollary is that, for a sensitive function, length generalization requires exact match to the minimum: the slightest deviation from the exact minimum will, in expectation, lead to failure when inputs get sufficient long, even if the minimum itself represents a length-generalizing solution. This provides, for the first time, a rigorous explanation why, despite the in-principle existence of length-generalizing transformers, transformers struggle with length generalization for PARITY (e.g. Bhattamishra et al., 2020).

Generalization on Unseen Data Another corollary is that, in expectation, the training loss landscape around an interpolating minimum places a constraint on a function's overall sensitivity, *even if not a single pair of Hamming neighbors* are in the training set. This is because (16) remains true, on average across randomly selected training sets, if replacing the expectation over the full input space with an expectation over the training set in (15). This means that, for long inputs, flat minima of the training loss generalize with bounded sensitivity. To the extent that gradient-based training tends to find flatter minima (e.g. Damian et al., 2023), this provides a theoretical justification for the empirical result that transformers' generalization behavior, when trained on a Boolean function on some training subset of $\{\pm 1\}^n$, shows a strong bias towards low average sensitivity (Bhattamishra et al., 2023).

Abbe et al. (2023) proposed a *min-degree* bias, that is, a generalization bias towards functions that are linear combinations of functions that each depend on only a few inputs. Any function $f : \{\pm 1\}^n \to \mathbb{R}$ can be uniquely written as a linear combination of the multilinear monomials $\chi_P(x) := \prod_{i\in P} x_i$, where $P \subseteq \{1,\dots,n\}$: $f(x) = \sum_P \lambda_P \chi_P(x)$. The coefficients λ_P define the Fourier-Walsh transform of f. For χ_P, both its degree as a polynomial, and its average sensitivity, are $|P|$. The degree profile, as defined by Abbe et al. (2023), is the tuple $(d_1, ..., d_n)$ where $d_i = \sum_{P:|P|=i} |\lambda_P|^2$. Minimization of degree profile then refers to setting as many of the later entries to zero, and minimizing the size of the last nonzero entry. Abbe et al. (2023) proved inductive biases towards functions with a low degree profile for a random features model and diagonal networks. Evidence in the case of transformers was limited to empirical data on three example functions. Our results entail prove an bound on degree profiles for the full transformer architecture, because average sensitivity is a summary statistic of the degree profile (O'Donnell, 2014):

$$as_n(f) = \sum_{P\subseteq\{1,\dots,n\}} \lambda_P^2 |P| = \sum_{i=0}^{n} i \cdot d_i \qquad (17)$$

Hence, functions with degree profile assigning substantial weight to degrees of order $\sim n$ are brittle in parameter space, corresponding to very sharp minima.

Intermediate Steps Reduce Sensitivity PARITY can be solved well with a scratchpad (Anil et al., 2022; Liu et al., 2023). Existing theoretical accounts of the benefit of intermediate steps for transformers' expressive capacity (e.g. Merrill and Sabharwal, 2023a; Feng et al., 2023) do not account for the benefit of intermediate steps for PARITY-like problems: While the theoretical models of transformer expressivenes used in these studies do predict versions with intermediate steps to be easy, they do not predict that computing PARITY in a single step would be hard, due to Fact 1. The concept of average sensitivity provides a simple explanation for the benefit of intermediate steps. Formally, we can consider the problem of simulating a finite automaton with state set $\mathcal{X}$ either translating to the final state t_n in one go (standard), or to autoregressively translate it into a sequence of states $t_1, ..., t_n$ (scratchpad). Then (proof in Appendix D):

Theorem 7. *Simulating an automaton with scratchpad has sensitivity $O(1)$ for each autoregressive step.*

7 Experiments

7.1 Setup

We conducted experiments to test the predictions made by our theory, specifically assessing predictions regarding loss landscapes and sharpness of the minima. In all experiments, we use the transformer encoder architecture, using the default implementation in PyTorch (Paszke et al., 2019). Each model is trained to fit a function f on a specific sequence length n. Each training input x is generated uniformly from $\{\pm 1\}^n$, and each input bit, treated as a separate token, is embedded using learned token and positional encodings.[3]

The representation of the last token is passed to a linear layer to generate the prediction $T_\theta(x)$. Parameters θ of the transformer are optimized for MSE loss between $T_\theta(x)$ and $f(x)$ using AdamW (Loshchilov and Hutter, 2017). For full details on hyperparameters and training setup, refer to Appendix E.1.

In implementation, we assumed versions of the functions outputting to $\{0, 1\}$; we rescaled sharpness values accordingly for comparability with the theory.

We analyzed models using the following metrics:

1. **Parameter Norm.** We compute the L2 norm of the entire model's parameter vector, excluding positional encoding matrices. We discard the norm of positional encodings, so that the norms of the models trained for different sequence lengths are comparable.

2. **LayerNorm Blowup.** This metric is computed by computing the maximum normalization factor (5) across the entire layer in each application of layer norm, and take the product over all applications of layer norm. This essentially corresponds to Blowup.[4]

3. **Sharpness.** In order to avoid committing to any specific ρ, we sample Δ in (15) not from the radius-ρ-sphere, but from a mean-zero Gaussian with STD $\rho = 0.02$. We estimate using $N_{s,p}$ perturbations and $N_{s,b}$ input strings x. This provides

[3] Code is available at `https://github.com/lacoco-lab/sensitivity-hardness`.

[4] In the theory, Blowup has an additional 1+... in each factor for technical reasons. This difference is immaterial, as we are interested in situations where $Blowup = \omega(1)$.

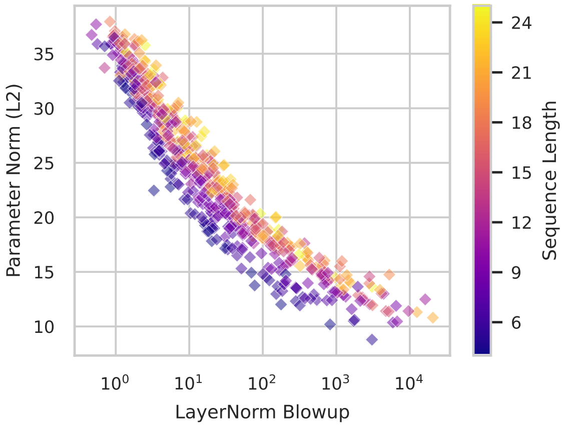


Figure 2: The tradeoff between parameter norm of Transformers trained to approximate PARITY and the blowup of their Layer Normalization layers. The tradeoff depends on the input length; blowup or parameter weights need to increase with the input length (in accordance with Corollary 5). This length dependency is not observed with low sensitivity functions (Appendix, Figures 9 and 10).

results equivalent to $L_{\rho,n}$ in the $\rho \to 0$ asymptotic, while avoiding committing to any specific ρ.

As we are interested in properties of models that compute given functions, runs that did not converge (evaluation MSE higher than 10^{-3}) were discarded.

7.2 Results

Higher Sensitivity Implies Sharper Minima. In this experiment, we train transformers to fit $f \in \{\text{PARITY}, \text{MAJORITY}, \text{FIRST}, \text{MEAN}\}$ on sequence lengths from 4 to 30. For each function and sequence length, we retrain the model 10 times from different random initializations.

For PARITY, sharpness stably increases with the input length (Figure 1). For the other functions, whose sensitivity grows more slowly with n, (a) the absolute value of sharpness is orders of magnitude lower than for PARITY; (b) there is little increase with n. More results are shown in Appendix E.2.

Tradeoff between Weight Norm and Layer Norm Blowup. At any fixed input length n, high sensitivity can be achieved by a combination of large weights and a large LayerNorm blowup. By Theorem 5, the product of C and squared blowup is bounded from below with some value $B_n(f)$. Hence, the product of $\sqrt{C}$ and blowup is bounded with $\sqrt{B_n(f)}$, and the sum of $\frac{1}{2}\log C$ and $\log$ Blowup is bounded with $\frac{1}{2}\log B_n(f)$. However, C depends exponentially on parameters, and thus we expect the parameter norm to trade off with the logarithm of the layer norm blowup. Moreover, for PARITY the value of $B_n(f)$ increases with n, and therefore sum of parameter norms and the logarithm of the blowup should also increase with n.

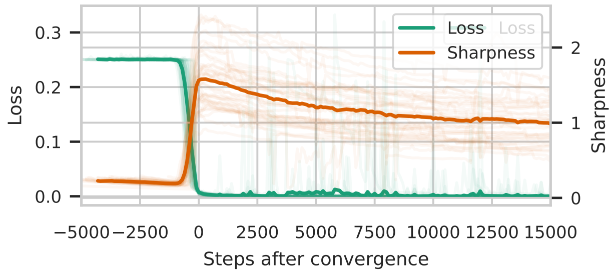


Figure 3: During training a transformer on PARITY, a sudden drop in the loss coincides with an increase in sharpness. Sharpness decreases again in further training, but asymptotes to a nontrivial value (Appendix, Figure 13). See corresponding curves for weight norm and LN Blowup in Appendix, Figure 12.

To test this prediction, for each function f we train a set of models with varying sequence lengths n, weight decay and learning rate parameters. It allows us to obtain datapoints with diverse values of LN Blowup and parameter norm.

Results for PARITY can be seen in Figure 2, and for other functions in Figure 9. For all functions, there is a clear log Blowup-Parameter Norm tradeoff. For PARITY, the shape of the tradeoff indeed depends on n, with transformers trained for high n located above others in the log Blowup-Parameter Norm coordinates. For other functions, dependency on n is not visible, at least at this range of n.

The Limits of Transformers' Generalization. As discussed above, Theorem 6 predicts that transformers will generalize with low sensitivity, as low-sensitivity functions will tend to have flatter minima.

To test this prediction, we created random functions $f := \{\pm 1\}^n \to \{\pm 1\}$, and sampled random training sets from $\{\pm 1\}^n$ of size 128/256/512. We fixed $n = 10$. For each f, we train a transformer T_1 on the training set and use it to label the whole input space of sequences of length n (1024 objects in our case). Replicating Bhattamishra et al. (2023), these extrapolated functions T_1 have lower average sensitivity than the original functions f, indicating a low-sensitivity bias in generalization (Figure 4). Now, in order to directly compare the sharpness of minima corresponding to the true function f and the extrapolated function T_1, we trained new transformers to fit both functions on the entire dataset $\{\pm 1\}^n$, and measured the sharpness for these two new transformers. The results were averaged over 10 random functions f, 10 training sets per f and training set size, and 5 new transformers per each T_1. Sharpness was indeed lower for the transformer fitting the extrapolated functions than for the transformer matching the original random functions f. This also held when measuring sharpness only on the training set (Appendix, Figure 15).

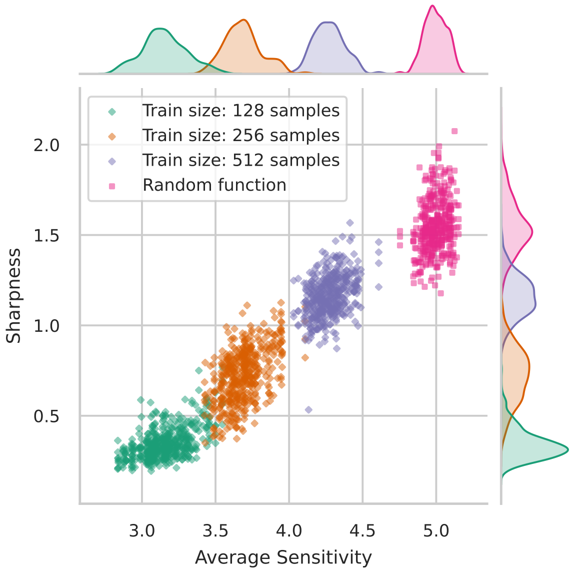


Figure 4: Generalization: When trained on data from a random Boolean function a subset of $\{\pm 1\}^n$ (here: n=10), transformers generalize with reduced sensitivity compared to the actual function. The solutions found have lower sharpness than a solution fitting the actual function. When the training size is smaller, the inferred function is less constrained, and learnt functions have even lower sensitivity.

Scratchpad Eliminates Sharpness. By Theorem 7, sensitivity of each autoregressive step when computing PARITY with scratchpad is $O(1)$. Hence, Theorem 6 provides no nontrivial lower bound for $L_{\rho,n}(T)$. We trained an Encoder-Decoder Transformer, predicting PARITY of i-th substring on i-th autoregressive step: $t_i = \text{PARITY}(x_{1:i}) = x_i \oplus t_{i-1}$ $(t_0 = 0)$. The visual dependency between sharpness and length of input for PARITY with a scratchpad is shown in Figure 11. Even for length around 300, sharpness is low and there is little increase with input length. Thus, decrease in sensitivity due to the scratchpad can explain why prior work (Anil et al., 2022) found that PARITY is easy for Transformers with scratchpad.

LayerNorm Blowup Enables Learning Figure 3 represents the evolution of loss and sharpness of Transformer models trained for PARITY with input length 25. The results are averaged across 39 converged runs. Similar curves for parameter norm and LayerNorm blowup are presented in Appendix 12.

A dramatic increase in sharpness occurs at exactly the time when loss falls to 0. This suggests the presence of a steep minimum in the loss landscape, and fitting PARITY requires the optimization procedure to find this minimum. Figure 12 (Appendix) shows further details of this process. During learning, there is a small but sharp increase in the blowup which makes the high sensitivity of the model possible, increasing the left-hand side of the inequality in Corollary 5. Following that, non-zero weight decay drives the parameter norm down, which – by the theoretically predicted tradeoff between blowup and parameter norm – is accompanied by an exponential increase in blowup.

8 Discussion

We have provided a rigorous theoretical explanation of the inductive bias towards low sensitivity observed in empirical research on transformers. Theorem 6 describes a fundamental inductive bias of the transformer architecture: for long inputs, fitting sensitive functions is only possible in sharp minima of the loss. This holds without assumptions often made in previous work about expressive capacity, such as non-infinite precision (e.g. Merrill and Sabharwal, 2023b; Angluin et al., 2023), hard attention (e.g. Hahn, 2020), or Lipschitz-continuous variants of layer norm (Edelman et al., 2022). We speculate that Theorem 6 reflects a fundamental limitation of parallelized differentiable computing with bounded depth. Our results show that it is overcome by scaling the number of computation steps with the input length. While we focused on functions outputting a single label; an interesting direction for future research is the extension of this theory to sequence-to-sequence transductions, for which some empirical data has been reported (e.g. Zhou et al., 2023), but theoretical understanding remains wide open.

Due to the relation between average sensitivity and Fourier analysis on the Boolean cube (Equation 17), a low-sensitivity bias can be viewed as a sequence modeling analogue of a spectral bias towards low frequencies observed in other neural architectures (e.g. Rahaman et al., 2019; Fridovich-Keil et al., 2022).

We note that, while our results show that sensitive transformers are very brittle, these results do not by themselves have implications for *real-world generalization*, as a low-sensitivity bias need not always be beneficial in real-world setups. Indeed, the relationship between sharpness and real-world generalization is not straightforward (e.g. Andriushchenko et al., 2023; Kaur et al., 2023). Our theory suggests that transformers generalize well to the extent that real-world data has bounded sensitivity (e.g. Hahn et al., 2021).

9 Conclusion

We have proven that, under the transformer architecture, high sensitivity in input space can only be achieved in very sharp minima. Empirical results confirm the predictions of the theory. Taken together, our results explain a diverse set of empirical observations about transformers not explained by previous theoretical work. They suggest shifting theoretical research from in-principle expressiveness considerations to studying quantitative bounds and the shape of the loss landscape in order to understand the abilities of transformers.

Limitations

A limitation of our results is that the theoretical results are asymptotic, providing statements about the limit of very long input sequences. Providing more quantitative bounds that tightly characterize finite-length behavior is an interesting problem for future research.

A second limitation is that we only target functions outputting a single value. Operationalizing sensitivity-like metrics for sequence-to-sequence functions may be required in order to expand the theory to such functions.

Third, our results apply to transformer encoders. It remains open if transformer decoders, with causal attention masking, face a different set of limitations than we have shown here in the absence of masking.

Fourth, our theoretical results concern the loss landscape, not the training dynamics itself. Further technical advances may be needed to directly prove corresponding results for training dynamics.

Acknowledgments

We thank Lena Strobl, Dana Angluin, and David Chiang for useful discussion. We thank David Chiang, Paul Lintilhac, and Yuval Pinter for spotting various errors in a previous version, and the anonymous ARR reviewers for useful feedback.

References

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, and Kevin Rizk. 2023. Generalization on the unseen, logic reasoning and degree curriculum. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31–60. PMLR.

Kwangjun Ahn, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie, and Suvrit Sra. 2023. Linear attention is (maybe) all you need (to understand transformer optimization).

Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. 2023. A modern look at the relationship between sharpness and generalization. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 840–902. PMLR.

Dana Angluin, David Chiang, and Andy Yang. 2023. Masked hard-attention transformers and boolean rasp recognize exactly the star-free languages. *arXiv preprint arXiv:2310.13897*.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, 35:38546–38556.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the ability and limitations of transformers to recognize formal languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7096–7116. Association for Computational Linguistics.

Satwik Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. 2023. Simplicity bias in transformers and their ability to learn sparse boolean functions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5767–5791. Association for Computational Linguistics.

David Chiang and Peter Cholak. 2022. Overcoming a theoretical limitation of self-attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7654–7664. Association for Computational Linguistics.

David Chiang, Peter Cholak, and Anand Pillay. 2023. Tighter bounds on the expressivity of transformer encoders. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 5544–5562. PMLR.

Alex Damian, Eshaan Nichani, and Jason D. Lee. 2023. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Ronald De Wolf. 2008. A brief introduction to fourier analysis on the boolean cube. *Theory of Computing*, pages 1–20.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. 2023. Neural networks and the chomsky hierarchy.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. 2022. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pages 5793–5831. PMLR.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2023. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2020. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*.

Sara Fridovich-Keil, Raphael Gontijo Lopes, and Rebecca Roelofs. 2022. Spectral bias in practice: The role of function frequency in generalization. In *Advances in Neural Information Processing Systems*, volume 35, pages 7368–7382. Curran Associates, Inc.

Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171.

Michael Hahn, Dan Jurafsky, and Richard Futrell. 2021. Sensitivity as a complexity measure for sequence classification tasks. *Transactions of the Association for Computational Linguistics*, 9:891–908.

Yiding Hao, Dana Angluin, and Robert Frank. 2022. Formal language recognition by hard attention transformers: Perspectives from circuit complexity. *Transactions of the Association for Computational Linguistics*, 10:800–810.

Johan Håstad. 1986. *Computational limitations for small depth circuits*. Ph.D. thesis, Massachusetts Institute of Technology.

Pooya Hatami, Raghav Kulkarni, and Denis Pankratov. 2010. Variations on the sensitivity conjecture. *Theory of Computing*, 4:1–27.

Yiding Jiang, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. 2020. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*.

Stasys Jukna. 2012. *Boolean Function Complexity: Advances and Frontiers*.

J. Kahn, G. Kalai, and N. Linial. 1988. The influence of variables on boolean functions. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 68–80.

Simran Kaur, Jeremy Cohen, and Zachary Chase Lipton. 2023. On the maximum hessian eigenvalue and generalization. In *Proceedings on*, pages 51–65. PMLR.

Shengqiao Li. 2010. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics & Statistics*, 4(1):66–70.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and stability in in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19565–19594. PMLR.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2023. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

William Merrill and Ashish Sabharwal. 2023a. The expressive power of transformers with chain of thought. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*.

William Merrill and Ashish Sabharwal. 2023b. A logic for expressing log-precision transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*.

William Merrill and Ashish Sabharwal. 2023c. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545.

William Merrill, Ashish Sabharwal, and Noah A. Smith. 2022. Saturated transformers are constant-depth threshold circuits. *Trans. Assoc. Comput. Linguistics*, 10:843–856.

Ryan O'Donnell. 2014. *Analysis of Boolean Functions*. Cambridge University Press.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. 2019. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR.

Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. 2023. Randomized positional encodings boost length generalization of transformers.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. 2023. Representational strengths and limitations of transformers. *CoRR*, abs/2306.02896.

Lena Strobl. 2023. Average-hard attention transformers are constant-depth uniform threshold circuits. *CoRR*, abs/2308.03212.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. 2023. Transformers as recognizers of formal languages: A survey on expressivity. *CoRR*, abs/2311.00208.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. 2022. On layer normalizations and residual connections in transformers. *arXiv preprint arXiv:2206.00330*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2021. Thinking like transformers. In *International Conference on Machine Learning*, pages 11080–11090. PMLR.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. 2022. How does sharpness-aware minimization minimize sharpness? *CoRR*, abs/2211.05729.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. 2021. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2019. Are transformers universal approximators of sequence-to-sequence functions?

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. 2023. What algorithms can transformers learn? a study in length generalization. *arXiv preprint arXiv:2310.16028*.

A Simple Constructions for PARITY

While it is well-known that PARITY can be expressed using uniform circuits with majority gates (Håstad, 1986), it may not be obvious just how simple the construction can be, and how it can easily embed into formalisms like RASP. In order to highlight the existence of very simple constructions for PARITY in various formalisms, and to make discussion here self-contained, we provide a simple proof. The key is the following insight:

Lemma 8 (Folklore)**.** *A bit string is odd if and only if (†) for the (strict) majority of positions i holding a 1, it holds that strictly more than 1/2 of 1s appear no later than at i.*

This fact appears to be well known in the circuit complexity community, though we are not aware of the original published reference. We provide a proof for self-containedness.

Proof. As only positions holding a one play a role, it is sufficient to consider strings of the form $x \in 1^*$. First, consider the case where $x = 1^{2n}$:

Index	1	2	...	n	n+1	...	2n
A: # of 1s no later than at i	1	2	...	n	n+1	...	2n
B: # of 1s later than at i	2n-1	2n-2	...	n	n-1	...	0
A>B?	no	no	...	no	yes	...	yes

Here, $A > B$ holds at exactly n positions, less than the strict majority of positions. On the other hand, when $x = 1^{2n+1}$:

Index	1	2	...	n	n+1	...	2n	2n+1
A: # of 1s no later than at i	1	2	...	n	n+1	...	2n	2n+1
B: # of 1s later than at i	2n	2n-1	...	n+1	n	...	1	0
A>B?	no	no	...	no	yes	...	yes	yes

Here, $A > B$ holds at exactly $n+1$ positions, a strict majority of positions. □

Proof of Fact 1. The property (†) from Lemma 8 is straightforwardly formalized in RASP. To formalize it in FO[M] (Merrill and Sabharwal, 2023b), we note that FO[M] permits defining constructs of the form "for the majority of positions x satisfying $\phi(x)$, it holds that $\psi(x)$", for any formula ψ. FO[M] formulas have a known translation to (highly uniform) majority circuits. This concludes the proof of Fact 1.

□

Lemma 8 shows that, in terms of expressive capacity, PARITY is tightly linked to MAJORITY. While the components of (†), MAJORITY and position comparison, are easily learned by transformers, the composite function is hard to learn. This observation suggests that function classes satisfying typical closure properties, as satisfied by typical logic and circuit classes, cannot model which functions transformers learn easily. On the other hand, average sensitivity elegantly explains this difference: The expression (†) uses two nested "majority" operations, each introducing an average sensitivity of $\sqrt{n}$, multiplying to the asymptotically larger sensitivity n of the composite function.

This result does not exclude a role of minimizing description length in formalisms such as RASP as a factor impacting transformers' inductive biases (as suggested empirically by Zhou et al. (2023)), but suggests that description length and expressive capacity might be unable to account for important aspects of transformers' inductive biases, including the low-sensitivity bias and difficulty learning PARITY.

B Proof of Theorem 4 and Corollary 5

For a bitstring $x \in \{\pm 1\}^n$, we write $\mathrm{I}_i[f](x)$ to denote the *absolute influence* of the i-th bit:

$$\mathrm{I}_i[f](x) := \left\|f(x) - f(x^{\oplus i})\right\|_2 = \sqrt{\sum_s |f(x)_s - f(x^{\oplus i})_s|^2} \tag{18}$$

which simplifies to the absolute value $|f(x) - f(x^{\oplus i})|$ if $f(x) \in \mathbb{R}$. We will suppress the argument x where it is contextually given. When f outputs a scalar, then we have by (6):

$$s(f,x) = \frac{1}{4}\sum_{i=1}^{n} \mathrm{I}_i[f](x)^2 \tag{19}$$

We begin by noting

$$\|f(x) - f(x^{\oplus q})\|_2^2 \leq \|f(x) - f(x^{\oplus q})\|_2 \cdot \max_x \|f(x) - f(x^{\oplus q})\|_2 \leq 2\max_x \|f(x)\|_2 \cdot \|f(x) - f(x^{\oplus q})\|_2 \tag{20}$$

and hence

$$\frac{1}{2}\mathrm{I}_q[f]^2(x) = \frac{1}{2}\|f(x) - f(x^{\oplus q})\|_2^2 \leq (\max_x \|f(x)\|_2)\|f(x) - f(x^{\oplus q})\|_2 = (\max_{x'} \|f(x')\|_2)\,\mathrm{I}_q[f](x) \tag{21}$$

B.1 Bounding the Sensitivity of the Attention Heads

The first step will be bound the sensitivity of the outputs of individual attention heads. At a high level, such bounds are also part of the existing pointwise Lipschitzness bounds (Hahn, 2020; Edelman et al., 2022; Li et al., 2023). Our key innovation in this part of the proof is to replace a *pointwise* bound with a *high-probability* bound that, at the input layer, is independent of the key-query matrix. The key novel part is Lemma 11. In order to make the proof self-contained, we also include proofs for the other steps.

We will refer to the following bounds, for suitably defined constants $1 \leq C_V, C_P, C_A^{(k)}, C_{MLP}, L_{f^{MLP}}, C^{(k)}, C < \infty$:

$$\|V_{k,h}\|_2 \leq C_V \tag{22}$$

$$\left(\max_w \|y_w^{(k)}\|_2\right) \leq \sqrt{d} \text{ when } k \geq 1 \tag{23}$$

$$\left(\max_w \|y_w^{(0)}\|_2\right) \leq C_P \tag{24}$$

$$a_{ij}^{(k,h)} \leq C_A^{(k)} := d \cdot \max_h \|K_{k,h}^T Q_{k,h}\|_{spectral} \qquad \text{if } k > 1 \tag{25}$$

$$C_{MLP} := \sup_{x:\|x\|_2 \leq (H+1)C_V \min\{C_P,\sqrt{d}\}} \|f^{MLP}(x)\|_2 \tag{26}$$

$$L_{f^{MLP}} := \sup_{\|x\|_2,\|x'\|_2 \leq (H+1)\sqrt{d}} \frac{\|f^{MLP}(x) - f^{MLP}(x')\|_2}{\|x - x'\|_2} \tag{27}$$

$$C^{(0)} := 150 L_{f^{MLP}} C_V^2 C_P H \tag{28}$$

$$C^{(1)} = 2\left(\max_w \|Y_w^{(1)}\|_2\right) \cdot L_{f^{MLP}} \leq 2C_{MLP}(H+1)C_V C_P \cdot L_{f^{MLP}} \tag{29}$$

$$C^{(k)} := 12 L_{f^{MLP}}(H+1)\|V_{k,h}\|_1 4\sqrt{d}\exp(4C_A^{(k)}) \qquad (\text{for } k > 1) \tag{30}$$

$$\|Y_j^{(1)}\|_2 \leq^{(2)} C_{MLP}(H+1)C_V C_P \tag{31}$$

$$C := 25\sqrt{d}H(1 + \|v_{out}\|)\left(\prod_{l=0}^{k} C^{(k)}\right) \tag{32}$$

We first bound the influence of any bit on an attention head's activation in terms of its influence on the attention values and the activations at the preceding layer:

Lemma 9 (Sensitivity of an Attention Head)**.** *We have*

$$\mathrm{I}_q[b_{j,h}^{(k)}] \leq C_V \sum_{w=1}^{n} \hat{a}_{j,w}^{k,h}\,\mathrm{I}_q[y_w^{(k-1)}] + C_V\left(\max_w \|y_w^{(k-1)}\|_2\right)\sum_{w=1}^{n} \mathrm{I}_q[\hat{a}_{j,w}^{k,h}] \tag{33}$$

The main task will be to bound $\sum_{w=1}^{n} \mathrm{I}_q[\hat{a}_{j,w}^{k,h}]$. We separately bound this for $k = 1$ and $k > 1$.

Remark 10. *We note that the conceptually similar Lemma 4.3 in Edelman et al. (2022) might suggest a seemingly stronger bound where $\sum_w \mathrm{I}_q[\hat{a}_{j,w}^{k,h}]$ is replaced (using their Lemma A.6) by the* maximum *influence on the* logits*: $\max_{w=1,\dots,N}\mathrm{I}_q[a_{j,w}^{k,h}]$ using the fact that softmax has a Lipschitz constant ≤ 1. However, $s(f,x)$ requires summing over q (not relevant to Edelman et al. (2022)[5]), and the expression $\sum_q \max_{w=1,\dots,N}\mathrm{I}_q[a_{j,w}^{k,h}]$ can easily be $\Theta(n)$, for instance, if $a_{j,w}^{1,h} \equiv 1_{x_w=1}$. Importantly, at the lowest layer, we find sublinear bounds for $\sum_q\sum_w \mathrm{I}_q[\widehat{a}_{j,w}^{k,h}]$ that, at $k = 1$, involve no bound on the logits $a_{j,w}^{k,h}$ at all.*

Proof of Lemma 9. Using the definition of b:

$$b_{i,h}^{(k)} = \sum_{j=1}^{n} \hat{a}_{i,j}^{(k,h)} V_{k,h} y_j^{(k-1)} \tag{34}$$

[5] They considered Lipschitzness in the parameter space, with parameters that grow at most sublinearly with n. In contrast, we investigate sensitivity in the input space.

and the triangle inequality, we find:

$$
\begin{aligned}
\mathrm{I}_q[b_{j,h}^{(k)}] =& \left\| \sum_{j=1}^{n} \hat{a}_{i,j}^{(k,h)}(x) \cdot V_{k,h} \cdot y_j^{(k-1)}(x) - \sum_{j=1}^{n} \hat{a}_{i,j}^{(k,h)}(x^{\oplus q}) \cdot V_{k,h} \cdot y_j^{(k-1)}(x^{\oplus q}) \right\|_2 \\
\leq& \sum_{j=1}^{n} \left\| \hat{a}_{i,j}^{(k,h)}(x) V_{k,h} y_j^{(k-1)}(x) - \hat{a}_{i,j}^{(k,h)}(x^{\oplus q}) V_{k,h} y_j^{(k-1)}(x^{\oplus q}) \right\|_2 \\
\leq& \sum_{j=1}^{n} |\hat{a}_{i,j}^{(k,h)}(x)| \left\| V_{k,h} y_j^{(k-1)}(x) - V_{k,h} y_j^{(k-1)}(x^{\oplus q}) \right\|_2 + \left| \hat{a}_{i,j}^{(k,h)}(x) - \hat{a}_{i,j}^{(k,h)}(x^{\oplus q}) \right| \left\| V_{k,h} y_j^{(k-1)}(x) \right\|_2 \\
\leq& C_V \sum_{w=1}^{n} \hat{a}_{j,w}^{k,h} \mathrm{I}_q[y_w^{(k-1)}] + C_V \sum_{w=1}^{n} \|y_w^{(k-1)}\|_2 \, \mathrm{I}_q[\hat{a}_{j,w}^{k,h}] \\
\leq& C_V \sum_{w=1}^{n} \hat{a}_{j,w}^{k,h} \mathrm{I}_q[y_w^{(k-1)}] + C_V \left(\max_w \|y_w^{(k-1)}\|_2 \right) \sum_{w=1}^{n} \mathrm{I}_q[\hat{a}_{j,w}^{k,h}]
\end{aligned}
$$

□

Bounding the Sensitivity of Attention (First Layer) First, we bound the sensitivity of the attention distribution in the first layer. Here, it will be crucial to provide high-probability bounds independent of $\|K_{1,h}^T Q_{1,h}\|$, unlike the pointwise Lipschitzness bounds of Hahn (2020); Li et al. (2023). The resulting Lemma 11 is one of the key innovations compared to the pointwise Lipschitzness bounds from prior work, and the main technical innovation of Section B.1.

Lemma 11. *Let $\delta > 0$. With probability $\geq 1 - \frac{H}{n^{\delta-2}}$ over the choice of x, the following holds: For each $i = 1, \dots, n$, $h = 1, \dots, H$,*

$$
\sum_{q=1}^{n} \sum_{j=1}^{n} \mathrm{I}_q[\widehat{a}_{i,j}^{(1,h)}] \leq (8 + 32\delta) \log n + 10 \tag{35}
$$

Proof of Lemma 11. Let x_i denote the token at position i (-1 or 1), and $e(x_i)$ the corresponding word embedding.

Throughout, we will suppress the index $h = 1, \dots, H$ for notational simplicity.

If $e(x_i)$ is the word embedding for the token x_i and p_i is the i-th positional embedding, then:

$$
\begin{aligned}
a_{ij}^{(1)} &= \left(p_i^T + e(x_i)^T\right) \cdot Q^T K \cdot (p_j + e(x_j)) \\
&= \begin{pmatrix} p_i^T & e(x_i)^T \end{pmatrix} \begin{pmatrix} I_d & I_d \end{pmatrix}^T Q^T K \begin{pmatrix} I_d & I_d \end{pmatrix} \begin{pmatrix} p_j \\ e(x_j) \end{pmatrix} \\
&= \begin{pmatrix} p_i^T & e(x_i)^T \end{pmatrix} \begin{pmatrix} A & B \\ C & W \end{pmatrix} \begin{pmatrix} p_j \\ e(x_j) \end{pmatrix} \\
&= \underbrace{p_i^T A p_j}_{A^{(ij)}} + \underbrace{p_i^T C}_{C_i} e(x_j) + e(x_i)^T \underbrace{B p_j}_{B_j} + e(x_i)^T W e(x_j)
\end{aligned}
$$

for appropriate matrices A, B, C, W. Note that this decomposition holds for any linear combination of Note that *adding* positional and word embeddings is a special case, where Q and K can each be written as products of two matrices, where the second one has the form $\begin{pmatrix} I_d & I_d \end{pmatrix}$. Then

$$
\begin{aligned}
\widehat{a}_{i,j}^{(1,h)} =& \frac{\exp(q_i(x_i) k_j(x_j))}{\sum_{w=1}^{n} \exp(q_i(x_i) k_w(x_w))} \\
=& \frac{\exp(C_i e(x_j) + e(x_i)^T B_j + e(x_j) W e(x_i) + A^{(ij)})}{\sum_{w=1}^{n} \exp(Q_i e(x_w) + e(x_i)^T B_w + e(x_w) W e(x_i) + A^{(iw)})}
\end{aligned}
$$

We can write, taking $\widehat{x}_i \in \{0, 1\}$ to be the indicator that $x_i = 1$[6]:

$$
\exp(e(x_i)^T B_w + A^{(iw)} + C_i e(x_w) + e(x_i)^T W e(x_w))
$$

[6] I.e., $\widehat{x}_i = 1$ of $x_i = 1$ and else 0.

$$
\begin{aligned}
=&\exp\left(e(x_i)^T B_w + A^{(iw)}\right)\cdot\exp\left(C_i e(x_w) + e(x_i)^T W e(x_w)\right)\\
=&\underbrace{\exp(e(x_i)^T B_w + A^{(iw)} + C_i e(-1) + e(x_i)^T W e(-1))}_{\gamma_{iw}}\underbrace{\exp(C_i e(x_w) + e(x_i)^T W e(x_w) - C_i e(-1) - e(x_i)^T W e(-1))}_{1+\widehat{x}_w\rho_i}
\end{aligned}
$$

where

$$\rho_i = \exp(C_i e(1) + e(x_i)^T W e(1) - C_i e(-1) - e(x_i)^T W e(-1)) - 1$$

Next, we may assume w.l.o.g. that ρ_i is nonnegative, by, if ρ_i otherwise were negative, renaming the input bits across the entire input string for a fixed i. So we have

$$\widehat{a}_{i,j}^{(1,h)} = \frac{\gamma_{ij}(1+\widehat{x}_j\rho_i)}{\sum_w \gamma_{iw}(1+\widehat{x}_w\rho_i)}$$

We split the summation over j and q into three cases: $q \neq i, j$; $q = j, q \neq i$; $q = i$:

$$\sum_{j=1}^{n}\sum_{q=1}^{n} = \sum_{j,q:q\neq i,j} + \sum_{j,q:q=j,q\neq i} + \sum_{j,q:q=i} \tag{36}$$

First, for $q \neq i, j$:

$$
\begin{aligned}
\mathrm{I}_q[\widehat{a}_{i,j}^{(1,h)}] =& \frac{\gamma_{ij}(1+\widehat{x}_j\rho_i)}{\gamma_{iq} + \sum_{w\neq q}\gamma_{iw}(1+\widehat{x}_w\rho_i)} - \frac{\gamma_{ij}(1+\widehat{x}_j\rho_i)}{\gamma_{iq}(1+\rho_i) + \sum_{w\neq q}\gamma_{iw}(1+\widehat{x}_w\rho_i)}\\
=&\frac{\gamma_{ij}(1+\widehat{x}_j\rho_i)\gamma_{iq}\rho_i}{(\gamma_{iq}+\sum_{w\neq q}\gamma_{iw}(1+\widehat{x}_w\rho_i))(\gamma_{iq}(1+\rho_i)+\sum_{w\neq q}\gamma_{iw}(1+\widehat{x}_w\rho_i))}\\
\leq&\frac{\gamma_{ij}(1+\widehat{x}_j\rho_i)\gamma_{iq}(1+\rho_i)}{(\gamma_{iq}+\sum_{w\neq q}\gamma_{iw}(1+\widehat{x}_w\rho_i))(\gamma_{iq}(1+\rho_i)+\sum_{w\neq q}\gamma_{iw}(1+\widehat{x}_w\rho_i))}\\
=&\frac{\gamma_{ij}(1+\widehat{x}_j\rho_i)}{(\gamma_{iq}+\sum_{w\neq q}\gamma_{iw}(1+\widehat{x}_w\rho_i))}\frac{\gamma_{iq}(1+\rho_i)}{(\gamma_{iq}(1+\rho_i)+\sum_{w\neq q}\gamma_{iw}(1+\widehat{x}_w\rho_i))}\\
=&\widehat{a}_{i,j}^{(1,h)}(x:\widehat{x}_q=0)\widehat{a}_{i,q}^{(1,h)}(x:\widehat{x}_q=1)
\end{aligned}
$$

where $\widehat{a}_{i,j}^{(1,h)}(x:\widehat{x}_q=0)$ is $\widehat{a}_{i,j}^{(1,h)}$ evaluated at the input $x = x_1\dots x_{q-1}(-1)x_{q+1}\dots x_n$, and similarly for $\widehat{a}_{i,q}^{(1,h)}(x:\widehat{x}_q=1)$.

We may assume w.l.o.g., by appropriate reordering given a fixed i, that $\gamma_{i1}\geq\gamma_{i2}\geq\gamma_{i3}\geq\dots$. Then,

$$
\begin{aligned}
\sum_q\sum_{j\neq q}\mathrm{I}_q[\widehat{a}_{i,j}^{(1,h)}] \leq& \sum_q\sum_{j\neq q}\widehat{a}_{i,j}^{(1,h)}(x:\widehat{x}_q=0)\widehat{a}_{i,q}^{(1,h)}(x:\widehat{x}_q=1)\\
=&\sum_q\widehat{a}_{i,q}^{(1,h)}(x:\widehat{x}_q=1)\sum_{j\neq q}\widehat{a}_{i,j}^{(1,h)}(x:\widehat{x}_q=0)\\
\leq&\sum_q\widehat{a}_{i,q}^{(1,h)}(x:\widehat{x}_q=1)\\
=&\sum_q\frac{\gamma_{iq}(1+\rho_i)}{\gamma_{iq}(1+\rho_i)+\sum_{w\neq q}\gamma_{iw}(1+\widehat{x}_w\rho_i)}\\
\leq^{(*)}&\sum_{q>16\delta\log n}\frac{\gamma_{iq}(1+\rho_i)}{\gamma_{iq}(1+\rho_i)+\gamma_{iq}\sum_{w<q}(1+\frac{1}{4}\rho_i)} + \sum_{q\leq 16\delta\log n}\frac{\gamma_{iq}(1+\rho_i)}{\gamma_{iq}(1+\rho_i)}\\
\leq&\sum_{q>16\delta\log n}\frac{(1+\rho_i)}{(1+\rho_i)+\sum_{w<q}(1+\frac{1}{4}\rho_i)} + 16\delta\log n\\
\leq&\sum_{q>16\delta\log n}\frac{(1+\rho_i)}{(1+\rho_i)+q(\frac{1+\rho_i}{4})} + 16\delta\log n\\
=&\sum_{q>16\delta\log n}\frac{4}{4+q} + 16\delta\log n\\
\leq&4\sum_{q>0}\frac{1}{q} + 16\delta\log n
\end{aligned}
$$

$$\leq (4+16\delta)\log n + 4$$

where $(*)$ holds with overwhelming probability over the choice of x, say, $1-\frac{1}{n^{\delta}}$ (multiplicative Chernoff bound for binomial random variables), for any individual q, and $1-\frac{1}{n^{\delta-1}}$ for the whole expression by a union bound over all q. We further used $\sum_{i=1}^{n}\frac{1}{i} \leq \log n + 1$.

Second, consider the case where $q = j$, $q \neq i$:

$$\begin{aligned}\sum_{q=j;q\neq i} \mathrm{I}_j[\widehat{a}_{i,j}^{(1,h)}] &= \sum_{q=j;q\neq i;q>16\delta\log n} \mathrm{I}_j[\widehat{a}_{i,j}^{(1,h)}] + \sum_{q=j;q\neq i;q\leq 16\delta\log n} \mathrm{I}_j[\widehat{a}_{i,j}^{(1,h)}] \\ &\leq \sum_{q=j;q\neq i;q>16\delta\log n} \mathrm{I}_j[\widehat{a}_{i,j}^{(1,h)}] + 16\delta\log n\end{aligned}$$

Then:

$$\begin{aligned}\sum_{q=j;q\neq i;q>16\delta\log n} \mathrm{I}_j[\widehat{a}_{i,j}^{(1,h)}] &= \sum_{q=j;q\neq i} \frac{\gamma_{ij}(1+1\rho_i)}{\gamma_{ij}(1+\rho_i)+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i)} - \frac{\gamma_{ij}}{\gamma_{ij}+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i)} \\ &= \sum_{q=j;q\neq i;q>16\delta\log n} \frac{\gamma_{ij}\rho_i\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i)}{(\gamma_{ij}+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i))(\gamma_{ij}+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i)+\gamma_{ij}\rho_i+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i))} \\ &\leq \sum_{q=j;q\neq i;q>16\delta\log n} \frac{\gamma_{ij}\rho_i\cdot\left(\gamma_{ij}+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i)+\gamma_{ij}\rho_i+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i)\right)}{(\gamma_{ij}+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i))(\gamma_{ij}+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i)+\gamma_{ij}\rho_i+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i))} \\ &= \sum_{q=j;q\neq i;q>16\delta\log n} \frac{\gamma_{ij}\rho_i}{(\gamma_{ij}+\sum_{w\neq j}\gamma_{iw}(1+\widehat{x}_w\rho_i))} \\ &\leq \sum_{q=j;q\neq i;q>16\delta\log n} \frac{\gamma_{ij}\rho_i}{(\gamma_{ij}+\sum_{w<j}\gamma_{iw}(1+\widehat{x}_w\rho_i))} \\ &\leq \sum_{q=j;q\neq i;q>16\delta\log n} \frac{\gamma_{ij}\rho_i}{(\gamma_{ij}+\gamma_{ij}\sum_{w<j}(1+\widehat{x}_w\rho_i))} \\ &\leq^{(*)} \sum_{q=j;q\neq i} \frac{\gamma_{ij}\rho_i}{(\gamma_{ij}+\gamma_{ij}j\frac{1+\rho_i}{4}))} \\ &= \sum_{q=j;q\neq i} \frac{\rho_i}{(1+j\frac{1+\rho_i}{4}))} \\ &\leq \sum_{q=j;q\neq i} \frac{4\rho_i}{4+j+j\rho_i} \\ &\leq \sum_{q=j;q\neq i} \frac{4\rho_i}{j+j\rho_i} \\ &= \frac{4\rho_i}{1+\rho_i}\sum_{q=j;q\neq i}\frac{1}{j} \\ &\leq 4\log n + 4\end{aligned}$$

where again $(*)$ holds with probability $1-\frac{1}{n^{\delta-1}}$ by a Chernoff bound for each j, and a union bound across all j. Importantly, the third inequality used the fact that $\gamma_{iw} \geq \gamma_{ij}$ when $w < j$. So overall

$$\sum_{q=j;q\neq i} \mathrm{I}_j[\widehat{a}_{i,j}^{(1,h)}] \leq 4\log n + 4 + 16\delta\log n$$

with probability $1-\frac{1}{n^{\delta-1}}$.

Third, for the case $q = i$:

$$\begin{aligned}\sum_{j\neq q} \mathrm{I}_q[\widehat{a}_{q,j}^{(1,h)}] &= \sum_{j\neq q} |\widehat{a}_{q,j}^{(1,h)}(x) - \widehat{a}_{q,j}^{(1,h)}(x^{\oplus q})| \\ &\leq \sum_{j\neq q} |\widehat{a}_{q,j}^{(1,h)}(x)| + \sum_{j\neq q} |\widehat{a}_{q,j}^{(1,h)}(x^{\oplus q})|\end{aligned}$$

$$\leq 2$$

Overall, we get

$$\sum_{q=1}^{n}\sum_{j=1}^{n} \mathrm{I}_q[\widehat{a}_{i,j}^{(1,h)}] \leq (8+32\delta)\log n + 10 \tag{37}$$

for each i,h, with probability $1-\frac{H}{n^{\delta-2}}$, via a union bound. □

Remark 12. *To illustrate why the statement can only hold with high probability, rather than pointwise at every x, we consider a one-layer transformer where the first layer at each position computes the OR function, accomplished by setting*

$$a_{ij}^{(1)} = \begin{cases} \log n & \text{if } x_j = 1 \\ -\log n & \text{if } x_j = 0 \end{cases} \tag{38}$$

That is, any occurrence of a 1 draws attention to it, allowing a subsequent MLP to test whether a 1 appears in the string. Consider the input $x = 0^n$. Here, for any Hamming neighbor $x' = x^{\oplus q}$:

$$\widehat{a}_{i,j}^{(1,h)} = \begin{cases} \frac{n}{n+\frac{n-1}{n}} = \frac{1}{1+1/n-1/n^2} \geq 1-\frac{1}{n} & \text{if } j = q \\ \frac{1/n}{n+\frac{n-1}{n}} = \frac{1}{n^2+n-1} \leq \frac{1}{n^2} & \text{else} \end{cases} \tag{39}$$

Hence,

$$\mathrm{I}_q[\widehat{a}_{i,j}^{(1,h)}] \geq \begin{cases} 1-\frac{2}{n} & \text{if } j = q \\ \frac{1}{n}-\frac{1}{n^2} & \text{else} \end{cases} \tag{40}$$

and

$$\sum_{q=1}^{n}\sum_{j=1}^{n} \mathrm{I}_q[\widehat{a}_{i,j}^{(1,h)}] \geq \sum_{q=1}^{n} \mathrm{I}_q[\widehat{a}_{i,q}^{(1,h)}] \geq n-2 \tag{41}$$

The bound (35) thus cannot hold for all input strings x. However, the lemma shows that it holds for the vast majority of inputs. The intuitive reason is that the vast majority of strings $x \in \{0,1\}^n$ have a much less skewed distribution of ones and zeros. For a string where a substantial number of both ones and zeros appear, the influence of any individual input bit is low; the proof of Lemma 11 makes this idea rigorous using a simple concentration bound argument. This intuition roughly matches the sensitivity properties of the OR function: $s(f_{OR}, 0^n) = n$, but $as_n(f_{OR}) = o(1)$, because for the vast majority of strings, flipping any bit cannot flip the OR output. Lemma 11 generalizes this idea to arbitrary assignments of attention logits, and to the continuous output of the softmax operation, as opposed to discrete Boolean functions such as OR.

Bounding the Sensitivity of Attention: Higher Layers At higher layers, the logits a_{ij} may be highly correlated across j, impeding the use of concentration bounds. In fact, for our purposes here, a pointwise Lipschitzness bound as in Hahn (2020); Li et al. (2023) will be sufficient here (roughly corresponding to Lemmas B.1, B.2 of Li et al. (2023)). While no novelty is claimed for Lemma 13, we include its proof for completeness.

Lemma 13. *For $k > 1$ and $q \in \{1,\dots,n\}$,*

$$\mathrm{I}_q[\widehat{a}_{i,j}^{(k,h)}] \leq 2\exp(4C_A^{(k)}) \left(\mathrm{I}_q[y_i^{(k-1)}] + \frac{1}{n}\sum_{j=1}^{n} \mathrm{I}_q[y_j^{(k-1)}] \right) \tag{42}$$

Remark 14. *Note that this bound depends on $\|K^TQ\|_{spectral}$, unlike the first layer, where we found a bound independent of $\|K^TQ\|_{spectral}$. We do not know whether the exponential dependency on $\|K^TQ\|_{spectral}$ is optimal. However, some dependency on $\|K^TQ\|_{spectral}$ is unavoidable. To see why, consider a transformer where $y_n^{(L)}$ computes a function close to $\frac{PARITY}{n}$ at $\varepsilon = 1$ (Chiang and Cholak, 2022), and $y_1^{(L)}$ computes its negation. Then adding another layer that, by rescaling logits with a factor of n^2, attends to $y_1^{(L)}$ or $y_N^{(n)}$ depending on the input's parity, can compute a function very close to PARITY, if the two activations also provide some disambiguating positional information. This holds even for high-probability or on-average bounds. However, Lemma 11 shows that such dependency can be eliminated at the lowest layer.*

Proof. Fixing k, h, i, we write

$$c_u = \exp\left(a_{iu}^{(k,h)}(x)\right) \leq^{(25)} \exp(C_A^{(k)}) \tag{43}$$

$$d_u = \exp\left(a_{iu}^{(k,h)}(x)\right) - \exp\left(a_{iu}^{(k,h)}(x^{\oplus q})\right) \leq^{(25)} 2\exp(C_A^{(k)}) \tag{44}$$

Then:

$$\begin{aligned}
\left|\widehat{a}_{i,u}^{(k,h)}(x) - \widehat{a}_{i,u}^{(k,h)}(x^{\oplus q})\right| &= \left|\frac{c_u}{\sum_{y=1}^n c_y} - \frac{c_u + d_u}{\sum_{y=1}^n c_y + d_y}\right| \\
&= \left|\frac{c_u(\sum_{y=1}^n c_y + d_y) - (c_u + d_u)\sum_{y=1}^n c_y}{(\sum_{y=1}^n c_y)(\sum_{y=1}^n c_y + d_y)}\right| \\
&= \left|\frac{c_u \sum_{y=1}^n d_y - d_u \sum_{y=1}^n c_y}{(\sum_{y=1}^n c_y)(\sum_{y=1}^n c_y + d_y)}\right| \\
&\leq \frac{c_u \sum_{y=1}^n \mathrm{I}_q[c_y] + \mathrm{I}_q[c_u] \sum_{y=1}^n c_y}{n^2 \exp(-2C_A^{(k)})}
\end{aligned}$$

Now using

$$\begin{aligned}
\mathrm{I}_q[c_u] &= \mathrm{I}_q\left[\exp\left(a_{iu}^{(k,h)}(x)\right)\right] \\
&\leq \exp(C_A^{(k)})\, \mathrm{I}_q\left[a_{iu}^{(k,h)}(x)\right] \\
&\leq \exp(C_A^{(k)}) \|K_{k,h}^T Q_{k,h}\| \left(\mathrm{I}_q\left[y_i^{(k-1)}(x)\right] + \mathrm{I}_q\left[y_u^{(k-1)}(x)\right]\right)
\end{aligned} \tag{45}$$

By $x \leq \exp(x)$, we can bound $\|K_{k,h}^T Q_{k,h}\|_2 \leq \exp(C_A^{(k)})$. Now taking a sum over u yields:

$$\begin{aligned}
\sum_u \left|\widehat{a}_{i,u}^{(k,h)}(x) - \widehat{a}_{i,u}^{(k,h)}(x^{\oplus q})\right| &\leq \sum_{u=1}^n \frac{c_u \sum_{y=1}^n \mathrm{I}_q[c_y] + \mathrm{I}_q[c_u] \sum_{y=1}^n c_y}{n^2 \exp(-2C_A^{(k)})} \\
&\leq^{(45,43)} \exp(2C_A^{(k)}) \frac{2n^2 \mathrm{I}_q\left[y_i^{(k-1)}(x)\right] + 2n \sum_{u=1}^n \mathrm{I}_q\left[y_u^{(k-1)}(x)\right]}{n^2 \exp(-2C_A^{(k-1)})} \\
&\leq 2\exp(4C_A^{(k-1)}) \left(\mathrm{I}_q[y_i^{(k-1)}] + \frac{1}{n} \sum_{j=1}^n \mathrm{I}_q[y_j^{(k-1)}]\right)
\end{aligned}$$

□

Bounding the Sensitivity of the Attention Heads We now put together Lemmas 9,13 in order to bound the sensitivity of any individual attention head's output.

Lemma 15. *For $k > 1$, the following holds for each j and h:*

$$\mathrm{I}_q[b_{j,h}^{(k)}] \leq 4\sqrt{d} \exp(4C_A^{(k)}) \left[\mathrm{I}_q[y_j^{(k-1)}] + \frac{1}{n} \sum_{w=1}^n \mathrm{I}_q[y_w^{(k-1)}]\right] \tag{46}$$

Proof. In the higher layers ($k > 1$):

$$\begin{aligned}
\mathrm{I}_q[b_{j,h}^{(k)}] &\leq^{(33)} \sum_{w=1}^n \hat{a}_{j,w}^{k,h} \mathrm{I}_q[y_w^{(k-1)}] + \left(\max_w \|y_w^{(k-1)}\|\right) \sum_{w=1}^n \mathrm{I}_q[\hat{a}_{j,w}^{k,h}] \\
&\leq^{(42)} \frac{\exp(2C_A^{(k)})}{n} \sum_{w=1}^n \mathrm{I}_q[y_w^{(k-1)}] + \left(\max_w \left\|y_w^{(k-1)}\right\|\right) 2\exp(4C_A^{(k)}) \left(\mathrm{I}_q[y_j^{(k-1)}] + \frac{1}{n} \sum_{j=1}^n \mathrm{I}_q[y_j^{(k-1)}]\right) \\
&= \frac{\exp(2C_A^{(k)}) + \sqrt{d} 2\exp(4C_A^{(k)})}{n} \sum_{w=1}^n \mathrm{I}_q[y_w^{(k-1)}] + \sqrt{d} 2\exp(4C_A^{(k)}) \left(\mathrm{I}_q[y_j^{(k-1)}]\right) \\
&\leq 4\sqrt{d} \exp(4C_A^{(k)}) \left[\mathrm{I}_q[y_j^{(k-1)}] + \frac{1}{n} \sum_{w=1}^n \mathrm{I}_q[y_w^{(k-1)}]\right]
\end{aligned}$$

where the second step uses the fact that

$$\hat{a}^{k,h}_{j,w} \leq^{(43)} \frac{\exp(2C_A^{(k)})}{n}, \tag{47}$$

and the third step uses the fact that the outcome of layer norm has an L2 norm bounded by $\sqrt{d}$. □

B.2 Impact of Layer Norm

Next, we investigate the effect that layer norm has on sensitivity. We note that layer norm was not fully included in any of the pointwise Lipschitzness bounds (Hahn, 2020; Edelman et al., 2022; Li et al., 2023): Hahn (2020); Li et al. (2023) do not seem to mention layer norm, and Edelman et al. (2022) assumes a Lipschitz-continuous proxy, projection on the unit ball. We find that standard layer norm indeed can increase sensitivity substantially, which will be key to our overall results:

Lemma 16. *Let $Y(x) \in \mathbb{R}^d$, and $y(x) := \mathit{LayerNorm}(Y(x))$. Define $N(x) := \sqrt{\sigma^2(Y(x)) + \varepsilon}$. Let $C > 1$ be such that $\|Y(x)\|_2 \leq C$. Then*

$$\mathrm{I}_i[y] \leq 2 \cdot \mathrm{I}_i[Y] \cdot C \cdot \left[1 + \frac{1}{N(x)}\right] \cdot \left[1 + \frac{1}{N(x^{\oplus i})}\right] \tag{48}$$

Proof. First, we note

$$\mathrm{I}_i[(Y - \mu(Y))] \leq \mathrm{I}_i[Y] + \frac{1}{d}\sum_s \mathrm{I}_i[Y_s] = (1 + \frac{1}{d})\,\mathrm{I}_i[Y] \leq 2\,\mathrm{I}_i[Y]$$

Thus, at a factor of 2, it is sufficient to consider the case where $\mu(Y) = 0$, because $\|Y - \mu(Y)\| \leq \|Y\|$. Then

$$\begin{aligned}
\mathrm{I}_i[y] =& \|y(x) - y(x^{\oplus i})\|_2 \\
=& \left\| \frac{Y(x)}{N(x)} - \frac{Y(x^{\oplus i})}{N(x^{\oplus i})} \right\|_2 \\
=& \frac{\left\| N(x^{\oplus i})Y(x) - N(x)Y(x^{\oplus i}) \right\|_2}{N(x)N(x^{\oplus i})} \\
\leq& \frac{\|Y(x)\|_2 \cdot |N(x^{\oplus i}) - N(x)| + N(x^{\oplus i}) \left\| Y(x) - Y(x^{\oplus i}) \right\|_2}{N(x)N(x^{\oplus i})} \\
\leq& \frac{\|Y(x)\|_2 \cdot |N(x^{\oplus i}) - N(x)|}{N(x)N(x^{\oplus i})} + \frac{\left\| Y(x) - Y(x^{\oplus i}) \right\|_2}{N(x)} \\
\leq& \frac{C|N(x^{\oplus i}) - N(x)|}{N(x)N(x^{\oplus i})} + \frac{\left\| Y(x) - Y(x^{\oplus i}) \right\|_2}{N(x)} \\
\leq& \mathrm{I}_i[Y] \cdot \left[\frac{C}{N(x)N(x^{\oplus i})} + \frac{1}{N(x)} \right] \\
=& \mathrm{I}_i[Y] \cdot \frac{1}{N(x)} \cdot \left[\frac{C}{N(x^{\oplus i})} + 1 \right] \\
\leq& \mathrm{I}_i[Y] \cdot C \cdot \left[1 + \frac{1}{N(x)}\right] \cdot \left[1 + \frac{1}{N(x^{\oplus i})}\right]
\end{aligned}$$

where $C \geq 1$ is a bound on $\|Y_i\|_2$. Here, we used $\mathrm{I}_i[N] \leq \mathrm{I}_i[Y]$ in the last step: Note that Y is mean zero; hence, standard deviation equals the L2 norm, and

$$\sqrt{d}\,\mathrm{I}_i[N] = |\|Y(x)\|_2 - \|Y(x^{\oplus i})\|_2| \leq \|Y(x) - Y(x^{\oplus i})\|_2 \leq \|Y(x) - Y(x^{\oplus i})\|_2 = \mathrm{I}_i[Y]$$

□

B.3 Layerwise Bounds on Sensitivity

Putting together Lemmas 9 and 16, we bound the influence of the i-th bit on the k-th layer in terms of its influence on the $k-1$-th layer.

Lemma 17. *With probability $\geq 1 - \frac{H}{n^2}$ over the choice of x, the following holds for each j:*

$$\boxed{\sum_{i=1}^{n} \mathrm{I}_i[Y_j^{(1)}] \leq 150 L_{f MLP} C_V^2 C_P H \log n = C^{(0)} \log n} \tag{49}$$

and

$$\boxed{\mathrm{I}_i[y_j^{(1)}] \leq 2 \left(\max_w \|Y_w^{(1)}\|_2\right) \cdot N_j^{(1)}(x) \cdot N_j^{(1)}(x^{\oplus i}) \cdot L_{f MLP} \cdot \mathrm{I}_i[Y_j^{(1)}]} \tag{50}$$

For $k > 1$:

$$\boxed{\mathrm{I}_i[y_j^{(k)}] \leq \frac{C^{(k)}}{3} \cdot N_j^{(k)}(x) \cdot N_j^{(k)}(x^{\oplus i}) \cdot \left(\sum_{w=1}^{n} (\delta_{w,j} + \frac{1}{n}) \mathrm{I}_i[y_w^{(k-1)}]\right)} \tag{51}$$

Intuitively, the $\delta_{w,j}$ contribution reflects the skip connection (residual stream); the $\frac{1}{n}$ contribution reflects the role of soft attention.

Proof. We first obtain:

$$\mathrm{I}_s[y_j^{(k)}] \leq^{(48,2,4,26)} N_k^{(k)}(x) \cdot N_k^{(k)}(x^{\oplus s}) \cdot C_{MLP} \cdot L_{f MLP} \cdot \left(\mathrm{I}_s[y_j^{(k-1)}] + \sum_{q=1}^{H} \mathrm{I}_s[b_{i,q}^{(k)}]\right) \tag{52}$$

For $k > 1$, we find, putting together the previous results:

$$\begin{aligned}
\mathrm{I}_i[y_j^{(k)}] &\leq^{(48)} \mathrm{I}_i[Y_j] \cdot N_j^{(k)}(x) \cdot N_j^{(k)}(x^{\oplus i}) \\
&\leq^{(52)} (L_{f MLP} C_{MLP} \cdot \left(\max_w \|Y_w^{(k)}\|_2\right) \cdot \left(\mathrm{I}_i[y_j^{(k-1)}] + \sum_{q=1}^{H} \mathrm{I}_i[b_{i,q}^{(k)}]\right)) \cdot N_j^{(k)}(x) \cdot N_j^{(k)}(x^{\oplus i}) \\
&\leq^{(46)} (L_{f MLP} C_{MLP} \cdot C_{MLP}(H+1) C_V C_P \cdot \left(\mathrm{I}_i[y_j^{(k-1)}] + H 4\sqrt{d} \exp(4C_A^{(k)}) \left[\mathrm{I}_i[y_j^{(k-1)}] + \frac{1}{n} \sum_{w=1}^{n} \mathrm{I}_i[y_w^{(k-1)}]\right]\right)) \cdot N_j^{(k)}(x) \cdot N_j^{(k)}(x^{\oplus i}) \\
&=^{(30)} \frac{C^{(k)}}{3} \cdot N_j^{(k)}(x) \cdot N_j^{(k)}(x^{\oplus i}) \cdot \left(\sum_{w=1}^{n} (\delta_{w,j} + \frac{1}{n}) \mathrm{I}_i[y_w^{(k-1)}]\right)
\end{aligned}$$

In the lowest layer, the influence is bounded instead in terms of the influence on the attention weights. With high probability, a logarithmic bound on the summed influences over all bits results. Crucially, the key-value-matrix plays no role here. For $k = 1$ we find, setting $\delta = 4$ in (35):

$$\begin{aligned}
\mathrm{I}_i[Y_j^{(1)}] &\leq^{(2)} \mathrm{I}_i \left[f_{MLP} \left(y_j^{(0)} + \sum_{h=1}^{H} b_{j,h}^{(1)}\right)\right] \\
&\leq^{(27)} L_{f MLP} \left[\mathrm{I}_i \left[y_j^{(0)}\right] + \sum_{h=1}^{H} \mathrm{I}_i \left[b_{j,h}^{(1)}\right]\right] \\
&\leq^{(33)} L_{f MLP} \left[\mathrm{I}_i \left[y_j^{(0)}\right] + \sum_{h=1}^{H} C_V \left[\sum_{w=1}^{n} \hat{a}_{j,w}^{1,h} \mathrm{I}_i[y_w^{(0)}] + C_V \left(\max_w \|y_w^{(0)}\|_2\right) \sum_{w=1}^{n} \mathrm{I}_i[\hat{a}_{j,w}^{1,h}]\right]\right] \\
&\leq^{(24)} L_{f MLP} \left[2 C_P \delta_{ij} + C_V \sum_{h=1}^{H} \left[\sum_{w=1}^{n} \hat{a}_{j,w}^{1,h} C_P \delta_{iw} + C_V C_P \sum_{w=1}^{n} \mathrm{I}_i[\hat{a}_{j,w}^{1,h}]\right]\right] \\
&\leq L_{f MLP} C_V^2 C_P \left[2\delta_{ij} + \sum_{h=1}^{H} \hat{a}_{j,i}^{1,h} + \sum_{h=1}^{H} \sum_{w=1}^{n} \mathrm{I}_i[\hat{a}_{j,w}^{1,h}]\right]
\end{aligned}$$

and hence, by summing over i,

$$\begin{aligned}
\sum_{i=1}^{n} \mathrm{I}_i[Y_j^{(1)}] &\leq L_{f MLP} C_V^2 C_P \left[2 + \sum_{h=1}^{H} \sum_{i=1}^{n} \hat{a}_{j,i}^{1,h} + \sum_{h=1}^{H} \sum_{i=1}^{n} \sum_{w=1}^{n} \mathrm{I}_i[\hat{a}_{j,w}^{1,h}]\right] \\
&\leq^{(35)} L_{f MLP} C_V^2 C_P [2 + H + H 136 \log n + 10H] \\
&\leq 150 L_{f MLP} C_V^2 C_P H \log n
\end{aligned}$$

with probability $\geq 1 - \frac{H}{n^2}$.[7]

□

B.4 Relating Influence at First and Last Layers

At this point, we have derived bounds on the influence of any bit on any individual activation $y_j^{(k)}$ in terms of the parameter norms, the layer-norm-induced blowup, and influence the bit has on the activations at the preceding layer. We now derive bounds linking these quantities to the overall sensitivity $s(f,x)$ and the average sensitivity $as_n(f)$. In order to do this, we link influences on the top layer to influences at the first layer. Recall

$$\text{Blowup}_K(x) = \prod_{k=1}^{K} \tau^{(k)}(x) \tag{53}$$

We first find that

Lemma 18.

$$\mathrm{I}_i[y_j^{(k)}] \leq C^{(1)} \left(\prod_{l=2}^{k} C^{(k)} \right) \text{Blowup}_k(x)\, \text{Blowup}_k(x^{\oplus i}) \left(\frac{1}{n} \sum_{w=q}^{n} \mathrm{I}_i[Y_w^{(1)}] + \mathrm{I}_i[Y_j^{(1)}] \right) \tag{54}$$

Proof. By induction using (51). First, for $k = 2$,

$$\begin{aligned}
\mathrm{I}_i[y_j^{(2)}] \leq & C^{(2)} \cdot N_j^{(2)}(x) \cdot N_j^{(2)}(x^{\oplus i}) \cdot \left(\sum_{w=1}^{n} (\delta_{w,j} + \frac{1}{n})\, \mathrm{I}_i[y_w^{(1)}] \right) \\
\leq & C^{(2)} \cdot N_j^{(2)}(x) \cdot N_j^{(2)}(x^{\oplus i}) \cdot \left(\sum_{w=1}^{n} (\delta_{w,j} + \frac{1}{n}) 2 \left(\max_w \|Y_w^{(1)}\|_2 \right) \cdot N_w^{(1)}(x) \cdot N_w^{(1)}(x^{\oplus i}) \cdot L_{fMLP} \cdot \mathrm{I}[Y_w^{(1)}] \right) \\
\leq & C^{(2)} 2 \left(\max_w \|Y_w^{(1)}\|_2 \right) \cdot L_{fMLP} \cdot \text{Blowup}_2(x) \cdot \text{Blowup}_2(x^{\oplus i}) \cdot \left(\sum_{w=1}^{n} (\delta_{w,j} + \frac{1}{n}) \cdot \mathrm{I}[Y_w^{(1)}] \right) \\
\leq & C^{(1)} C^{(2)} \cdot \text{Blowup}_2(x) \cdot \text{Blowup}_2(x^{\oplus i}) \cdot \left(\sum_{w=1}^{n} (\delta_{w,j} + \frac{1}{n}) \cdot \mathrm{I}[Y_w^{(1)}] \right)
\end{aligned}$$

where

$$C^{(1)} = 2 \left(\max_w \|Y_w^{(1)}\|_2 \right) \cdot L_{fMLP}$$

Now, for $k > 2$, using the induction hypothesis (IH):

$$\begin{aligned}
\mathrm{I}_i[y_j^{(k)}] \leq & \frac{C^{(k)}}{3} \cdot N_j^{(k)}(x) \cdot N_j^{(k)}(x^{\oplus i}) \cdot \left(\sum_{w=1}^{n} (\delta_{w,j} + \frac{1}{n})\, \mathrm{I}_i[y_w^{(k-1)}] \right) \\
\leq^{(IH)} & \frac{1}{3} C^{(1)} \left(\prod_{l=2}^{k} C^{(k)} \right) \text{Blowup}_k(x)\, \text{Blowup}_k(x^{\oplus i}) \cdot \left(\sum_{w=1}^{n} \sum_{v=1}^{n} (\delta_{w,j} + \frac{1}{n})(\delta_{v,w} + \frac{1}{n})\, \mathrm{I}_i[Y_v^{(1)}] \right) \\
= & \frac{1}{3} C^{(1)} \left(\prod_{l=2}^{k} C^{(k)} \right) \text{Blowup}_k(x)\, \text{Blowup}_k(x^{\oplus i}) \cdot \left(\mathrm{I}_i[Y_j^{(1)}] + 3 \sum_{v=1}^{n} \frac{1}{n} \mathrm{I}_i[Y_v^{(1)}] \right) \\
\leq & C^{(1)} \left(\prod_{l=2}^{k} C^{(k)} \right) \text{Blowup}_k(x)\, \text{Blowup}_k(x^{\oplus i}) \cdot \left(\mathrm{I}_i[Y_j^{(1)}] + \sum_{v=1}^{n} \frac{1}{n} \mathrm{I}_i[Y_v^{(1)}] \right)
\end{aligned}$$

□

B.5 Deriving Almost-Everywhere Pointwise Sensitivity Bounds

Putting together the previous findings, we are now in a position to link the sensitivity to layer norm blowup and parameter norms. We find that the sensitivity on an input x is bounded in terms of the blowup on x itself and on its Hamming neighbors, up to sublinear factors $\sqrt{n \log n}$:

Theorem 19 (Repeated from Theorem 4)**.** *With probability at least* $1 - \frac{H}{n^{-2}}$ *over the choice of* $x \in \{\pm 1\}^n$*, we have*

$$\frac{s(f,x)}{C\sqrt{n \log n}} \leq \text{Blowup}(x)^2 + \frac{1}{n} \sum_i \text{Blowup}(x^{\oplus i})^2$$

[7] The last inequality holds whenever $n > 1$. If $n = 1$, the probability bound evaluates to 0, making the statement trivially true.

Proof. The most immediate idea is to sum both sides of (54) across i to get a bound on $s(f,x) = \frac{1}{4}\sum_{i=1}^{n} \mathrm{I}_i[y_n^{(L)}]^2$. In fact, it will pay off to *average* the influence across the input bits, and only later eliminate the $1/n$ factor. Formally,

$$\begin{aligned}
\frac{1}{n}s(f,x) &=^{(19)} \frac{1}{4n}\sum_{i=1}^{n} \mathrm{I}_i[v^T y_n^{(L)}]^2 \\
&\leq^{(18)} \frac{1}{4n}\sum_{i=1}^{n} \mathrm{I}_i[y_n^{(L)}]^2 \|v_{out}\|_2^2 \\
&\leq^{(21)} \frac{1}{2n}\sum_{i=1}^{n} \mathrm{I}_i[y_n^{(L)}] \|v_{out}\|_2^2 \sqrt{d} \\
&\leq^{(54)} \frac{\sqrt{d}\|v_{out}\|^2}{2}\left(\prod_{l=2}^{k} C_k\right) \mathrm{Blowup}(x) \sum_i \frac{1}{n}\mathrm{Blowup}(x^{\oplus i}) \left[\frac{1}{n}\sum_w \mathrm{I}_i[Y_w^{(1)}] + \mathrm{I}_i[Y_n^{(1)}]\right] \\
&\leq^{(CSB)} \frac{\sqrt{d}\|v_{out}\|^2}{2}\left(\prod_{l=2}^{k} C_k\right) \mathrm{Blowup}(x) \sqrt{\sum_i \frac{1}{n}\mathrm{Blowup}(x^{\oplus i})^2} \left[\sqrt{\sum_i \frac{1}{n}\frac{1}{n}\sum_w \mathrm{I}_i[Y_w^{(1)}]^2} + \sqrt{\sum_i \frac{1}{n}\mathrm{I}_i[Y_n^{(1)}]^2}\right]
\end{aligned}$$

where the last inequality uses Cauchy-Schwarz-Bunyakovsky (CSB). Now, we find for any $w \in \{1,\dots,n\}$:

$$\sum_i \mathrm{I}_i[Y_w^{(1)}]^2 \leq^{(21)} 2\left(\max_x \|Y_w^{(1)}\|_2\right)\sum_i \mathrm{I}_i[Y_w^{(1)}] \leq^{(49)} \left(\max_x \|Y_w^{(1)}\|_2\right) C_0 \log n \tag{55}$$

with probability (simultaneously over w) $1 - \frac{H}{n^{\delta-2}}$ over the choice of x. Setting $\delta = 4$ and defining and plugging in (55) together with the fact $\sqrt{x} \leq x$ when $x \geq 1$, we get:

$$\frac{1}{n}s(f,x) \leq C\,\mathrm{Blowup}(x)\sqrt{\frac{\log n}{n}}\sqrt{\sum_i \frac{1}{n}\mathrm{Blowup}(x^{\oplus i})^2}$$

Rearranging, we find

$$\begin{aligned}
\frac{1}{C}s(f,x)\sqrt{\frac{1}{n\log n}} &\leq \mathrm{Blowup}(x)\sqrt{\sum_i \frac{1}{n}\mathrm{Blowup}(x^{\oplus i})^2} \\
&\leq \mathrm{Blowup}(x)^2 + \sum_i \frac{1}{n}\mathrm{Blowup}(x^{\oplus i})^2
\end{aligned}$$

where the last step follows by Young's Inequality ($ab \leq \frac{a^2+b^2}{2}$ for $a,b \geq 0$). As we chose $\delta = 4$, this is true with probability $1 - \frac{H}{n^2}$ over the choice of x. □

B.6 Deriving On-Average Sensitivity Bounds

Finally, we can convert the high-probability statement from the preceding lemma, which bounded the layer norm blowup on x itself *or* its Hamming neighbors, into an on-average bound over the entire input space:

Corollary 20 (Repeated from Corollary 4)**.**

$$C \cdot \mathbb{E}[\mathrm{Blowup}(x)^2] \geq \frac{as_n(f)}{\sqrt{n\log n}} - \frac{H}{n} \tag{56}$$

Proof. Recall from the preceding lemma that, with probability $1 - \frac{H}{n^2}$ over the choice of x, we have

$$\frac{s(f,x)}{C\sqrt{n\log n}} \leq \mathrm{Blowup}(x)^2 + \frac{1}{n}\sum_i \mathrm{Blowup}(x^{\oplus i})^2 \tag{57}$$

We first note the following, for any function $\phi : \{\pm 1\}^n \rightarrow \mathbb{R}$:

$$\mathbb{E}[\phi(x)] = \frac{1}{n}\mathbb{E}_x \sum_{i=1}^{n} \phi(x^{\oplus i})$$

since every one of the maps $x \mapsto x^{\oplus i}$ ($i = 1,\dots,n$) is a bijection on $\{\pm 1\}$. Hence,

$$\mathbb{E}[\phi(x)] = \mathbb{E}_x\left[\frac{1}{2}\phi(x) + \frac{1}{2}\sum_{i=1}^{n} \phi(x^{\oplus i})\right]$$

Taking $\phi(x) := \mathrm{Blowup}(x)^2$ yields:

$$\mathbb{E}[\mathrm{Blowup}(x)^2] = \mathbb{E}_x \left[\frac{1}{2} \mathrm{Blowup}(x)^2 + \frac{1}{2}\frac{1}{n} \sum_i \mathrm{Blowup}(x^{\oplus i})^2 \right] \tag{58}$$

Let A be the set of x satisfying (57); $|A| \geq 2^n(1 - \frac{H}{n^2})$. Then

$$\begin{aligned}
\mathbb{E}[\mathrm{Blowup}(x)^2] =& \frac{1}{2^n} \sum_{x \in \{\pm 1\}^n} \left[\frac{1}{2} \mathrm{Blowup}(x)^2 + \frac{1}{2}\frac{1}{n} \sum_i \mathrm{Blowup}(x^{\oplus i})^2 \right] \\
\geq& \frac{1}{2^n} \sum_{x \in A} \left[\frac{1}{2} \mathrm{Blowup}(x)^2 + \frac{1}{2}\frac{1}{n} \sum_i \mathrm{Blowup}(x^{\oplus i})^2 \right] \\
\geq& \frac{1}{C\sqrt{n \log n}} \frac{1}{2^n} \sum_{x \in A} s(f,x) \\
\geq& \frac{1}{C\sqrt{n \log n}} \frac{1}{2^n} \sum_{x \in \{\pm 1\}^n} s(f,x) - \frac{1}{C\sqrt{n \log n}} \frac{1}{2^n} \sum_{x \in \in A} s(f,x) \\
\geq& \frac{1}{C\sqrt{n \log n}} as_n(f) - \frac{1}{C\sqrt{\log n}} \frac{H}{n^{3/2}}
\end{aligned}$$

Rearranging, we find:

$$C \cdot \mathbb{E}[\mathrm{Blowup}(x)^2] \geq \frac{1}{\sqrt{n \log n}} as_n(f) - \frac{1}{\sqrt{\log n}} \frac{H}{n^{3/2}} \geq \frac{1}{\sqrt{n \log n}} as_n(f) - \frac{H}{n}$$

□

C Theorem 6

Before proving Theorem 6, we require the following lemma in the analysis of Boolean functions. Informally, the lemma describes how the RMSE between two functions can be lower-bounded in terms of their average sensitivities.

For any function $f : \{\pm 1\} \to \mathbb{R}$, we write $\|f\|_2 := \sqrt{\frac{1}{2^n} \sum_{x \in \{\pm 1\}^n} f(x)^2} = \mathbb{E}_x[f(x)^2]$. Then the lemma states:

Lemma 21. *For any $f, f' : \{\pm 1\}^n \to \mathbb{R}$, we have for any $\alpha > 1$:*

$$\sqrt{\mathbb{E}[(f(x) - f'(x))^2]} \geq \sqrt{\frac{(1 - \frac{1}{\alpha}) as_n(f)}{(n - \frac{as_n(f)}{\alpha \cdot \|f\|_2^2})}} - \|f\|_2 \sqrt{\alpha \frac{as_n(f')}{as_n(f)}} \tag{59}$$

Note that the optimal α depends on f, f'. The choice $\alpha = 2$ will be sufficient for our needs:

$$\sqrt{\mathbb{E}[(f(x) - f'(x))^2]} \geq \sqrt{\frac{as_n(f)}{2n - \frac{as_n(f)}{\|f\|_2^2}}} - \|f\|_2 \sqrt{2 \frac{as_n(f')}{as_n(f)}} \tag{60}$$

Proof. Recall the discussion of the Fourier-Walsh decomposition preceding (17). The proof idea is that, when functions have different average sensitivities, their degree profiles must be quite different, which then entails that they must be substantially distinct in behavior.

Let $1 \leq \lambda_{Min} \leq n$; we will fix it later. Let Π be the orthogonal projection on the Fourier components with degree at least λ_{Min}, formally given as the unique linear map satisfying

$$\Pi \chi_P = \begin{cases} \chi_P & \text{if } |P| \geq \lambda_{Min} \\ 0 & \text{else} \end{cases} \tag{61}$$

Then, for any $g : \{\pm 1\}^n \to \mathbb{R}$, by (17) and Parseval's theorem for the Fourier-Walsh transform (O'Donnell, 2014):[8]

$$\lambda_{Min} \|\Pi g\|_2^2 \leq as_n(g) \leq \lambda_{Min}(\|g\|_2^2 - \|\Pi g\|_2^2) + n\|\Pi g\|_2^2 \tag{62}$$

[8] If $d_0 \ldots, d_n$ is the degree profile of g, then $\lambda_{Min} \|\Pi g\|_2^2 = \sum_{i \geq \lambda_{Min}} \lambda_{Min} d_i \leq as_n(g) \leq \sum_{0 < i < \lambda_{Min}} \lambda_{Min} d_i + \sum_{i \geq \lambda_{Min}} n d_i = \lambda_{Min}(\|g\|_2^2 - \|\Pi g\|_2^2) + n\|\Pi g\|_2^2$, where the inequalities follow from (17) and the equalities follow from Parseval's theorem applied to Πg and to g.

Rearranging and inserting f, f' for g in the two inequalities, respectively:

$$\begin{aligned} \frac{as_n(f) - \lambda_{Min}\|f\|_2^2}{n - \lambda_{Min}} \leq & \|\Pi f\|_2^2 \\ \|\Pi f'\|_2^2 \leq & \frac{as_n(f)}{\lambda_{Min}} \end{aligned}$$

Hence

$$\|\Pi f\|_2 - \|\Pi f'\|_2 \geq \sqrt{\frac{as_n(f) - \lambda_{Min}\|f\|_2^2}{(n - \lambda_{Min})}} - \sqrt{\frac{as_n(f')}{\lambda_{Min}}}$$

Now at $\lambda_{Min} = \frac{as_n(f)}{\alpha\|f\|_2^2}$, we get

$$\|\Pi f\|_2 - \|\Pi f'\|_2 \geq \sqrt{\frac{(1-\frac{1}{\alpha})as_n(f)}{(n - \frac{as_n(f)}{\alpha\|f\|_2^2})}} - \|f\|_2\sqrt{\alpha\frac{as_n(f')}{as_n(f)}}$$

Now by the reverse triangle inequality

$$\|\Pi f\|_2 - \|\Pi f'\|_2 \leq \|\|\Pi f - \Pi f'\|_2 \leq \|f - f'\|_2 \tag{63}$$

and the claim follows.

□

Proof of the Theorem. The proof can be decomposed into the following steps.

Step 1: Decomposing Δ We have $(F \cdot d + G \cdot d^2) \cdot L$ parameters (where F, G are constant). We are assuming that Δ is from $\rho\mathbb{S}$. For each of the L instances layer norm, we consider the part of Δ corresponding to the immediately preceding bias; we will name this $\Delta_1, \dots, \Delta_L$.

Step 2: Effect of Perturbations on C_θ We will make explicit the dependence of C on θ by writing C_θ. We next note that, for any fixed θ, as $\rho \to 0$, the effect of a small perturbation Δ on C_θ is bounded, because C_θ is locally Lipschitz in θ.[9] Thus, as $\rho \to 0$:

$$C_{\theta+\Delta} = C_\theta \cdot (1 + O(\rho)) \tag{64}$$

uniformly across all Δ with $\|\Delta\| = \rho$, where $O(\cdot)$ includes constants depending on θ, but not ρ, Δ, or n.

Step 2: Lower Bounding Error in Terms of Sensitivity Drop Set

$$\mu := \liminf_{n\to\infty} \frac{as_n(T_\theta)}{n} \tag{65}$$

Necessarily, $\mu \in [0,1]$. The claim of the theorem is nontrivial if and only if $\mu > 0$. Lemma (21) shows that the difference between f_θ and $f_{\theta+\Delta}$ can be lower-bounded in terms of their sensitivities. First, we note, as $n \to \infty$, with $\alpha = 2$,

$$\frac{as_n(f)}{2n - \frac{as_n(f)}{\|f\|_2^2}} = \frac{\mu}{2 - \frac{\mu}{\|f\|_2^2}} + o(1) \geq \frac{\mu}{2} + o(1) \tag{66}$$

where we used $\frac{\mu}{\|f\|_2^2} \in [0,1]$ up to $o(1)$. Indeed, if the output of ±1, we have $\|f\|_2^2 = 1$, and the term indeed is lower-bounded by μ:

$$\frac{as_n(f)}{2n - \frac{as_n(f)}{\|f\|_2^2}} = \frac{\mu}{2-\mu} + o(1) \geq \mu + o(1) \tag{67}$$

[9] This follows from the fact that in (8), for C_P, ℓ^1, ℓ^∞ norms are locally Lipschitz; for C_{MLP} because it can be chosen to be the product of parameter matrix norms in the MLP; for the exponential term because the spectral norm is locally Lipschitz.

showing that the 2 factor can be eliminated in the case of Boolean output. Next, as $n \rightarrow \infty$, for $\alpha = 2$ as above, and using $0 \leq \frac{as_n(T_\theta)}{n} \leq \|T_\theta\|_2^2 \leq 1$, we find

$$
\begin{aligned}
&\mathbb{E}_\Delta \mathbb{E}[(T_\theta(x) - T_{\theta+\Delta}(x))^2] \\
\geq &\mathbb{E}_\Delta \underbrace{\frac{(1-\frac{1}{\alpha})as_n(T_\theta)}{(n - \frac{as_n(T_\theta)}{\alpha\|T_\theta(x)\|_2^2})}}_{\text{independent of } \Delta} + \underbrace{\|T_\theta\|_2^2 \alpha \mathbb{E}_\Delta \frac{as_n(T_{\theta+\Delta})}{as_n(T_\theta)}}_{\geq 0} - 2\sqrt{\frac{(1-\frac{1}{\alpha})as_n(T_\theta)}{(n - \frac{as_n(T_\theta)}{\alpha\|T_\theta(x)\|_2^2})}} \|T_\theta\|_2 \mathbb{E}_\Delta \sqrt{\alpha \frac{as_n(T_{\theta+\Delta})}{as_n(T_\theta)}} \\
\geq &\frac{as_n(T_\theta)}{2n - \frac{as_n(T_\theta)}{\|T_\theta(x)\|_2^2}} - 2\sqrt{\frac{1}{(n - \frac{1}{2\|T_\theta(x)\|_2^2})}} \|T_\theta\|_2 \mathbb{E}_\Delta \sqrt{as_n(T_{\theta+\Delta})} \\
\geq &\frac{as_n(T_\theta)}{2n - \frac{as_n(T_\theta)}{\|T_\theta(x)\|_2^2}} - 2\mathbb{E}_\Delta \sqrt{\frac{as_n(T_{\theta+\Delta})}{(n - \frac{1}{2\|T_\theta(x)\|_2^2})}} \\
\geq^{(\dagger)} &\frac{as_n(T_\theta)}{2n - \frac{as_n(T_\theta)}{\|T_\theta(x)\|_2^2}} - 4\mathbb{E}_\Delta \sqrt{\frac{as_n(T_{\theta+\Delta})}{n}} \\
= &\frac{as_n(T_\theta)}{2n - \frac{as_n(T_\theta)}{\|T_\theta(x)\|_2^2}} - 4\mathbb{E}_\Delta \sqrt{\frac{\mathbb{E}_x s_n(T_{\theta+\Delta}, x)}{n}} \\
\geq &\frac{\mu}{2} - 4\sqrt{\frac{\mathbb{E}_\Delta \mathbb{E}_x s(T_{\theta+\Delta}, x)}{n}} + o(1)
\end{aligned}
\tag{68}
$$

where $(\dagger)$ uses $(n - \frac{1}{2\|T_\theta(x)\|_2^2}) \geq n - 1/2 \geq n/4$, and the final step used Jensen's inequality applied to the concave function $\sqrt{\cdot}$. By the reasoning above, in Eq. (67), the factor 2 in the first term can be eliminated when the output is ± 1. In order to lower-bound (68), we want to show the following:

Take any $\zeta > \frac{2L}{d}$. For each x, with probability at least $(1 - O(n^{2-(d-1)\zeta/L}) - L\exp(-\Theta(d)))$ (where O contains constants depending on ρ, d, L, but not depending on x and n), the blowup is simultaneously bounded on x and its radius-1 Hamming ball:

$$\tau_{\theta+\Delta}^{(k)}(X) \leq 1 + n^{\zeta/L}, \quad \forall X \in \{x, x^{\oplus 1}, x^{\oplus 2}, \ldots, x^{\oplus n}\} \tag{69}$$

We will prove this below in Step 4. We may then bound $\tau_{\theta+\Delta}^{(k)}(X) \leq 2n^{\zeta/L}$ for large n. By (12), if (69) is proven, then:

$$\frac{s(T_{\theta+\Delta}, x)}{C_{\theta+\Delta}\sqrt{n \log n}} \leq \text{Blowup}(x)^2 + \frac{1}{n}\sum_{i=1}^{n} \text{Blowup}(x^{\oplus i})^2 \leq 4n^{2\zeta}$$

and hence

$$s(T_{\theta+\Delta}, x) \leq 4n^{2\zeta+.5} C_{\theta+\Delta}\sqrt{\log n} \tag{70}$$

and hence

$$\frac{s(T_{\theta+\Delta}, x)}{n} \leq 4n^{2\zeta-.5} C_{\theta+\Delta}\sqrt{\log n} \tag{71}$$

with probability $1 - \frac{H}{n^2}$ over x and $1 - O(Ln^{2-(d-1)\zeta/L}) - L\exp(-\Theta(d))$ over Δ, where we used a union bound over the L layers. Let us now fix $\zeta = \frac{3L}{d}$. Then, as $n \rightarrow \infty$,

$$\mathbb{E}_\Delta \mathbb{E}_x \frac{s(T_{\theta+\Delta}, x)}{n} \leq 4n^{6L/d-.5} C_{\theta+\Delta}\sqrt{\log n} + O(n^{2-3\frac{d-1}{d}}) + \frac{H}{n^2} + L\exp(-\Omega(d)) + o(1)$$

Under the hypothesis $12L < d$, all terms except for $L\exp(-\Omega(d))$ are $o(1)$ as $n \rightarrow \infty$. Inserting this into (68), the claim follows.

Step 4: Effect on Layer Norm It remains to show (69). We proceed inductively from the input layer upwards, examining the effect of perturbing parameters. In the inductive step, we consider a transformer where all the parameters below the bias immediately preceding the i-th layer norm have already been perturbed.

Fix an input string $x \in \{\pm 1\}^n$. Write $Y_i^{(k)}$ for the activation computed using the perturbed parameters $\theta + \Delta$ for all parameters except the bias of interest, for which θ is used here. Thus, after applying the perturbation to that bias, the activation is $Y_i^{(k)} + \Delta_i$. The aim is to lower-bound the SD of $Y_i^{(k)} + \Delta_i$ with high probability over the choice of Δ.

Let $center(x) := x - mean(x)$, i.e. the pre-scaling component of layer norm. First,

$$\mathbb{P}\left(\|center(\Delta_i)\| \leq \frac{\rho}{2}\sqrt{\frac{d}{D}}\right) \leq \exp(-\Omega(d))$$

by standard concentration arguments, where $\Omega(d)$ scales positively with d.[10] Let $\Pi_{\mathbb{S}} : \mathbb{R}^d \to \mathbb{S}^{d-1} \subset \mathbb{R}^d$ be the projection on the sphere around 0 of radius $\frac{\rho}{2}\sqrt{\frac{d}{D}}$:

$$\Pi_{\mathbb{S}}(x) = \frac{\rho}{2}\sqrt{\frac{d}{D}}\frac{x}{\|x\|} \tag{72}$$

Let $\Pi_{\mathbb{B}} : \mathbb{R}^d \to B_{\frac{\rho}{2}\sqrt{\frac{d}{D}}}(0)$ be the projection on the ball around 0 of radius $\frac{\rho}{2}\sqrt{\frac{d}{D}}$. $\Pi_{\mathbb{B}}$ is a contraction (referred to below with (†)[11]). Conditional on $\|center(\Delta_i)\| \geq \frac{\rho}{2}\sqrt{\frac{d}{D}}$, we have $\Pi_{\mathbb{B}} center(\Delta_i) = \Pi_{\mathbb{S}} center(\Delta_i)$ and its distribution is, by symmetry, uniform on the ball's surface. We now bound the probability that the SD of the perturbed activation is large:

$$\begin{aligned}
&\mathbb{P}\left(\exists j : SD(\Delta_i + Y_i^{(k)}) < n^{-\zeta/L}\right)\\
=&\mathbb{P}\left(\exists j : \|center(\Delta_i + Y_i^{(k)})\| < \sqrt{d}n^{-\zeta/L}\right)\\
=&\mathbb{P}\left(\exists j : center(\Delta_i) \in B_{n^{-\zeta/L}}\left(-center(Y_i^{(k)})\right)\right)\\
\leq&\sum_{j=1}^{n}\mathbb{P}\left(center(\Delta_i) \in B_{\sqrt{d}n^{-\zeta/L}}\left(-center(Y_i^{(k)})\right)\right)\\
\leq&\mathbb{P}(\|center(\Delta_i)\| \leq \frac{\rho}{2}\sqrt{\frac{d}{D}}) + \sum_{j=1}^{n}\mathbb{P}\left(center(\Delta_i) \in B_{\sqrt{d}n^{-\zeta/L}}\left(-center(Y_i^{(k)})\right) \middle| \|center(\Delta_i)\| > \frac{\rho}{2}\sqrt{\frac{d}{D}}\right)\\
\leq^{\dagger}&\mathbb{P}(\|center(\Delta_i)\| \leq \frac{\rho}{2}\sqrt{\frac{d}{D}}) + \sum_{j=1}^{n}\mathbb{P}\left(\Pi_{\mathbb{B}} center(\Delta_i) \in B_{\sqrt{d}n^{-\zeta/L}}\left(-\Pi_{\mathbb{B}} center(Y_i^{(k)})\right) \middle| \|center(\Delta_i)\| > \frac{\rho}{2}\sqrt{\frac{d}{D}}\right)\\
\leq&\exp(-\Omega(d)) + \sum_{j=1}^{n}\mathbb{P}\left(\Pi_{\mathbb{B}} center(\Delta_i) \in B_{\sqrt{d}n^{-\zeta/L}}\left(-\Pi_{\mathbb{B}} center(Y_i^{(k)})\right) \middle| \|center(\Delta_i)\| > \frac{\rho}{2}\sqrt{\frac{d}{D}}\right)\\
\leq&\exp(-\Omega(d)) + \sum_{j=1}^{n}\mathbb{P}\left(\Pi_{\mathbb{S}} center(\Delta_i) \in B_{\sqrt{d}n^{-\zeta/L}}\left(-\Pi_{\mathbb{S}} center(Y_i^{(k)})\right) \middle| \|center(\Delta_i)\| > \frac{\rho}{2}\sqrt{\frac{d}{D}}\right)\\
=&\exp(-\Omega(d)) + \sum_{j=1}^{n}\frac{Area\left(B_{\sqrt{d}n^{-\zeta/L}}\left(-\Pi_{\mathbb{S}} center(Y_i^{(k)})\right)\right)}{Area\left(\frac{\rho}{2}\sqrt{\frac{d}{D}}\mathbb{S}\right)}
\end{aligned}$$

where $Area(B_{\sqrt{d}n^{-\zeta/L}}\left(-\Pi_{\mathbb{S}} center(Y_i^{(k)})\right))$ is the area of a hypersphere cap of radius $\sqrt{d}n^{-\zeta/L}$ on a hypersphere of radius $\frac{\rho}{2}\sqrt{\frac{d}{D}}$, which is at most $n^{-(d-1)\zeta/L}$ times some constant depending on d, D, ρ but not n, x[12]. The denominator is constant in n. Overall, as $n \to \infty$,

$$\mathbb{P}\left(\exists j : \|center(\Delta_i + Y_i^{(k)})\| < n^{-\zeta/L}\right) \leq \exp(-\Omega(d)) + O(n^{1-(d-1)\zeta/L})$$

[10] One way of obtaining this is by parameterizing Δ in terms of first sampling D independent $X_1, \ldots, X_D \sim \mathcal{N}(0, \frac{1}{D})$, and dividing the resulting vector by its norm, before multiplying by ρ in the end. Before normalization, the squared norms of the entire vector and the norm of the part corresponding to Δ_i concentrate with exponential tail bounds in d, around 1 and $\frac{d}{D}$, respectively. Furthermore, the mean of Δ_i concentrates around 0. Thus, the norm of $center(\Delta_i)$ concentrates around $\rho\sqrt{\frac{d}{D}}$.

[11] The contractivity of $\Pi_{\mathbb{B}}$ is proven, in a different context, as Lemma A.9 in Edelman et al. (2022).

[12] A nonasymptotic formula for the area of the hypersphere cap is given by Li (2010). A simple derivation of the asymptotic relationship used here is by approximating, when n is large and thus $n^{-\zeta/L}$ is small in relation to d, the cap as a disk on the $d-1$-dimensional tangent space to the hypersphere. This approximation is valid in this limit since the radius of the hypersphere, and thus its curvature, are independent of n.

If the event denoted by this probability is avoided, then under the parameter setting $\theta + \Delta$,

$$\tau_k(x) \leq 1 + n^{\zeta/L} \tag{73}$$

for both x and its Hamming ball of radius 1, with probability (by a union bound over the radius-1 Hamming ball $B_1(x)$ around x):

$$\begin{aligned}
&\mathbb{P}\left(\forall x' \in B_1(x) : \forall j : \|center(\Delta_i + Y_i^{(k)}(x'))\| \geq n^{-\zeta/L}\right) \\
\geq& 1 - \mathbb{P}\left(\|center(\Delta_i)\| \leq \frac{\rho}{2}\sqrt{\frac{d}{D}}\right) - \sum_{x' \in B_1(x)} \mathbb{P}\left(center(\Delta_i) \in B_{n^{-\zeta/L}}\left(-center(Y_i^{(k)}(x'))\right) \middle| \|center(\Delta_i)\| > \frac{\rho}{2}\sqrt{\frac{d}{D}}\right) \\
\geq& (1 - O(n^{2-(d-1)\zeta/L}) - \exp(-\Theta(d)))
\end{aligned}$$

This concludes the proof.

Remark Note that we assumed for notational simplicity that layer norm only applies a single time, following the MLP (4). Depending on the implementation, LN may appear in other cases, e.g. directly following attention (e.g., in Vaswani et al. (2017)). In this case, we can analogously consider the the effect of $V + \Delta_V$, using the properties of the random matrix Δ_V and the effects of the perturbation on the preceding MLP (or the token embeddings). First, we note that $\eta_i := \sum_h \sum_j \widehat{a}_{ij}^{(k,h)} y_j^{(k-1)}$ is, after adding Δ to the parameters involved, very unlikely to have norm $o(n^{1/(\zeta L)})$, because of the random perturbation applied to the MLP bias in the preceding layer (or to the token embeddings, if $k = 1$), provided $d \gg H$. Second, considering the spectral properties of the random matrix Δ_V, the perturbed head activation $y_i^{(k-1)} + (V + \Delta_i)\eta_i$ is unlikely to have much smaller SD, hence a bound analogous to (73) follows with high probability.

□

D Theorem 7

Proof. Each autoregressive step consists in determining, based on a sequence $x_1 \dots x_N t_1 \dots t_t$ the state $t_{t+1} = T(t_t, x_t)$. Assuming $\log|\Sigma|$ and $\log|X|$ bits are used to encode x_i and t_i, respectively, the function thus only depends on $\log|\Sigma||X|$ inputs, and is represented by a polynomial of that degree, independent of N. □

E Further Experimental Results

E.1 The Details of Experimental Setup

Parameter Name	Value
Learning Rate	0.0003
Weight Decay	0.1
Batch Size	1024
Training Steps	10,000
Optimizer	AdamW
ρ (in computation of sharpness)	0.02

Parameter Name	Value
Hidden Dimension	128
Transformer Layers	2
Attention Heads	2
Dropout Rate	0

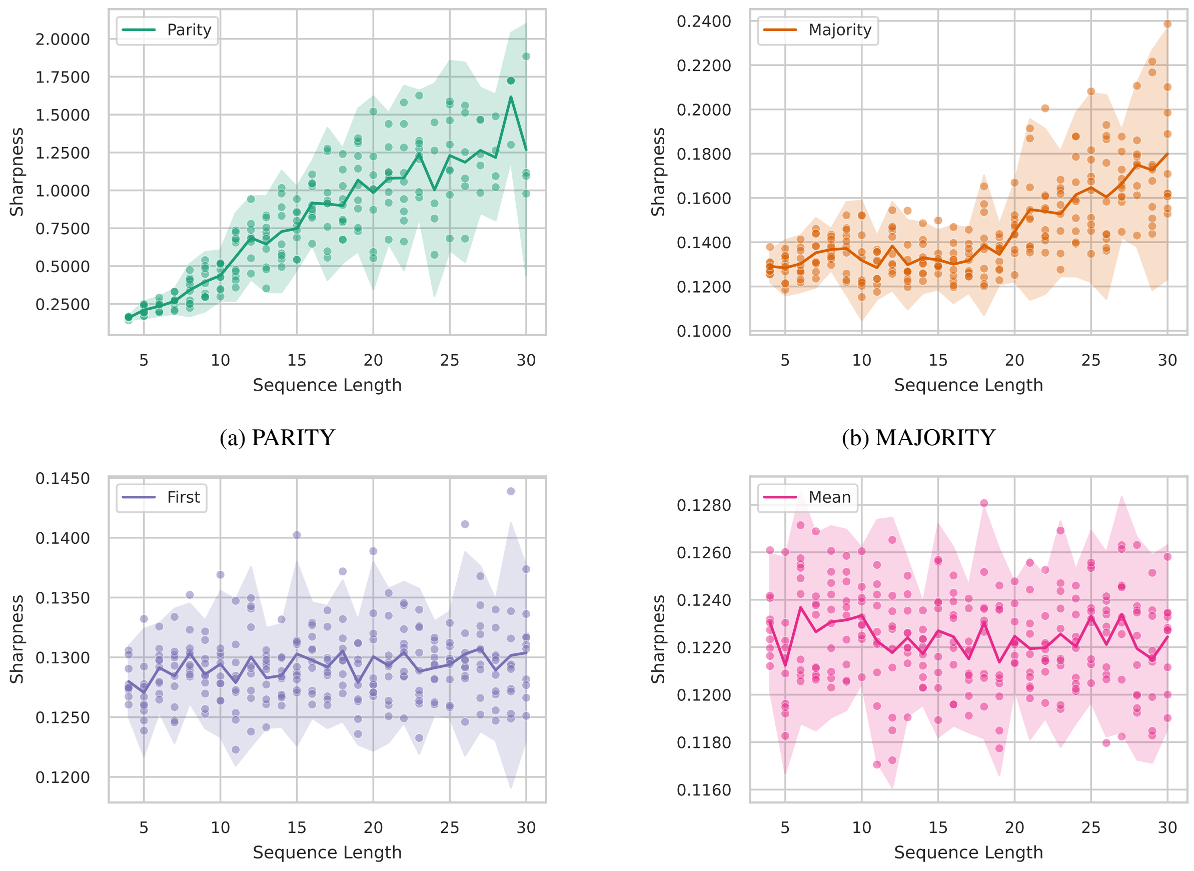
Table 1: In all the experiments, the hyperparameters above were used unless stated otherwise.

In the experiment presenting the tradeoff between weight norm and LayerNorm blowup, we uniformly sampled weight decay between 0 and 0.4 and learning rate between 0.0001 and 0.0005.

In the scratchpad experiment, we used an encoder-decoder Transformer with 2 attention heads per block, hidden dimension 32 and 2 layers.

In the experiment including training the models for high sequence lengths (Figures 7 and 8) we used hidden dimension 4.

E.2 Results



(a) PARITY

(b) MAJORITY

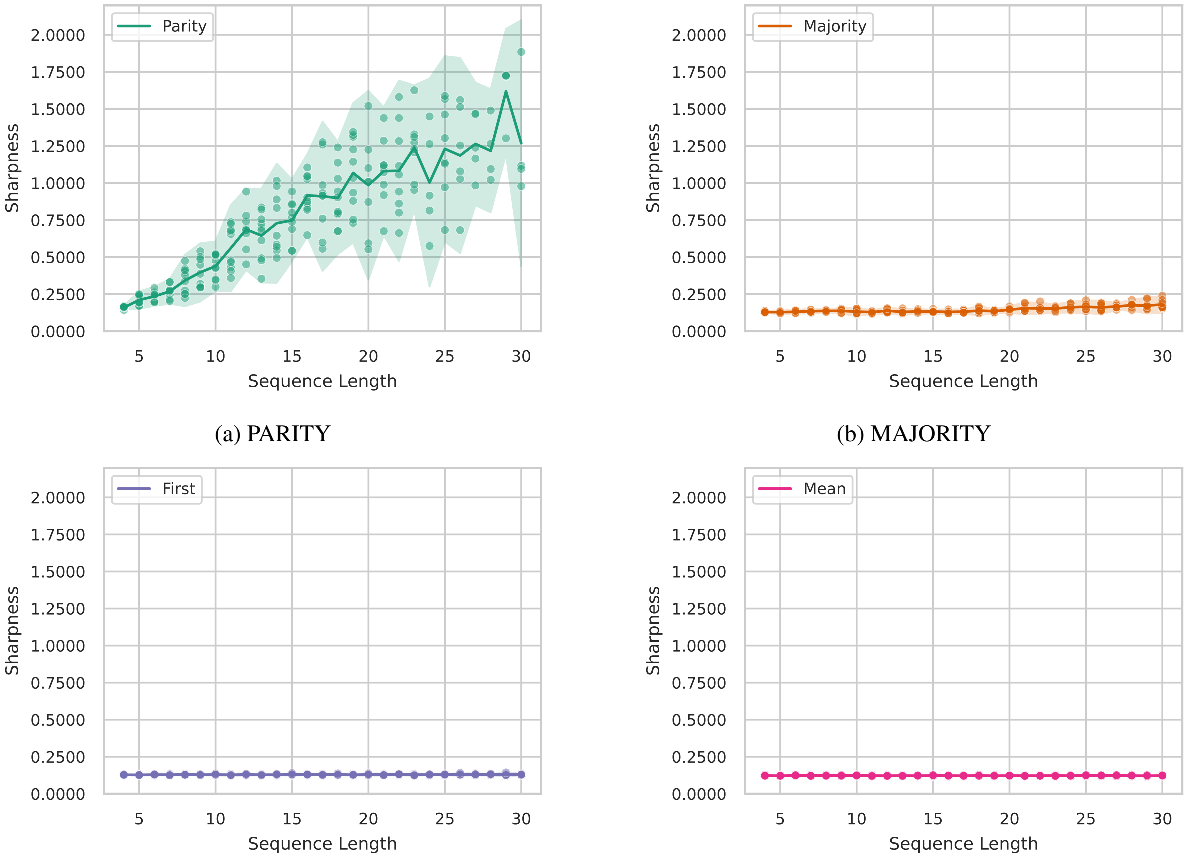
(c) FIRST

(d) MEAN

Figure 5: Sharpness as a function of sequence length for all the functions discussed in the paper. As predicted by Theorem 6, parameters fitting PARITY have substantial sharpness as inputs get longer. For functions with lower sensitivity, sharpness barely increases with the input length. For PARITY, the sharpness approaches the theoretical asymptotic lower bound of 1 from Theorem 6 already at $n \approx 30$. See Figure 6 for a version with aligned y-axes.

Function	**Regression Slope**	**Pearson Correlation**	**p-value** (H_0 : slope $= 0$)
PARITY	$5.0 \cdot 10^{-2}$	0.88	$6 \cdot 10^{-72}$
MAJORITY	$1.8 \cdot 10^{-3}$	0.67	$2 \cdot 10^{-36}$
FIRST	$6.4 \cdot 10^{-5}$	0.16	0.0075
MEAN	$-1.9 \cdot 10^{-5}$	-0.07	0.22

Table 2: The statistical properties of the length-sharpness relationship computed for $n \in [4, 30]$. For all the functions except for MEAN, sequence length and minima sharpness are statistically significantly correlated within this range of n; though for PARITY the slope is 2 orders of magnitude higher than for other functions. See the visualisations in Figure 5.



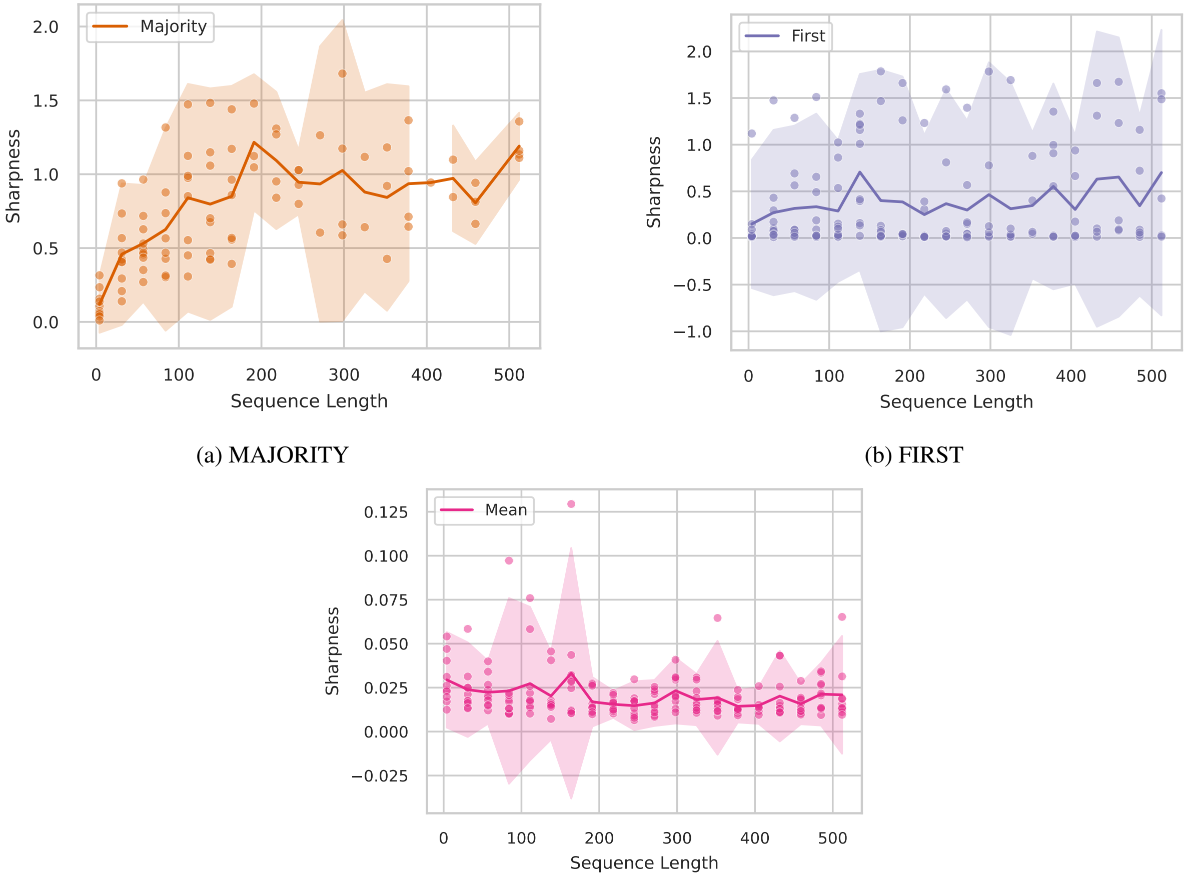
(a) PARITY

(b) MAJORITY

(c) FIRST

(d) MEAN

Figure 6: Same as Figure 5, but *y*-axes are aligned.

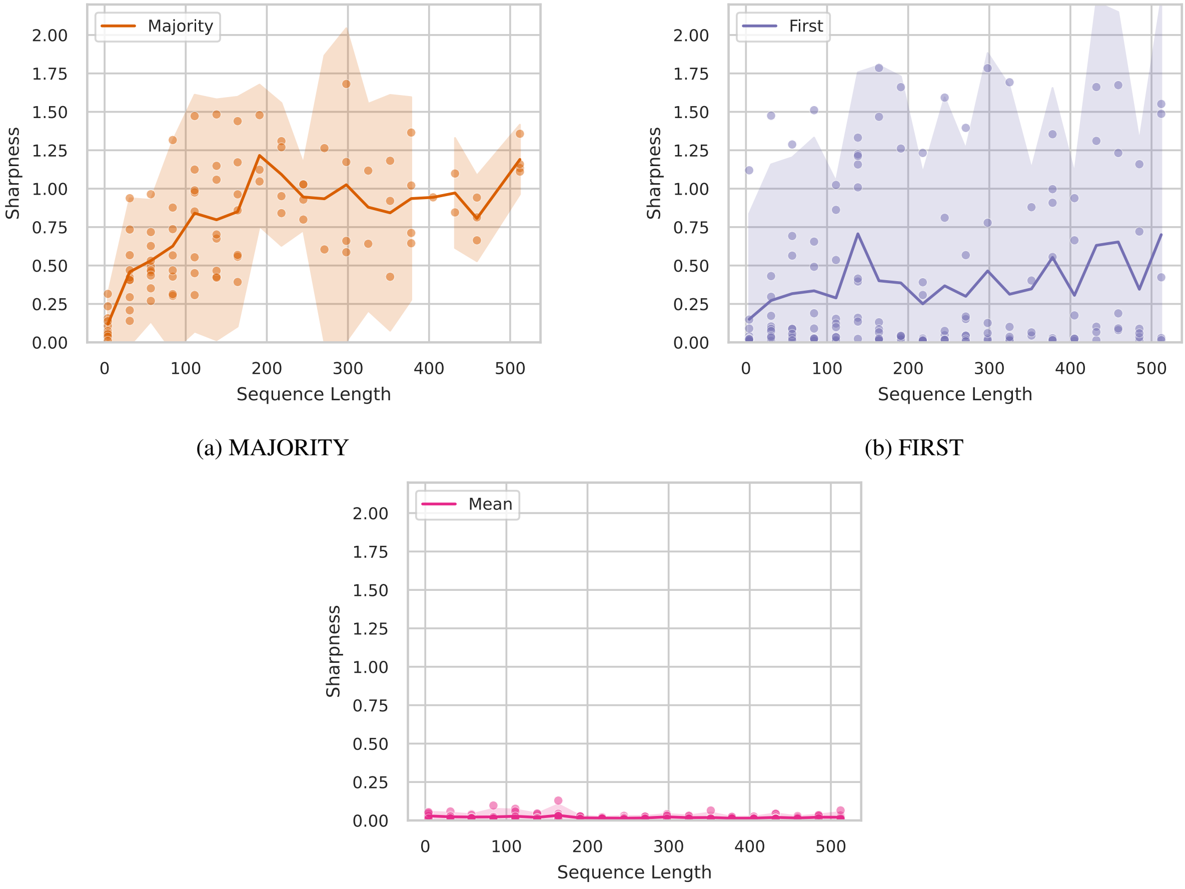


(a) MAJORITY

(b) FIRST

(c) MEAN

Figure 7: Length-Sharpness dependency for high sequence lengths. Note that training a comparable setup for PARITY at such input lengths was not feasible. For MAJORITY, sharpness does increase, in line with its higher (though still sublinear) asymptotic average sensitivity. For FIRST and MEAN, whose average sensitivity does not increase with n, sharpness shows little discernible dependency on length. We note that the absolute sharpness values for high input lengths (as in this figure) and for low input lengths (as in Figure 5) are incomparable, as different model hyperparameters were used, due to the computational cost of fitting models at high sequence lengths.

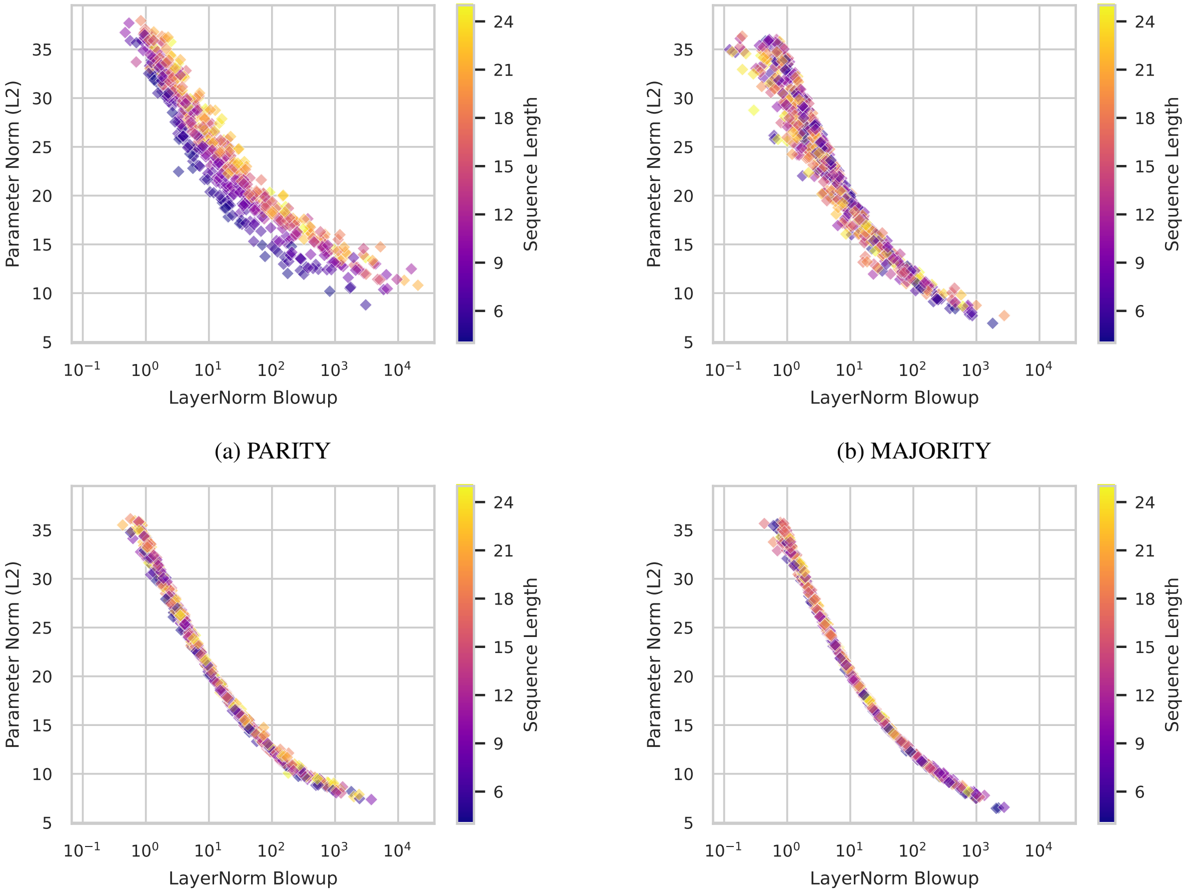


(a) MAJORITY

(b) FIRST

(c) MEAN

Figure 8: Same as Figure 7, but *y*-axes are aligned.



(a) PARITY

(b) MAJORITY

(c) FIRST

(d) MEAN

Figure 9: Tradeoff between layer norm blowup and parameter norm: When varying the layer norm penalty, transformers find different tradeoffs between these two quantities. For PARITY, the tradeoff depends on the input length; blowup or parameter weights need to increase with the input length (in accordance with Corollary 5). For functions with lower sensitivity, little dependency on the input length is observed.

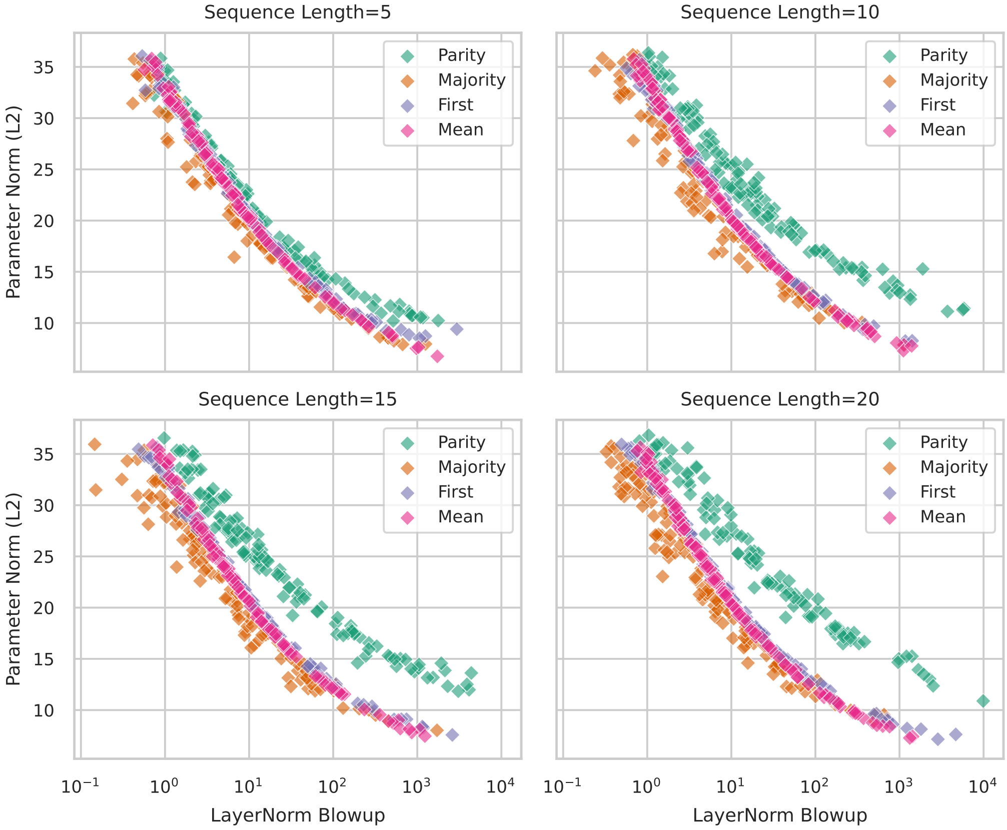


Figure 10: Parameter Norm/Blowup tradeoff for input lengths 5, 10, 15 and 20. At input length 5, PARITY behaves similarly to the other functions. However, for higher input lengths, fixed LN Blowup requires higher weight norm for PARITY than for other functions, and vice versa. This aligns with the theory, as increasing input length for PARITY increases the lower bound of weight norm and LN blowup in (13).

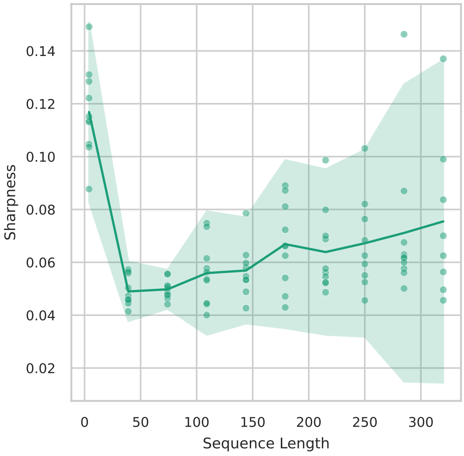
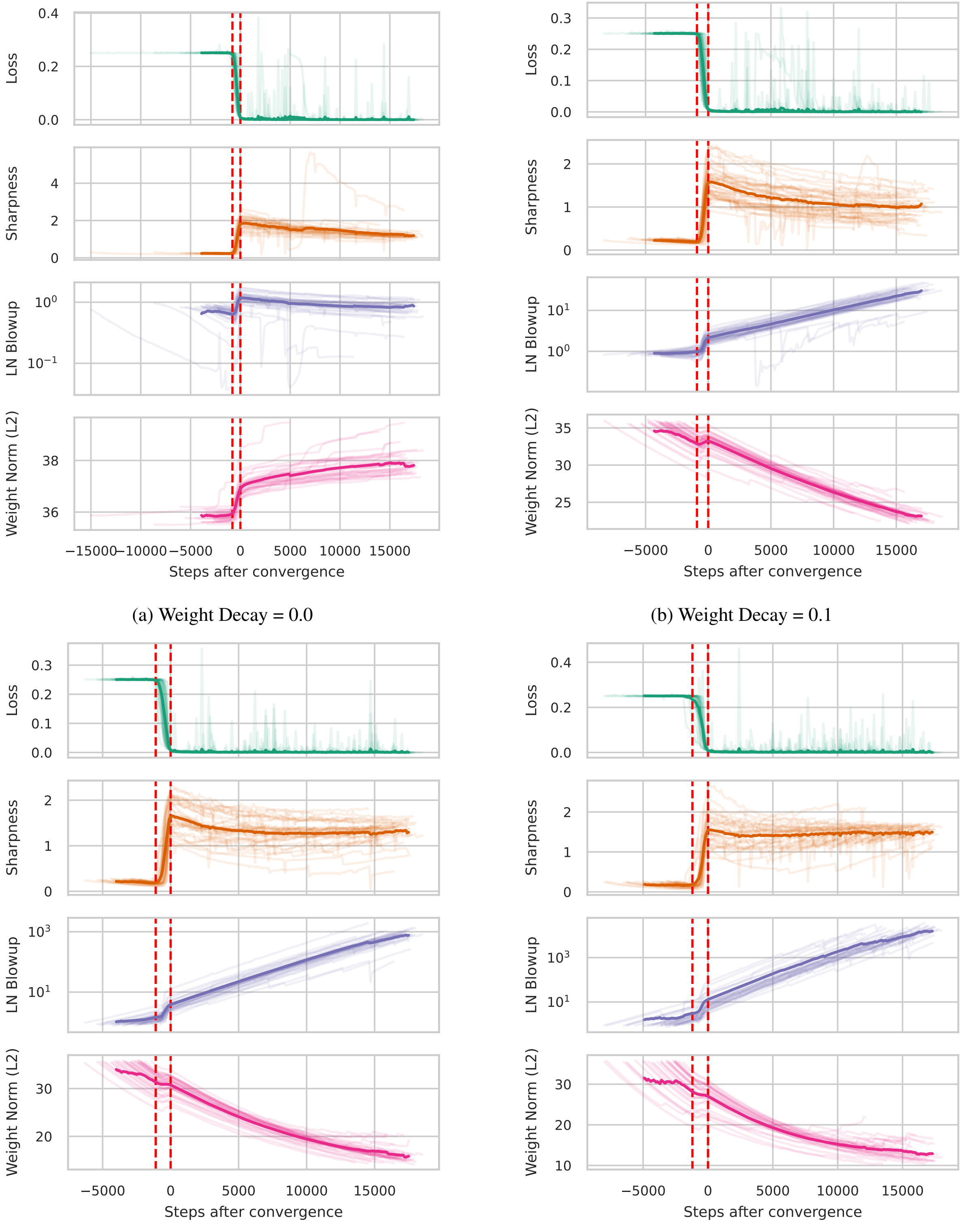


Figure 11: Sharpness of a Transformer with scratchpad trained for PARITY. Despite approximating a highly sensitive function, sharpness stays low even at hundreds of bits and shows no clear increase with length.



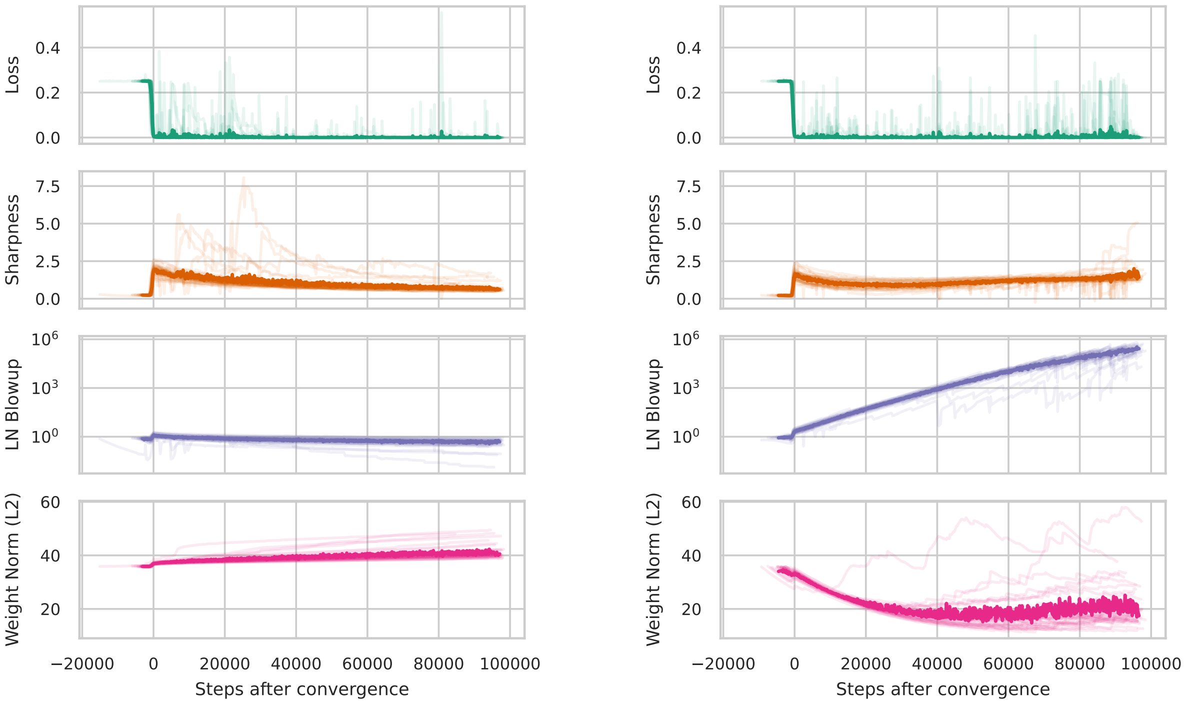
(a) Weight Decay = 0.0

(b) Weight Decay = 0.1

(c) Weight Decay = 0.2

(d) Weight Decay = 0.3

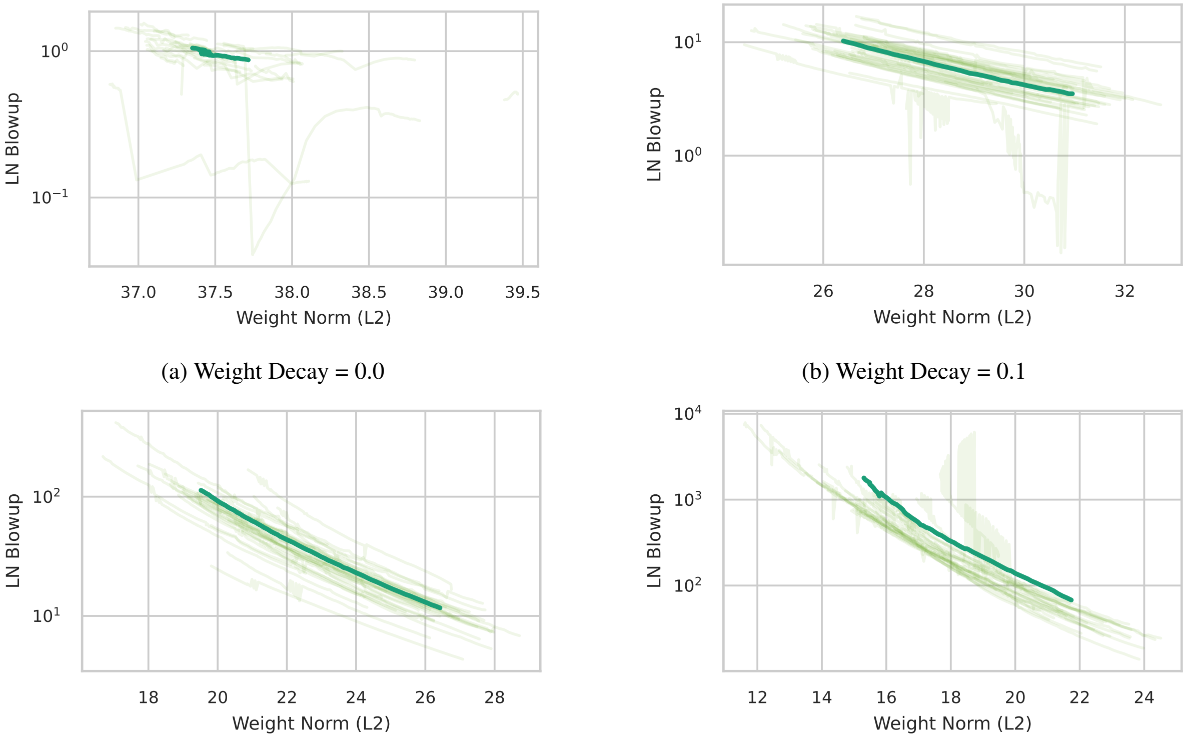
Figure 12: Training dynamics of a Transformer for PARITY at different values of weight decay. Sharpness always increases dramatically at the same time as training loss falls to 0. For non-zero weight decay, after the learning phase parameter norm starts to decrease, causing the appropriate increase in LayerNorm blowup, which does not have a significant effect on sharpness.



(a) Weight Decay = 0.0

(b) Weight Decay = 0.1

Figure 13: Training dynamics at weight decay 0.0 (left) and 0.1 (right), 100k training steps, PARITY. As predicted by Corollary 5, over the course of training, layer norm blowup and parameter norm trade off. Zero weight decay ultimately enables a lower sharpness, but even after 100k steps, it remains substantially higher than the low sharpness quickly reached on low-sensitivity functions.



(a) Weight Decay = 0.0

(b) Weight Decay = 0.1

(c) Weight Decay = 0.2

(d) Weight Decay = 0.3

Figure 14: Tradeoff between Weight Norm and LayerNorm Blowup when training with different weight decay values. In the course of training, non-zero weight decay drives down parameter norm, and that makes LN Blowup to increase. The observed dependency is log-linear, similar to the results in Figure 10.

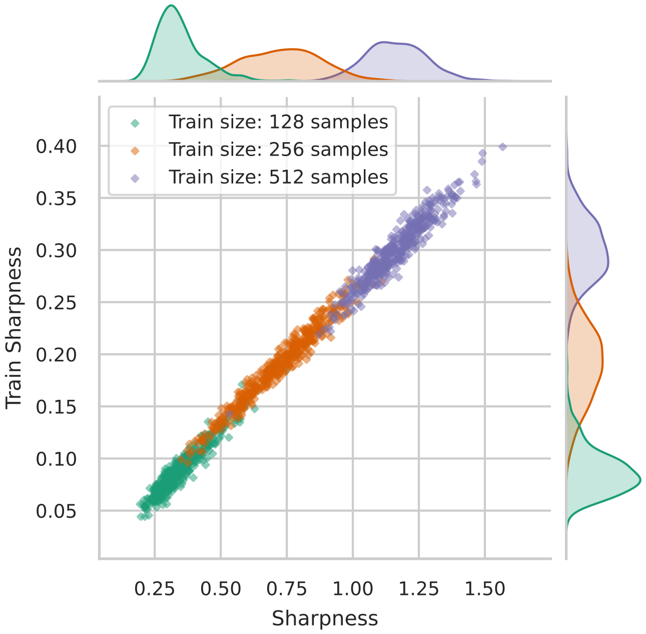


Figure 15: Generalization: The sharpness of the model when restricting sharpness (15) to the train set is almost the same as the sharpness on the whole input space.