Quantitative Bounds for Length Generalization in Transformers

Zachary Izzo [*1] **Eshaan Nichani** [*2] **Jason Lee** [2]

Abstract

We provide quantitative bounds on the length of sequences required to be observed during training for a transformer to length generalize, e.g., to continue to perform well on sequences unseen during training. Our results improve on Huang et al. (2024), who show that there is a finite training length beyond which length generalization is guaranteed, but for which they do not provide quantitative bounds.

1. Introduction

An important problem that has arisen in the training of large language models (LLMs) is length generalization, which is the ability to generalize to input sequences longer than those encountered during training. Prior works have studied the ability of transformers to length generalize on simple testbed tasks (Anil et al., 2022; Kazemnejad et al., 2023), yet the success of length-generalization varies widely from task to task. Recent theoretical work has thus sought to characterize which tasks admit length generalization.

In particular, Zhou et al. (2023) introduced the RASP-L conjecture, which states that transformers can length generalize on tasks which are expressible by a "simple" RASP-L program (a variant of the RASP language introduced in Weiss et al. (2021)). Huang et al. (2024) later formalized and partially proved the conjecture, showing that tasks expressible by a limiting object called a "limit transformer," which includes tasks expressible by a C-RASP program (Yang & Chiang, 2024), admit length generalization at some finite training length. However, the results in Huang et al. (2024) are asymptotic in nature; for a fixed task f on which length generalization is possible, it is not specified what the minimum training length is for length generalization to occur.

In this paper, we aim to characterize how long training sequences need to be in order for a transformer to generalize to sequences of arbitrary length. Specifically, we adopt the limit transformer formulation from Huang et al. (2024), and aim to understand the minimum N such that two limit transformers f, g which agree on inputs of length $\leq N$ approximately agree on inputs of arbitrary length. Our main result, Theorem 4.1, is that for one-layer limit transformers, the minimum such N scales monotonically with the parameter norms L, positional embedding periodicity Δ, "locality" parameter τ, and inverse error ε^{-1}.

Our results rely on the use of finite precision calculations in the attention patterns for the transformer. It is known that finite precision is necessary for the transformer to be identifiable from finite input sequences (Huang et al., 2024). This assumption results in a hard attention pattern mechanism for sequences past a certain length. A careful analysis of the possible hard attention patterns allows us to construct auxiliary shorter input sequences on which the two transformers match and whose output must be similar to the output on the original input, leading to the length generalization result.

Altogether, our results make progress towards both characterizing a natural hierarchy of "difficulty" among length-generalizable tasks, and more practically speaking, developing a better understanding of how to scale training context length for LLMs.

[*]Equal contribution [1]NEC Labs America [2]Princeton University. Correspondence to: Zachary Izzo <zach@nec-labs.com>, Eshaan Nichani <eshnich@princeton.edu>.

Accepted at Methods and Opportunities at Small Scale (MOSS), ICML 2025, Vancouver, Canada.

2. Related Work

A number of works have empirically studied the ability of transformers to length generalize on various tasks. Bhattamishra et al. (2020) studies the ability of transformers to length generalize on various formal language tasks. Anil et al. (2022) show that transformers fail to generalize on certain reasoning tasks, unless certain scratchpad prompting techniques are used. Kazemnejad et al. (2023) study the role of various positional encoding schemes on length generalization. Zhou et al. (2023) study length generalization on various algorithmic tasks, and observe that tasks with a short RASP program (Weiss et al., 2021) have better length generalization, leading to their RASP-L conjecture. This is supported by works such as Jelassi et al. (2024), who observe that for the string copying task, transformers can length generalize when there are no repeated tokens, but fail once the string has repeats.

In light of these length generalization challenges, recent works have designed specific positional encoding schemes, such as Alibi (Press et al., 2021) or Abacus (McLeish et al., 2024) to improve length generalization. Prior works have also considered modifying the input with a scratchpad or extra positional information to improve length generalization on arithmetic tasks (Lee et al., 2023; Shen et al., 2023). Most recently, architectural modifications such as looping (Fan et al., 2024) or recurrence (McLeish et al., 2024) have led to length generalization improvements.

Theoretically, Huang et al. (2024) partially resolves the RASP-L conjecture for tasks expressible by limit transformers. Wang et al. (2024) proves that 1-layer transformers trained with GD length generalize on a sparse token selection task. Ahuja & Mansouri (2024) show that a model resembling a self-attention head can length generalize. Golowich et al. (2025) show that an abstraction of the self-attention head can length generalize on tasks which depend on a sparse subset of input tokens.

3. Problem Formulation

3.1. Limit Transformers

We are interested in considering the ability of transformers to generalize to sequences of arbitrary length, but real transformer architectures are limited by a bounded context length. To address this issue, (Huang et al., 2024) introduced the concept of a *limit transformer*. These objects have an infinite context length and generalized positional embeddings, allowing them to distinguish between arbitrarily many positions in their context. The computation of a limit transformer proceeds as follows:

$$\boldsymbol{y}_i^{(0)} = \boldsymbol{E}_{x_i} + \boldsymbol{p}_i, \quad i = 1, \ldots, |x|, \qquad a_{i,j}^{(l,h)} = (\boldsymbol{y}_j^{(l-1)})^\top \boldsymbol{K}_{l,h}^\top \boldsymbol{Q}_{l,h} \boldsymbol{y}_i^{(l-1)} + \phi_{l,h}(j,i),$$

$$\boldsymbol{Y}_i^{(l)} = \boldsymbol{y}_i^{(l-1)} + \sum_{h=1}^{H} \frac{\sum_{j=1}^{i} \exp\left(\log|x| \cdot a_{i,j}^{(l,h)}\right) \boldsymbol{V}_{l,h} \boldsymbol{y}_j^{(l-1)}}{\sum_{j=1}^{i} \exp\left(\log|x| \cdot a_{i,j}^{(l,h)}\right)},$$

$$\boldsymbol{y}_i^{(l)} = \boldsymbol{Y}_i^{(l)} + \boldsymbol{B}_l \cdot \psi_l(\boldsymbol{A}_l \boldsymbol{Y}_i^{(l)} + \boldsymbol{b}_l), \qquad T(x)_i = \boldsymbol{U} \boldsymbol{y}_i^{(L)}.$$

Here x is the input sequence of tokens in a finite vocabulary Σ with token $x_i \in \Sigma$ in the i-th position, $\boldsymbol{E}_{x_i} \in \mathbb{R}^d$ is the embedding of the i-th token, $\boldsymbol{p}_i$ is the i-th (absolute) positional embedding vector. The super- and sub-scripts (l,h) denote the l-th layer of the transformer and the h-th attention head. $a_{i,j}^{(l,h)}$ is the (l,h) attention logit between token i and j, $\boldsymbol{K}_{l,h}$, $\boldsymbol{Q}_{l,h}$, and $\boldsymbol{V}_{l,h}$ are the the (l,h) key, query, and value embedding matrices, respectively. The functions $\phi_{l,h}(j,i)$ allow for modifications to the attention pattern which cannot be captured by positional embedding vectors alone. $\boldsymbol{Y}_i^{(l)}$ denote the pre-activation features for layer l at position i, and $\boldsymbol{y}_i^{(l)}$ denote the post-activation features which have been passed through a single-hidden-layer MLP with activation ψ_l, plus a residual connection; $\boldsymbol{A}_l$ and $\boldsymbol{b}_l$ denote the hidden layer weights and bias term for this MLP, and $\boldsymbol{B}_l$ denotes the output layer weights. Finally, $T(x)_i$ denotes the output logits at position i which are computed via the unembedding matrix $\boldsymbol{U}$.

Without additional constraints, a limit transformer cannot be recovered without seeing arbitrarily long input sequences. Thus, (Huang et al., 2024) also make two additional assumptions. First, the limit transformers in question are assumed to be Δ-*periodic*, defined as $\boldsymbol{p}_i = \boldsymbol{p}_{i+\Delta}$ for all i. Second, the limit transformers are also *translation-invariant*, defined as $\phi_{l,h}(j,i) = \phi_{l,h}(j+t, i+t)$ for all t and τ-*local*, defined as $\phi_{l,h}(j,i) = 0$ whenever $i > j + \tau$.

3.2. Finite-Precision Attention

The final assumption placed on limit transformers is also the key tool for simplifying our analysis. Specifically, (Huang et al., 2024) assume that all of the transformer parameters, as well as the softmax attention, are computed at p finite bits of

precision. This is motivated by (Merrill & Sabharwal, 2023).

The precise instantiation of this assumption that we will assume that all quantities of absolute value $\leq 2^{-p}$ are rounded to 0 during each intermediate computation of the limit transformer. Even this definition requires further clarification, particularly for the computation of the softmax. This is because the softmax (at infinite precision) is invariant to a constant shift in all of the logits; thus, in principal, the softmax may be computed as a collection of terms each of which has absolute value less than 2^{-p}, in which case it is unclear what to do. To avoid this problem, we take the usual step for improving the numerical stability of softmax and perform computations with the largest logit shifted to 0. Equivalently, we subtract the largest logit from every logit in the softmax. After this standardization, all terms in the softmax (post exponentiation) with absolute value at most 2^{-p} are rounded to 0, then the computation proceeds as usual.

The impact of this assumption is as follows. Let f be a single-layer limit transformer which is τ-local, Δ-periodic, and translation invariant as defined above. We can define the attention matrix $A \in \mathbb{R}^{\Delta|\Sigma|\times\Delta|\Sigma|}$ indexed by pairs (y, i) for $y \in \Sigma$ and $i \in \mathbb{Z}/\Delta$, where

$$A_{(y,i),(z,j)} := (\boldsymbol{E}_z + \boldsymbol{p}_i)^\top K^\top Q(\boldsymbol{E}_y + \boldsymbol{p}_j).$$

For $y \in \Sigma$ and $i \in \mathbb{Z}/\Delta$, define

$$\mathcal{A}_{(y,i)} = \{A_{(y,i),(z,i-k)} + \phi(1, k+1) \mid z \in \Sigma,\, k = 0, \ldots, \tau\}.$$

Note that $\mathcal{A}_y$ contains all of the possible attention logits that we can observe when processing a token $x_i = y$. We then define the *logit margin* $\gamma(f)$ of f by

$$\gamma(f) := \min_{\substack{y\in\Sigma \\ i\in\mathbb{Z}/\Delta}} \min_{\substack{a,a'\in\mathcal{A}_{(y,i)} \\ a-a'>0}} a - a',$$

where the minimum over an empty set is defined as $+\infty$. The quantity $\gamma(f)$ is the smallest nonzero gap we can observe between a maximal attention logit and any non-maximal logit.

Now let x be any input sequence and suppose that $N = |x| \geq 2^{p/\gamma(f)}$. Consider an individual term in the softmax, post-exponentiation but before the rounding procedure. These have the form

$$\begin{aligned} s_j &= \exp\Big(\log N \cdot [(A_{(x_N,N),(x_j,j)} + \phi(j, N)) - (A_{(x_N,N),(x_{j^*},j^*)} + \phi(j^*, N))]\Big) \\ &= \exp\Big(\log N \cdot [(A_{(x_N,N),(x_j,j)} + \phi(1, N-j+1)) - (A_{(x_N,N),(x_{j^*},j^*)} + \phi(1, N-j^*+1))]\Big) \\ &= \exp(\log N \cdot (a - a^*)), \end{aligned}$$

where $j^* \in \operatorname{argmax}_{j'=1,\ldots,i} A_{(x_N,N),(x_{j'},j')} + \phi(j', N)$ is an index with the largest attention logit and $a, a^* \in \mathcal{A}_{(x_N,N)}$ are simply a renaming of the logits to emphasize that these are quantities in $\mathcal{A}_{(x_N,N)}$. The second equation follows by the translation invariance of ϕ.

There are now two cases. If $a = a^*$ (i.e., the j-th position attains maximal attention for the input sequence), then $s_j = \exp(0) = 1$ and this contribution to the softmax will not be affected by the rounding procedure. On the other hand, if $a \neq a^*$ (i.e., the j-th position attains strictly sub-maximal attention for the input sequence), then by definition of $\gamma(f)$, $a - a^* \leq -\gamma(f)$ and we have

$$s_j = \exp(\log N \cdot (a - a^*)) \leq \exp\left(-\frac{p \log 2}{\gamma(f)}\gamma(f)\right) = 2^{-p}.$$

Thus, this term will be rounded to 0. It follows that for sequences x of length $N \geq 2^{p/\gamma(f)}$, *softmax attention acts as a hardmax* and the computation is performed as a uniform average over the tokens with argmax attention.

As can be seen from this analysis, while these design choices may seem like minutiae, they have outsized effects on the analysis, and this fact has been observed in previous work (Jerad et al., 2025). There is also empirical evidence that attention does indeed concentrate on only a few tokens (Bietti et al., 2023; Rogers et al., 2021).

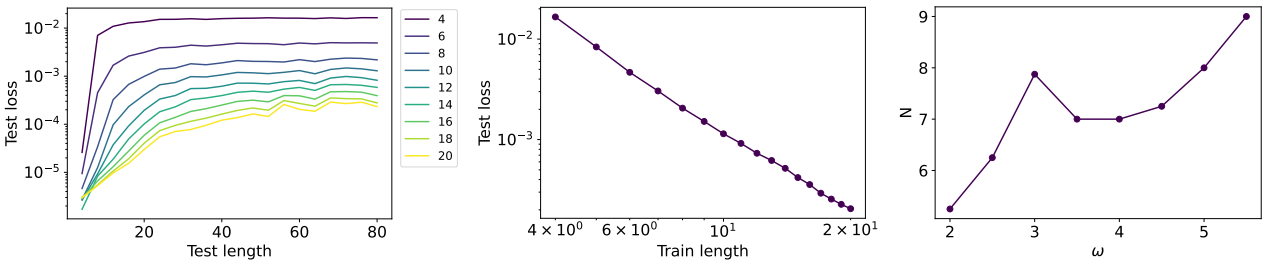

Figure 1. Experiments on SimpleTask. **Left:** For fixed training length, as test length increases, the test loss plateaus at a finite value. **Middle:** The value the test loss plateaus at decreases monotonically with training length. **Right:** As the frequency ω increases (and consequently, the value of L for the equivalent limit transformer increases as $L = \Theta(\omega)$), the minimum training length N needed to obtain fixed test loss ε (here $\varepsilon = 10^{-2}$) increases linearly with ω. Results are averaged over 8 random seeds.

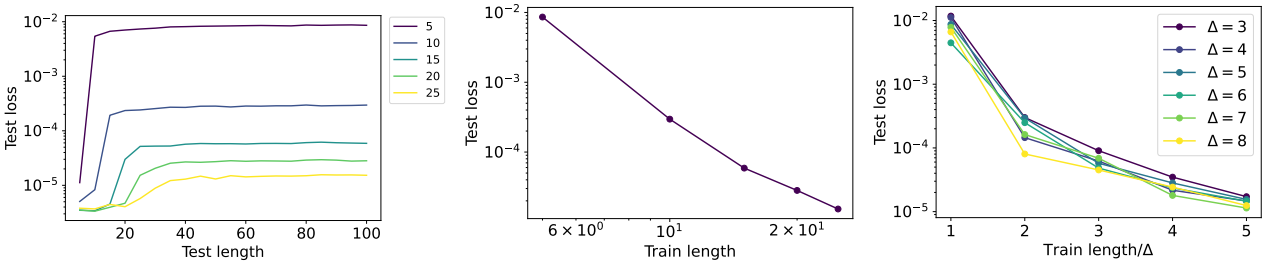

Figure 2. Experiments on ModPTask. **Left:** For fixed training length, as test length increases, the test loss plateaus at a finite value. **Middle:** The value the test loss plateaus at decreases monotonically with training length. **Right:** We plot the limiting test loss as a function of train length for varying values of $\Delta = p$. We observe that the different curves approximately overlap, implying that the training length needed for the limiting test loss to reach some fixed quantity ε scales linearly with Δ. Results are averaged over 8 random seeds.

4. Main Results

Our main result is the following bound on the length of sequences required to appear in training in order for the resulting transformer to length generalize.

Let f and g be two single-layer limit transformers which are τ-local, Δ-periodic, translation invariant, and operate at p finite bits of precision as described in Section 3. Let $\boldsymbol{V}_f, \boldsymbol{E}_s^f, (\boldsymbol{A}_f, \boldsymbol{B}_f)$ be the value matrix, token embedding, and MLP weights for f (and analogously defined for g), and define $L = \max\{\|\boldsymbol{A}_f\|_F \|\boldsymbol{B}_f\|_F \|\boldsymbol{V}_f \boldsymbol{E}_s^f\|,\ \|\boldsymbol{A}_g\|_F \|\boldsymbol{B}_g\|_F \|\boldsymbol{V}_g \boldsymbol{E}_s^g\| \ :\ s \in \Sigma\}$. Finally, let $\gamma = \min\{\gamma(f), \gamma(g)\}$, with $\gamma(f)$ and $\gamma(g)$ as defined in Section 3.2.

Theorem 4.1. *There exists an* $N = O(\max\{2^{p/\gamma}, \frac{\tau \Delta L^2}{\varepsilon^2}\})$ *such that* $|f(x) - g(x)| \leq \delta$ *for all* $|x| \leq N$ *implies that* $|f(x) - g(x)| = O(\delta + \varepsilon)$ *for* any *sequence* x.

Remarks. Theorem 4.1 shows that, assuming that the input sequences are sufficiently long ($N \gtrsim 2^{p/\gamma}$), the desired training length scales polynomially in the periodicity parameter Δ, the parameter norms L, and the inverse accuracy ε. The $N \gtrsim 2^{p/\gamma}$ constraint is to ensure that the softmax in the self-attention behaves as a "hardmax," i.e. uniform attention over a subset of tokens, as discussed in Section 3.2. Indeed, it is possible for this hardmax behavior to occur at smaller training lengths, implying that the training length N need only scale with $\tau \Delta L^2 / \varepsilon^{-2}$. See Section 5 for empirical support of this claim.

Proof Sketch. To prove this result, given any input string x with $|x| > N$, we construct an auxiliary string z of length $|z| \leq N$ which simultaneously approximates the attention patterns of f and g on x. Since $f(z) \approx g(z)$ by assumption, this in turn implies that $f(x) \approx g(x)$. The complete proof is given in Appendix A.1.

5. Experiments

We next provide empirical support for the conclusions of Theorem 4.1 on two synthetic tasks:

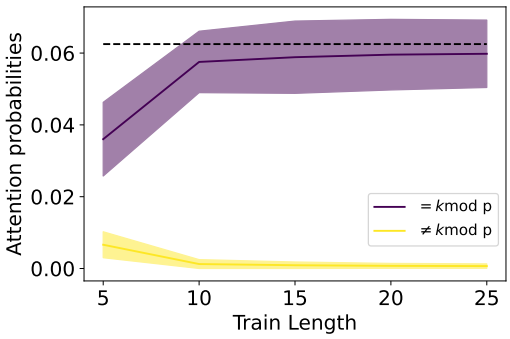

Figure 3. For the $\mathrm{ModPTask}$, the softmax attention weight approximates uniform attention on all positions $\equiv k \mod p$.

- $\mathrm{SimpleTask}$: The vocabulary is $\Sigma = \{0, 1, 2\}$. Given an input sequence $x_{1:T} = (x_1, \ldots, x_T) \in \Sigma^T$, define $c_s(x) = \sum_{t=1}^{T} \mathbf{1}(x_t = s)$ to count the number of tokens equal to s. For some function $\sigma : \mathbb{R} \to \mathbb{R}$, the output is $f^*(x_{1:T}) = \sigma\left(\frac{c_0(x) - c_1(x)}{c_0(x) + c_1(x)}\right)$. We will specifically consider the link function $\sigma(z) = \sin(\omega z)$ for some $\omega \in \mathbb{R}$. One observes that f^* is expressible by a one-layer limit transformer with no positional embeddings and $L = \Theta(\omega)$.
- $\mathrm{ModPTask}$: The vocabulary is $\Sigma = \{0, 1\}$. Given a period p and index k, the output is defined to be the average of all tokens in positions which are $k \mod p$: $f^*(x_{1:T}) = (\sum_{t=1}^{T} \mathbf{1}(x_t = 1, t \equiv k \mod p))/(\sum_{t=1}^{T} \mathbf{1}(t \equiv k \mod p))$. One observes that f^* is expressible by a limit transformer with $\Delta = p$ and $L = \Theta(1)$.

We show that these tasks are expressible by a single-layer limit transformer in Appendix A.2. We train depth 1 transformers (consisting of a single self-attention layer followed by an MLP layer) on $\mathrm{SimpleTask}$ for varying frequencies ω and $\mathrm{ModPTask}$ for varying periods p. For a fixed training length N, we train models on sequences of length $T \leq N$, and compute the test loss on sequences of length $T' \geq N$. For more details, see Appendix B.

Results for $\mathrm{SimpleTask}$ and $\mathrm{ModPTask}$ are presented in Figures 1 and 2 respectively. In the left panes of both figures, we observe that the test loss plateaus as the test length increases. In the middle panes of both figures, we see that the value at which the test loss plateaus at decreases monotonically with the training length. This provides qualitative support for the conclusions of Theorem 4.1, in particular that (i) given a target accuracy ε, tasks expressible by a one-layer limit transformer have a finite N such that a model which fits the task on sequences up to length N achieves ε error on sequences of all length and (ii) the value of this N increases monotonically as ε increases. Moreover, the rightmost pane in Figure 1 shows that N scales with the parameter norm L, while the right pane in Figure 2 shows that N scales with the periodicity parameter Δ.

The proof of Theorem 4.1 relies on the assumption that the softmax attention acts as a "hardmax", uniformly attending to a subset of tokens. To check the validity of this assumption, we consider models trained on the $\mathrm{ModPTask}$ with $p = 5$ for varying training lengths. For each trained model, we look at the post-softmax attention probabilities on a batch of test sequences of length 80. In Figure 3, we plot the mean and standard deviation of these probabilities for tokens in the $k \mod p$ position and tokens not in the $k \mod p$ position. The positions not equal to $k \mod p$ receive near zero attention probabilities while those in positions equal to $k \mod p$ receive nearly the same attention probability of 1/16 (the dashed black line). This is evidence that the models are indeed operating in the hardmax regime, given sufficient training length.

6. Conclusion

In this work, we made quantitative the results of Huang et al. (2024) for single-layer transformers. Our main result, Theorem 4.1, shows that the minimum training length to acheive length generalization scales as $\tau \Delta L^2 / \varepsilon^2$, for parameter norm L, error ε, and periodicity Δ. Qualitative support for these scalings is presented in Figure 1 and Figure 2.

One interesting direction of future work is to extend our results to transformers with larger depth. In particular, it would be interesting to relate the minimum training length N to other notions of complexity such as the length of the corresponding C-RASP program. Moreover, the results in Huang et al. (2024) and our main theorem assume that we have a limit transformer that agrees with the target task on all sequences of length $\leq N$. An important direction is to extend our analysis to functions which have training error on average over some input distribution.

References

Ahuja, K. and Mansouri, A. On provable length and compositional generalization. *arXiv preprint arXiv:2402.04875*, 2024.

Anil, C., Wu, Y., Andreassen, A., Lewkowycz, A., Misra, V., Ramasesh, V., Slone, A., Gur-Ari, G., Dyer, E., and Neyshabur, B. Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, 35: 38546–38556, 2022.

Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Bhattamishra, S., Ahuja, K., and Goyal, N. On the ability and limitations of transformers to recognize formal languages. *arXiv preprint arXiv:2009.11264*, 2020.

Bietti, A., Cabannes, V., Bouchacourt, D., Jegou, H., and Bottou, L. Birth of a transformer: A memory viewpoint. *Advances in Neural Information Processing Systems*, 36:1560–1588, 2023.

Fan, Y., Du, Y., Ramchandran, K., and Lee, K. Looped transformers for length generalization. *arXiv preprint arXiv:2409.15647*, 2024.

Golowich, N., Jelassi, S., Brandfonbrener, D., Kakade, S. M., and Malach, E. The role of sparsity for length generalization in transformers. *arXiv preprint arXiv:2502.16792*, 2025.

Huang, X., Yang, A., Bhattamishra, S., Sarrof, Y., Krebs, A., Zhou, H., Nakkiran, P., and Hahn, M. A formal framework for understanding length generalization in transformers. *arXiv preprint arXiv:2410.02140*, 2024.

Jelassi, S., Brandfonbrener, D., Kakade, S. M., and Malach, E. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv:2402.01032*, 2024.

Jerad, S., Svete, A., Li, J., and Cotterell, R. Unique hard attention: A tale of two sides. *arXiv preprint arXiv:2503.14615*, 2025.

Kazemnejad, A., Padhi, I., Natesan Ramamurthy, K., Das, P., and Reddy, S. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36:24892–24928, 2023.

Lee, N., Sreenivasan, K., Lee, J. D., Lee, K., and Papailiopoulos, D. Teaching arithmetic to small transformers. *arXiv preprint arXiv:2307.03381*, 2023.

McLeish, S., Bansal, A., Stein, A., Jain, N., Kirchenbauer, J., Bartoldson, B., Kailkhura, B., Bhatele, A., Geiping, J., Schwarzschild, A., et al. Transformers can do arithmetic with the right embeddings. *Advances in Neural Information Processing Systems*, 37:108012–108041, 2024.

Merrill, W. and Sabharwal, A. A logic for expressing log-precision transformers. *Advances in neural information processing systems*, 36:52453–52463, 2023.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Rogers, A., Kovaleva, O., and Rumshisky, A. A primer in bertology: What we know about how bert works. *Transactions of the association for computational linguistics*, 8:842–866, 2021.

Shen, R., Bubeck, S., Eldan, R., Lee, Y. T., Li, Y., and Zhang, Y. Positional description matters for transformers arithmetic. *arXiv preprint arXiv:2311.14737*, 2023.

Wang, Z., Wei, S., Hsu, D., and Lee, J. D. Transformers provably learn sparse token selection while fully-connected nets cannot. *arXiv preprint arXiv:2406.06893*, 2024.

Weiss, G., Goldberg, Y., and Yahav, E. Thinking like transformers. In *International Conference on Machine Learning*, pp. 11080–11090. PMLR, 2021.

Yang, A. and Chiang, D. Counting like transformers: Compiling temporal counting logic into softmax transformers. *arXiv preprint arXiv:2404.04393*, 2024.

Yang, G., Hu, E. J., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

Zhou, H., Bradley, A., Littwin, E., Razin, N., Saremi, O., Susskind, J., Bengio, S., and Nakkiran, P. What algorithms can transformers learn? a study in length generalization. *arXiv preprint arXiv:2310.16028*, 2023.

A. Proofs

A.1. Proof of Theorem 4.1

In this section, we give the proof of our main theorem, which we restate for convenience.

Theorem 4.1. *There exists an* $N = O(\max\{2^{p/\gamma}, \frac{\tau \Delta L^2}{\varepsilon^2}\})$ *such that* $|f(x) - g(x)| \leq \delta$ *for all* $|x| \leq N$ *implies that* $|f(x) - g(x)| = O(\delta + \varepsilon)$ *for* any *sequence* x.

Proof. Consider two limit transformers f and g and an input string x. Let $\mathrm{Lip}(\mathrm{MLP}_f)$ be the Lipschitz constant of the MLP in f. One can bound $\mathrm{Lip}(\mathrm{MLP}_f) \leq \|\boldsymbol{B}_f\|_F \|\boldsymbol{A}_f\|_F$, where $(\boldsymbol{B}_f, \boldsymbol{A}_f)$ are the MLP weights for f; the analogous result holds for g.

Let A_f be the positions attended to by f and A_g be the positions attended to by g *in the* τ*-prefix of* x and assume WLOG that $|A_f| \leq |A_g|$. Let $S_f = \{x_i \,:\, i \in A_f\}$ be the set of tokens which f attends to and similarly $S_g = \{x_i \,:\, i \in A_g\}$. For any $s \in \Sigma$, let n_s be the number of times s occurs in the τ-prefix of x. We construct the auxiliary string z as follows. The τ-suffix of z is always equal to the τ-suffix of x. If $|A_f|, |A_g| \leq \tau/\varepsilon$, then the attention pattern in the τ-prefix of x can be directly recreated simultaneously for f and g using at most $2\tau/\varepsilon$ tokens by just copying the union of the tokens in attention for f and g into z, then we must have $|f(x) - g(x)| = |f(z) - g(z)| \leq \delta$. Thus, we will assume that at least $|A_g| \geq \tau/\varepsilon$.

We first recreate the attention pattern of f. Let m_s denote the number of times a token s occurs in the τ-prefix of z. If $|A_f| \leq \tau/\varepsilon$, then we simply set $z_{1:|A_f|} = x_{A_f}$ (i.e., we set the first $|A_f|$ tokens of z equal to the attention pattern of f on the τ-prefix of x). The tokens which we will add later do not belong to A_f; thus, we will clearly have $f(x) = f(z)$.

If $|A_f| > \tau/\varepsilon$, then for each $s \in S_f$ we define

$$m_s = \left\lfloor \frac{n_s}{|A_f|} \cdot \frac{\tau}{\varepsilon} \right\rfloor.$$

For each m_s, we have $\frac{n_s}{|A_f|}\frac{\tau}{\varepsilon} - 1 \leq m_s \leq \frac{n_s}{|A_f|}\frac{\tau}{\varepsilon}$. Since $\sum_{s \in S_f} n_s = |A_f|$, it follows that

$$\frac{\tau}{\varepsilon} - |S_f| \leq \sum_{s \in S_f} m_s \leq \frac{\tau}{\varepsilon}.$$

Here we have used the fact that $r - 1 \leq \lfloor r \rfloor \leq r$ for any real number r. We will make use of this inequality repeatedly throughout the proof. Thus, for each $s \in S_f$, we have

$$\frac{n_s}{|A_f|} - \frac{\varepsilon}{\tau} = \frac{\frac{n_s}{|A_f|}\frac{\tau}{\varepsilon} - 1}{\tau/\varepsilon} \leq \frac{m_s}{\sum_{s \in S_f} m_s} \leq \frac{\frac{n_s}{|A_f|}\frac{\tau}{\varepsilon}}{\tau/\varepsilon - |S_f|} \leq \frac{n_s}{|A_f|} + \frac{2|S_f|}{\tau}\varepsilon$$

provided that $|S_f|\varepsilon/\tau \leq 1/2$, which will hold for small enough ε. In particular, we have

$$\left| \frac{m_s}{\sum_{s' \in S_f} m_{s'}} - \frac{n_s}{|A_f|} \right| \leq \frac{2|\Sigma|\varepsilon}{\tau} = O(\varepsilon). \tag{1}$$

We can use these inequalities to bound the difference between $f(x)$ and $f(z)$. Since the τ-suffix contributes at most τ $O(L)$ terms to the computation of $f(x)$ and $f(z)$, and $|A_f|$ and $\sum_{s \in S_f} m_s$ are both $\Omega(\tau/\varepsilon)$ terms, the τ-suffix terms contribute at most $O(L\varepsilon)$-sized terms to each of $f(x)$ and $f(z)$. Thus we have

$$\begin{aligned}
\|f(x) - f(z)\| &\leq \mathrm{Lip}(\mathrm{MLP}_f) \left\| \frac{\sum_{s \in S_f} n_s \boldsymbol{V}_f \boldsymbol{E}_s^f}{|A_f|} - \frac{\sum_{s \in S_f} m_s \boldsymbol{V}_f \boldsymbol{E}_s^f}{\sum_{s' \in S_f} m_{s'}} \right\| + O(L\varepsilon) \\
&\leq \|\boldsymbol{B}_f\|_F \|\boldsymbol{A}_f\|_F \sum_{s \in S_f} \left| \frac{n_s}{|A_f|} - \frac{m_s}{\sum_{s' \in S_f} m'_s} \right| \left\| \boldsymbol{V}_f \boldsymbol{E}_s^f \right\| + O(L\varepsilon) \\
&= O(\frac{L|\Sigma|\varepsilon}{\tau}) + O(L\varepsilon) = O((\frac{|\Sigma|}{\tau} + 1)L\varepsilon)
\end{aligned}$$

since we have assumed $\|\boldsymbol{B}_f\|_F \|\boldsymbol{A}_f\|_F \|\boldsymbol{V}_f \boldsymbol{E}_s^f\| = O(L)$ and $|S_f| \leq |\Sigma| = O(1)$. We will refer to the portion of z which has been defined up to now as the *f-prefix* of z.

It now remains to extend z so that it can simulate the behavior of g *without* adding any tokens in S_f so as to preserve the previous calculations. There are now two cases. If $|A_f \cap A_g|/|A_g| \leq \varepsilon$, then we for each $s \in S_g \setminus S_f$ we can set

$$m_s = \left\lfloor \frac{n_s}{|A_g|} \cdot \frac{\tau}{\varepsilon^2} \right\rfloor.$$

Now, observe that

$$\begin{aligned}
\sum_{s \in S_g \setminus S_f} m_s &= \sum_{s \in S_g \setminus S_f} \left\lfloor \frac{n_s}{|A_g|} \frac{\tau}{\varepsilon^2} \right\rfloor \\
&\geq \sum_{s \in S_g \setminus S_f} \frac{n_s}{|A_g|} \frac{\tau}{\varepsilon^2} - |S_g \setminus S_f| \\
&= \frac{|A_g \setminus A_f|}{|A_g|} \frac{\tau}{\varepsilon^2} - |S_g \setminus S_f| \\
&= \frac{|A_g| - |A_g \cap A_f|}{|A_g|} \frac{\tau}{\varepsilon^2} - |S_g \setminus S_f| \\
&= \frac{\tau}{\varepsilon^2} - O(1/\varepsilon).
\end{aligned}$$

We also have

$$\sum_{s \in S_g \setminus S_f} m_s \leq \sum_{s \in S_g} \frac{n_s}{|A_g|} \frac{\tau}{\varepsilon^2} = \frac{\tau}{\varepsilon^2},$$

so $\sum_{s \in S_g \setminus S_f} m_s \approx \tau/\varepsilon^2$ up to lower-order terms. From this we can also deduce that

$$\left| \frac{m_s}{\sum_{s' \in S_g \setminus S_f} m_{s'}} - \frac{n_s}{|A_g|} \right| = O(\frac{\varepsilon^2}{\tau})$$

by roughly the same logic which we used to deduce (1). Since there are at most τ terms from the τ-suffix of z and at most τ/ε terms from the f-prefix of z, these will contribute an $O(L\varepsilon)$ term to the computation of $g(z)$ (as by the computations we have just completed, the denominator for $g(z)$ is roughly τ/ε^2). Since $|A_f \cap A_g|/|A_g| \leq \varepsilon$ and $|A_g| \geq \tau/\varepsilon$, the tokens in $A_f \cap A_g$ *and* the tokens in the τ-suffix together contribute at most $O(L\varepsilon)$ to the computations of $g(x)$. Thus we have

$$\|g(x) - g(z)\| \leq \mathrm{Lip}(\mathrm{MLP}_g) \sum_{s \in S_g \setminus S_f} \left| \frac{m_s}{\sum_{s' \in S_g \setminus S_f} m_{s'}} - \frac{n_s}{|A_g|} \right| \|\boldsymbol{V}_g \boldsymbol{E}_s^g\| + O(\varepsilon) = O(L\varepsilon^2) + O(L\varepsilon) = O(L\varepsilon).$$

This completes the case when $|A_f \cap A_g|/|A_g| \leq \varepsilon$.

Otherwise we have $|A_f \cap A_g|/|A_g| > \varepsilon$. In this case, we have $|A_g| < |A_f \cap A_g|/\varepsilon \leq |A_f|/\varepsilon$. Let $s^* = \mathrm{argmax}_{s \in A_f} n_s$. For $s \in A_g \setminus A_f$, we define m_s by

$$m_s = \left\lfloor \frac{m_{s^*}}{n_{s^*}} \cdot n_s \right\rfloor.$$

Note that since $|A_f| \geq \tau/\varepsilon$ and $|A_f| \leq |\Sigma|$, we must have $n_{s^*} \geq \tau/|\Sigma|\varepsilon$. This now allows us to bound the scaling ratio m_{s^*}/n_{s^*}. We have $m_{s^*} = \lfloor \frac{n_{s^*}}{|A_f|} \frac{\tau}{\varepsilon} \rfloor$, so

$$\frac{\tau/\varepsilon}{|A_f|} - \frac{\varepsilon|\Sigma|}{\tau} \leq \frac{\frac{n_{s^*}}{|A_f|} \frac{\tau}{\varepsilon} - 1}{n_{s^*}} \leq \frac{m_{s^*}}{n_{s^*}} \leq \frac{\frac{n_{s^*}}{|A_f|} \frac{\tau}{\varepsilon}}{n_{s^*}} = \frac{\tau/\varepsilon}{|A_f|}.$$

First, we have

$$\sum_{s \in S_g \setminus S_f} m_s \leq \sum_{s \in S_g \setminus S_f} \frac{m_{s^*}}{n_{s^*}} n_s = \frac{m_{s^*}}{n_{s^*}} |A_g \setminus A_f| \leq \frac{m_{s^*}}{n_{s^*}} |A_g| \leq \frac{\tau/\varepsilon}{|A_f|} \frac{|A_f|}{\varepsilon} = \tau/\varepsilon^2.$$

In particular, this implies that this construction can be completed by adding at most τ/ε^2 tokens to z, so in all cases the length of z is $O(\tau/\varepsilon^2)$ as desired.

Next, for $s \in S_g \setminus S_f$, we have the following bounds on $\frac{m_s}{m_{s^*}} = \frac{\lfloor m_{s^*} n_s / n_{s^*} \rfloor}{m_{s^*}}$:

$$\frac{n_s}{n_{s^*}} - \frac{|\Sigma|\varepsilon}{\tau - |\Sigma|\varepsilon} \leq \frac{n_s}{n_{s^*}} - \frac{1}{m_{s^*}} \leq \frac{\lfloor m_{s^*} n_s / n_{s^*} \rfloor}{m_{s^*}} \leq \frac{n_s}{n_{s^*}}.$$

The leftmost inequality uses the fact that $m_{s^*} = \lfloor \frac{n_{s^*}}{|A_f|} \frac{\tau}{\varepsilon} \rfloor \geq \lfloor \frac{1}{|\Sigma|} \frac{\tau}{\varepsilon} \rfloor \geq \frac{\tau}{|\Sigma|\varepsilon} - 1$. In particular, this means that $|n_s/n_{s^*} - m_s/m_{s^*}| = O(\varepsilon)$ for $s \in S_g \setminus S_f$.

For $s \in S_g \cap S_f$, we have $m_s/m_{s^*} = \lfloor \frac{n_s}{|A_f|} \frac{\tau}{\varepsilon} \rfloor / \lfloor \frac{n_{s^*}}{|A_f|} \frac{\tau}{\varepsilon} \rfloor$ and a similar bound can be established. For the lower bound, we have

$$\frac{\left\lfloor \frac{n_s}{|A_f|} \frac{\tau}{\varepsilon} \right\rfloor}{\left\lfloor \frac{n_{s^*}}{|A_f|} \frac{\tau}{\varepsilon} \right\rfloor} \geq \frac{\frac{n_s}{|A_f|} \frac{\tau}{\varepsilon} - 1}{\frac{n_{s^*}}{|A_f|} \frac{\tau}{\varepsilon}} \geq \frac{n_s}{n_{s^*}} - \frac{|\Sigma|\varepsilon}{\tau}.$$

For the upper bound, we have

$$\begin{aligned}
\frac{\left\lfloor \frac{n_s}{|A_f|} \frac{\tau}{\varepsilon} \right\rfloor}{\left\lfloor \frac{n_{s^*}}{|A_f|} \frac{\tau}{\varepsilon} \right\rfloor} &\leq \frac{\frac{n_s}{|A_f|} \frac{\tau}{\varepsilon}}{\frac{n_{s^*}}{|A_f|} \frac{\tau}{\varepsilon} - 1} \\
&= \frac{n_s}{n_{s^*}} \left(1 + \frac{1}{\frac{n_{s^*}}{|A_f|} \frac{\tau}{\varepsilon} - 1} \right) \\
&\leq \frac{n_s}{n_{s^*}} + \frac{2|\Sigma|\varepsilon}{\tau}
\end{aligned}$$

for ε small enough ($\tau/|\Sigma|\varepsilon \geq 2$ suffices). Thus we have $|n_s/n_{s^*} - m_s/m_{s^*}| = O(\varepsilon)$ for $s \in S_g \cap S_f$ as well (in fact $O(\frac{|\Sigma|\varepsilon}{\tau})$). Observe that this means, for any $s \in S_g$:

$$\begin{aligned}
\left| \frac{m_s}{\sum_{s' \in S_g} m_{s'}} - \frac{n_s}{\sum_{s' \in S_g} n_{s'}} \right| &= \frac{m_s/m_{s^*}}{\sum_{s' \in S_g} m_{s'}/m_{s^*}} - \frac{n_s/n_{s^*}}{\sum_{s' \in S_g} n_{s'}/n_{s^*}} \\
&= \left| \frac{n_s/n_{s^*} + O(\varepsilon)}{\sum_{s' \in S_g} (n_{s'}/n_{s^*} + O(\varepsilon))} - \frac{n_s/n_{s^*}}{\sum_{s' \in S_g} n_{s'}/n_{s^*}} \right| \\
&= O(\varepsilon).
\end{aligned}$$

Now we compare $g(x)$ and $g(z)$. As before, the effect of the τ-prefix contributes at most $O(L\varepsilon)$ to $\|g(x) - g(z)\|$, so we have

$$\|g(x) - g(z)\| \leq \mathrm{Lip}(\mathrm{MLP}_g) \sum_{s \in S_g} \left| \frac{m_s}{\sum_{s' \in S_g} m_{s'}} - \frac{n_s}{\sum_{s' \in S_g} n_{s'}} \right| \|\boldsymbol{V}_g \boldsymbol{E}_s^g\| + O(L\varepsilon) = O(L\varepsilon).$$

In all cases, we have constructed z such that $\|f(x) - f(z)\| = O(L\varepsilon)$ and $\|g(x) - g(z)\| = O(L\varepsilon)$, and the length of z is $O(\tau/\varepsilon^2)$. Thus, provided that f and g differ by at most δ on inputs up to a length $N_0 = O(\tau/\varepsilon^2)$, we have

$$\|f(x) - g(x)\| \leq \|f(z) - g(z)\| + \|f(x) - f(z)\| + \|g(x) - g(z)\| \leq \delta + O(L\varepsilon).$$

The proof is completed by substituting $\varepsilon \mapsto \varepsilon/L$, whereby we see that we can obtain error $O(\delta + \varepsilon)$ with $N_0 = O(\tau L^2/\varepsilon^2)$ as desired.

Including positional embedding vectors The setting with positional embedding vectors can be reduced to the general vocabulary case at the cost of an additional factor of Δ by considering each possible $(\text{token}, \text{position mod } \Delta)$ combination as its own token without positional embedding vectors. □

A.2. Expressivity of Synthetic Tasks

We sketch the constructions for each of the synthetic tasks in Section 5.

SimpleTask: Set $\boldsymbol{p}_i = 0$, and let $\boldsymbol{E}_0, \boldsymbol{E}_1, \boldsymbol{E}_2$ be orthogonal. Choose $\boldsymbol{K}, \boldsymbol{Q}$ so that $a_{i,j} = \infty$ when $j = 0, 1$ and $a_{i,j} = 0$ when $j = 2$. The attention probabilities will then be uniform over all 0 and 1 tokens, and thus the output of self-attention becomes $\boldsymbol{Y}_T = \boldsymbol{E}_{x_T} + \frac{c_0(x)}{c_0(x)+c_1(x)}\boldsymbol{V}\boldsymbol{E}_0 + \frac{c_1(x)}{c_0(x)+c_1(x)}\boldsymbol{V}\boldsymbol{E}_1$. We can then set $\boldsymbol{V}\boldsymbol{E}_0 = -\boldsymbol{V}\boldsymbol{E}_1$. It suffices to approximate the one-dimensional function $z \mapsto \sin(\omega z)$ with an MLP; it is well known (Barron, 1993) that this can be done with weight norms $\Theta(\omega)$, as desired.

ModPTask: Let $\{\boldsymbol{q}_i\}_{i\in[\Delta]}$ be some fixed set of orthogonal embeddings, and let $\boldsymbol{p}_i$ be equal to $\boldsymbol{q}_j$, where $i \neq j \mod p$. These are periodic embeddings with periodicity $\Delta = p$. Choose $\boldsymbol{K}, \boldsymbol{Q}$ so that $a_{i,j}$ equals ∞ if $j \equiv k \mod p$ and 0 otherwise. The attention probabilities will then be uniform over all positions which are $k \mod p$. Choosing $\boldsymbol{V}$ so that $\boldsymbol{V}\boldsymbol{q}_j = 0$ for all j, the output of self-attention becomes $\boldsymbol{Y}_T = \boldsymbol{y}_T + f^*(x_{1:T})\boldsymbol{V}\boldsymbol{E}_1 + (1 - f^*(x_{1:T})\boldsymbol{V}\boldsymbol{E}_0$. Choosing the readout layer appropriately, we can ensure that $T(x)_T = f^*(x_{1:T})$, as desired.

B. Experimental Methodology

Training Procedure: The model architecture is one layer of a single self-attention head followed by an MLP. The embedding dimension is $d = 16$ and the MLP width is 256. We use the μP initialization (Yang et al., 2022), and train using the Adam optimizer with learning rate $\eta = 10^{-2}/d$ for the hidden layers and $\eta = 10^{-2}$ for the embedding layers. We train all of the models using online SGD (sampling a fresh batch of size 1024 at each step), until the training loss crosses below 10^{-5}.

Data Generation:

- $\mathrm{SimpleTask}$: Each sequence $x_{1:T}$ is generated by first sampling a probability vector $\mathbf{p} \in \mathbb{R}^3$ uniformly at random over the simplex, then sampling each x_i i.i.d, where $x_i = s$ with probability $\mathbf{p}_s$. This ensures that $\mathrm{Var}(f^*) = \Theta(1)$.
- $\mathrm{ModPTask}$: Each sequence $x_{1:T}$ is generated by first generating $q_0, \ldots, q_{p-1}$ i.i.d uniformly from $[0, 1]$. Then, each x_i is sampled from $\mathrm{Bernoulli}(p_k)$, where $k \equiv i \mod p$. This ensures that $\mathrm{Var}(f^*) = \Theta(1)$, and also that attending to incorrect positions mod p cannot help the model.