Compact Proofs of Model Performance via Mechanistic Intepretability

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work, we propose using mechanistic interpretability – techniques for reverse engineering model weights into human-interpretable algorithms – to derive and compactly prove formal guarantees on model performance. We prototype this approach by formally lower bounding the accuracy of 151 small transformers trained on a Max-of-k task. We create 102 different computer-assisted proof strategies and assess their length and tightness of bound on each of our models. Using quantitative metrics, we show that shorter proofs seem to require and provide more mechanistic understanding, and that more faithful mechanistic understanding leads to tighter performance bounds. We confirm these connections by qualitatively examining a subset of our proofs. Finally, we identify compounding structureless noise as a key challenge for using mechanistic interpretability to generate compact proofs on model performance.

1 Introduction

One approach to ensuring the safety and reliability of powerful AI systems is via formally verified proofs of model performance [36, 8]. If we hope to deploy formal verification on increasingly large models [18, 21] with powerful emergent capabilities [45], we will need *compact* proofs of *global robustness* properties. However, existing approaches tend to use proof strategies that suffer from bad asymptotic complexity, while verifying either *local* robustness properties or the correctness or generalization properties of training procedures instead of particular resulting models of interest.

One key challenge is that neural network architectures are highly expressive [39, 47], and models with similar training procedure and performance may still have learned significantly different weights [28, 7]. Thus, using only knowledge of the architecture or training procedure, it can be challenging to develop efficient proof strategies for verifying model performance. We propose verifying model performance using understanding derived from *mechanistic interpretability* (Section 2) – that is, reverse engineering the *specific implementation* of the algorithm from the learned weights of particular models. Knowledge of the specific implementation allows us to construct less lossy simplifications of the model, and more efficiently reason about model performance over possible inputs.

In this work, we provide a case study of translating mechanistic interpretations into compact proofs. We train a set of 151 attention-only transformers on a Max-of-k task (Section 3), and then reverse engineer the models using standard mechanistic interpretability techniques. We use our understanding to define a set of 102 different computer-assisted proof strategies with varying tightness of bound and with different asymptotic complexity and number of required floating-point operations (Section 4).

We define a quantitative metric to assess the mechanistic understanding used in a proof strategy by the dimensionality of the function space that the proof strategy must consider, which we deem the *unexplained dimensionality* of the proof strategy (Subsection 5.1, Appendix L). Using this metric,

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

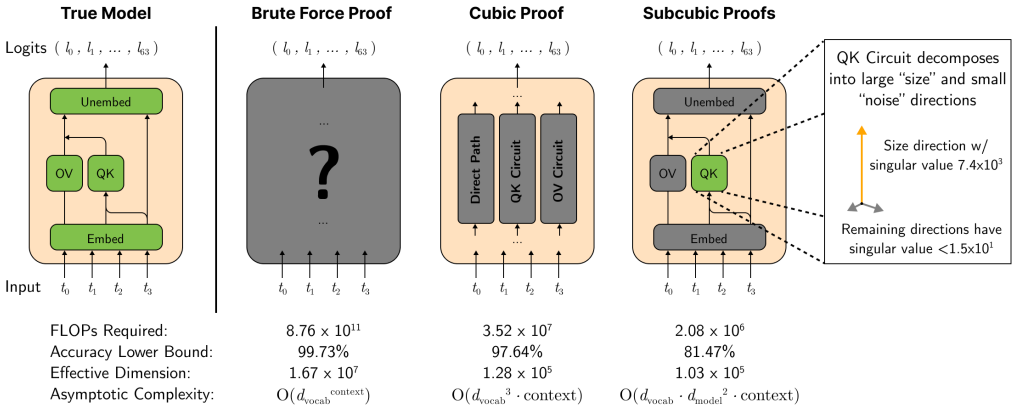


Figure 1: We construct proofs using different degrees of mechanistic interpretation. (Left) The models we consider in this paper are one-layer attention-only transformers, and so contain three "paths": the OV circuit, the QK circuit, and the direct path. (Right) For the brute force proof (Subsubsection 4.2.1), we treat the model as a black box and thus need to check all possible combinations of inputs. For the cubic proof (Subsubsection 4.2.1), we decompose the model into its three corresponding paths, but still check the correctness of each path via brute force. Finally, in some subcubic proofs (Subsection 4.2), we use all parts of the mechanistic interpretation presented in Section 3. (Bottom) For each of the three categories of proof, we report the number of FLOPs used in computing the certificate (lower=better), lower bound on model accuracy (higher=better), effective dimension of the mechanistic understanding used (lower=better), and asymptotic complexity of the proof strategy as we scale the inputs and model (lower=better). Using more mechanistic understanding leads to much shorter proofs, but with worse bounds on accuracy (as our understanding is not fully faithful to the model internals).

we find a negative relationship between proof length and degree of understanding. We qualitatively examine proof strategies to confirm and explain this relationship, finding that shorter proofs both require and provide more mechanistic understanding. We also find suggestive evidence that the trade-off between proof length and tightness of bound is mediated by the faithfulness of the mechanistic understanding used to derive the proof (Subsection 5.2).[1]

However, we also identify compounding structureless noise as a key challenge for generating compact proofs on model behavior (Subsection 5.3). The implementation of algorithms inside of neural networks may contain components that defy mechanistic understanding and appear to us as "noise". When we don't know how noise composes across model components, establishing a bound requires pessimizing over the ways the composition could occur. Worst-case noise can quickly grow over components even in cases when the empirical noise is small, and lead to vacuous performance bounds.

2 Mechanistic interpretability for proofs

In the style of mechanistic interpretability evalution work [5], we target theorem templates that establish bounds on the expected **global performance** of the model. Let $\mathcal{M} : X \to Y$ be a model (here assumed to be a neural network), $\mathcal{D}$ be a probability distribution over inputs $\mathbf{x} \in X$, and $f : X \times Y \to \mathbb{R}$ be a scoring function for evaluating the performance of the model. Then, we seek to establish lower bounds b on the expected s as the form:

$$s := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[f(\mathbf{x}, \mathcal{M}(\mathbf{x}))\right] \geq b. \tag{1}$$

As f can be any metric, this is a fully general template for theorems that can capture any aspect of model performance for which we have a formal specification. However, in this work we restrict f to be the 0-1 loss, so our theorems lower bound the accuracy of the model.

The proofs in this work have two components: a computational component C : model weights $\to \mathbb{R}$ and a non-computational component Q arguing that for any model $\mathcal{M}'$, $C(\mathcal{M}') \leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} f(\mathbf{x}, \mathcal{M}'(\mathbf{x}))$, thus implying that C generates a valid lower bound for the performance of $\mathcal{M}$. The whole proof is Q paired with a trace of running C that certifies its output on $\mathcal{M}$. Here, $b = C(\mathcal{M})$. As even the size of the model parameters is much larger than any reasonable Q, we approximate the length of a proof pair C, Q by the length of a trace of $C(\mathcal{M})$.

[1] Code for reproducing our results can be found at https://github.com/gbmi-team/gbmi.

Proof compactness vs. tightness of bound Different proof strategies make different tradeoffs between compactness and tightness of bound. For example, consider two extreme proof strategies: We can "prove" a vacuous bound using a null proof. On the other hand, in the brute-force proof, we simply run the model on the entirety of $\mathcal{D}$ to achieve $b = s$, albeit with a very long proof.

We quantify the length of $C(\mathcal{M})$ using two metrics: the *asymptotic time complexity* of C as we scale the size of the model and the input $\mathbf{x}$, as well as the empirical average *number of floating point operations* required to evaluate $C(\mathcal{M}')$ over a given set of models $\{\mathcal{M}_i\}$. We measure *tightness of bound* of $C(\mathcal{M})$ using the ratio of the bound to the true accuracy: b/s.

Proof as pessimal ablation A standard way of assessing the faithfulness of mechanistic interpretability is by ablating the parts of the model that your interpretation does not explain [43, 5, 17]. In this framework, proofs can be thought of as performing a *pessimal ablation* over the unexplained parts of the model – we set the remaining components of the model (the "noise") with values drawn from $\mathcal{D}$ that minimize the performance of the model. Compact proofs will also often involve performing a *relaxation* over input sequences, such that establishing performing pessimal ablations on a smaller number of relaxed input sequences is sufficient to lower bound the performance on $\mathcal{D}$.

3 Experimental setting

We study our approach to generating compact proofs in a simple toy setting: one-layer transformers trained to output the max of k integers.

Model Architecture We study one-layer, one-head, attention-only transformers with no biases but with learned positional embeddings, with vocabulary size d_{vocab}, model and head dimension $d = d_{\text{model}} = d$, and context length n_{ctx}. The model parameters consist of the $n_{\text{ctx}} \times d_{\text{model}}$ positional embedding P; the $d_{\text{vocab}} \times d_{\text{model}}$ token embed E; the $d_{\text{model}} \times d_{\text{model}}$ query, key, value, and output matrices of the attention head Q, K, V, and O; as well as the $d_{\text{model}} \times d_{\text{vocab}}$ unembed matrix U. We assume (as is standard in language modeling) that $d_{\text{model}} < d_{\text{vocab}}$.

For an $n_{\text{ctx}} \times d_{\text{vocab}}$ one-hot encoded input sequence $\mathbf{x} = [t_0, t_1, ..., t_{n_{\text{ctx}}-1}]^T$, we compute the logits of the model as follows:

$$
\begin{aligned}
h^{(0)} &= \mathbf{x}E + P && \text{Initial residual stream } (n_{\text{ctx}} \times d_{\text{model}}) \\
\alpha &= h^{(0)} Q K^T {h^{(0)}}^T / \sqrt{d} && \text{Attention matrix } (n_{\text{ctx}} \times n_{\text{ctx}}) \\
h^{(1)} &= \sigma^*(\alpha) \cdot h^{(0)} V O + h^{(0)} && \text{Final residual stream } (n_{\text{ctx}} \times d_{\text{model}}) \\
\mathcal{M}(\mathbf{x}) &= \ell = h^{(1)}_{n_{\text{ctx}}-1} U && \text{Final sequence position logits } (d_{\text{vocab}})
\end{aligned}
$$

where σ^* is the masked softmax function used in causal attention. Because we only look at outputs of the model above the final sequence position $i = n_{\text{ctx}} - 1$, we also denote this position as the query position query and the value of the token in this position as t_{query}. The model's prediction is the token corresponding to the max-valued logit $\ell_{\max}$.

Task Specifically, we study the setting with $n_{\text{ctx}} = k = 4$ because it is the largest sequence length for which we can feasibly evaluate the brute force proof. We set hidden dimension $d_{\text{model}} = 32$ and a vocabulary of size $d_{\text{vocab}} = 64$ comprising integers between 0 and 63 inclusive. For an input sequence $\mathbf{x} = [t_0, t_1, t_2, t_3]^T$, we denote the *true* maximum of the by $t_{\max}$. We trained 151 models on this task. Models achieved an average accuracy of 0.9992 ± 0.0015 over the entire data distribution.

Path decomposition Following prior work [9], we expand the logits of the model and split the paths through the model into three components – the QK circuit, the OV circuit, and the direct path:

$$
\mathcal{M}(\mathbf{x}) = \sigma^*\Big(\underbrace{\left(t_{\text{query}} E + P_{\text{query}}\right) Q K^T \left(\mathbf{x}E + P\right)^T}_{\text{QK circuit}} / \sqrt{d} \Big) \cdot \underbrace{\left(\mathbf{x}E + P\right) V O U}_{\text{OV circuit}} + \underbrace{\left(t_{\text{query}} E + P_{\text{query}}\right) U}_{\text{direct path}}
$$

Intuitively, the QK circuit determines *which* tokens the model attends to from a particular query token and sequence position, while the OV circuit *processes* the tokens and sequence positions that model attends to. The direct path is simply the skip connection around the attention head.

We further divide the QK and OV circuits into token (position-independent) and position-dependent components. Let $P_{\text{avg}} = \frac{1}{n_{\text{ctx}}} \sum_i P_i$ be the average position embeds across positions (of size d_{model}), and let $\bar{\mathbf{P}} = \mathbf{1}_{n_{\text{ctx}}} \otimes P_{\text{avg}}$ represent the result of broadcasting P_{avg} back into the shape of P (that is, $n_{\text{ctx}} \times d_{\text{model}}$). Similarly, let $\mathbf{P}_q = \mathbf{1}_{n_{\text{ctx}}} \otimes P_{\text{query}}$ be the result of broadcasting P_{query}. Then with a slight abuse of notation, we can rewrite the QK and OV circuits, as well as the direct path, as follows:[2]

$$\text{QK circuit} = t_{\text{query}} \left(\underbrace{\mathbf{E}_q Q K^T \bar{\mathbf{E}}^T}_{\text{EQKE}} \mathbf{x}^T + \underbrace{\mathbf{E}_q Q K^T \hat{\mathbf{P}}^T}_{\text{EQKP}} \right)$$

$$\text{OV circuit} = \mathbf{x} \underbrace{\bar{\mathbf{E}} V O U}_{\text{EVOU}} + \underbrace{\hat{\mathbf{P}} V O U}_{\text{PVOU}} \qquad \text{Direct Path} = t_{\text{query}} \underbrace{\mathbf{E}_q U}_{\text{EU}}$$

where $\hat{\mathbf{P}} = P - \bar{\mathbf{P}}$ and $\mathbf{x}\bar{\mathbf{E}} = \mathbf{x}E + \bar{\mathbf{P}}$ and $\mathbf{x}\mathbf{E}_q = \mathbf{x}E + \mathbf{P}_q$ (since $h^{(0)} = \mathbf{x}\bar{\mathbf{E}} + \hat{\mathbf{P}}$).

3.1 Mechanistic interpretation of learned models

Using standard empirical mechanistic interpretability techniques, we interpret one of our learned models (our "mainline" model) by independently examining the QK and OV circuits and the direct path. We find that the model outputs the largest logit on the true max token t_{max} by attending more to larger tokens via the QK circuit and copying the tokens it attends to via the OV circuit. We then quantitatively confirm that these interpretations hold for all 151 models by reporting the mean plus minus standard deviation for various summary statistics. Plots for this section are available in Appendix C.

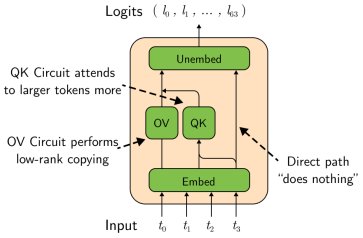


Figure 2: The models in our setting implement max-of-K by attending exponentially more to larger tokens and copying the attended-to tokens (Subsection 3.1).

QK circuit By qualitatively examining the position-independent QK component EQKE, we find the amount of pre-softmax attention paid to a key token is approximately independent of the value of the query token t_{query}, and increases monotonically based on the size of the key token. We confirm this hypothesis by performing a singular-value decomposition (SVD) of the EQKE matrices (Appendix G.3), and find that it contains a single large rank-one component with singular value around 7800 ± 380, around 620 ± 130 times larger than the second largest component with singular value 13 ± 3. The left (query-side) singular vector is approximately constant in all dimensions, with value $0.1243 \pm 0.0003 \approx \frac{1}{8} = \frac{1}{\sqrt{d_{\text{vocab}}}}$. The right (key-side) singular vector of this component is monotonically increasing as we increase the size of the key token, with ($1/\sqrt{d}$-scaled) pre-softmax attention increasing by an average of 1.2176 when the key token increases by 1.[3]

In comparison, each $1/\sqrt{d}$-scaled entry of the position-dependent QK component EQKP has negligible size (average 0.31 ± 0.18), suggesting that EQKP is unimportant to the functioning of the model. We confirm this by zero ablating EQKP. Combined with our interpretation of EQKE, this implies that the attention pattern of the model depends only on the token values and not the ordering of the sequence.

OV circuit Then, by qualitatively examining the position-independent OV component EVOU, we see that it has large positive entries along the diagonal. In fact, the entry along the diagonal is the largest in the row for all rows corresponding to $t > 6.6 \pm 1.2$. Since each entry in the sequence is uniformly sampled and $d_{\text{vocab}} = 64$, this means that EVOU is a good approximation for the identity matrix for all but $(7/64)^4 \approx 1.2 \times 10^{-4}\%$ of the sequences.

As with the position-dependent QK component, the position-dependent OV component PVOU also has negligible size and is unimportant to model performance. Taken together with the above results on EVOU, this suggests that the attention head copies the tokens it attends to.

[2] Including the mean position embed into the token (position-independent) component is a standard technique in prior mechanistic interpretability work, see for example [28, 7].

[3] This implies that the ratio of attention paid to token t and $t-1$ is approximately $\exp(1.2176) = 3.379$.

Table 1: We report the proof complexity, normalized accuracy bound, and estimated flops required (Equation 2), as well as unexplained dimensonality (Section 5). We round the FLOP and unexplained dimension counts to the closest power of 2, and report the mean/standard deviation of the bound averaged across all 151 models. As we include more aspects of the mechanistic interpretation (reflected by a lower number of unexplained dimensions), we get more compact proofs (in terms of both asymptotic complexity and FLOPs), albeit with worse bounds. For space reasons, we use $k := n_{\text{ctx}}$, $d := d_{\text{model}}$, and $v := d_{\text{vocab}}$.

Description of Proof	Complexity Cost	Bound	Est. FLOPs	Unexplained Dimensions
Brute force	$\mathcal{O}(v^{k+1}kd)$	0.9992 ± 0.0015	2^{40}	2^{30}
Cubic	$\mathcal{O}(v^3k^2)$	0.9845 ± 0.0041	2^{25}	2^{14}
Sub-cubic	$\mathcal{O}(v^2 \cdot k^2 + v^2 \cdot d)$	0.832 ± 0.011	2^{21}	2^{13}
without mean+diff		0.758 ± 0.039	2^{21}	2^{13}
Low-rank QK	$\mathcal{O}(v^2k^2 + vd^2 +$	0.797 ± 0.013	2^{22}	2^{12}
SVD only	(EU&OV) $v^2d)$	0.643 ± 0.044	2^{22}	2^{12}
Low-rank EU	$\mathcal{O}(v^2k^2 + vd^2 +$	0.662 ± 0.061	2^{21}	2^{13}
SVD only	(QK&OV) $v^2d)$	$(3.38 \pm 0.06) \times 10^{-6}$	2^{21}	2^{13}
Low-rank QK&EU	$\mathcal{O}(v^2k^2 + vd^2 +$	0.627 ± 0.060	2^{22}	2^{13}
SVD only	(OV) $v^2d)$	$(3.38 \pm 0.06) \times 10^{-6}$	2^{22}	2^{13}

Direct path As with the two position-dependent components, the entries in EU have small absolute magnitude 2.54 ± 0.20,[4] and contribute negligibly to model performance.

4 Proofs of model performance

In this section we describe intuitions for three categories of proof that are developed around different mechanistic interpretations and methods for using the interpretations. The strategies result in proofs of different complexity offering with varying bound tightness (Table 1). We provide detailed theorem statements, proofs, algorithms, and explanations of proof search in the appendices.

The brute-force baseline We start by considering the brute force proof (Appendix D), which treats the model as a black box and evaluates it on all possible sequences. However, this proof strategy quickly becomes untenable as the length of inputs increases. So in subsequent sections, we use knowledge of the model drawn from the interpretation in Subsection 3.1 to derive shorter proofs.

4.1 A cubic proof

Next, we use the fact that the model is composed of the direct path and the QK and OV circuits (Section 3) to decrease the number of sequences that we need to consider, and the fact that only the position-independent components EQKE and EVOU contribute meaningfully to performance (Subsection 3.1) to pessimize over sequence ordering.

First, let a pure sequence ξ be a sequence with at most three distinct tokens: the max token t_{max}, the query token $t_{\text{query}} \leq t_{\text{max}}$, and optionally a third token $t' < t_{\text{max}}$, and let Ξ^{pure} be the set of all pure sequences in $\mathcal{D}$. For a given input sequence $\mathbf{x}$, the define the adjacent pure sequences $\text{Adj}(\mathbf{x})$ as the set of sequences that share the same max and query token, and only take on values in $\mathbf{x}$:

$$\text{Adj}(\mathbf{x}) = \{\xi \in \Xi^{\text{pure}} \mid \max_i \xi_i = t_{\text{max}},\ \xi_{\text{query}} = t_{\text{query}},\ \forall i < n_{\text{ctx}}\ \xi_i \in \mathbf{x}\}$$

Using the convexity of softmax and the fact that the model contains three paths, we can show that one-layer attention-only transformers satisfies a variant of the following convexity property: for a given $\mathbf{x}$, if $\mathcal{M}(\xi)$ is correct for all $\xi \in \text{Adj}(\mathbf{x})$, then $\mathcal{M}(\mathbf{x})$ is correct. That is, for these transformers, we can bound the accuracy on all sequences by evaluating $\mathcal{M}$ on only the $O({d_{\text{vocab}}}^3(n_{\text{ctx}} - 1)!)$ pure sequences. This allows us to bound the accuracy of our actual $\mathcal{M}$ on all ${d_{\text{vocab}}}^{n_{\text{ctx}}}$ sequences, while evaluating it on $O({d_{\text{vocab}}}^3(n_{\text{ctx}} - 1)!)$ sequences.

[4] For comparison, the average off-diagonal element of EVOU is 21.68 ± 0.83 below the corresponding diagonal element.

We can reduce the number of sequences that we need to evaluate by pessimizing over the order of a sequence. For a given tuple of $(t_{\text{max}}, t_{\text{query}}, t')$, there are $(n_{\text{ctx}} - 1)!$ pure sequences, corresponding to the permutations of the tuple. Pessimizing over the order of sequences reduces the number of sequences to consider for each $(t_{\text{max}}, t_{\text{query}}, t')$ tuple to the number of t' in the pure sequence, and the total number of sequences to $O({d_{\text{vocab}}}^3 n_{\text{ctx}})$. By precomputing the five component matrices EU, EQKE, EQKP, EVOU, PVOU and cleverly caching intermediate outputs, we can reduce the additional work of each sequence to the $O(n_{\text{ctx}})$ required to compute the softmax over n_{ctx} elements, resulting in asymptotic complexity $O({d_{\text{vocab}}}^3 {n_{\text{ctx}}}^2)$ (Theorem 12, additional details in Appendix E).

4.2 Sub-cubic proof

We now consider proofs that are more compact than $O({d_{\text{vocab}}}^3)$. These require avoiding iterating over any set of size $O({d_{\text{vocab}}}^3)$ (e.g. the set of pure sequences) and performing operations that take $O(d_{\text{vocab}})$ time on each of $O({d_{\text{vocab}}}^2)$ combinations. Unfortunately, some methods of avoiding these operations can lead to vacuous bounds (i.e. accuracy lower bounds near 0%). In order to recover non-vacuous bounds, we introduce two tricks: the "mean+diff" trick to better approximate the sum of two components with unequal variance, and the "max row diff trick" to improve upon the low-rank approximations for EU and EQKE. We consider applying variants of these tricks at different locations in the naive subcubic proof, leading to 100 distinct subcubic proof strategies. See Appendix G for a formal description of these strategies.

4.2.1 Removing cubic-time computatations

Reducing the number of cases by pessimizing over sufficiently small t' Previously, we consider $\Theta({d_{\text{vocab}}}^3 n_{\text{ctx}})$ pure sequences ξ, with each ξ parameterized by $(t_{\text{max}}, t_{\text{query}}, t', c)$. Recall from our mechanistic interpretation in Subsection 3.1 that the pre-softmax attention paid from t_{query} to a key token t' is invariant to t_{query} and increases linearly with the size of t'. This allows us to pessimize over the OV circuit over all "sufficiently small" tokens.

More formally, suppose we are given some gap $g \in \mathbb{N}$. For each pure sequence ξ with max token t_{max}, query token $t_{\text{query}} \leq t_{\text{max}} - g$, and c copies of the third token type $t \leq t_{\text{max}} - g$, we pessimally ablate the OV circuit over the set $\Xi^{\text{pure}}(t_{\text{max}}, t_{\text{query}}, c; g)$ of pure sequences ξ' with the same max and query tokens and c copies of the third token type $t' \leq t_{\text{max}} - g$. If the model gets all sequences in $\Xi^{\text{pure}}(t_{\text{max}}, t_{\text{query}}, c; g)$ correct, then we can conclude that it gets ξ correct, otherwise, we treat the model as having gotten ξ wrong. This means that it suffices to only consider the $O({d_{\text{vocab}}}^2 n_{\text{ctx}})$ pessimal pure sequences of each of the $O({d_{\text{vocab}}}^2 n_{\text{ctx}})$ sets of the form $\Xi^{\text{pure}}(t_{\text{max}}, t_{\text{query}}, c; g)$.

Decoupling and pessimizing computations that require $O({d_{\text{vocab}}}^3)$ **computations** Many parts of our cubic certificate require iterating through $O({d_{\text{vocab}}}^2)$ cases parameterized by t_{max} and t_{query} or t_{max} and t'. For example, as part of the pessimization procedure over pure sequences, for each of the d_{vocab} possible t_{max}s, we need to consider the relative effects on the d_{vocab}-sized logits of attending to each of the $O(d_{\text{vocab}})$ other tokens $t' < t_{\text{max}}$, and for each t_{max} and t_{query}, we need to check that the contribution of the direct path on logits t_{query}EU is not sufficiently large as to overwhelm the contribution from t_{max}EVOU. We independently pessimize over each of these components over one of the d_{vocab}-sized axes: for example, instead of computing $t_{\text{max}}\text{EVOU} + t_{\text{query}}\text{EU}$ for each t_{max}, t_{query} pair, we first pessimally ablate the direct path along the query token (which takes $O({d_{\text{vocab}}}^2)$ time as it does not depend on the t_{max}, and then consider the sum $t_{\text{max}}\text{EVOU} + \max_{t'} t'\text{EU}$. Since this sum no longer depends on t_{query}, we only need to perform it $O(d_{\text{vocab}})$ times, for a total cost of $O({d_{\text{vocab}}}^2)$.

Low rank approximations to EQKE and EU Recall from Subsection 3.1 that EQKE is approximately rank 1, where the sole direction of variation is the size of the key token. By computing only the rank 1 or rank 2 approximation to EQKE, we can much more cheaply compute the most significant component of the behavior in the QK circuit. To bound the remaining error, we can use the fact that after pulling off the first principle component from each of the four matrices we multiply, very little structure remains.

We can find the rank 1/2 approximations by performing SVD on EQKE. We can efficiently compute the SVD in $\mathcal{O}(d_{\text{vocab}} {d_{\text{model}}}^2)$ time by using the fact that EQKE can be written as the product

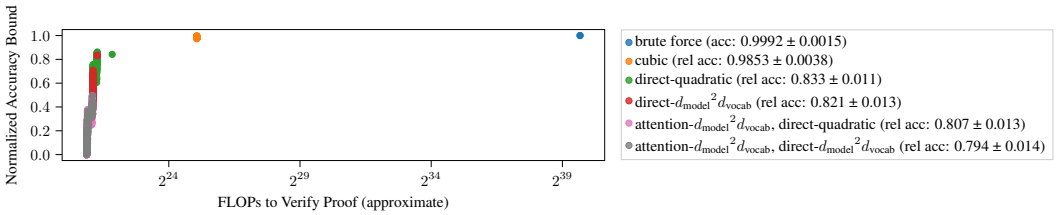


Figure 3: For each of the proofs in Section 4, we plot the number of FLOPs used to compute the certificate, as well as the normalized accuracy lower-bound (b/s). For clarity's sake, we exclude The brute-force proof (Section 4) computes the exact performance, but uses orders of magnitude more compute than other approaches. The cubic proof (Subsection 4.2) uses a small amount of mechanistic understanding and less compute, while still retaining good accuracy lower bounds. Finally, subcubic proofs (Subsection 4.2) use the entirety of the mechanistic interpretation of the model, which further reduces compute costs, but achieve worse bounds.

of a $d_{\text{vocab}} \times d_{\text{model}}$ matrix and a $d_{\text{model}} \times d_{\text{vocab}}$ matrix. This allows us to avoid performing the $\mathcal{O}({d_{\text{vocab}}}^2 d_{\text{model}})$-cost matrix multiplications to explicitly compute EQKE.

Similarly, we can more efficiently check that the direct path EU contributes negligibly to the model outputs, by using SVD to decompose EU into a sum of rank 1 products (which we can evaluate exactly) and a high-rank error term that we can cheaply bound.

4.3 Additional subcubic proof strategies

Tighter bounds for sums of variables with unequal variance via the "mean+diff trick" Suppose we want to lower bound the minimum of the sum of two functions over three variables $h(x, y, z) = f(x, y) + g(y, z)$, while only iterating over two variables at a time. The naive way is to minimize $f(x, y)$ and $g(x, y)$ independently:

$$\min_{x,y,z} h(x, y, z) \geq \min_{x,y} f(x, y) + \min_{y,z} g(y, z)$$

Here, the error comes from setting the ys in f and g to different values. But in cases where $g(y, z)$ varies only by ε with z but more with y, then rewriting g as a sum of a component that is independent of z (and only varies along y), as well a component that depends on z, yields a better lower bound:

$$\min_{x,y,z} h(x, y, z) \geq \min_{x,y} \left(f(x, y) + \mathbb{E}'_z g(y, z')\right) + \min_{y,z}(g(y, z) - \mathbb{E}'_z g(y, z'))$$

This estimate will have error at most ε, while the naive estimator can have arbitrarily large error. We refer to this rewrite as the "mean+diff trick".[5] From the mechanistic interpretation in Subsection 3.1, we know that some of the components barely vary among one or more axes. So we can apply the mean+diff trick to get tighter lower bounds.

Avoiding matrix multiplications using the "max row-diff trick" Using properties of linear algebra, we derive a cheap approximation to the max row-diff for the product of matrices AB in terms of the product of the max-row diff of B and the absolute value of A, we deem the "max row-diff" trick. We apply this trick to get a better cheap bound on the error terms of low-rank approximations, without having to multiply out the full matrices. See Appendix F for more details.

5 Results

We run each of 151 transformers on the various proof strategies of different asymptotic complexity,and analyze these proofs to empirically examine the relationship between proof length, bound tightness, and degree of understanding. For each proof on each transformer, we approximate the length of the proof by estimating the number of FLOPs used, and plot this against the ratio of certified bound the true accuracy b/s (Equation 2) in Figure 3. There exists a clear trade-off between bound tightness and compactness of the proof – more compact proofs yield looser bounds, and tighter bounds are associated with more expensive proofs.

[5] In fact, this is the motivation behind the standard rewrites of QK and OV into position-independent and position-dependent components, in cases where the behavior does not vary much across positions(Section 3).

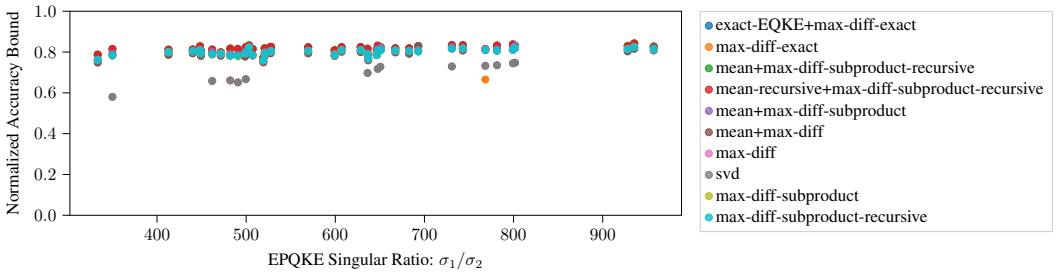


Figure 4: To study whether more faithful interpretations lead to tighter bounds even holding proof length fixed, we plot the normalized accuracy bound versus the ratio of first and second singular values of EQKE, for various types of subcubic proofs that depend on a rank-1 approximation EQKE. The closer EQKE is to rank-1, the tighter the accuracy bound.

5.1 Shorter proofs both require and provide mechanistic understanding

Quantifying mechanistic understanding using unexplained dimensionality We first quantify the amount of mechanistic understanding used in a proof by measuring its **unexplained dimensionality** – the number of free parameters required to fully describe model behavior, assuming the structural assumptions of the proof are correct. More detailed mechanistic interpretations will leave fewer free parameters that need to be filled in via empirical observation. (Details in Appendix L). In Figure 5, we plot the two axes and find a suggestive correlation – that is, proofs based on less mechanistic understanding are longer.

More mechanistic understanding allows for more compact proofs In addition to the constructions in Section 4, the parts of proofs we were unable to compact seem to correspond to components that we do not mechanistically understand. For example, we could not cheaply bound the behavior of EVOU without multiplying out the matrices, and this seems in part because we do have a mechanistic understanding of how EVOU implements low-rank copying.

Compact proofs seem to provide understanding By examining compact proofs, we can extract understanding about the model. For example, the fact that replacing each row of EU with its average across rows has little effect on the bound implies that EU does not vary much based on t_{query}.

5.2 The trade-off between proof length and bound tightness is mediated by faithfulness of interpretation

Compact proofs are less faithful to model internals To derive more compact proofs, we use our mechanistic understanding to simplify the model computation in ways that diverge from the original model internals. For example, in some subcubic proofs (Subsection 4.2), we approximate EQKE with a rank-1 approximation corresponding to the “size direction". However, while other components are small, they’re nonzero, and this approximation harms model internals.

Less faithful interpretations lead to worse bounds on performance To confirm that faithfulness of understanding affects the tightness of bound independent of proof length, we plot the normalized accuracy bound of subcubic proofs that perform a rank-1 approximation to EQKE, versus the ratio of the first two singular components. A larger ratio between the components implies that the rank-1 approximation is more faithful. From the results in Figure 4, we see a positive correlation between the two axes – that is, when the interpretation is more faithful, the bounds are tighter, even at a fixed proof length.

5.3 Compounding structureless noise poses a serious challenge to compacting proofs of global behavior

Pessimal error terms compound in the absence of known structure Approximating EQKE with a rank-1 matrix has small error. However, when approximating each of the constituent matrices E, Q, K with rank-1 approximations, pessimizing over the worst way to composing the individual small error terms leads to a bound on the error term of EQKE that is orders of magnitude larger than the actual error term. Because we don’t understand the matrices compose in a way that doesn’t cause noise to compound (without just multiplying out the matrices), this approximation leads to a trivial bound on performance. We speculate that in many cases, there is no short human-interpretable

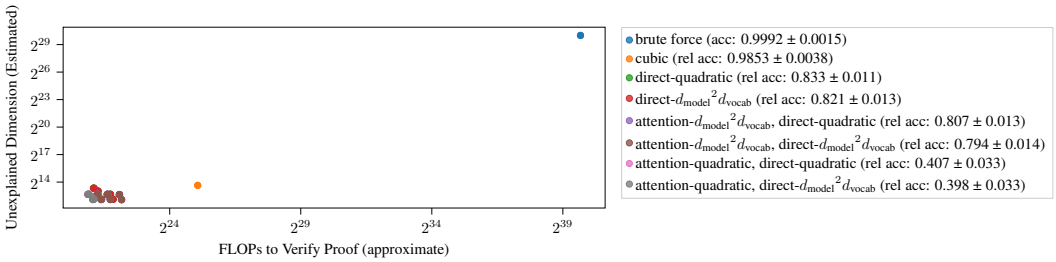


Figure 5: For each proof, we plot the approximate number of flops required to evaluate the proofs, versus the unexplained dimensionality (Subsection 5.1). Shorter proofs seem to related to proofs that contain more mechanistic understanding (and thus leave fewer dimensions unexplained).

description for why random noise or approximation errors do not compound across layers of neural networks (e.g., see the error correction results on *randomly initialized* neural networks from Vaintrob et al. [40]), and thus that compounding structureless noise may be an issue in practice.

6 Related Work

Generalization Bounds Prior work in the PAC-Bayes framework [47, 26] proves generalization bounds over learning procedures, which are similar to the global performance bounds we consider in this work. These proofs tend to provide statistical guarantees [19] about the outputs of a known stochastic training procedure, while we seek to bound the performance of particular trained models.

Formally verifying neural networks Most prior work formally verifies neural networks either via model checking [22, 6] or by relaxing the problem setting and taking an automated theorem proving approach [13, 38, 14, 25, 32] to verify *local* robustness properties. These proof strategies tend to be derived by examining only the network architecture. We take an approach more akin to interactive theorem proving [16] and verify *global* performance properties by reverse-engineering the neural network weights.

Mechanistic Interpretability Finally, mechanistic interpretability is the subfield of the broader field of understanding model internals [34], which is too large to faithfully summarize. Our work takes most direct inspiration from efforts to deeply understand how either toy models [28, 7, 42, 2] or small pretrained text transformers [43, 15] implement algorithmic tasks, generally by performing ablations and SVD. In contrast, we formally prove that a transformer implements an algorithm.

[29] proves that, in a significantly simplified 2-layer, 1-head attention-only transformer model and for the task of in-context bigram statistics, gradient descent will create induction heads [30]. Our results concern transformers with fixed weights. In concurrent work, Michaud et al. [24] use techniques inspired by mechanistic interpretability to perform automated program synthesis on 2-dimensional RNNs, while our work works with significantly larger transformer models.

7 Conclusion and Future Work

Summary In this work, we used a max-of-k setting to prototype using mechanistic interpretability to derive compact proofs of model behavior. Using varying amounts of understanding, we derive more efficient proof computations lower bounding model accuracy. We find suggestive evidence that mechanistic understanding can compactify proofs, and that we can use the tightness of the lower bound to assess the faithfulness our understanding. Finally, we identify compounding structureless noise as a key obstacle to deriving compact proofs of model behavior.

Limitations and future work We study one-layer attention-only transformers on a toy algorithmic tasks. Future work should explore the viability of deriving proofs via interpretability using larger models featuring MLPs or layernorm on more complex domains. In addition, we were unable to significantly compact the part of the proof involving the OV circuit, which future work can explore. The proofs we explored in this work also did not lead to qualitatively novel insights; future work may be able to derive such insights with improved techniques. Finally, future work can address the problem of compounding structureless noise, perhaps by relaxing the worst-case assumption used in our pessimal ablations.

References

[1] Behzad Akbarpour, Amr Abdel-Hamid, Sofiène Tahar, and John Harrison. Verifying a synthesized implementation of IEEE-754 floating-point exponential function using HOL. *Comput. J.*, 53:465–488, May 2010. doi: 10.1093/comjnl/bxp023.

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

[3] Andrew Appel and Ariel Kellison. VCFloat2: Floating-point error analysis in Coq. In *Proceedings of the 13th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2024, pages 14–29, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704888. doi: 10.1145/3636501.3636953.

[4] Sylvie Boldo and Guillaume Melquiond. Flocq: A unified library for proving floating-point algorithms in Coq. In *2011 IEEE 20th Symposium on Computer Arithmetic*, pages 243–252, July 2011. doi: 10.1109/ARITH.2011.40.

[5] Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldwosky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. Causal scrubbing, a method for rigorously testing interpretability hypotheses. *AI Alignment Forum*, 2022. https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN/causal-scrubbing-a-method-for-rigorously-testing.

[6] Chih-Hong Cheng, Georg Nührenberg, and Harald Ruess. Maximum resilience of artificial neural networks. In *Automated Technology for Verification and Analysis: 15th International Symposium, ATVA 2017, Pune, India, October 3–6, 2017, Proceedings 15*, pages 251–268. Springer, 2017.

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

[8] David Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, et al. Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems. *arXiv preprint arXiv:2405.06624*, 2024.

[9] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

[10] Martín H. Escardó. Synthetic topology of data types and classical spaces. *Electronic Notes in Theoretical Computer Science*, 87:21–156, November 2004.

[11] Martín H. Escardó. Infinite sets that admit fast exhaustive search. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS 2007)*, Wrocław, Poland, July 2007.

[12] Martín H. Escardó. Seemingly impossible functional programs, 2007. URL https://math.andrej.com/2007/09/28/seemingly-impossible-functional-programs/. Accessed: 2024-05-15.

[13] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, Los Alamitos, CA, USA, may 2018. IEEE Computer Society. doi: 10.1109/SP.2018.00058. URL https://doi.ieeecomputersociety.org/10.1109/SP.2018.00058.

[14] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

[15] Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than. *Interpreting mathematical abilities in a pre-trained language model*, 2:11, 2023.

[16] John Harrison, Josef Urban, and Freek Wiedijk. History of interactive theorem proving. In *Handbook of the History of Logic*, volume 9, pages 135–214. Elsevier, 2014.

[17] Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024.

[18] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[19] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *arXiv preprint arXiv:1812.08342*, 2018. URL https://arxiv.org/abs/1812.08342.

[20] Plotly Technologies Inc. Collaborative data science, 2015. URL https://plot.ly.

[21] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[22] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pages 97–117. Springer, 2017.

[23] A. E. Kellison, A. W. Appel, M. Tekriwal, and D. Bindel. LAProof: A library of formal proofs of accuracy and correctness for linear algebra programs. In *2023 IEEE 30th Symposium on Computer Arithmetic (ARITH)*, pages 36–43, Los Alamitos, CA, USA, September 2023. IEEE Computer Society. doi: 10.1109/ARITH58626.2023.00021.

[24] Eric J. Michaud, Isaac Liao, Vedang Lad, Ziming Liu, Anish Mudide, Chloe Loughridge, Zifan Carl Guo, Tara Rezaei Kheirkhah, Mateja Vukelić, and Max Tegmark. Opening the AI black box: program synthesis via mechanistic interpretability, 2024.

[25] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3578–3586. PMLR, 2018.

[26] Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[27] Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

[28] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint*, 2023. doi: 10.48550/arXiv.2301.05217.

[29] Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2402.14735.

[30] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

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

[32] Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. *Advances in neural information processing systems*, 31, 2018.

[33] Tahina Ramananandro, Paul Mountcastle, Benoıˆt Meister, and Richard Lethin. A unified Coq framework for verifying C programs with floating-point computations. In *Proceedings of the 5th ACM SIGPLAN Conference on Certified Programs and Proofs*, CPP 2016, pages 15–26, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341271. doi: 10.1145/2854065.2854066.

[34] Tilman Rauker, Stephen Casper, Anson Ho, and Dylan Hadfield-Menell. Toward transparent AI : A survey on interpreting the inner structures of deep neural networks. In *First IEEE Conference on Secure and Trustworthy Machine Learning*, 2022.

[35] Alex Rogozhnikov. Einops: Clear and reliable tensor manipulations with Einstein-like notation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=oapKSVM2bcj.

[36] Sanjit A Seshia, Dorsa Sadigh, and S Shankar Sastry. Toward verified artificial intelligence making ai more trustworthy with a formal methods-based approach to ai system verification and validation.

[37] Alex K. Simpson. Lazy functional algorithms for exact real functionals. In Luboš Brim, Jozef Gruska, and Jiří Zlatuška, editors, *Mathematical Foundations of Computer Science 1998*, pages 456–464, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-68532-6.

[38] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL), jan 2019. doi: 10.1145/3290354. URL https://doi.org/10.1145/3290354.

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[40] Dimtry Vaintrob, Karell Hanni, and Jake Mendel. Toward a mathematical framework for computation in superposition, 2023. URL https://www.lesswrong.com/posts/2roZtSr5TGmLjXMnT/toward-a-mathematical-framework-for-computation-in-superposition. Accessed: 2024-05-22.

[41] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

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

[43] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint*, 2022. doi: 10.48550/arXiv.2211.00593.

[44] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

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

[46] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pages 5286–5295. PMLR, 2018.

[47] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115, 2021.

A Training details

To train each model, we generate 384,000 random sequences of 4 integers picked uniformly at random, corresponding to less than 2.5% of the input distribution. We use AdamW with batch_size $= 128$, lr $= 0.001$, betas $= (0.9, 0.999)$, weight_decay left at the default 0.01. We train for 1 epoch (3000 steps). Over our 151 seeds, models trained with this procedure achieve $(99.92 \pm 0.15)\,\%$ train accuracy and a loss of 0.004 ± 0.008. (Numbers reported as mean across training runs $\pm$ std dev across training runs of mean accuracy and loss.) When qualitatively examining a single model (for example in Subsection 3.1 or Appendix C), we use the model with config seed 123, model seed 613947648 (which is deterministically pseudorandomly derived from 123).

As our models as sufficiently small, we did not have to use any GPUs to accelerate training our inference. Each training run takes less than a single CPU-hour to complete. In total, the experiments in this paper took less than 1000 CPU-hours in total.

We use the folllowing software packages in our work: [31, 20, 27, 35, 41**?** , 44]

B Mathematical defintions

On the following page, we provide a detailed breakdown of the mathematical notation used in the appendix. Note that while in the main body, $\mathcal{M}(\mathbf{x})$ referred to the pre-softmax output logits, in the appendix we abuse notation and occasionally use it to refer to maximum token indicated by the logits where appropriate.

Figure 6: Definitions of the model behavior

Let $d = \sqrt{d}$, $\mathbf{e}_t = (E)_t$, $\mathbf{p}_i = (P)_i$, α be the pre-softmax attention scores, ℓ^{EU} be the contribution to the logits from the skip connection, ℓ^k be the contributions to the logits via attention to the input token at index k, ℓ be the logits of the model, $\Delta\ell_t$ be the difference between the logit of token t and the maximum token, and $\mathcal{M}$ be the model output:

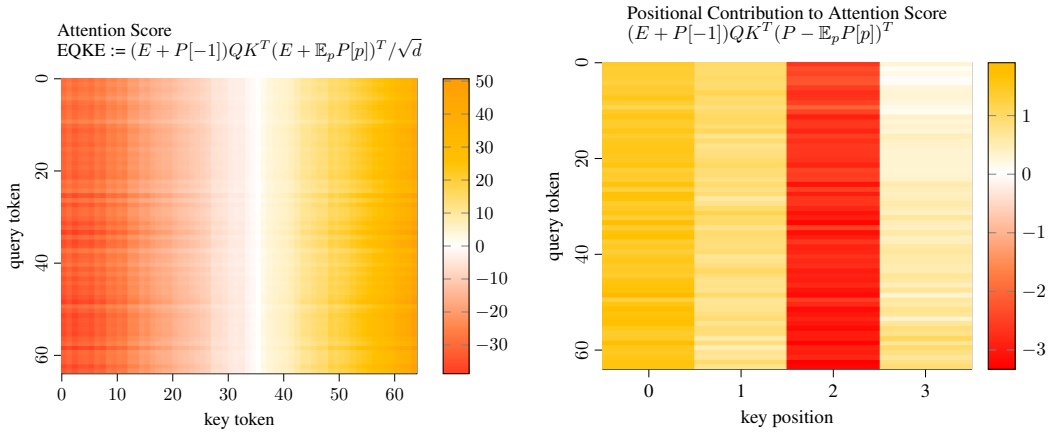
$$
\begin{aligned}
\text{EQKE}(x_{-1}, x_i) &:= \frac{1}{\sqrt{d}}(\mathbf{e}_{x_{-1}} + \mathbf{p}_{-1})QK^T\mathbf{e}_{x_i}^T \\
\text{EQKP}(x_{-1}, i) &:= \frac{1}{\sqrt{d}}(\mathbf{e}_{x_{-1}} + \mathbf{p}_{-1})QK^T\mathbf{p}_i^T \\
\alpha(x_{-1}, x_i, i) &:= \text{EQKE}(x_{-1}, x_i) + \text{EQKP}(x_{-1}, i) \\
\alpha_k^*(\mathbf{x}) &:= \frac{1}{\sum_i e^{\alpha(x_{-1}, x_i, i)}} e^{\alpha(x_{-1}, x_k, k)} \\
\text{EVOU}(x_k) &:= \mathbf{e}_{x_k}VOU \\
\text{PVOU}(k) &:= \mathbf{p}_k VOU \\
\text{EPVOU}(x_k, k) &:= \text{EVOU}(x_k) + \text{PVOU}(k) \\
\ell^{\text{EU}}(x_{-1}) &:= (\mathbf{e}_{x_{-1}} + \mathbf{p}_{-1})U \\
\ell^k(\mathbf{x}) &:= \alpha_k^*(\mathbf{x})\text{EPVOU}(x_k, k) \\
\ell^{\text{EVOU},k}(\mathbf{x}) &:= \alpha_k^*(\mathbf{x})\text{EVOU}(x_k) \\
\ell^{\text{PVOU},k}(\mathbf{x}) &:= \alpha_k^*(\mathbf{x})\text{PVOU}(k) \\
\mathcal{M}(\mathbf{x}) := \ell(\mathbf{x}) &:= \ell^{\text{EU}}(x_{-1}) + \textstyle\sum_{k=0}^{n_{\text{ctx}}-1} \ell^k(\mathbf{x}) \\
\Delta\ell_t(\mathbf{x}) &:= \ell(\mathbf{x})_t - \ell(\mathbf{x})_{\max_i x_i} \\
\Delta\ell_t^{\text{EU}}(x_{-1}, \max_i x_i) &:= \ell^{\text{EU}}(x_{-1})_t - \ell^{\text{EU}}(x_{-1})_{\max_i x_i} \\
\Delta\ell_t^k(\mathbf{x}) &:= \ell^k(\mathbf{x})_t - \ell^k(\mathbf{x})_{\max_i x_i} \\
\Delta\ell_t^{\text{EVOU},k}(\mathbf{x}) &:= \ell^{\text{EVOU},k}(\mathbf{x})_t - \ell^{\text{EVOU},k}(\mathbf{x})_{\max_i x_i} \\
\Delta\ell_t^{\text{PVOU},k}(\mathbf{x}) &:= \ell^{\text{PVOU},k}(\mathbf{x})_t - \ell^{\text{PVOU},k}(\mathbf{x})_{\max_i x_i} \\
\Delta\ell(\mathbf{x}) &:= \max_{i \neq \max_j x_j} \Delta\ell_i(\mathbf{x}) \\
\Delta\ell^{\text{EU}}(x_{-1}, \max_i x_i) &:= \max_{j \neq \max_i x_i} \Delta\ell_i^{\text{EU}}(x_{-1}, \max_i x_i) \\
\Delta\ell^k(\mathbf{x}) &:= \max_{i \neq \max_j x_j} \Delta\ell_i^k(\mathbf{x}) \\
\Delta\ell^{\text{EVOU},k}(\mathbf{x}) &:= \max_{i \neq \max_j x_j} \Delta\ell_i^{\text{EVOU},k}(\mathbf{x}) \\
\Delta\ell^{\text{PVOU},k}(\mathbf{x}) &:= \max_{i \neq \max_j x_j} \Delta\ell_i^{\text{PVOU},k}(\mathbf{x})
\end{aligned}
$$

We might also inline the equations and write

$$
\begin{aligned}
\alpha(x_{-1}, x_i, i) &:= \frac{1}{\sqrt{d}}(\mathbf{e}_{x_{-1}} + \mathbf{p}_{-1})QK^T(\mathbf{e}_{x_i} + \mathbf{p}_i)^T \\
\mathcal{M}(\mathbf{x}) := \ell(\mathbf{x}) &:= \frac{1}{\sum_i e^{\alpha(x_{-1}, x_i, i)}} \sum_{k=0}^{n_{\text{ctx}}-1} \left[e^{\alpha(x_{-1}, x_k, k)}(\mathbf{e}_{x_k} + \mathbf{p}_k)VOU\right] + (\mathbf{e}_{x_{-1}} + \mathbf{p}_{-1})U
\end{aligned}
$$

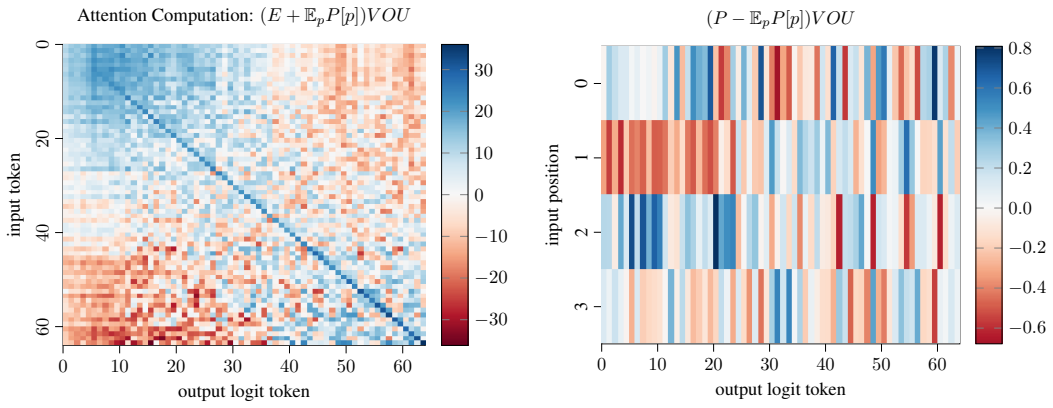
C Additional details supporting our mechanistic interpretation of the model.

We provide heatmaps of the matrices corresponding to the five components described/defined in 3, for the mainline model.



(a) EQKE $= \mathbf{E}_q Q K^T \bar{\mathbf{E}}^T \mathbf{x}^T$ **(b)** EQKP $= \mathbf{E}_q Q K^T \hat{\mathbf{P}}^T$

Figure 7: The QK circuit can be decomposed into the position-independent and position-dependent components EQKE and EQKP, and computes the pre-softmax attention score for the model. The positional contribution to the attention score, as shown in (b), is minimal. In figure (a), the gradient from left to right along the key axis indicates that the single attention head pays more attention to larger tokens. The uniformity along the query axis suggests that this behavior is largely independent of the query token. Further, the light and dark bands imply that some queries are better than others at focusing more on larger tokens.



(a) EVOU $= \mathbf{x}(\bar{\mathbf{E}}VOU)$ **(b)** PVOU $= \hat{\mathbf{P}}VOU$

Figure 8: The OV circuit is a sum of EVOU and PVOU. In figure (a) we see that EVOU "copies" — with the exception of input tokens 5 and under — by virtue of the fact that above 5, the diagonal is larger than all the other elements in the same row. We see that the range on figure (b) is much smaller than figure (a), incidcating that positional contribution to the copying is minimal.

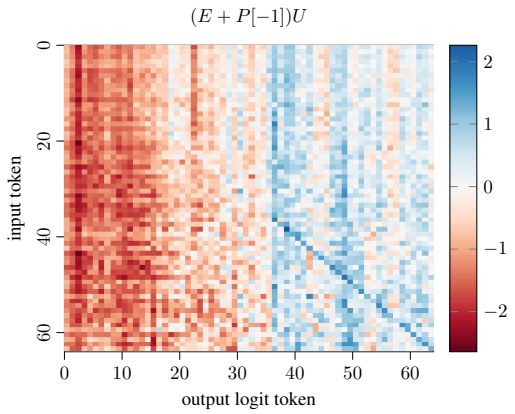


Figure 9: Direct Path $= t_{\text{query}}(\mathbf{E}_q U)$. These values matter a bit more that PVOU, being only $\approx 10\times$ smaller than the typical EVOU difference. They don't matter that much, though, being so small. Additionally, the vertical banding indicates that the primary effect of this is a largely-query-independent bias towards larger numbers, reflecting the fact that the input distribution is biased towards larger numbers being the maximum. The weak diagonal pattern indicates a slight bias towards upweighting the query token itself as a (possible) maximum token.

D Brute-force proof

Theorem 1.

$$\mathbb{E}_{\mathbf{x}\sim U(0,1,\ldots,d_{vocab}-1)^{n_{ctx}}}\left[\mathcal{M}(\mathbf{x}) = \max_i x_i\right] \geq \text{BRUTE-FORCE}(d_{vocab}, n_{ctx})$$

Proof. By definition and reflexivity of $\geq$. □

Algorithm 1 Counting Correct Sequences By Brute Force

```
function CORRECTNESS(input-sequence)
    return MODEL-BEHAVIOR(input-sequence) == MAX(input-sequence)
end function
function BRUTE-FORCE(d_vocab, n_ctx)
    return SUM(CORRECTNESS(tokens) for tokens ∈ (RANGE(d_vocab))^n_ctx) / d_vocab^n_ctx
end function
```

E Details of cubic proof

In this section, we prove formally the result used in Subsection 4.1, A cubic proof.

At its heart, the convexity of softmax is an extension to a simple idea: a weighted average of scalar values is extremized by putting 100% of the weight on an extremal value.

Using this simple version of the theorem, however, gives a useless bound of 0% accuracy: if we pay no attention to the maximum of the sequence, of course we're going to get the wrong answer. Since in fact the space of possible weightings we may see in practice is much smaller (finite, in fact, with at most $d_{\text{vocab}}{}^{n_{\text{ctx}}}$ values), we may look for a more general version of this idea that gives us tighter bounds that still cover the space of possible weightings.

Since the weights are *not* linearly independently choosable (softmax is non-linear), extremal values do not necessarily result from putting maximal attention on the worst token: it may be, when trying to find the worst case, that some positions are so dis-preferred that it makes more sense to choose a token that is "less bad" for those positions, if it draws enough attention away from the correct token. See Lemma 3 for details.

We thus spend this section characterizing a relaxation of the constraints on weights:

1. that contains all actually possible weightings,
2. that is extremized at weights that still correspond to some notion of "put the most weight on the extremal tokens", and
3. for which computing the extremal weightings is computationally efficient.

Before diving in, let's recall the proof that a weighted average of scalar values is extremized by putting 100% of the weight on extremal values:

Theorem 2 (Extremizing weighted averages)**.** *Fix a set of values $v_i \in \mathbb{R}$. The weighted average is bounded by the extremal values: for any w_i such that $\sum_i w_i = 1$ and $0 \le w_i \le 1$,*

$$\min_i v_i \le \sum_i w_i v_i \le \max_i v_i$$

Proof. The proof is simple. We have

$$\sum_i w_i v_i - \min_i v_i = \sum_i w_i (v_i - \min_j v_j) \ge 0$$

and

$$\max_i v_i - \sum_i w_i v_i = \sum_i w_i (\max_j v_j - v_i) \ge 0$$

so the result follows. □

E.1 Proof strategy

Outputting the correct behavior is equivalent to outputting logits ℓ such that $\Delta\ell_{t'} := \ell_{t'} - \ell_{\max} < 0$ for all $t' \neq \max$. As a result, it suffices to lower-bound the proportion of sequences where (an upper bound on) the logit difference is negative for all non-max outputs. In particular, we will upper-bound the contribution from incorrect tokens t to the logit difference between incorrect (t') and correct ($\max$) tokens $\Delta\ell_{t'} = \ell^t_{t'} - \ell^t_{\max}$.

We do this by arguing that the logit difference $\Delta\ell_{t'}$ satisfies a certain notion of convexity over the space of a relaxation of sequences (Theorem 6), and constructing a set of $\Theta({d_{\text{vocab}}}^3 n_{\text{ctx}})$ "extremal" relaxed sequences where the position and token embedding components of attention are pessimized independently.

We start by first rewriting the contribution of each token through the attention head to the logit difference into the contributions involving PVOU and EVOU:

$$\Delta\ell^k_t(\mathbf{x}) = \Delta\ell^{\text{PVOU},k}_t(\mathbf{x}) + \Delta\ell^{\text{EVOU},k}_t(\mathbf{x})$$

We then upper bound $\Delta\ell^{\text{PVOU},k}_t(\mathbf{x})$ by noting that because the softmax attention is a weighted average of PVOU,

$$\begin{aligned}
\Delta\ell^{\text{PVOU},k}_t(\mathbf{x}) &= \ell^{\text{PVOU},k}(\mathbf{x})_t - \ell^{\text{PVOU},k}(\mathbf{x})_{\max_j x_j} \\
&= \alpha^*_k(\mathbf{x})\text{PVOU}_{k,t} - \alpha^*_k(\mathbf{x})\text{PVOU}_{k,\max_j x_j} \\
&= \alpha^*_k(\mathbf{x}) \left(\text{PVOU}_{k,t} - \text{PVOU}_{k,\max_j x_j}\right) \\
&\le \alpha^*_k(\mathbf{x}) \max_k \left(\text{PVOU}_{k,t} - \text{PVOU}_{k,\max_j x_j}\right)
\end{aligned}$$

Since $\sum_k \alpha^*_k(\mathbf{x}) = 1$, we have

$$\sum_{k=0}^{n_{\text{ctx}}-1} \Delta\ell^{\text{PVOU},k}_t(\mathbf{x}) \le \max_k \left(\text{PVOU}_{k,t} - \text{PVOU}_{k,\max_j x_j}\right)$$

We then construct a set Ξ of "pure sequences" consisting of only three types of tokens in one of two orders, and show that for each input sequence $\mathbf{x}$ and readoff logit t, we bound the logit difference from the token embeddings $\Delta\ell_t^{\text{EVOU},k}(\mathbf{x})$ using a small subset $\mathcal{X}$ of Ξ:

$$\sum_{k=0}^{n_{\text{ctx}}-1} \Delta\ell_t^{\text{EVOU},k}(\mathbf{x}) \leq \max_{\xi\in\mathcal{X}} \sum_{k=0}^{n_{\text{ctx}}-1} \Delta\ell_t^{\text{EVOU},k}(\xi)$$

We construct a set $\mathcal{R}_{\text{relaxed}}$ of relaxed sequences, where each relaxed sequence consists of a sequence and a position $r = (\mathbf{x}, i)$, where $\Delta\ell_t(\mathbf{x}, i)$ is evaluated by separately considering the positional contribution through attention (that is, the attention weighted PVOU) and the token contribution (that is, the attention-weighted EVOU) and direct contribution (the logit difference through the skip connection $(\mathbf{e}_{t_{\text{query}}} + \mathbf{p}_{-1})U$).

Note that i indicates the position that pay 100% of the attention to for the PVOU contribution.

We argue that $\Delta\ell_t(\mathbf{x}, i)$ satisfies a certain notion of convexity over mixtures of sequences, such that we can evaluate it only on a set of $\Theta({d_{\text{vocab}}}^3 n_{\text{ctx}})$ "extremal" sequences in a way that takes $O({d_{\text{vocab}}}^3 n_{\text{ctx}})$ total time to bound $\Delta\ell_t(\mathbf{x}, i)$ for *every* possible input sequence.

We then use the extremal sequences that the model gets correct to lower bound the proportion of *all* sequences that the model will get correct.

Specifically, we argue that Algorithm 3 provides a valid lower bound on the proportion of sequences the model gets correct.

E.2 Proof outline

We now proceed to the main results of this section.

Math fact: For each token t, the logit difference $\Delta\ell_t$ for any sequence $\mathbf{x}$ can be decomposed into the direct contribution from the embeds ℓ^{EU}, the attention-weighted position contribution (PVOU), and the attention-weighted token contribution (EVOU). Therefore, it suffices to upper bound each of the three components independently, since summing these upper bounds gives a valid upper bound on the logit difference.

We can compute the direct contribution ℓ^{EU} exactly by first computing $(\mathbf{e}_{t_{\text{query}}} + \mathbf{p}_{-1})U = (P + E)U$ and then, for each max, subtracting the logit of the max token from each row of the matrix. No theorems needed.

For each max token, we can bound the position contribution by its maximum over positions (Theorem 6).

In order to upper bound the token contribution, we argue that any mixed sequence will be upper bounded by the maximum of the corresponding pure sequences (Theorem 7). We then argue that for pure sequences, it suffices to consider orderings where same tokens appear contiguously (Theorem 4).

E.3 Formal proof

For this subsection, all theorems are parameterized over the following quantities. Fix a token value function (à la a row difference in EVOU) $v : \mathbb{N} \to \mathbb{R}$ and a token attention function (à la EQKE for a fixed query token) $a : \mathbb{N} \to \mathbb{R}$. Fix a position value function (à la a row difference in PVOU) $w : \mathbb{N} \to \mathbb{R}$ and a position attention function (à la EQKP for a fixed query token) $b : \mathbb{N} \to \mathbb{R}$.

Definition 1. *We can* define a sequence of tokens via sorted tokens and a position permutation *by specifying a non-decreasing sequence of tokens* $t_0 \leq \cdots \leq t_{n_{ctx}-1} \in \mathbb{N}^{<d_{vocab}}$ *paired with a permutation* $\sigma : \mathbb{N}^{<n_{ctx}} \to \mathbb{N}^{<n_{ctx}}$.

Definition 2. *Given a non-decreasing sequence of tokens* $t_0 \leq \cdots \leq t_{n_{ctx}-1} \in \mathbb{N}^{<d_{vocab}}$ *and a permutation* $\sigma : \mathbb{N}^{<n_{ctx}} \to \mathbb{N}^{<n_{ctx}}$ *define the* sequence score $s_{t_0,\ldots,t_{n_{ctx}-1},\sigma}$ *as:*

$$s_{t_0,\ldots,t_{n_{ctx}-1},\sigma} := \sum_{0\leq i<n_{ctx}} v_{t_i} e^{a_{t_i}+b_{\sigma(i)}} \Bigg/ \sum_{0\leq i<n_{ctx}} e^{a_{t_i}+b_{\sigma(i)}}$$

We will drop the token subscript, writing only s_σ*, when the token values are unambiguous by context.*

Definition 3. *Given a permutation $\sigma : \mathbb{N}^{<n_{ctx}} \to \mathbb{N}^{<n_{ctx}}$ of the n_{ctx} positions and two indices $0 \leq i, j < n_{ctx}$, define the* swap permutation $\sigma_{i \leftrightarrow j}$ *to be the permutation that is σ except swapping i and j:*

$$\sigma_{i \leftrightarrow j}(k) = \begin{cases} \sigma(i) & \text{if } k = j \\ \sigma(j) & \text{if } k = i \\ \sigma(k) & \text{otherwise} \end{cases}$$

Lemma 3 (Characterization of swapping tokens)**.** *Fix a non-decreasing sequence of tokens $t_0 \leq \cdots \leq t_{n_{ctx}-1} \in \mathbb{N}$. Fix $\sigma : \mathbb{N} \to \mathbb{N}$ be a permutation of the n_{ctx} positions. Fix indices $0 \leq i, j < n_{ctx}$. Define $\Delta_{\sigma, i \leftrightarrow j}$ to be the difference in sequence scores when you swap i and j:*

$$\Delta_{\sigma, i \leftrightarrow j} := s_{\sigma_{i \leftrightarrow j}} - s_\sigma$$

Then there are two cases for $\operatorname{sign}(\Delta_{\sigma, i \leftrightarrow j})$*:*

1. *If $a_{t_i} = a_{t_j}$ then* $\operatorname{sign}(\Delta_{\sigma, i \leftrightarrow j}) = -\operatorname{sign}\left(b_{\sigma(i)} - b_{\sigma(j)}\right) \operatorname{sign}\left(v_{t_i} - v_{t_j}\right)$.
2. *Otherwise,* $\operatorname{sign}(\Delta_{\sigma, i \leftrightarrow j}) = \operatorname{sign}\left(a_{t_i} - a_{t_j}\right) \operatorname{sign}\left(b_{\sigma(i)} - b_{\sigma(j)}\right) \operatorname{sign}\left(s_\sigma - \frac{v_{t_i} e^{a_{t_i}} - v_{t_j} e^{a_{t_j}}}{e^{a_{t_i}} - e^{a_{t_j}}}\right)$.

Intuitively, Lemma 3 says that, if the token contribution to attention is equal between tokens t_i and t_j, then the impact of swapping their positions $\sigma(i)$ and $\sigma(j)$ is entirely determined by how much attention is paid to the positions of i and j and the relative difference in their value. (Notably, by swapping these tokens, we don't affect the attention paid on other tokens, and so the effect of the change does not depend on the values of the other tokens.) Alternatively, if the attentions are not equal, then swapping the positions changes the allocation of attention to other tokens in the sequence, and so it may the case that this change in allocation in attention dominates the attention-weighted values of these two tokens.

Proof. First note that the theorem is trivial for $i = j$.

For the rest of the proof, we take $i \neq j$.

The proof proceeds just by algebraic manipulation with no deep insight. We first list the facts we use, the proceed to computing $\operatorname{sign}(\Delta_{\sigma, i \leftrightarrow j})$. We abbreviate $\sigma_{i \leftrightarrow j}$ as σ' for brevity.

$$\operatorname{sign}\left(e^{b_{\sigma(i)}} - e^{b_{\sigma(j)}}\right) = \operatorname{sign}\left(b_{\sigma(i)} - b_{\sigma(j)}\right)$$

$$\begin{aligned} \operatorname{sign}(\Delta_{\sigma, i \leftrightarrow j}) &= \operatorname{sign}(s_{\sigma'} - s_\sigma) \\ &= \operatorname{sign}\left(\frac{\sum_{0 \leq p < n_{\text{ctx}}} v_{t_p} e^{a_{t_p} + b_{\sigma'(p)}}}{\sum_{0 \leq p < n_{\text{ctx}}} e^{a_{t_p} + b_{\sigma'(p)}}} - s_\sigma\right) \end{aligned}$$

Now multiply through by the denominator, which is positive

$$\begin{aligned} &= \operatorname{sign}\left(\sum_{0 \leq p < n_{\text{ctx}}} v_{t_p} e^{a_{t_p} + b_{\sigma'(p)}} - s_\sigma \sum_{0 \leq p < n_{\text{ctx}}} e^{a_{t_p} + b_{\sigma'(p)}}\right) \\ &= \operatorname{sign}\left(\sum_{0 \leq p < n_{\text{ctx}}} v_{t_p} e^{a_{t_p} + b_{\sigma(p)}} - v_{t_i} e^{a_{t_i}} \left(e^{b_{\sigma(i)}} - e^{b_{\sigma'(i)}}\right) - v_{t_j} e^{a_{t_j}} \left(e^{b_{\sigma(j)}} - e^{b_{\sigma'(j)}}\right)\right. \\ &\qquad \left. - s_\sigma \sum_{0 \leq p < n_{\text{ctx}}} e^{a_{t_p} + b_{\sigma(p)}} + s_\sigma e^{a_{t_i}} \left(e^{b_{\sigma(i)}} - e^{b_{\sigma'(i)}}\right) + s_\sigma e^{a_{t_j}} \left(e^{b_{\sigma(j)}} - e^{b_{\sigma'(j)}}\right)\right) \\ &= \operatorname{sign}\left(\cancel{\sum_{0 \leq p < n_{\text{ctx}}} v_{t_p} e^{a_{t_p} + b_{\sigma(p)}}} - v_{t_i} e^{a_{t_i}} \left(e^{b_{\sigma(i)}} - e^{b_{\sigma(j)}}\right) - v_{t_j} e^{a_{t_j}} \left(e^{b_{\sigma(j)}} - e^{b_{\sigma(i)}}\right)\right. \end{aligned}$$

$$
\begin{aligned}
&\left. - \cancel{\sum_{0 \le p < n_{\text{ctx}}} v_{t_p} e^{a_{t_p} + b_{\sigma(p)}}} + s_\sigma e^{a_{t_i}} \left(e^{b_{\sigma(i)}} - e^{b_{\sigma(j)}}\right) + s_\sigma e^{a_{t_j}} \left(e^{b_{\sigma(j)}} - e^{b_{\sigma(i)}}\right) \right) \\
&= \operatorname{sign}\left(\left(v_{t_j} e^{a_{t_j}} - v_{t_i} e^{a_{t_i}}\right)\left(e^{b_{\sigma(i)}} - e^{b_{\sigma(j)}}\right) + s_\sigma \left(e^{a_{t_i}} - e^{a_{t_j}}\right)\left(e^{b_{\sigma(i)}} - e^{b_{\sigma(j)}}\right)\right) \\
&= \operatorname{sign}\left(e^{b_{\sigma(i)}} - e^{b_{\sigma(j)}}\right) \operatorname{sign}\left(\left(v_{t_j} e^{a_{t_j}} - v_{t_i} e^{a_{t_i}}\right) + s_\sigma \left(e^{a_{t_i}} - e^{a_{t_j}}\right)\right) \\
&= \operatorname{sign}\left(b_{\sigma(i)} - b_{\sigma(j)}\right) \operatorname{sign}\left(s_\sigma \left(e^{a_{t_i}} - e^{a_{t_j}}\right) - \left(v_{t_i} e^{a_{t_i}} - v_{t_j} e^{a_{t_j}}\right)\right)
\end{aligned}
$$

Divide through by non-zero values when possible

$$
\begin{aligned}
&= \operatorname{sign}\left(b_{\sigma(i)} - b_{\sigma(j)}\right) \\
&\quad \cdot \begin{cases} \operatorname{sign}\left(v_{t_i} - v_{t_j}\right) & \text{if } a_{t_i} = a_{t_j} \\ \operatorname{sign}\left(e^{a_{t_i}} - e^{a_{t_j}}\right) \operatorname{sign}\left(s_\sigma - \frac{v_{t_i} e^{a_{t_i}} - v_{t_j} e^{a_{t_j}}}{e^{a_{t_i}} - e^{a_{t_j}}}\right) & \text{otherwise} \end{cases} \\
&= \begin{cases} -\operatorname{sign}\left(b_{\sigma(i)} - b_{\sigma(j)}\right) \operatorname{sign}\left(v_{t_i} - v_{t_j}\right) & \text{if } a_{t_i} = a_{t_j} \\ \operatorname{sign}\left(a_{t_i} - a_{t_j}\right) \operatorname{sign}\left(b_{\sigma(i)} - b_{\sigma(j)}\right) \operatorname{sign}\left(s_\sigma - \frac{v_{t_i} e^{a_{t_i}} - v_{t_j} e^{a_{t_j}}}{e^{a_{t_i}} - e^{a_{t_j}}}\right) & \text{otherwise} \end{cases}
\end{aligned}
$$

□

Definition 4. *Fix a set of fixed indices $F \subseteq \mathbb{N}^{<n_{ctx}}$ and an assigment of token values to each of the fixed positions $t_F : F \to \mathbb{N}^{<d_{vocab}}$. Fix a non-decreasing sequence of tokens $t_0 \le \cdots \le t_{n_{ctx}-1} \in \mathbb{N}$.*

Given a permutation $\sigma : \mathbb{N}^{<n_{ctx}} \to \mathbb{N}^{n_{ctx}}$, say that σ fixes F (relative to $t_0, \ldots, t_{n_{\text{ctx}}-1}$) *if $t_i = t_F(\sigma(i))$ whenever $\sigma(i) \in F$.*

Definition 5. *Fix a set of fixed indices $F \subseteq \mathbb{N}^{<n_{ctx}}$ and an assigment of token values to each of the fixed positions $t_F : F \to \mathbb{N}^{<d_{vocab}}$.*

Define the position-sorting permutation fixing indices in F *$\sigma_s : \mathbb{N}^{<n_{ctx}} \to \mathbb{N}^{<n_{ctx}}$ to be the permutation that sorts the indices not in F according to b: for $0 \le i, j < n_{ctx}$ with $i, j \notin F$, $b_i \le b_j$ whenever $\sigma_s(i) < \sigma_s(j)$; and $\sigma_s(i) = i$ for $i \in F$.*

Theorem 4 (Pessimization over sequence ordering is possible and results in contiguous sequences)**.** *Fix a set of fixed indices $F \subseteq \mathbb{N}^{<n_{ctx}}$ and an assigment of token values to each of the fixed positions $t_F : F \to \mathbb{N}^{<d_{vocab}}$. Fix a non-decreasing sequence of tokens $t_0 \le \cdots \le t_{n_{ctx}-1} \in \mathbb{N}$.*

Let $\sigma_{\min}, \sigma_{\max} : \mathbb{N} \to \mathbb{N}$ be permutations of the n_{ctx} positions, fixing positions in F, satisfying the following property: For all $\sigma : \mathbb{N} \to \mathbb{N}$ a permutation fixing F, we have

$$s_{\sigma_{\min}} \le s_\sigma \le s_{\sigma_{\max}} \tag{2}$$

(Such permutations are guaranteed to exist because the permutation group on n_{ctx} elements is finite.)

Then $\sigma_{\max}$ and $\sigma_{\min}$ may be taken to be contiguous on equal tokens*. That is, there exist $\sigma_{\max}$ and $\sigma_{\min}$ satisfying the property of Equation 2 which additionally satisfy the property that for $0 \le i, j, k < n_{ctx}$ with $t_i = t_j \ne t_k$ and $i, j, k \notin \sigma^{-1}(F)$, it is never the case that $\sigma_s(\sigma(i)) < \sigma_s(\sigma(k)) < \sigma_s(\sigma(j))$ for $\sigma \in \{\sigma_{\max}, \sigma_{\min}\}$.*

The basic idea is that we will assume that one of $\sigma_{\max}$ and $\sigma_{\min}$ cannot be contiguous on equal tokens and derive a contradiction. We will pick the extremal permutation that is closest to being contiguous, take a contiguity violation, and then show that either we can correct the contiguity violation without changing the score—thus violating the presumption that the permutation is *closest* to being contiguous—or we will find one swap of indices that decreases the score and another swap of indices that increases the score, thus violating the presumption of extremality.

In slightly more detail, but still informally, we will consider the sign of the difference between scores of our purported extremal permutation and a permutation that has swapped some indices. The theorem follows from showing that there exists a triple of indices i, j, k such that the sign of the score difference from swapping i and j is different from the sign of the score difference from swapping j and k.

First, a definition and some helpful facts about it.

Definition 6. *Fix a set of fixed indices $F \subseteq \mathbb{N}^{<n_{ctx}}$ and an assigment of token values to each of the fixed positions $t_F : F \to \mathbb{N}^{<d_{vocab}}$. Fix a non-decreasing sequence of tokens $t_0 \leq \cdots \leq t_{n_{ctx}-1} \in \mathbb{N}$.*

Say that a permutation σ is contiguous on equally-attended positions *if, for all $0 \leq i < n_{ctx}$ with $i \notin \sigma^{-1}(F)$, the sorting order according σ_s on the contiguous block of positions with contribution to the attention score equal to that of $\sigma(i)$, $\left\{\sigma(j) \,\middle|\, b_{\sigma(j)} = b_{\sigma(i)} \text{ and } \sigma(j) \notin F\right\}$, is the same as the sorting order according to the fraction of tokens equal to t_j with b-values greater than $b_{\sigma(i)}$, with ties broken by the value of t_j. Equationally, this second sorting order is defined by the score*

$$\left(\left|\left\{k \,\middle|\, t_k = t_j \text{ and } b_{\sigma(k)} > b_{\sigma(i)} \text{ and } \sigma(k) \notin F\right\}\right| + \frac{t_j}{d_{vocab}}\right) \Big/ \left|\{k \,|\, t_k = t_j \text{ and } \sigma(k) \notin F\}\right| .$$

Most importantly, any permutation that is contiguous on equally-attended positions has the property that for any indices $0 \leq i, j, k < n_{\text{ctx}}$ with $i, j, k \notin \sigma^{-1}(F)$ and $t_i = t_j \neq t_k$ and $\sigma_s(\sigma(i)) < \sigma_s(\sigma(k)) < \sigma_s(\sigma(j))$, we will have the *strict* inequality $b_{\sigma(i)} < b_{\sigma(k)} < b_{\sigma(j)}$. Additionally, we may always sort equally-attended positions to make any permutation contiguous on equally-attended positions.

We will define an additional notion of contiguity-violations which we avoid up-front by arbitrarily swapping involved indices without changing the score s_σ.

Definition 7. *Fix a set of fixed indices $F \subseteq \mathbb{N}^{<n_{ctx}}$ and an assigment of token values to each of the fixed positions $t_F : F \to \mathbb{N}^{<d_{vocab}}$. Fix a non-decreasing sequence of tokens $t_0 \leq \cdots \leq t_{n_{ctx}-1} \in \mathbb{N}$.*

Say that a permutation σ is needlessly non-contiguous at i, j, k *(for $i, j, k \notin \sigma^{-1}(F)$) if $\Delta_{\sigma, i \leftrightarrow k} = 0$ or $\Delta_{\sigma, j \leftrightarrow k} = 0$, for $0 \leq i, j, k < n_{ctx}$ with $i, j, k \notin \sigma^{-1}(F)$ with $t_i = t_j \neq t_k$ and $\sigma_s(\sigma(i)) < \sigma_s(\sigma(k)) < \sigma_s(\sigma(j))$.*

Say that a permutation σ is needlessly non-contiguous *if it is needlessly non-contiguous at any $i, j, k \notin \sigma^{-1}(F)$.*

Lemma 5. *Fix a set of fixed indices $F \subseteq \mathbb{N}^{<n_{ctx}}$ and an assigment of token values to each of the fixed positions $t_F : F \to \mathbb{N}^{<d_{vocab}}$. Fix a non-decreasing sequence of tokens $t_0 \leq \cdots \leq t_{n_{ctx}-1} \in \mathbb{N}$.*

Any needlessly non-contiguous sequence σ which fixes F can be made into a sequence σ' which still fixes F and is both simultaneously contiguous on equally-attended positions and not needlessly non-contiguous, and for which $s_\sigma = s_{\sigma'}$.

Proof. First, sort regions of equally-attended positions to make σ contiguous on equally-attended positions. If the resulting permutation is not needlessly non-contiguous, then we are done.

Otherwise, we have $\Delta_{\sigma, i \leftrightarrow k} = 0$ or $\Delta_{\sigma, j \leftrightarrow k} = 0$ for some i, j, k, for $0 \leq i, j, k < n_{\text{ctx}}$ with $i, j, n \notin \sigma^{-1}(F)$ and $t_i = t_j \neq t_k$ and $\sigma_s(\sigma(i)) < \sigma_s(\sigma(k)) < \sigma_s(\sigma(j))$. Since the sequence is contiguous on equally-attended positions, we have the strict inequality $b_{\sigma(i)} < b_{\sigma(k)} < b_{\sigma(j)}$.

By Lemma 3, we have two cases. Noting that $t_i = t_j$, we can write them as

1. $v_{t_k} = v_{t_i}$ and $a_{t_i} = a_{t_k}$
2. $a_{t_i} \neq a_{t_k}$ and $s_\sigma = \frac{v_{t_i} e^{a_{t_i}} - v_{t_k} e^{a_{t_k}}}{e^{a_{t_i}} - e^{a_{t_k}}}$

In the first case, we may fully freely interchange tokens equal to t_i with tokens equal to t_k without changing the score; in this case we may use the token value as a sorting tie-breaker and swap tokens until there are no more needlessly non-contiguous triples falling into case (1).

In the second case, since swapping tokens does not change s_σ, the property will continue to hold for these tokens after the swap. We may then swap tokens, again using token value as a tie-breaker, until there are no more needlessly non-contiguous triples falling into case (2). □

We can now finally make our argument for Theorem 4 more precise.

Proof of Theorem 4. Choose $\sigma_{\max}$ and $\sigma_{\min}$ to be contiguous on equally-attended positions and not needlessly non-contiguous, and suppose that we have $\sigma \in \{\sigma_{\max}, \sigma_{\min}\}$ such that for some

$0 \leq i, j, k < n_{\text{ctx}}$ with $i, j, k \notin \sigma^{-1}(F)$ and $t_i = t_j \neq t_k$, we have $b_{\sigma(i)} < b_{\sigma(k)} < b_{\sigma(j)}$. We will derive a contradiction with the presumption that σ is extremal by showing that we can swap i and k to change the score in one direction and that we can swap j and k to change the score in the other direction.

Take σ'_0 to be σ but swapping i and k, and take σ'_1 to be σ but swapping j and k.

Now we will consider the cases for the sign of the score difference $\Delta_0 := s_{\sigma'_0} - s_\sigma$ and $\Delta_1 := s_{\sigma'_1} - s_\sigma$. By the presumption of not being needlessly non-contiguous, $\Delta_z \neq 0$ for $z \in \{0, 1\}$. If we can show that the sign of Δ_0 is distinct from the sign of Δ_1, then we will have a contradiction with extremality because we will have either $s_{\sigma'_0} < s_\sigma < s_{\sigma'_1}$ or $s_{\sigma'_1} < s_\sigma < s_{\sigma'_0}$. That is, we would be able to swap $i \leftrightarrow k$ and $j \leftrightarrow k$ to get a lower and higher score, making σ not extremal.

Noting that $t_i = t_j$,

$$\operatorname{sign}(\Delta_0) = \operatorname{sign}\left(b_{\sigma(i)} - b_{\sigma(k)}\right) \begin{cases} \operatorname{sign}(v_{t_k} - v_{t_i}) & \text{if } a_{t_i} = a_{t_k} \\ \operatorname{sign}(a_{t_i} - a_{t_k}) \operatorname{sign}\left(s_\sigma - \frac{v_{t_i} e^{a_{t_i}} - v_{t_k} e^{a_{t_k}}}{e^{a_{t_i}} - e^{a_{t_k}}}\right) & \text{otherwise} \end{cases}$$

$$\operatorname{sign}(\Delta_1) = \operatorname{sign}\left(b_{\sigma(j)} - b_{\sigma(k)}\right) \begin{cases} \operatorname{sign}(v_{t_k} - v_{t_i}) & \text{if } a_{t_i} = a_{t_k} \\ \operatorname{sign}(a_{t_i} - a_{t_k}) \operatorname{sign}\left(s_\sigma - \frac{v_{t_i} e^{a_{t_i}} - v_{t_k} e^{a_{t_k}}}{e^{a_{t_i}} - e^{a_{t_k}}}\right) & \text{otherwise} \end{cases}$$

Noting that the product is non-zero by presumption, that right multiplicand is equal for Δ_0 and Δ_1, and $\operatorname{sign}\left(b_{\sigma(i)} - b_{\sigma(k)}\right) = -1$ and $\operatorname{sign}\left(b_{\sigma(j)} - b_{\sigma(k)}\right) = 1$, we have our desired contradiction. □

Note that the proof of Theorem 4 does not go through if we include the position value function w in the score, because we may trade off the position value function against the token value function. We now show that we can *independently* pessimize over positional attention.

Definition 8. *Given a non-decreasing sequence of tokens $t_0 \leq \cdots \leq t_{n_{ctx}-1} \in \mathbb{N}^{<d_{vocab}}$ and a permutation $\sigma : \mathbb{N}^{<n_{ctx}} \to \mathbb{N}^{<n_{ctx}}$ define the* full sequence score *$s'_{t_0,\ldots,t_{n_{ctx}-1},\sigma}$ as:*

$$s_{t_0,\ldots,t_{n_{ctx}-1},\sigma} := \sum_{0 \leq i < n_{ctx}} (v_{t_i} + w_{\sigma(i)}) e^{a_{t_i} + b_{\sigma(i)}} \Bigg/ \sum_{0 \leq i < n_{ctx}} e^{a_{t_i} + b_{\sigma(i)}}$$

We will drop the token subscript, writing only s'_σ, when the token values are unambiguous by context.

Theorem 6 (Independent pessimization over positional contributions is possible)**.** *Fix a set of fixed indices $F \subseteq \mathbb{N}^{<n_{ctx}}$ and an assigment of token values to each of the fixed positions $t_F : F \to \mathbb{N}^{<d_{vocab}}$. Fix a non-decreasing sequence of tokens $t_0 \leq \cdots \leq t_{n_{ctx}-1} \in \mathbb{N}$. Let $\sigma_{\min}, \sigma_{\max} : \mathbb{N} \to \mathbb{N}$ be as in Theorem 4.*

Define relaxed extremal sequence scores *$r_{\sigma_{\max}}$, $r_{\sigma_{\min}}$:*

$$r_{\sigma_{\min}} := s_{\sigma_{\min}} + \min_{0 \leq i < n_{ctx}} w_i$$

$$r_{\sigma_{\max}} := s_{\sigma_{\max}} + \max_{0 \leq i < n_{ctx}} w_i$$

Then $r_{\sigma_{\min}} \leq s'_{\sigma_{\min}}$ and $s'_{\sigma_{\max}} \leq r_{\sigma_{\max}}$.

Proof. This proof follows straightforwardly from the softmax weighting being an affine weighting.

$$\begin{aligned} s'_\sigma &= \sum_{0 \leq i < n_{\text{ctx}}} (v_{t_i} + w_{\sigma(i)}) e^{a_{t_i} + b_{\sigma(i)}} \Bigg/ \sum_{0 \leq i < n_{\text{ctx}}} e^{a_{t_i} + b_{\sigma(i)}} \\ &= \frac{\sum_i v_{t_i} e^{a_{t_i} + b_{\sigma(i)}}}{\sum_i e^{a_{t_i} + b_{\sigma(i)}}} + \frac{\sum_i w_{\sigma(i)} e^{a_{t_i} + b_{\sigma(i)}}}{\sum_i e^{a_{t_i} + b_{\sigma(i)}}} \\ &= s_\sigma + \frac{\sum_i w_{\sigma(i)} e^{a_{t_i} + b_{\sigma(i)}}}{\sum_i e^{a_{t_i} + b_{\sigma(i)}}} \end{aligned}$$

$$= s_\sigma + \sum_i w_{\sigma(i)} \frac{e^{a_{t_i}+b_{\sigma(i)}}}{\sum_j e^{a_{t_i}+b_{\sigma(i)}}}$$

$$s_\sigma + \sum_i w_{\sigma(i)} \frac{e^{a_{t_i}+b_{\sigma(i)}}}{\sum_j e^{a_{t_i}+b_{\sigma(i)}}} = s'_\sigma = s_\sigma + \sum_i w_{\sigma(i)} \frac{e^{a_{t_i}+b_{\sigma(i)}}}{\sum_j e^{a_{t_i}+b_{\sigma(i)}}}$$

$$s_\sigma + \min_k w_{\sigma(k)} \sum_i \frac{e^{a_{t_i}+b_{\sigma(i)}}}{\sum_j e^{a_{t_i}+b_{\sigma(i)}}} \le s'_\sigma \le s_\sigma + \max_k w_{\sigma(k)} \sum_i \frac{e^{a_{t_i}+b_{\sigma(i)}}}{\sum_j e^{a_{t_i}+b_{\sigma(i)}}}$$

Since

$$\sum_i \frac{e^{a_{t_i}+b_{\sigma(i)}}}{\sum_j e^{a_{t_i}+b_{\sigma(i)}}} = \frac{\sum_i e^{a_{t_i}+b_{\sigma(i)}}}{\sum_j e^{a_{t_i}+b_{\sigma(i)}}} = 1$$

we get

$$s_\sigma + \min_k w_{\sigma(k)} \le s'_\sigma \le s_\sigma + \max_k w_{\sigma(k)}$$

and hence $r_{\sigma_{\min}} \le s'_{\sigma_{\min}}$ and $s'_{\sigma_{\max}} \le r_{\sigma_{\max}}$ as desired. □

Theorem 7 (For a fixed ordering, softmax is convex over token counts and only pure sequences need be considered)**.** *Fix a set of fixed indices $F \subseteq \mathbb{N}^{<n_{ctx}}$ and an assigment of token values to each of the fixed positions $t_F : F \to \mathbb{N}^{<d_{vocab}}$. Let n denote the number of fixed positions: $n := |F|$. Fix a set $S \subseteq \mathbb{N}^{<d_{vocab}}$ of valid other tokens in the sequence.*

Define a comparison on non-negative integers less than d_{vocab}:

$$c := \sum_{i \in F} v_{t_F(i)} e^{a_{t_F(i)}+b_i} \qquad d := \sum_{i \in F} e^{a_{t_F(i)}+b_i} \qquad f := \sum_{\substack{0 \le i < n_{ctx} \\ i \notin F}} e^{b_i}$$

$$\mathrm{cmp}(x, y) := \mathrm{sign}\left(d(e^{a_x} v_x - e^{a_y} v_y) - c(e^{a_x} - e^{a_y}) + f e^{a_x+a_y}\left(v_x e^{a_x+a_y} - v_y e^{a_x+a_y}\right)\right)$$

Let $t_{\min}$ and $t_{\max}$ be the minimum and maximum elements of S according to cmp.[6]

For a given choice of a non-decreasing sequence of tokens $t_0 \le \cdots \le t_{n_{ctx}-1} \in \mathbb{N}$ compatible with F and S and a given choice of permutation $\sigma : \mathbb{N} \to \mathbb{N}$ of the n_{ctx} positions fixing F ($t_i = t_F(\sigma(i))$ for $\sigma(i) \in F$; and $t_i \in S$ for $\sigma(i) \notin F$): let $s_{\sigma,\min}$ (and $s_{\sigma,\max}$) denote $s_{t_0,\ldots,t_{n_{ctx}-1},\sigma}$ when $t_i = t_{\min}$ for all $\sigma(i) \notin F$ (or $t_{\max}$, respectively).

Then for all such choices of sequence-permutation pairs,

$$s_{\sigma,\min} \le s_{t_0,\ldots,t_{n_{ctx}-1},\sigma} \le s_{\sigma,\max}.$$

This theorem follows by chaining two lemmas: that scores are extremized by considering pure sequences, and that the extremal pure sequences match the comparison function defined in the theorem statement.

Lemma 8 (Sequences scores are extremized on purer sequences)**.** *Fix all the same quantities as in Theorem 7.*

For any indices $0 \le i < j < n_{ctx}$, token values $x, y \in S$, the score for a sequence with $t_i = x \ne y = t_j$ is bounded on both sides by sequences with $t_i = t_j = x$ and $t_i = t_j = y$.

Proof. Let $s_{\alpha,\beta}$ be the sequence score with $t_i = \alpha$ and $t_j = \beta$, and define the score differences $\Delta_x := s_{x,x} - s_{x,y}$ and $\Delta_y := s_{y,y} - s_{x,y}$. It sufficies to show that $\mathrm{sign}(\Delta_x \Delta_y) \le 0$. To show this, we must only compute the sign of Δ_α for $\alpha \in \{x, y\}$ and show that whenever both Δ_x and Δ_y are non-zero, they have opposite signs.

We proceed by computation after defining some convenience variables for brevity:

$$C := \sum_{\substack{0 \le k < n_{\mathrm{ctx}} \\ k \ne i,j}} v_{t_k} e^{a_{t_k}+b_{\sigma(k)}} \qquad D := \sum_{\substack{0 \le k < n_{\mathrm{ctx}} \\ k \ne i,j}} e^{a_{t_k}+b_{\sigma(k)}}$$

[6] We will prove that cmp is transitive in the process of proving this theorem.

$$\tilde{\alpha} := \begin{cases} x & \text{if } \alpha = y \\ y & \text{if } \alpha = x \end{cases} \qquad i_\alpha := \begin{cases} i & \text{if } \alpha = x \\ j & \text{if } \alpha = y \end{cases} \qquad i_{\tilde{\alpha}} := \begin{cases} i & \text{if } \tilde{\alpha} = x \\ j & \text{if } \tilde{\alpha} = y \end{cases}$$

$$\begin{aligned} \operatorname{sign}(\Delta_\alpha) &= \operatorname{sign}\left(\frac{v_\alpha e^{a_\alpha + b_{\sigma(i)}} + v_\alpha e^{a_\alpha + b_{\sigma(j)}} + C}{e^{a_\alpha + b_{\sigma(i)}} + e^{a_\alpha + b_{\sigma(j)}} + D} - \frac{v_x e^{a_x + b_{\sigma(i)}} + v_y e^{a_y + b_{\sigma(j)}} + C}{e^{a_x + b_{\sigma(i)}} + e^{a_y + b_{\sigma(j)}} + D}\right) \\ &= \operatorname{sign}\left(\frac{v_\alpha e^{a_\alpha + b_{\sigma(i_\alpha)}} + v_\alpha e^{a_\alpha + b_{\sigma(i_{\tilde{\alpha}})}} + C}{e^{a_\alpha + b_{\sigma(i_\alpha)}} + e^{a_\alpha + b_{\sigma(i_{\tilde{\alpha}})}} + D} - \frac{v_\alpha e^{a_\alpha + b_{\sigma(i_\alpha)}} + v_{\tilde{\alpha}} e^{a_{\tilde{\alpha}} + b_{\sigma(i_{\tilde{\alpha}})}} + C}{e^{a_\alpha + b_{\sigma(i_\alpha)}} + e^{a_{\tilde{\alpha}} + b_{\sigma(i_{\tilde{\alpha}})}} + D}\right) \end{aligned}$$

Multiply through by positive denominators and simplify

$$\begin{aligned} &= \operatorname{sign}\Big(C\left(e^{a_{\tilde{\alpha}} + b_{\sigma(i_{\tilde{\alpha}})}} - e^{b_{\sigma(i_{\tilde{\alpha}})} + a_\alpha}\right) + D\left(v_\alpha e^{b_{\sigma(i_{\tilde{\alpha}})} + a_\alpha} - v_{\tilde{\alpha}} e^{a_{\tilde{\alpha}} + b_{\sigma(i_{\tilde{\alpha}})}}\right) \\ &\qquad + v_\alpha\left(e^{b_{\sigma(i_{\tilde{\alpha}})}} + e^{b_{\sigma(i_\alpha)}}\right) e^{a_{\tilde{\alpha}} + b_{\sigma(i_{\tilde{\alpha}})} + a_\alpha} - v_{\tilde{\alpha}}\left(e^{b_{\sigma(i_{\tilde{\alpha}})}} + e^{b_{\sigma(i_\alpha)}}\right) e^{a_{\tilde{\alpha}} + b_{\sigma(i_{\tilde{\alpha}})} + a_\alpha}\Big) \end{aligned}$$

Pulling out $e^{b_{\sigma(i_{\tilde{\alpha}})}}$

$$= \operatorname{sign}\left(e^{a_{\tilde{\alpha}} + a_\alpha}\left(e^{b_{\sigma(i_{\tilde{\alpha}})}} + e^{b_{\sigma(i_\alpha)}}\right)(v_\alpha - v_{\tilde{\alpha}}) + C\left(e^{a_{\tilde{\alpha}}} - e^{a_\alpha}\right) + D\left(e^{a_\alpha} v_\alpha - e^{a_{\tilde{\alpha}}} v_{\tilde{\alpha}}\right)\right)$$

Note that swapping α and $\tilde{\alpha}$ negates the sign. Hence, we have $\operatorname{sign}(\Delta_x) = -\operatorname{sign}(\Delta_y)$ and hence $s_{x,x} \le s_{x,y} \le s_{y,y}$ or $s_{y,y} \le s_{x,y} \le s_{x,x}$ as desired. □

Lemma 9 (Pure sequences are sorted according to cmp in Theorem 7)**.** *Fix all the same quantities as in Theorem 7.*

Fix tokens $x, y \in S$. Let $n := |F|$ be the number of non-fixed tokens. Fix sequences with n copies of x and y respectively: fix $t_{x,0} \le \cdots \le t_{x,n_{ctx}-1} \in \mathbb{N}$ and $t_{y,0} \le \cdots \le t_{y,n_{ctx}-1} \in \mathbb{N}$ compatible with F and S and given choices of permutations $\sigma_x, \sigma_y : \mathbb{N} \to \mathbb{N}$ of the n_{ctx} positions fixing F: $t_{x,i} = t_F(\sigma_x(i))$ for $\sigma_x(i) \in F$; $t_{y,i} = t_F(\sigma_y(i))$ for $\sigma_y(i) \in F$; $t_{x,i} = x$ for $\sigma_x(i) \notin F$; and $t_{y,i} = y$ for $\sigma_y(i) \notin F$.

Then

$$\operatorname{sign}((s_{\sigma_x, t_{x,0}, \ldots, t_{x,n_{ctx}-1}}) - (s_{\sigma_y, t_{y,0}, \ldots, t_{y,n_{ctx}-1}})) = \operatorname{cmp}(x, y)$$

Proof. The proof goes by straightforward computation.

$$\begin{aligned} &\operatorname{sign}((s_{\sigma_x, t_{x,0}, \ldots, t_{x,n_{\text{ctx}}-1}}) - (s_{\sigma_y, t_{y,0}, \ldots, t_{y,n_{\text{ctx}}-1}})) \\ &= \operatorname{sign}\left(\frac{v_x e^{a_x} f + c}{e^{a_x} f + d} - \frac{v_y e^{a_y} f + c}{e^{a_y} f + d}\right) \end{aligned}$$

Multiply through by non-negative denominators

$$\begin{aligned} &= \operatorname{sign}\left((v_x e^{a_x} f + c)(e^{a_y} f + d) - (v_y e^{a_y} f + c)(e^{a_x} f + d)\right) \\ &= \operatorname{sign}\left(-cfe^{a_x} + cfe^{a_y} + dfv_x e^{a_x} - dfv_y e^{a_y} + f^2 v_x e^{a_x + a_y} - f^2 v_y e^{a_x + a_y}\right) \end{aligned}$$

Use $f > 0$

$$\begin{aligned} &= \operatorname{sign}\left(-ce^{a_x} + ce^{a_y} + dv_x e^{a_x} - dv_y e^{a_y} + fv_x e^{a_x + a_y} - fv_y e^{a_x + a_y}\right) \\ &= \operatorname{sign}\left(c\left(e^{a_y} - e^{a_x}\right) + d\left(v_x e^{a_x} - v_y e^{a_y}\right) + f\left(v_x e^{a_x + a_y} - v_y e^{a_x + a_y}\right)\right) \\ &= \operatorname{cmp}(x, y) \end{aligned}$$

□

Corollary 10. *Define the relation $\le_{\text{cmp}}$ by $x \le_{\text{cmp}} y$ if and only if $\operatorname{cmp}(x, y) \in \{-1, 0\}$. The relation $\le_{\text{cmp}}$ is always transitive.*

Proof. Note that by Lemma 9, cmp is comparing two sequence scores. Since $\le$ is transitive over the reals, the relation $\le_{\text{cmp}}$ is also transitive. □

Figure 10: Recapitualtion of some relevant definitions from Figure 6

Let $d = \sqrt{d}$, $\mathbf{e}_t = (E)_t$, $\mathbf{p}_i = (P)_i$:

$$\begin{aligned}
\text{EQKE}(x_{-1}, x_i) &:= \frac{1}{\sqrt{d}}(\mathbf{e}_{x_{-1}} + \mathbf{p}_{-1})QK^T\mathbf{e}_{x_i}^T \\
\text{EQKP}(x_{-1}, i) &:= \frac{1}{\sqrt{d}}(\mathbf{e}_{x_{-1}} + \mathbf{p}_{-1})QK^T\mathbf{p}_i^T \\
\text{EVOU}(x_k) &:= \mathbf{e}_{x_k}VOU \\
\text{PVOU}(k) &:= \mathbf{p}_kVOU \\
\ell^{\text{EU}}(x_{-1}) &:= (\mathbf{e}_{x_{-1}} + \mathbf{p}_{-1})U \\
\Delta\ell_t^{\text{EU}}(x_{-1}, \max_i x_i) &:= \ell^{\text{EU}}(x_{-1})_t - \ell^{\text{EU}}(x_{-1})_{\max_i x_i}
\end{aligned}$$

Finally, we combine the previous lemmas to complete our proof of Theorem 7:

Proof of Theorem 7. Extremal sequences with scores $s_{\sigma,\min}$ and $s_{\sigma,\max}$ are guaranteed to exist because there are only finitely many elements of S and therefore only finitely many sequences. By Lemma 8, the extremal sequences must be pure (have $t_i = t_j$ whenever $\sigma(i), \sigma(j) \notin F$). By Lemma 9, the extremal sequences must have tokens that are extremal according to cmp. □

Algorithm 2 Counting Correct Sequences in Cubic Time, Full Version: Preliminaries

```
1: function CORRECTNESS(input-sequence)
2:     return MODEL-BEHAVIOR(input-sequence) == MAX(input-sequence)
3: end function
4: function MODEL-BEHAVIOR(input-sequence)
Require: input-sequence is a tensor of shape (n_ctx, ) with values in ℕ^<d_vocab
5:     t_max ← MAX(input-sequence)                          ▷ t_max ← max-token
6:     x ← input-sequence
7:     skip-score_t ← Δℓ_t^EU(x_{n_ctx−1}, t_max)
8:     attn-weights-unscaled_k ← EQKE(x_{n_ctx−1}, x_k) + EQKP(x_{n_ctx−1}, k)
9:     attn-weights ← SOFTMAX(attn-weights-unscaled)
10:    v_k ← EVOU(x_k)
11:    w_k ← PVOU(k)
12:    Δv_{k,i} ← v_{k,i} − v_{k,t_max}
13:    Δw_{k,i} ← w_{k,i} − w_{k,t_max}
14:    return max_{i≠t_max}(skip-score_i + Σ_{k=0}^{n_ctx−1}(Δv_{k,i} + Δw_{k,i}) · attn-weights_k)
15: end function
16: function CORRECTNESS-PESSIMIZING-OVER-POSITION-SLOW(input-sequence)
17:    x ← input-sequence
18:    return ALL(CORRECTNESS(perm + [x_{−1}]) for all perm ∈ PERMUTATIONS(x_{0:−1}))
19: end function
```

Theorem 11.

$$\mathbb{E}_{\mathbf{x}\sim U(0,1,\ldots,d_{vocab}-1)^{n_{ctx}}}\left[\mathcal{M}(\mathbf{x}) = \max_i x_i\right] \geq \text{CUBIC}(d_{vocab}, n_{ctx})$$

Proof. (sketch) Apply the previous theorems and lemmas to Algorithm 3. □

Theorem 12. *The running time of Algorithm 3, after using caching to avoid duplicate computations, is* $\mathcal{O}({d_{vocab}}^3 {n_{ctx}}^2)$.

Proof. The nested loops in CUBIC execute the innermost body $\mathcal{O}({d_{\text{vocab}}}^2 n_{\text{ctx}})$ times, and the summation on Line 42 costs $\mathcal{O}(n_{\text{ctx}})$ per iteration. What remains is to show that the call

Algorithm 3 Counting Correct Sequences in Cubic Time, Full Version. Lines are annotated with comments indicating the parameters for a cache to avoid duplicate computations.

function MODEL-BEHAVIOR-RELAXED(query-tok, max-tok, non-max-tok, n-copies-nonmax)
  $t_{\text{query}} \leftarrow$ query-tok, $t_{\text{max}} \leftarrow$ max-tok, $t' \leftarrow$ non-max-tok, $c \leftarrow$ n-copies-nonmax
Require: $0 \le t_{\text{query}} \le t_{\text{max}} < d_{\text{vocab}}$, $0 \le t' \le t_{\text{max}} < d_{\text{vocab}}$, $0 \le c < n_{\text{ctx}}$
Require: **if** n-copies-nonmax $= 0$ **then** non-max-tok $=$ max-tok
Require: **if** query-tok $\neq$ max-tok **then** n-copies-nonmax $< n_{\text{ctx}} - 1$
Ensure: **return** $\ge$ MODEL-BEHAVIOR($\mathbf{x}$) **for all** $\mathbf{x}$ with specified t_{query}, c copies of t' in non-query positions, and the remainder of the tokens equal to t_{max}
  skip-score$_t \leftarrow \Delta\ell_t^{\text{EU}}(t_{\text{query}}, t_{\text{max}})$ ▷ Cache by t_{max}, t_{query}, t
  $w_k \leftarrow \text{PVOU}(k)$ **for** $0 \le k < n_{\text{ctx}}$ ▷ Cache by k
  $\Delta w_{\max,i} \leftarrow \max_{0 \le k < n_{\text{ctx}}}(w_{k,i} - w_{k,t_{\text{max}}})$ ▷ Cache by t_{max}, i
  $v_k \leftarrow \text{EVOU}(k)$, $\Delta v_{k,i} \leftarrow v_{k,i} - v_{k,t_{\text{max}}}$ **for** $k \in \{t_{\text{query}}, t_{\text{max}}, t'\}$ ▷ Cache by t_{max}, k, i
  $a_k \leftarrow \text{EQKE}(t_{\text{query}}, k)$ **for** $k \in \{t_{\text{query}}, t_{\text{max}}, t'\}$ ▷ Cache by t_{query}, k
  $b_{n_{\text{ctx}}-1} \leftarrow \text{EQKP}(t_{\text{query}}, n_{\text{ctx}} - 1)$ ▷ Cache by t_{query}
  $b_{:-1} \leftarrow \text{SORT}(\text{EQKP}(t_{\text{query}}, : -1))$ ▷ Cache by t_{query}, k
  attn-weights-unscaled$_{:,n_{\text{ctx}}-1} \leftarrow a_{t_{\text{query}}} + b_{n_{\text{ctx}}-1}$ ▷ Cache by t_{query}
  attn-weights-unscaled$_{0,k} \leftarrow a_{t_{\text{max}}} + b_{k-c}$ **for** $c \le k < n_{\text{ctx}} - 1$ ▷ Cache by t_{query}, t_{max}, k, c
  attn-weights-unscaled$_{1,k} \leftarrow a_{t_{\text{max}}} + b_k$ **for** $c \le k < n_{\text{ctx}} - 1$ ▷ Cache by t_{query}, t_{max}, k, c
  attn-weights-unscaled$_{0,k} \leftarrow a_{t'} + b_{c+k}$ **for** $0 \le k < n_{\text{ctx}} - 1 - c$ ▷ Cache by t_{query}, t', k, c
  attn-weights-unscaled$_{1,k} \leftarrow a_{t'} + b_k$ **for** $0 \le k < n_{\text{ctx}} - 1 - c$ ▷ Cache by t_{query}, t', k, c
  attn-weights$_0 \leftarrow \text{SOFTMAX}(\text{attn-weights-unscaled}_0)$ ▷ Cache by t_{query}, t_{max}, t', k, c
  attn-weights$_1 \leftarrow \text{SOFTMAX}(\text{attn-weights-unscaled}_1)$ ▷ Cache by t_{query}, t_{max}, t', k, c
  if $c = 0$ **then** ▷ In this case, attn-weights$_{0,k}$ = attn-weights$_{1,k}$, so we drop the first subscript
    return $\max_{i \ne t_{\text{max}}}(\text{skip-score}_i + \Delta w_{\max,i} + \sum_{k=0}^{n_{\text{ctx}}-1} \Delta v_{k,i} \cdot \text{attn-weights}_k)$
  else
    $\Delta v_{k,i} \leftarrow \Delta v_{t_{\text{query}},i}$ **for** $c \le k < n_{\text{ctx}} - 1$
    $\Delta v_{k,i} \leftarrow \Delta v_{t',i}$ **for** $c \le k < n_{\text{ctx}} - 1$
    $\Delta v_{n_{\text{ctx}}-1,i} \leftarrow \Delta v_{t_{\text{query}},n_{\text{ctx}}-1}$
    return $\max_{i \ne t_{\text{max}}} \text{skip-score}_i + \max \begin{cases} \sum_{k=0}^{n_{\text{ctx}}-1} \max_{i \ne t_{\text{max}}}(\Delta w_{\max,i} + \Delta v_{k,i}) \cdot \text{attn-weights}_{0,k} \\ \sum_{k=0}^{n_{\text{ctx}}-1} \max_{i \ne t_{\text{max}}}(\Delta w_{\max,i} + \Delta v_{k,i}) \cdot \text{attn-weights}_{1,k} \end{cases}$
  end if
end function
function RELAXED-CORRECTNESS-PESSIMIZING-OVER-POSITION(t_{query}, t_{max}, t', c)
  ▷ runs the model on a relaxed variant of input sequences compatible with the arguments
Ensure: **return** is False **if** CORRECTNESS-PESSIMIZING-OVER-POSITION-SLOW($\mathbf{x}$) is False **for** *any* $\mathbf{x}$ with specified t_{query}, c copies of t' in non-query positions, and the remainder of the tokens equal to t_{max}
  return MODEL-BEHAVIOR-RELAXED$(t_{\text{query}}, t_{\text{max}}, t', c) < 0$
end function
function CUBIC(d_{vocab}, n_{ctx})
  count $\leftarrow 0$ ▷ # of correct sequences
  for $t_{\text{max}} \in$ RANGE(d_{vocab}) **do** ▷ $t_{\text{max}} \leftarrow$ max-token
    for $0 \le t_{\text{query}} \le t_{\text{max}}$ **do** ▷ $t_{\text{query}} \leftarrow$ query-token
      $c_{\max} \leftarrow n_{\text{ctx}} - 1$ **if** $t_{\text{query}} = t_{\text{max}}$ **else** $n_{\text{ctx}} - 2$ ▷ maximum copies of nonmax
      for $0 \le c \le c_{\max}$ **do** ▷ number of valid choices for the non-max token
        RCPOP$(\vec{\chi}) \leftarrow$ RELAXED-CORRECTNESS-PESSIMIZING-OVER-POSITION$(\vec{\chi})$
        if $c = 0$ **then**
          t-count $\leftarrow 1$ **if** RCPOP$(t_{\text{query}}, t_{\text{max}}, t_{\text{max}}, 0)$ **else** 0
        else
          t-count $\leftarrow \sum_{t'=0}^{t_{\text{max}}-1} 1$ **if** RCPOP$(t_{\text{query}}, t_{\text{max}}, t', c)$ **else** 0
        end if
        count $\leftarrow$ count $+ \sum_{i=0}^{c} \binom{n_{\text{ctx}}-1}{i} \cdot (\text{t-count})^i$ ▷ taking $0^0 = 1$ conventionally
      end for
    end for
  end for
  return count
end function

to RELAXED-CORRECTNESS-PESSIMIZING-OVER-POSITION$(t_{\text{query}}, t_{\text{max}}, t', c)$ costs $\mathcal{O}(n_{\text{ctx}})$ when $c \neq 0$ and at most $\mathcal{O}(d_{\text{vocab}} n_{\text{ctx}})$ when $c = 0$ and $t' = t_{\text{max}}$.

The matrix multiplications in EQKE, EQKP, EVOU, PVOU, and ℓ^{EU} can be cached upfront, costing $\mathcal{O}(\max(d_{\text{vocab}}, d_{\text{model}}, n_{\text{ctx}})^2 d_{\text{model}}) \leq \mathcal{O}({d_{\text{vocab}}}^3)$ since we assume $d_{\text{vocab}} > d_{\text{model}}$ and $d_{\text{vocab}} > n_{\text{ctx}}$.

The sorting on Line 9 can also be cached upfront (per t_{query}), costing $\mathcal{O}(d_{\text{vocab}} n_{\text{ctx}} \log n_{\text{ctx}})$.

Note that each variable assignment in RELAXED-CORRECTNESS-PESSIMIZING-OVER-POSITION can be cached into a table parameterized over at most three variables which range over d_{vocab} and over at most two variables that range over n_{ctx}.

What remains is the **return** statements.

When $c = 0$, we have on Line 18: **return** $\max_{i \neq t_{\text{max}}}(\text{skip-score}_i + \Delta w_{\text{max},i} + \sum_{k=0}^{n_{\text{ctx}}-1} \Delta v_{k,i} \cdot \text{attn-weights}_k)$. This is $\mathcal{O}(d_{\text{vocab}} n_{\text{ctx}})$ as desired.

When $c \neq 0$, we have on Line 23:
return $\max_{i \neq t_{\text{max}}} \text{skip-score}_i + \max \begin{cases} \sum_{k=0}^{n_{\text{ctx}}-1} \max_{i \neq t_{\text{max}}}(\Delta w_{\text{max},i} + \Delta v_{k,i}) \cdot \text{attn-weights}_{\text{min},k} \\ \sum_{k=0}^{n_{\text{ctx}}-1} \max_{i \neq t_{\text{max}}}(\Delta w_{\text{max},i} + \Delta v_{k,i}) \cdot \text{attn-weights}_{\text{max},k} \end{cases}$

We can cache $\max_{i \neq t_{\text{max}}} \text{skip-score}_i$ per t_{max} and t_{query}, costing $\mathcal{O}({d_{\text{vocab}}}^3 n_{\text{ctx}})$. We can cache $\max_{i \neq t_{\text{max}}}(\Delta w_{\text{max},i} + \Delta v_{k,i})$ per t_{max} and k costing $\mathcal{O}({d_{\text{vocab}}}^2 n_{\text{ctx}})$. Finally, we can compute the summation in cost $\mathcal{O}(n_{\text{ctx}})$ per loop iteration, as required. □

F Details of sub-cubic proof

In this section we fill in the details lacking from Subsection 4.2.

In Appendix E we proved an intricate version of convexity of softmax where, modulo pessimizing in unrealistic ways over the attention paid to positions for the computation done on positional encodings, all extremal relaxed sequences correspond to actual sequences.

When we only get a budget of $\mathcal{O}({d_{\text{vocab}}}^2 n_{\text{ctx}})$ extremal relaxed cases to consider, though, we must pessimize more, which gives us a simpler version of the convexity theorem and proof. Notably, when we restrict our sequences to have only two tokens (the max token t_{max} and the non-max token t'), most of the theorems from Appendix E.3 get significantly simpler.

Additionally, we must pessimize separately over the token value (v) and token attention (b) computations in order to allow efficient computation (Theorem 15).

F.1 Proof of baseline sub-cubic result

For this subsection, all theorems are parameterized over the following quantities. Fix a token value function (à la a row difference in EVOU) $v : \mathbb{N} \to \mathbb{R}$ and a token attention function (à la EQKE for a fixed query token) $a : \mathbb{N} \to \mathbb{R}$. Fix a position value function (à la a row difference in PVOU) $w : \mathbb{N} \to \mathbb{R}$ and a position attention function (à la EQKP for a fixed query token) $b : \mathbb{N} \to \mathbb{R}$. Fix a total number of tokens n_{ctx}.

Definition 9. *We can* define a sequence of tokens via mapping from positions *by specifying a subset of valid tokens $S \subseteq \mathbb{N}^{<d_{vocab}}$ paired with a function $T : \mathbb{N}^{<n_{ctx}} \to S$ specifying which token is in each position.*

Definition 10. *Given a subset of valid tokens $S \subseteq \mathbb{N}^{<d_{vocab}}$ and a function $T : \mathbb{N}^{<n_{ctx}} \to S$ specifying which token is in each position, define the* sequence score

$$s_T := \sum_{0 \leq i < n_{ctx}} v_{T(i)} e^{a_{T(i)} + b_i} \Bigg/ \sum_{0 \leq i < n_{ctx}} e^{a_{T(i)} + b_i}$$

Definition 11. *Given a subset of valid tokens $S \subseteq \mathbb{N}^{<d_{vocab}}$ and a function $T : \mathbb{N}^{<n_{ctx}} \to S$ specifying which token is in each position and two indices $0 \leq i, j < n_{ctx}$, define the* swaped mapping $T_{i \leftrightarrow j}$ *be the function that is T except swapping i and j:*

$$T_{i \leftrightarrow j}(k) = \begin{cases} T(i) & \text{if } k = j \\ T(j) & \text{if } k = i \\ T(k) & \text{otherwise} \end{cases}$$

Lemma 13 (Characterization of swapping tokens in a two-token sequence)**.** *Fix two tokens $t_0 < t_1 \in \mathbb{N}$ and a function $T : \mathbb{N}^{<n_{ctx}} \to \{t_0, t_1\}$ specifying which token is in each position.*

Define $\Delta_{T,i\leftrightarrow j}$ to be the difference in sequence scores when you swap i and j:

$$\Delta_{T,i\leftrightarrow j} := s_{T_{i\leftrightarrow j}} - s_T$$

Then

$$\operatorname{sign}\left(\Delta_{T,i\leftrightarrow j}\right) = -\operatorname{sign}\left(b_i - b_j\right)\operatorname{sign}\left(v_{T(i)} - v_{T(j)}\right)$$

Proof. Lemma 3 gives us the result directly when $a_{T(i)} = a_{T(j)}$. Otherwise, we get

$$\operatorname{sign}\left(\Delta_{T,i\leftrightarrow j}\right) = \operatorname{sign}\left(a_{T(i)} - a_{T(j)}\right)\operatorname{sign}\left(b_i - b_j\right)\operatorname{sign}\left(s_T - \frac{v_{T(i)}e^{a_{T(i)}} - v_{T(j)}e^{a_{T(j)}}}{e^{a_{T(i)}} - e^{a_{T(j)}}}\right)$$

Hence all that remains is to show that

$$\operatorname{sign}\left(s_T\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right) - v_{T(i)}e^{a_{T(i)}} + v_{T(j)}e^{a_{T(j)}}\right) = -\operatorname{sign}\left(v_{T(i)} - v_{T(j)}\right)$$

Define $\bar{v} := \frac{1}{2}(v_{T(i)} + v_{T(j)})$ and define $\Delta v := \frac{1}{2}(v_{T(i)} - v_{T(j)})$ so that $v_{T(i)} = \bar{v} + \Delta v$ and $v_{T(j)} = \bar{v} - \Delta v$. Assume WLOG that $T(i) = 0$ and $T(j) = 1$ so that $v_{T(p)} = \bar{v} + (-1)^{T(p)}\Delta v$ for all p.

Then we have

$$\begin{aligned}
&\operatorname{sign}\left(s_T\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right) - v_{T(i)}e^{a_{T(i)}} + v_{T(j)}e^{a_{T(j)}}\right)\\
&= \operatorname{sign}\left(s_T\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right) - \bar{v}\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right) - \Delta v e^{a_{T(i)}} - \Delta v e^{a_{T(j)}}\right)\\
&= \operatorname{sign}\left(\frac{\sum_{0\le p<n_{\text{ctx}}} v_{T(p)}e^{a_{T(p)}+b_p}}{\sum_{0\le p<n_{\text{ctx}}} e^{a_{T(p)}+b_p}}\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right) - \bar{v}\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right) - \Delta v\left(e^{a_{T(i)}} + e^{a_{T(j)}}\right)\right)\\
&= \operatorname{sign}\left(\frac{\sum_{0\le p<n_{\text{ctx}}} \left(\bar{v} + (-1)^{T(p)}\Delta v\right)e^{a_{T(p)}+b_p}}{\sum_{0\le p<n_{\text{ctx}}} e^{a_{T(p)}+b_p}}\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right) - \cancel{\bar{v}\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right)} - \Delta v\left(e^{a_{T(i)}} + e^{a_{T(j)}}\right)\right)\\
&= \operatorname{sign}(\Delta v)\operatorname{sign}\left(\frac{e^{a_{T(i)}}\sum_{\substack{0\le p<n_{\text{ctx}}\\ T(p)=T(i)}} e^{b_p} - e^{a_{T(j)}}\sum_{\substack{0\le p<n_{\text{ctx}}\\ T(p)=T(j)}} e^{b_p}}{\sum_{0\le p<n_{\text{ctx}}} e^{a_{T(p)}+b_p}}\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right) - e^{a_{T(i)}} - e^{a_{T(j)}}\right)\\
&= \operatorname{sign}(v_{T(i)} - v_{T(j)})\operatorname{sign}\left(\frac{e^{a_{T(i)}}\sum_{\substack{0\le p<n_{\text{ctx}}\\ T(p)=T(i)}} e^{b_p} - e^{a_{T(j)}}\sum_{\substack{0\le p<n_{\text{ctx}}\\ T(p)=T(j)}} e^{b_p}}{\sum_{0\le p<n_{\text{ctx}}} e^{a_{T(p)}+b_p}}\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right) - e^{a_{T(i)}} - e^{a_{T(j)}}\right)
\end{aligned}$$

Define

$$P_i := \sum_{\substack{0\le p<n_{\text{ctx}}\\ T(p)=T(i)}} e^{b_p} \qquad P_j := \sum_{\substack{0\le p<n_{\text{ctx}}\\ T(p)=T(j)}} e^{b_p}$$

so that we get

$$\operatorname{sign}\left(s_T\left(e^{a_{T(i)}} - e^{a_{T(j)}}\right) - v_{T(i)}e^{a_{T(i)}} + v_{T(j)}e^{a_{T(j)}}\right)$$

$$= \operatorname{sign}(v_{T(i)} - v_{T(j)}) \operatorname{sign}\left(\frac{e^{a_{T(i)}} P_i - e^{a_{T(j)}} P_j}{e^{a_{T(i)}} P_i + e^{a_{T(j)}} P_j} (e^{a_{T(i)}} - e^{a_{T(j)}}) - e^{a_{T(i)}} - e^{a_{T(j)}}\right)$$

Multiply through by the positive denominator and expand out so that we get

$$\begin{aligned}
&= \operatorname{sign}(v_{T(i)} - v_{T(j)}) \operatorname{sign}\left(-2e^{a_{T(i)}+a_{T(j)}} P_i - 2e^{a_{T(i)}+a_{T(j)}} P_j\right) \\
&= -\operatorname{sign}(v_{T(i)} - v_{T(j)}) \operatorname{sign}\left(e^{a_{T(i)}+a_{T(j)}} P_i + e^{a_{T(i)}+a_{T(j)}} P_j\right) \\
&= -\operatorname{sign}(v_{T(i)} - v_{T(j)})
\end{aligned}$$

□

Theorem 14 (Pessimization over sequence ordering for two-token sequences is simple)**.** *Let $\sigma_s : \mathbb{N} \to \mathbb{N}$ denote a permutation of the n_{ctx} positions that sorts them according to b: for $0 \leq i, j < n_{ctx}$, $b_i \leq b_j$ whenever $\sigma_s(i) < \sigma_s(j)$. Fix two tokens $t_0 < t_1 \in \mathbb{N}$.*

Let n_{t_0} be the number of $p \in [0, n_{ctx})$ with $T(p) = t_0$ and let n_1 be the number of $p \in [0, n_{ctx})$ with $T(p) = t_{t_1}$. Note that $n_{t_0} + n_{t_1} = n_{ctx}$.

Define $t_{\min} := \operatorname{argmin}_{t \in \{t_0, t_1\}} v_t$ and define $t_{\max} := \operatorname{argmax}_{t \in \{t_0, t_1\}} v_t$.

Define $T_{\min}, T_{\max} : \mathbb{N}^{<n_{ctx}} \to \{t_0, t_1\}$ to be the assignment of tokens to positions that pays the least (respectively, most) attention to $t_{\max}$:

$$T_{\min}(i) := \begin{cases} t_{\max} & \text{if } 0 \leq \sigma_s(i) < n_{t_{\max}} \\ t_{\min} & \text{if } n_{t_{\max}} \leq \sigma_s(i) < n_{ctx} \end{cases}$$

$$T_{\max}(i) := \begin{cases} t_{\min} & \text{if } 0 \leq \sigma_s(i) < n_{t_{\min}} \\ t_{\max} & \text{if } n_{t_{\min}} \leq \sigma_s(i) < n_{ctx} \end{cases}$$

Then we have that

$$s_{T_{\min}} \leq s_T \leq s_{T_{\max}}$$

Proof. The extremality of $s_{T_{\min}}$ and $s_{T_{\max}}$ follows straightforwardly from Theorem 4.

All that remains is $s_{T_{\min}} \leq s_{T_{\max}}$.

This follows from noting by Lemma 13 that swapping two tokens in $T_{\min}$ *increases* the sequence score, while the reverse is true of $s_{T_{\max}}$, thus showing that it must be $s_{T_{\min}}$ that is the minimum and $s_{T_{\max}}$ that is the maximum and not vice versa. □

Definition 12. *Given a subset of valid tokens $S \subseteq \mathbb{N}^{<d_{vocab}}$ and a function $T : \mathbb{N}^{<n_{ctx}} \to S$ specifying which token is in each position define the* full sequence score *s'_T:*

$$s'_T := \sum_{0 \leq i < n_{ctx}} (v_{T(i)} + w_i) e^{a_{T(i)}+b_i} \Bigg/ \sum_{0 \leq i < n_{ctx}} e^{a_{T(i)}+b_i}$$

Theorem 15 (Independent pessimization over positional contributions and token attention and token value is possible)**.** *Fix two tokens $t_0 < t_1 \in \mathbb{N}$. Let $T_{\min}, T_{\max} : \mathbb{N}^{<n_{ctx}} \to \{t_0, t_1\}$ and $t_{\max}$, $t_{\min}$ be as in Theorem 14. Fix a set S of valid tokens with $t_0, t_1 \in S$.*

Define relaxed versions $T'_{\max}, T'_{\min} : \mathbb{N}^{<n_{ctx}} \to S$ of $T_{\max}$ and $T_{\min}$:

$$T'_{\max}(i) := \begin{cases} T_{\max}(i) & \text{if } T_{\max}(i) = t_{\max} \\ \operatorname{argmin}_{\substack{j \in S \\ j \neq t_{\max}}} s_j & \text{otherwise} \end{cases}$$

$$T'_{\min}(i) := \begin{cases} T_{\min}(i) & \text{if } T_{\max}(i) = t_{\max} \\ \operatorname{argmax}_{\substack{j \in S \\ j \neq t_{\max}}} s_j & \text{otherwise} \end{cases}$$

That is, $T'_{\max}$ replaces $t_{\min}$ with whatever token in S draws the least attention away from $t_{\max}$, while $T'_{\min}$ replaces $t_{\min}$ with whichever token in S draws the most attention away from $t_{\max}$.

Define relaxed extremal sequence scores $r_{T_{\max}}$, $r_{T_{\min}}$*:*

$$r_{T_{\min}} := \min_{0 \le i < n_{ctx}} w_i + \left(\sum_{0 \le i < n_{ctx}} v_{T_{\min}(i)} e^{a_{T'_{\min}(i)} + b_i} \Big/ \sum_{0 \le i < n_{ctx}} e^{a_{T'_{\min}(i)} + b_i} \right)$$

$$r_{T_{\max}} := \max_{0 \le i < n_{ctx}} w_i + \left(\sum_{0 \le i < n_{ctx}} v_{T_{\min}(i)} e^{a_{T'_{\max}(i)} + b_i} \Big/ \sum_{0 \le i < n_{ctx}} e^{a_{T'_{\max}(i)} + b_i} \right)$$

Then $r_{T_{\min}} \le s'_{T_{\min}}$ *and* $s'_{T_{\max}} \le r_{T_{\max}}$.

Proof. (sketch) Essentially the same as the proof of Theorem 6. □

Note that in practice, we take S to be the set of all tokens less than $t_{\text{max}} - g$ for some minimum gap g. This allows us to share computation across the various maximum tokens to reduce overall computational complexity.

Algorithm 4 Counting Correct Sequences in Subcubic Time, Preliminaries

function INPUT-SEQUENCE-COMPATIBLE-WITH(input-sequence, d_{vocab}, n_{ctx}, t_{max}, t_{query}, c, g)
  $\mathbf{x} \leftarrow$ input-sequence
  return False **if** $\mathbf{x} \notin (\mathbb{N}^{<d_{\text{vocab}}})^{n_{\text{ctx}}}$ ▷ the sequence is not made of valid tokens
  return False **if** $x_{-1} \neq t_{\text{query}}$ ▷ wrong query token
  return False **if** $\max_i x_i \neq t_{\text{max}}$ ▷ wrong max token
  return False **if** $|\{i \in \mathbb{N}^{<n_{\text{ctx}}} \mid x_i \neq t_{\text{max}}\}| \neq c$ ▷ wrong count of non-max toks
  return ALL($x_i = t_{\text{max}}$ **or** $t_{\text{max}} - x_i \ge g$ **for** $0 \le i < n_{\text{ctx}}$) ▷ check gap on non-max toks
end function
function CORRECTNESS-PESSIMIZING-OVER-GAP-SLOW(d_{vocab}, n_{ctx}, t_{max}, t_{query}, c, g)
  return ALL(CORRECTNESS($\mathbf{x}$) **for all** $\mathbf{x}$ s.t. INPUT-SEQUENCE-COMPATIBLE-WITH($\mathbf{x}$, d_{vocab}, n_{ctx}, t_{max}, t_{query}, c, g))
end function
function SUBCUBIC(d_{vocab}, n_{ctx}, G)
  count $\leftarrow 0$ ▷ # of correct sequences
  $G_{t_{\text{max}}, t_{\text{query}}, c} \leftarrow \text{MIN}(t_{\text{max}}, \text{MAX}(1, G_{t_{\text{max}}, t_{\text{query}}, c}))$ ▷ Clip G to valid range
  $G^*_{t_{\text{max}}, c} \leftarrow \min_{t \le t_{\text{max}}} G_{t_{\text{max}}, t, c}$ ▷ Cache running minima
  for $t_{\text{max}} \in \text{RANGE}(d_{\text{vocab}})$ **do** ▷ $t_{\text{max}} \leftarrow$ max-token
    for $0 \le t_{\text{query}} \le t_{\text{max}}$ **do** ▷ $t_{\text{query}} \leftarrow$ query-token
      $c_{\text{max}} \leftarrow 0$ **if** $t_{\text{query}} = t_{\text{max}}$ **else** 1 ▷ minimum copies of nonmax
      $c_{\text{max}} \leftarrow \begin{cases} 0 & \textbf{if } t_{\text{max}} = 0 \\ n_{\text{ctx}} - 1 & \textbf{if } t_{\text{max}} = t_{\text{query}} \\ n_{\text{ctx}} - 2 & \textbf{else} \end{cases}$ ▷ maximum copies of nonmax
      for $c_{\text{min}} \le c \le c_{\text{max}}$ **do** ▷ number of valid choices for the non-max token
        $g \leftarrow G_{t_{\text{max}}, t_{\text{query}}, c}$
        $g^* \leftarrow G^*_{t_{\text{max}}, c}$
        q-gap $\leftarrow t_{\text{max}} - t_{\text{query}}$
        $\text{RCPOG}(\vec{\chi}) \leftarrow$ RELAXED-CORRECTNESS-PESSIMIZING-OVER-GAP($\vec{\chi}$)
        if (q-gap $= 0$ **or** q-gap $\ge g$) **and** $\text{RCPOG}(d_{\text{vocab}}, n_{\text{ctx}}, t_{\text{max}}, t_{\text{query}}, c, g, g^*)$ **then**
          $c' \leftarrow c$ **if** $t_{\text{query}} = t_{\text{max}}$ **else** $c - 1$ ▷ # of non-max non-query tokens
          count $\leftarrow$ count $+ \binom{n_{\text{ctx}}-1}{c'} (t_{\text{max}} - g)^{c'}$ ▷ taking $0^0 = 1$ conventionally
        end if
      end for
    end for
  end for
  return count
end function

Theorem 16. *For all G,*

$$\mathbb{E}_{\mathbf{x} \sim U(0,1,\ldots,d_{vocab}-1)^{n_{ctx}}} \left[\mathcal{M}(\mathbf{x}) = \max_i x_i \right] \ge \text{SUBCUBIC}(d_{vocab}, n_{ctx}, G)$$

Algorithm 5 Counting Correct Sequences in Subcubic Time

function MODEL-BEHAVIOR-RELAXED-OVER-GAP(t_{max}, t_{query}, c, g, g^*)
Ensure: CORRECTNESS-PESSIMIZING-OVER-GAP-SLOW is False $\implies$ result is False
Require: $0 \leq g^* \leq G \leq t_{\text{max}}$
Require: **if** $c = 0$ **then** $t_{\text{query}} = t_{\text{max}}$
    skip-score $\leftarrow \max_i \ell^{\text{EU}}(t_{\text{query}})_i - \min_i \ell^{\text{EU}}(t_{\text{query}})_i$ ▷ Cache by t_{query}
    $v_k \leftarrow \text{EVOU}(k)$
    $w_k \leftarrow \text{PVOU}(k)$
    $\Delta w_{\max,i} \leftarrow \max_p w_{p,i} - w_{p,t_{\text{max}}}$ ▷ Cache by t_{max}, i
    $\Delta w_{\max,\max} \leftarrow \max_i \Delta w_{\max,i}$ ▷ Cache by t_{max}
    $\Delta v_k \leftarrow \max_i v_{k,i} - \min_i v_{k,i}$ ▷ Cache by k
    $\Delta v_{\max} \leftarrow \max_{0 \leq k \leq t_{\text{max}} - g^*} \Delta v_k$ ▷ Cache by t_{max}, c
    $\Delta v_i^{t_{\text{max}}} \leftarrow v_{t_{\text{max}},i} - v_{t_{\text{max}},t_{\text{max}}}$ ▷ Cache by t_{max}
    $\Delta v_{\max}^{t_{\text{max}}} \leftarrow \max_{i \neq t_{\text{max}}} \Delta v_i^{t_{\text{max}}}$ ▷ Cache by t_{max}
    if $c = 0$ **then**
        $\ell_i \leftarrow \ell^{\text{EU}}(t_{\text{max}})_i + v_{t_{\text{max}},i} + \Delta w_{\max,i}$
        return $\max_{i \neq t_{\text{max}}} (\ell_i - \ell_{t_{\text{max}}})$
    end if
    $b_{:,n_{\text{ctx}}-1} \leftarrow \text{EQKP}(t_{\text{query}}, n_{\text{ctx}} - 1)$ ▷ Cache by t_{query}
    $b_{0,:-1} \leftarrow \text{SORT}(\text{EQKP}(t_{\text{query}}, :-1))$ ▷ Cache by t_{query}, k
    $b_{1,:-1} \leftarrow \text{REVERSE}(b_{0,:-1})$
    $a_k \leftarrow \text{EQKE}(t_{\text{query}}, k)$ ▷ Cache by t_{query}, k
    $a_{\min,k} \leftarrow \min_{0 \leq i \leq k} a_i$ ▷ Cache by t_{query}, k, compute in amortized $\mathcal{O}({d_{\text{vocab}}}^2)$
    $a_{\max,k} \leftarrow \max_{0 \leq i \leq k} a_i$ ▷ Cache by t_{query}, k, compute in amortized $\mathcal{O}({d_{\text{vocab}}}^2)$
    $\Delta a_{\max} \leftarrow a_{t_{\text{max}}} - a_{\min,t_{\text{max}}-g}$ ▷ Cache by t_{query}, t_{max}, c
    $\Delta a_{\min} \leftarrow a_{t_{\text{max}}} - a_{\max,t_{\text{max}}-g}$ ▷ Cache by t_{query}, t_{max}, c
    idx-set $\leftarrow \{0, \ldots, n_{\text{ctx}} - c - 1\}$ **if** $t_{\text{max}} \neq t_{\text{query}}$ **else** $\{0, \ldots, n_{\text{ctx}} - c - 2, n_{\text{ctx}} - 1\}$
    attn-weights-unscaled$_{0,k} \leftarrow \Delta a_{\min} + b_{0,k}$ **if** $k \in$ idx-set
    attn-weights-unscaled$_{1,k} \leftarrow \Delta a_{\max} + b_{0,k}$ **if** $k \in$ idx-set ▷ Cache by t_{query}, t_{max}, k, c
    attn-weights$_0 \leftarrow \text{SOFTMAX}(\text{attn-weights-unscaled}_0)$ ▷ Cache by t_{query}, t_{max}, k, c
    attn-weights$_1 \leftarrow \text{SOFTMAX}(\text{attn-weights-unscaled}_1)$ ▷ Cache by t_{query}, t_{max}, k, c
    attn-max$_0 \leftarrow \sum_{i \in \text{idx-set}}$ attn-weights$_{0,i}$
    attn-max$_1 \leftarrow \sum_{i \in \text{idx-set}}$ attn-weights$_{1,i}$
    attn-max $\leftarrow$ attn-max$_0$ **if** $v_{\max}^{t_{\text{max}}} \geq \Delta v_{\max}$ **else** attn-max$_1$
    return skip-score $+ \Delta w_{\max,\max} +$ attn-max$v_{\max}^{t_{\text{max}}} + (1 -$ attn-max$)\Delta v_{\max}$
end function
function RELAXED-CORRECTNESS-PESSIMIZING-OVER-GAP(d_{vocab}, n_{ctx}, t_{max}, t_{query}, c, g, g^*)
        ▷ runs the model on a relaxed variant of input sequences compatible with the arguments
Ensure: CORRECTNESS-PESSIMIZING-OVER-GAP-SLOW is False $\implies$ result is False
Ensure: **return** is False **if** CORRECTNESS-PESSIMIZING-OVER-GAP-SLOW($\mathbf{x}$) is False **for** *any* $\mathbf{x}$ with specified t_{max}, t_{query}, and c tokens not equal to t_{max}
    return MODEL-BEHAVIOR-RELAXED-OVER-GAP$(t_{\text{max}}, t_{\text{query}}, c, g, g^*) < 0$
end function

Proof. (sketch) Apply preceding lemmas and theorems to Algorithm 5 □

Theorem 17. *The running time of Algorithm 5, after using caching to avoid duplicate computations, is* $\mathcal{O}(d_{vocab}{}^2 d_{model} + d_{vocab}{}^2 n_{ctx}{}^2)$.

Proof. (sketch) Sum the complexities indicated along the right side of Algorithm 3. The $d_{\text{vocab}}{}^2 d_{\text{model}}$ term comes from the precomputing $EVOU$, EU, and EQKP. The $d_{\text{vocab}}{}^2 n_{\text{ctx}}{}^2$ term comes from the softmax over n_{ctx} tokens for $O(d_{\text{vocab}}{}^2 n_{\text{ctx}})$ pessimized pure sequences. Confirming that none of the complexities on the right side exceeds $\mathcal{O}(d_{\text{vocab}}{}^2 d_{\text{model}} + d_{\text{vocab}}{}^2 n_{\text{ctx}}{}^2)$ completes the proof.

□

F.2 The mean+diff trick

Suppose we have quantities $f_{x,y}$ and $g_{y,z}$ and we want to pessimize (WLOG, suppose minimize) the quantity $f_{x,y} + g_{y,z}$ over x, y, and z in time less than $\mathcal{O}(n_x n_y n_z)$, say we allow $\mathcal{O}(n_x n_y + n_y n_z + n_x n_z)$. Also suppose the variation of f over the y axis is much larger than the variation of f over the x-axis.

We can of course say

$$\min_{x,y} f_{x,y} + \min_{y,z} g_{y,z} \leq f_{x,y} + g_{y,z}$$

But we can do better!

Note that

$$f_{x,y} = \mathbb{E}_x f_{x,y} + (f_{x,y} - \mathbb{E}_x f_{x,y})$$

Suppose that $f_{x,y}$ varies much less over x than it does over y, and much less than $g_{y,z}$ varies over either of y and z. This will make the following bound a good approximation, though the bound is sound even without this assumption. We can write

$$\begin{aligned}
f_{x,y} + g_{y,z} &\geq \min_{x,y,z}[f_{x,y} + g_{y,z}] \\
&= \min_{x,y,z}[\mathbb{E}_x f_{x,y} + g_{y,z} + f_{x,y} - \mathbb{E}_x f_{x,y}] \\
&\geq \min_{x,y,z}[\mathbb{E}_x f_{x,y} + g_{y,z}] + \min_{x,y,z}[f_{x,y} - \mathbb{E}_x f_{x,y}] \\
&= \min_{y,z}[\mathbb{E}_x f_{x,y} + g_{y,z}] + \min_{x,y}[f_{x,y} - \mathbb{E}_x f_{x,y}]
\end{aligned}$$

By averaging the variation over certain axes, we have

Theorem 18 (Mean+Diff)**.**

$$\begin{aligned}
\min_{x,y,z} f_{x,y} + g_{y,z} &\geq \min_{y,z}[\mathbb{E}_x f_{x,y} + g_{y,z}] + \min_{x,y}[f_{x,y} - \mathbb{E}_x f_{x,y}] \\
\max_{x,y,z} f_{x,y} + g_{y,z} &\leq \max_{y,z}[\mathbb{E}_x f_{x,y} + g_{y,z}] + \max_{x,y}[f_{x,y} - \mathbb{E}_x f_{x,y}]
\end{aligned}$$

and the RHSs can be computed in time $\mathcal{O}(n_x n_y + n_y n_z + n_x n_z)$ *for* n_x*,* n_y*, and* n_z *the number of possible values of* x*,* y*, and* z*, respectively.*

Example for how this helps with small variation:

Take any function $k(y)$ and then take

$$\begin{aligned}
f_{x,y} &:= k(y) + \varepsilon_1(x,y) \\
g_{y,z} &:= -k(y) + \varepsilon_2(y,z)
\end{aligned}$$

Then we have

$$\begin{aligned}
\min_{x,y,z}[f_{x,y} + g_{y,z}] &= \min_{x,y,z}[\varepsilon_1(x,y) + \varepsilon_2(y,z)] \\
\min_{x,y} f_{x,y} + \min_{y,z} g_{y,z} &= \min_y k(y) + \min_y -k(y) + \min_{x,y} \varepsilon_1(x,y) + \min_{y,z} \varepsilon_2(y,z) \\
&= \min_y k(y) - \max_y k(y) + \min_{x,y} \varepsilon_1(x,y) + \min_{y,z} \varepsilon_2(y,z)
\end{aligned}$$

$$\min_{x,y}[f_{x,y} - \mathbb{E}_x f_{x,y}] + \min_{y,z}[g_{y,z} + \mathbb{E}_x f_{x,y}] = \min_{x,y} \varepsilon_1(x,y) + \min_{y,z}[\varepsilon_2(y,z) + \mathbb{E}_x \varepsilon_1(x,y)]$$

If ε_1 and ε_2 are small compared to $\min_y k(y) - \max_y k(y)$, then using $\mathbb{E}_x f_{x,y}$ gives a much better bound.

Note, though, that this could be a worse bound if the assumption of small variation does not hold.

Note also that this trick is not restricted to adding and subtracting $\mathbb{E}_x f_{x,y}$. If f is a matrix indexed by x and y, we might also try taking SVD and using the first principle component instead. Compactly, the more general theorem is:

A basic application of the triangle inequality gives the following result:

Theorem 19 (Summarize+Diff)**.** *For any h_y which can be computed in time $\mathcal{O}(n_h)$,*

$$\min_{x,y,z} f_{x,y} + g_{y,z} \geq \min_{y,z}[h_y + g_{y,z}] + \min_{x,y}[f_{x,y} - h_y]$$
$$\max_{x,y,z} f_{x,y} + g_{y,z} \leq \max_{y,z}[h_y + g_{y,z}] + \max_{x,y}[f_{x,y} - h_y]$$

and the RHSs can be computed in time $\mathcal{O}(n_x n_y + n_y n_z + n_h)$ for n_x, n_y, and n_z the number of possible values of x, y, and z, respectively.

We see that if the variation of f in the x-axis is indeed much smaller than the variation in the y-axis, then letting

$$\begin{aligned} f_{x,y} &= h_y + \varepsilon_{x,y} \\ &\quad |\min_{x,y,z} f_{x,y} + g_{y,z} - \min_{y,z}[h_y + g_{y,z}] - \min_{x,y}[f_{x,y} - h_y]| \\ &\leq |\min_{x,y,z}[f_{x,y} + g_{y,z}] - \min_{y,z}[h_y + g_{y,z}]| + |\min_{x,y}[\varepsilon_{x,y}]| \\ &\leq 2 \max_{x,y} |\varepsilon_{x,y}| \end{aligned}$$

so indeed this bound doesn't worsen too much and we are able to compute it in quadratic rather than cubic time.

G Details of SVD of QK proof

As discussed in Subsubsection 4.2.1, to further reduce the computation cost of proof, we need to avoid computing the residual stream, EVOU, and EPQKE matrices fully. Using mechanistic insight or otherwise, we observe that these matrices (apart from EVOU) can be well-approximated by rank 1 matrices. This will remove the dominant computation cost of $\mathcal{O}({d_{\text{vocab}}}^2 \cdot d_{\text{model}})$.

G.1 Comments on relationship between mechanistic insight and proof size

Up to this point, we haven't really said much about what the model is doing. All the mechanistic insight has been of the form "the model varies more along this axis than this other axis" or "the input data is distributed such that handling these inputs is more important than handling these other inputs" or, at best, "the model computes the answer by attending to the maximum token of the sequence; everything else is noise".

Here, finally, our proof-size constraints are tight enough that we will see something that we could plausibly call "how the model pays attention to the maximum token more than anything else", i.e., (if we squint a bit) "the model pays more attention to larger tokens in general.

G.2 The max row diff trick

As stated above, we are breaking matrices into their rank 1 approximation and some noise. To bound the noise, i.e. to bound expressions of the form $\prod_i (A_i + E_i) - \prod_i A_i$, where E_i denote the matrix errors, we can use the following trick:

Lemma 20 (Max Row Diff (vector-matrix version))**.** *For a row vector $\mathbf{a}$ and a matrix B,*

$$\max_{i,j} ((\mathbf{a}B)_i - (\mathbf{a}B)_j) \leq \sum_k |a_k| \max_{i,j} (B_{k,i} - B_{k,j})$$

Moreover, for a collection of n row vectors A_r, if the shape of B is $m \times p$, the right hand side can be computed for all r in time $\mathcal{O}(nm + mp)$.

Proof.

$$\begin{aligned}
&\max_{i,j}(\mathbf{a}B)_i - (\mathbf{a}B)_j \\
&= \max_{i,j} \sum_k a_k \left(B_{k,i} - B_{k,j}\right) \\
&\leq \sum_k \max_{i,j} a_k \left(B_{k,i} - B_{k,j}\right) \\
&= \sum_k a_k \begin{cases} \max_{i,j} \left(B_{k,i} - B_{k,j}\right) & \text{if } a_k \geq 0 \\ \min_{i,j} \left(B_{k,i} - B_{k,j}\right) & \text{if } a_k < 0 \end{cases} \\
&= \sum_k a_k \begin{cases} \max_{i,j} \left(B_{k,i} - B_{k,j}\right) & \text{if } a_k \geq 0 \\ -\max_{i,j} \left(B_{k,i} - B_{k,j}\right) & \text{if } a_k < 0 \end{cases} \\
&= \sum_k |a_k| \max_{i,j} \left(B_{k,i} - B_{k,j}\right)
\end{aligned}$$

The asymptotic complexity of computing the result follows from caching the computation of $\max_{i,j} \left(B_{k,i} - B_{k,j}\right)$ for each k independently of r, as the computation does not depend on A_r. □

Theorem 21 (Max Row Diff)**.** *For matrices A and B,*

$$\max_{r,i,j} \left((AB)_{r,i} - (AB)_{r,j}\right) \leq \max_r \sum_k |A_{r,k}| \max_{i,j} \left(B_{k,i} - B_{k,j}\right)$$

Proof. By taking the max of Lemma 20 over rows r of A. □

Lemma 20 can also be applied recursively for a product of more than two matrices.

Lemma 22 (Max Row Diff (vector-matrix recursive version))**.** *For a row vector $\mathbf{a}$ and a sequence of n matrices B_p of shapes $r_p \times c_p$,*

$$\max_{i,j} \left(\left(\mathbf{a} \prod_p B_p \right)_i - \left(\mathbf{a} \prod_p B_p \right)_j \right) \leq \sum_{k_0} |a_{k_0}| \cdots \sum_{k_n} \left| (B_n)_{k_{n-1},k_n} \right| \max_{i,j} \left(B_{k_n,i} - B_{k_n,j}\right)$$

Moreover, for a collection of q row vectors A_α, the right hand side can be computed for all α in time $\mathcal{O}(qr_0 + \sum_p r_p c_p)$.

Proof. We proceed by induction on n.

For $n = 1$, the statement is identical to Lemma 20.

Suppose the theorem holds for all positive $n = s$; we show the theorem holds for $n = s + 1$. We reassociate the matrix multiplication as

$$\begin{aligned}
&\max_{i,j} \left(\left(\mathbf{a} \prod_{p=1}^{s+1} B_p \right)_i - \left(\mathbf{a} \prod_{p=1}^{s+1} B_p \right)_j \right) \\
&= \max_{i,j} \left((\mathbf{a}B_1) \left(\left(\prod_{p=2}^{s+1} B_p \right)_i - \left(\prod_{p=2}^{s+1} B_p \right)_j \right) \right)
\end{aligned}$$

Using the induction hypothesis gives

$$\leq \sum_{k_1} \left| \sum_{k_0} a_{k_0} (B_1)_{k_0,k_1} \right| \sum_{k_2} |(B_2)_{k_1,k_2}| \cdots \sum_{k_{s+1}} \left| (B_{s+1})_{k_s,k_{s+1}} \right| \max_{i,j} \left(B_{k_{s+1},i} - B_{k_{s+1},j}\right)$$

The triangle inequality gives

$$\leq \sum_{k_1} \sum_{k_0} |a_{k_0}(B_1)_{k_0,k_1}| \sum_{k_2} |(B_2)_{k_1,k_2}| \cdots \sum_{k_{s+1}} \left|(B_{s+1})_{k_s,k_{s+1}}\right| \max_{i,j} \left(B_{k_{s+1},i} - B_{k_{s+1},j}\right)$$

and algebra gives

$$= \sum_{k_0} |a_{k_0}| \sum_{k_1} |(B_1)_{k_0,k_1}| \sum_{k_2} |(B_2)_{k_1,k_2}| \cdots \sum_{k_{s+1}} \left|(B_{s+1})_{k_s,k_{s+1}}\right| \max_{i,j} \left(B_{k_{s+1},i} - B_{k_{s+1},j}\right)$$

The asymptotic complexity of computing the right hand side also follows straightforwardly by induction. □

Theorem 23 (Max Row Diff (recursive))**.** *For a sequence of $n+1$ matrices $A_0, \ldots, A_n$,*

$$\max_{r,i,j} \left(\left(\prod_p A_p \right)_{r,i} - \left(\prod_p A_p \right)_{r,j} \right) \leq \max_r \sum_{k_0} |(A_0)_{r,k_0}| \cdots \sum_{k_n} \left|(A_{n-1})_{k_{n-1},k_n}\right| \max_{i,j} \left((A_n)_{k_n,i} - (A_n)_{k_n,j}\right)$$

Proof. By taking the max of Lemma 22 over rows r of A_0. □

Note that Theorem 21 is compatible with the mean+diff trick of Appendix F.2.

Theorem 24 (Combined Mean+Diff and Max Row Diff)**.** *For matrices A and B, and any column-wise summary vector H_k of A (for example we may take $H_k := \mathbb{E}_r A_{r,k}$)*

$$\max_{r,i,j} \left((AB)_{r,i} - (AB)_{r,j}\right) \leq \left(\max_{i,j} \sum_k H_k \left(B_{k,i} - B_{k,j}\right) \right) + \max_r \sum_k |A_{r,k} - H_k| \max_{i,j} \left(B_{k,i} - B_{k,j}\right)$$

Proof.

$$\begin{aligned}
&\max_{r,i,j} \left((AB)_{r,i} - (AB)_{r,j}\right) \\
&= \max_{r,i,j} \sum_k A_{r,k} \left(B_{k,i} - B_{k,j}\right) \\
&= \max_{r,i,j} \sum_k \left(H_k + (A_{r,k} - H_k)\right) \left(B_{k,i} - B_{k,j}\right) \\
&= \max_{i,j} \left(\sum_k H_k \left(B_{k,i} - B_{k,j}\right) + \max_r \sum_k \left(A_{r,k} - H_k\right) \left(B_{k,i} - B_{k,j}\right) \right) \\
&\leq \left(\max_{i,j} \sum_k H_k \left(B_{k,i} - B_{k,j}\right) \right) + \max_r \sum_k \max_{i,j} \left(A_{r,k} - H_k\right) \left(B_{k,i} - B_{k,j}\right) \\
&\leq \left(\max_{i,j} \sum_k H_k \left(B_{k,i} - B_{k,j}\right) \right) + \max_r \sum_k |A_{r,k} - H_k| \max_{i,j} \left(B_{k,i} - B_{k,j}\right)
\end{aligned}$$

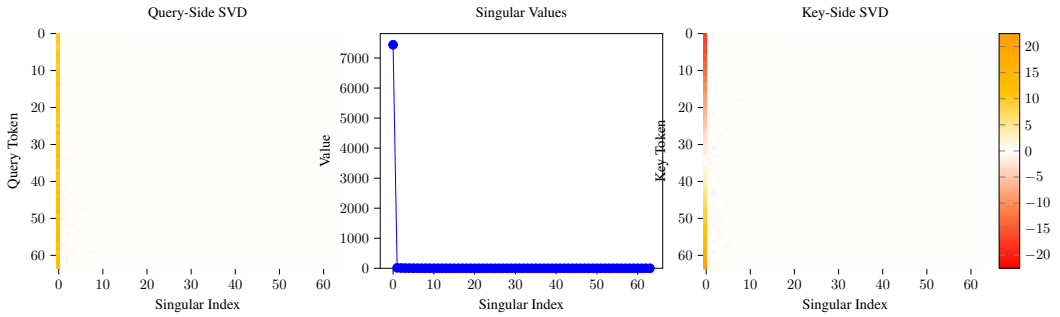
□

G.3 Exploring rank 1 approximation via SVD

Let us first look at

$$\text{EQKE}[q,k] := (E[q] + P[-1])QK(E[k] + \mathbb{E}_{\text{dim}=0} P[:-1]^T.$$

From Figure 7a, we see that there is not much variation along long query token direction. We can confirm this by performing a singular value decomposition (SVD) on EQKE, and plotting the resulting matrices, scaled by singular value:



The first singular value is just over 7440, while the second singular value is just under 15. There's really not much going on here beyond the first singular component.[7]

Call the first singular component of EQKE the "query direction" d_q and the "size direction" d_k on the query-side and key-side, respectively.

There are two ways that we can decompose EQKE into a low-rank component that we can compute exactly, and a full-rank error term that we approximate bounds for.

G.4 The simple SVD decomposition of QK

In time $\mathcal{O}(d_{\text{vocab}}{d_{\text{model}}}^2)$ we can perform SVD on each of the four component matrices $E+P[-1]$, Q, K, $E+\mathbb{E}_p P[p]$ and perform low-rank SVD on the matrix product $(E+P[-1])QK^T(E+\mathbb{E}_p P[p])^T$.

G.5 The complicated SVD decomposition of QK

We can decompose E into a part parallel to d_q and a part orthogonal to d_q, say $E+P[-1] = E_q+E_q^\perp$, and similarly $E+\mathbb{E}_p P[p] = E_k + E_k^\perp$. Note that E_q and E_k are both rank 1, and hence can be multiplied with other matrices of shape $d_{\text{model}}\times a$ in time $\mathcal{O}(d_{\text{model}}a)$ rather than time $\mathcal{O}(d_{\text{vocab}}d_{\text{model}}a)$. While the "most mechanistic" interpretation would proceed with the analysis in terms of E_q and E_k, perhaps decomposing them further, we can get more bang for our buck by extracting out all the low-rank structure available E, Q, and K, so as to make our error bounds as tight as possible.

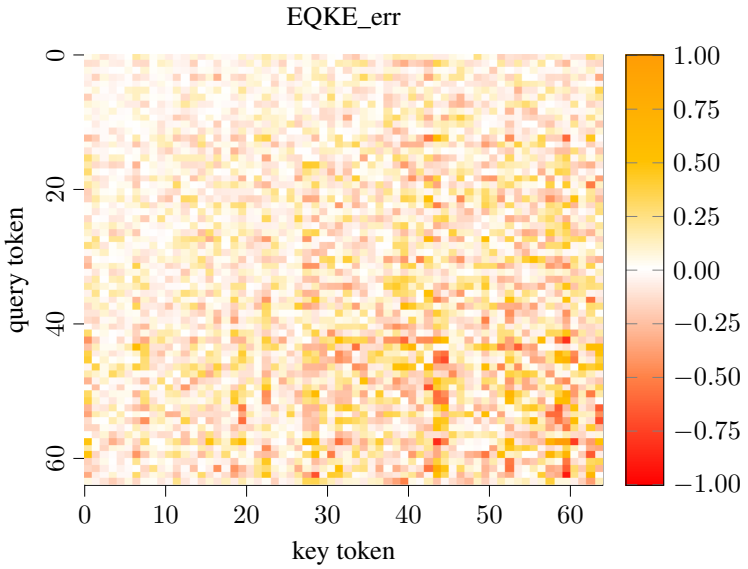
To this end, we perform SVD on $E_q^\perp$, $E_k^\perp$, Q, and K and peel off the first singular components so as to get the decomposition

$$\begin{aligned} E+P[-1] &= E_q + E_{q,2} + E_{q,2}^\perp \\ E+\mathbb{E}_p P[p] &= E_k + E_{k,2} + E_{k,2}^\perp \\ Q &= Q_0 + Q^\perp \\ K &= K_0 + K^\perp \end{aligned}$$

Then EQKE, a product of these four matrices, can be expressed as a sum of $2^2 3^2 - 1 = 35$ rank one products and one high-rank error term. We can compute the sum of the rank one products in time $\mathcal{O}({d_{\text{vocab}}}^2)$ and express EQKE as, say, $\text{EQKE}_2 + E_{q,2}^\perp Q^\perp (E_{k,2}^\perp K^\perp)^T$. Call the second term EQKE_err. We must now bound for each q and m the quantity $\max_{i\le m-G} \text{EQKE\_err}[q,i] - \text{EQKE\_err}[q,m]$.

How big is this?

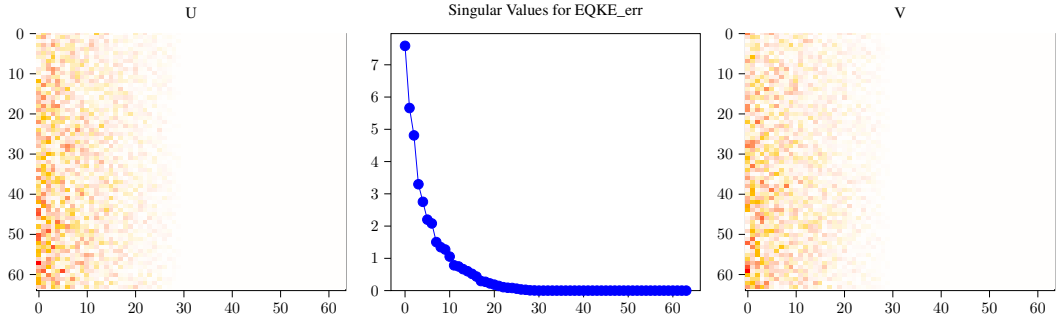
[7] We might be tempted to keep analyzing the SVD, and notice that the query direction is mostly uniform, while the key direction is monotonic (nearly linear, even). But the proof complexity doesn't demand this level of analysis, yet, and so we can't expect that any automated compact proof discovery system will give it to us.



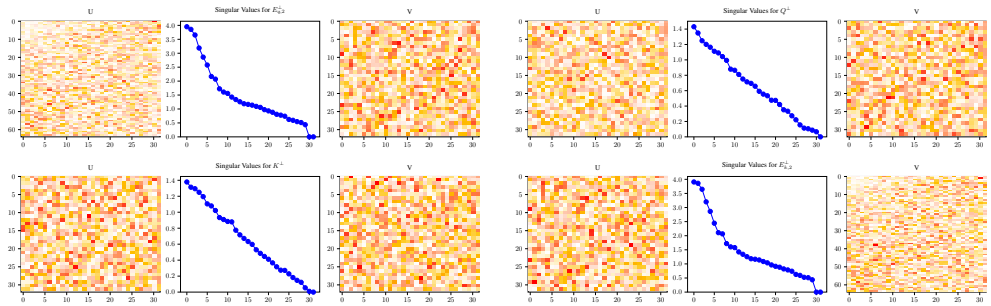
Even if we relax to $\max_{i,j} \text{EQKE\_err}[q,i] - \text{EQKE\_err}[q,j]$, the maximum such value across all rows is under 1.85. And the rows don't have any particular structure to them; the maximum absolute element of the entire matrix is just barely over 1, so doubling that doesn't give too bad an estimate.

But we somehow need to compute this value without multiplying out the four matrices.

One option is to try to use singular value decomposition again. Since $\sigma_1(M) = \sup_x \|Mx\| / \|x\|$, considering vectors with one 1, one -1, and zero elsewhere, the maximum difference between elements in a row upper bounded by $\sqrt{2}\sigma_1(M)$. The largest singular value of EQKE_err is just under 7.6, giving a row-diff bound of about 10.7, which is large but not unusably so.



If we perform SVD before multiplying out the matrices, however, their first singular values are about 4, 1.4, 1.4, and 4, giving a product of about 30, which when multiplied by $\sqrt{2}$ is about 43. This works because $\sigma_1(AB) \leq \sigma_1(A)\sigma_1(B)$, but note that we can do factored SVD without needing to use this technique. This bound is still usable, but pretty big.



Note that using anything close to this method to drop below $d_{\text{vocab}}{d_{\text{model}}}^2$ seems infeasible; the best bound we know on the largest singular value that can be verified even in the worst-case in strictly less time than it takes to compute the full SVD is the Frobenius norm, which is defined as $\text{tr}(MM^T)$, can be computed in $d_{\text{model}}d_{\text{vocab}}$ time, and is equal to the square root of the sum of the squares of the singular values. While the Frobenius norm of EQKE_err is only about 12 (giving a bound of about 17 on the row diff), the Frobenius norms of the four multiplicand matrices are a bit over 10, 4, 4, and 10, giving a product of 1932 and a bound of 2732(!). This is unusably large.

However, we can get a much better bound on the max row diff of EQKE_err without having to multiply out all four matrices. We can use an approach vaguely similar to the mean+diff trick, as follows.

If we want to compute the max row diff of a product of matrices AB, we can compute by Theorem 21

$$\max_{r,i,j}\left((AB)_{r,i}-(AB)_{r,j}\right)\leq\max_{r}\sum_{k}|A_{r,k}|\max_{i,j}\left(B_{k,i}-B_{k,j}\right) \tag{3}$$

or by combining this approximation with Theorem 18 via Theorem 24 we may compute

$$\begin{aligned}&x\max_{r,i,j}\left((AB)_{r,i}-(AB)_{r,j}\right)\\&\leq\left(\max_{i,j}\sum_{k}\mathbb{E}_{r}A_{r,k}\left(B_{k,i}-B_{k,j}\right)\right)+\max_{r}\sum_{k}|A_{r,k}-\mathbb{E}_{r}A_{r,k}|\max_{i,j}\left(B_{k,i}-B_{k,j}\right)\end{aligned}$$

taking whichever bound is better.

The first gives us a bound of 7.94 on the maximum row diff, which is better than we can get by doing SVD on the product of the matrices! We can get an even better bound by peeling off the first two singular values of all four matrices before multiplying them; this gives us a bound of 5.67. Combining it with the avg+diff trick wouldn't give us much (8.05 and 5.66 respectively), as we've effectively already done this by peeling off the leading singular contributions; the mean of EQKE_err over dimension zero has norm 0.025.

Although this noise bound is no longer the leading asymptotic bottleneck, we can peek ahead to what we get if we want to be linear in parameter count. In this case, we can apply the recursive version of Equation 3 via Theorem 23, giving a bound of 97.06 on the maximum row diff.

The mechanistic understanding we get here is roughly "for any given basis vector of the residual stream, the difference between the overlap of any two input tokens with this direction is small once we factor out the first two singular components", and this is sufficient to drive a low error term overall if we factor out the leading singular components in other places. We don't mechanistically understand how to combine the EQK (without multiplying them out) in a way that allows getting a good bound, though, which corresponds to our inability to drop below $d_{\text{vocab}}{d_{\text{model}}}^2$ here.

If we use this trick on QK only, and use the mean+diff trick on final attention handling (without which we lose about 3 %), we can achieve a bound of 0.7810.

If we use this trick on the skip connection (EU) only, we can achieve a bound of 0.6805.

Using this trick on both EU and QK drops us down only to 0.6379.

If we use this trick on EU and use the recursive version of this trick on QK, we get a bound of 0.3064.

Unfortunately, it's not clear how this trick would apply to EVOU. A fancier convex hull checking algorithm seems required, and an analysis thereof is in progress.

H Justification of pessimization choices

In Subsection 4.2 we make a number of choices about which axes of variation are more or less important to track at various points in the bound computation.

Here we do some more traditional mechanistic interpretability analysis to justify that the choices that we made could be expected to lead to reasonably good bounds.

EVOU := $EVOU$ (weighted by sequence count)
$(\text{EVOU} - \text{EVOU.diag}()[:, \text{None}]).\max(\dim = -1)$ (excluding diagonal)
$\bar{x} \pm \sigma$: -9.3 ± 3.8; range: 0 ± 13

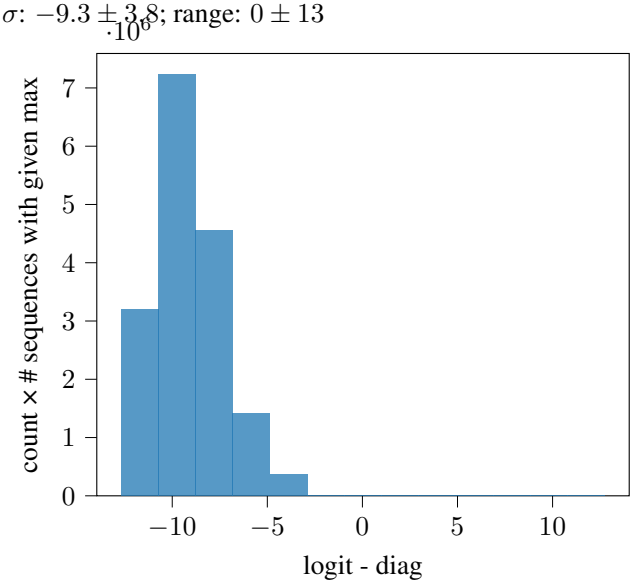


Figure 11
The attention computation weighted by the number of sequences with the particular max.

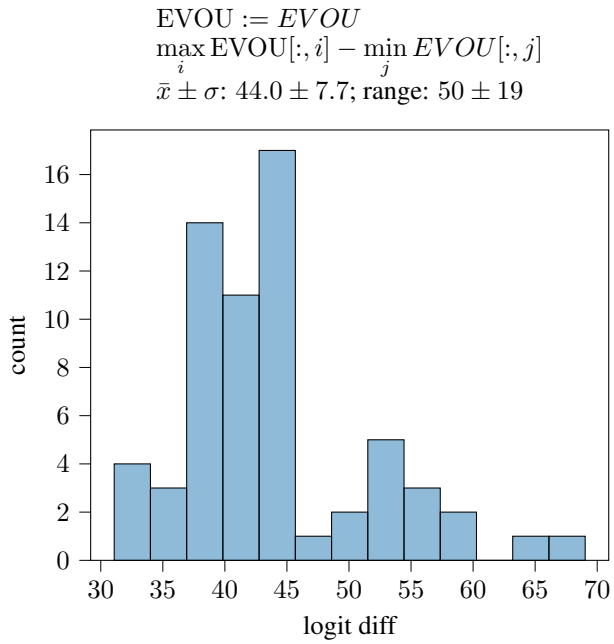
H.1 Justifying the gap

We take advantage of the fact that attention is mostly monotonically increasing in input integers and that for most sequences, the attentional contribution of the particular query token matters much more than the particular non-max token in the sequence.

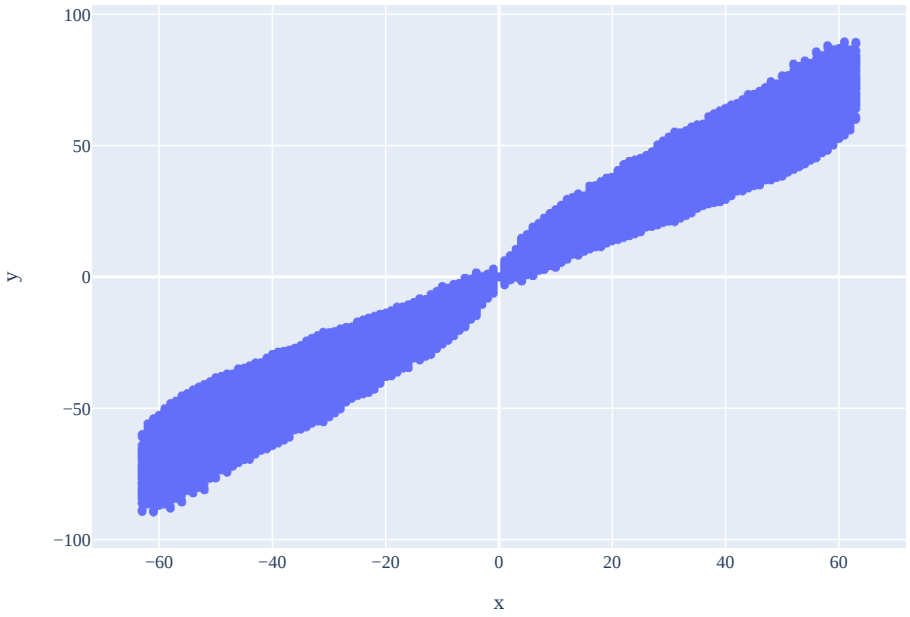
We justify this as follows.

We can look at the typical diff, when attending to the max token, between the largest non-max logit and the max logit. As shown in Figure 11, the largest difference between an off-diagonal entry of EVOU and the diagonal of that row is typically at most -5.[8] The typical worst contribution to the wrong logit from a non-max token (this is typical over non-max tokens, worst over choice of output token-logit index) is around 44:

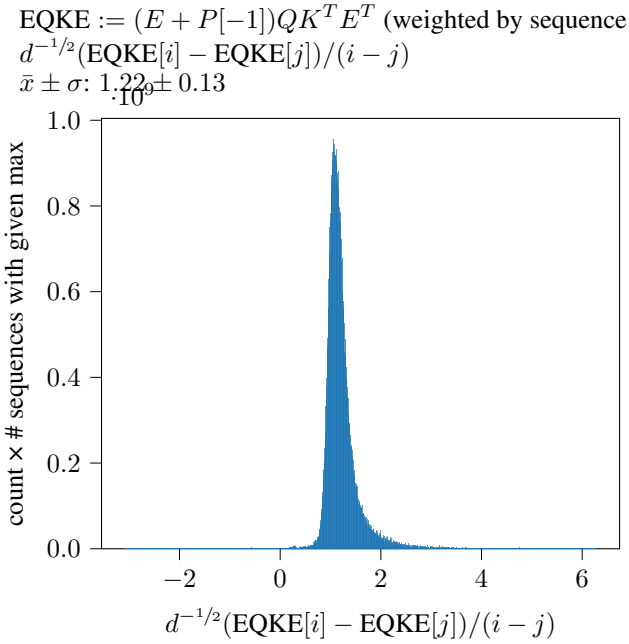
[8] "Typically" here means about $98\,\%$ of the time.



The difference in attention between tokens is approximately linear in the gap between the tokens



The slope of the line, that is, the difference in pre-softmax attention scores divided by the gap between the key token and the max token, is approximately 1.2:

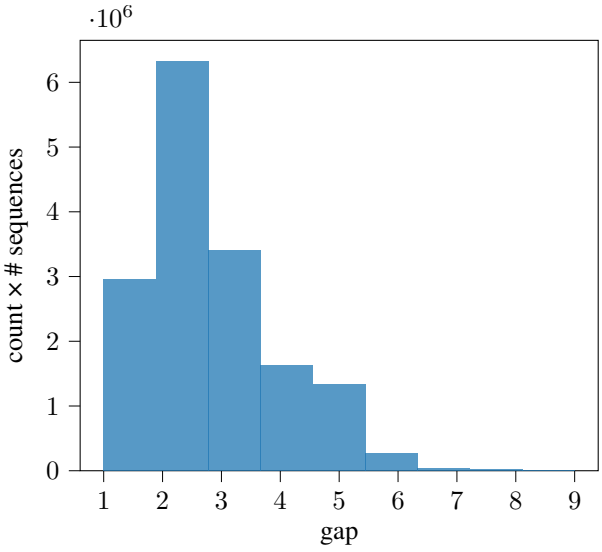


Exponentiating, the post-softmax attention paid to the max is typically about $3\times$ larger than to the token one below the max; here the logit difference between the max and non-max token is significant, typically being around 13 $(44/3)$ for the worst output logit. But by the time the gap is 3, this difference has dropped to about 1.1, and by the time the gap is 4 it is around 0.3.

So for sequences where the largest non-max and the max are close together, the particular structure of the non-max EVOU matters a lot; but when the max is separated from the largest non-max by a modest gap, the structure of the non-max EVOU does not matter so much.

The upshot is that to handle most sequences, we need only ask an oracle for the minimum gap $g > 0$ between the max token t_{max} and largest non-max tokens $t' \neq t_{\text{max}}$, such that the model outputs the correct answer for all sequences where the non-max, non-query tokens have value at most $t_{\text{max}} - g$

While computing this gap may be expensive (and indeed the naïve computation of the oracle takes longer than the brute force proof—though it should be very easy to optimize), we don't have to pay the cost of computing the gap in the size of the proof, only the cost of storing the gap table ($\mathcal{O}({d_{\text{vocab}}}^2 n_{\text{ctx}})$) and of verifying the gap. Empirically, gaps are typically 1–6:



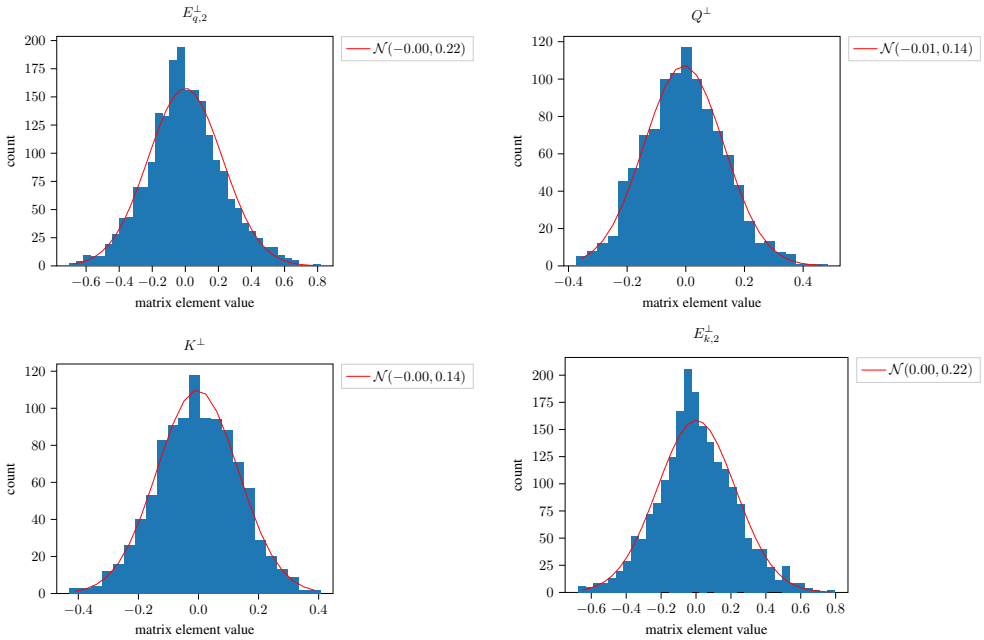


Figure 12: The distribution of entries of the four residual matrices (after removing two principle components from $E + P$ and one principle component from Q and K). Distributions look pretty close to normal.

If we rely on the gaps, this results in leaving behind about $7.9\,\%$ of sequences.

We can compute a non-max $\rightarrow$ max $\rightarrow$ largest logit contribution of non-max to maxes $\leq$ the max ($\mathcal{O}({d_{\text{vocab}}}^2)$) (and whether it's to a token within or outside of window); compute a table of pre-softmax attention diffs between tokens i and $i+1$ ($\mathcal{O}({d_{\text{vocab}}}^2)$); then sort the queries by overlap with the query direction; compute for each number of queries handled (where we assume we handle all queries with greater overlap than the current one) and for each max how many of the queries fall strictly below the max (and whether the max being the query makes the cut); compute a table of # queries handled $\rightarrow$ $i \rightarrow$ min more attn paid to $i+1$ than to i ($\mathcal{O}({d_{\text{vocab}}}^2)$); compute max $\rightarrow$ non-max $\rightarrow$ upper bound on amount more attention paid to non-max than to max by Oracle-permitted queries (indexed only on max) ($\mathcal{O}({d_{\text{vocab}}}^2)$); compute per num queries permitted then for each max, non-max, num copies nonmax, compute if the non-max contributes little enough to the bad logit that even with the worst skip connection things are fine.

H.2 Stopping after 1–2 principle components of QK

Did we miss out on any structure in the noise of EQKE? The distribution of entries of the four matrices looks pretty close to normal as seen in Figure 12.

If we sample elements randomly, we get (sample size 100) that the maximum row diff of the product of the matrices is approximately 1.31 ± 0.13 (sampling without replacement from the empirical distribution) or 1.31 ± 0.14 (sampling from the normal distribution). So in fact our max row diff is unusually high (by about 4σ).[9]

I Convex relaxation

We construct convex relaxation to perform pessimal ablations in our proofs. The following is a formal description of the argument.

[9]This shows up in the bias towards having larger values (both positive and negative) in the lower-right corner of the plot, indicating that noise is larger for larger query and key values. We hypothesize that this is due to the distribution of data: larger values are more likely to have more space between the maximum and next-most-maximum token, so a bit of noise matters less for larger maxes than for smaller ones.

For a set of inputs X_i, we define a set of "relaxed" inputs X_i^{relaxed} with an injection $T_i : X_i \hookrightarrow X_i^{\text{relaxed}}$ mapping input, and a function $h_i : X_i^{\text{relaxed}} \to \mathbb{R}$ such that for all $\mathbf{x} \in X_i$, we have $f(\mathbf{x}, \mathcal{M}(\mathbf{x})) \geq h_i(T_i(\mathbf{x}))$. We proceed by finding a small subset of "boundary" examples $B_i \subset X_i^{\text{relaxed}}$, proving that if $h_i(\mathbf{x}^{\text{relaxed}}) \geq b_i$ for all $\mathbf{x}^{\text{relaxed}} \in B_i$ then $h_i(\mathbf{x}^{\text{relaxed}}) \geq b_i$ for all $\mathbf{x}^{\text{relaxed}} \in X_i^{\text{relaxed}}$. C then validates that that $h_i(\mathbf{x}^{\text{relaxed}}) \geq b_i$ for some b_i for all $\mathbf{x}^{\text{relaxed}} \in X_i^{\text{relaxed}}$. This allows us to conclude that $f(\mathbf{x}, \mathcal{M}(\mathbf{x})) \geq b_i$ for all $\mathbf{x} \in X_i$.

J IEEE 754 vs. $\mathbb{R}$

In Section 2 we defined C and Q and glossed over whether we were reasoning over reals or floats. Here we clarify this point that we've so far been sweeping under the rug.

Let $\mathbb{F}$ denote the set of the relevant flavor of IEEE 754 Floating Point numbers (generally 32-bit for our concrete models, but everything would hold just as well for 64-bit). Let $\mathbb{F}^*$ denote $\mathbb{F}$ restricted to finite numbers (that is, without NaNs and without $\pm\infty$).

Parameterize C, $\mathcal{M}$, and $\mathcal{D}$ over the real field they operate on, so that, e.g., C_F : model weights $\to F$.

Then we have Q establishing that for any model $\mathcal{M}'$, $C_{\mathbb{R}}(\mathcal{M}'_{\mathbb{R}}) \leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbb{R}}} f_{\mathbb{R}}(\mathbf{x}, \mathcal{M}'_{\mathbb{R}}(\mathbf{x}))$, and we have a trace demonstrating that $C_{\mathbb{F}}(\mathcal{M}_{\mathbb{F}}) = b$.

Let $i : \mathbb{F}^* \to \mathbb{R}$ be any injection such that maps each floating point number to some real number that it is "closest to". Supposing that $b \in \mathbb{F}^*$ and thus $b \in \mathbb{R}$, we need two additional components of the proof. We need to find $\varepsilon, \varepsilon' \in \mathbb{R}^+$ prove that

$$|C_{\mathbb{R}}(\mathcal{M}_{\mathbb{R}}) - i\left(C_{\mathbb{F}}(\mathcal{M}_{\mathbb{F}})\right)| < \varepsilon$$

and

$$|(\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbb{R}}} f_{\mathbb{R}}(\mathbf{x}, \mathcal{M}_{\mathbb{R}}(\mathbf{x}))) - i\left(\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbb{F}}} f_{\mathbb{F}}(\mathbf{x}, \mathcal{M}_{\mathbb{F}}(\mathbf{x}))\right)| < \varepsilon'$$

Then we can chain these proofs to prove that

$$i\left(\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathbb{F}}} f_{\mathbb{F}}(\mathbf{x}, \mathcal{M}_{\mathbb{F}}(\mathbf{x}))\right) \geq b - \varepsilon - \varepsilon'$$

Such ε-ball robustness proofs should be well within the scope of existing approaches to formal methods on neural nets, see, e.g., [33, 3, 4, 23, 1, 46]. We leave actually dealing with the gap between floating point numbers and real numbers to future work.

K Infinite distributions

When we described the basic brute-force proof strategy in Appendix D, we talked about running the model on the entirety of $\mathcal{D}$. This is straightforward when X is finite. Perhaps surprisingly, we can do this even if X is infinite as long as the PDF $X \to \mathbb{R}$ of $\mathcal{D}$ is computable and the natural computational topology of X is compact [12, 11, 10], because integration of computable functions on computable reals is computable [37].

L Computing effective dimensionality reduction

We claim in Figure 5 that we can use unexplained dimensionality as a metric for understanding. Here we describe how we compute the unexplained dimensionality of a proof strategy.

As in Figure 1, for any given proof, we can separate our treatment of transformer components into "black-box" (e.g., matrix multiplication) and "white-box" components (e.g., specifying that the QK circuit is approximately rank one; pessimizing over non-max tokens). Considering the performance score as a large white-box component which may reference black-boxes internally, we define the unexplained dimensionality of a single black-box computation as the log-cardinality of it function space (so, e.g, $2 \cdot 64$ for a function $\underline{64} \to \mathbb{R}^2$, whose cardinality is $(\mathbb{R}^2)^{\underline{64}}$, where $\underline{64}$ denotes the finite set on 64 elements). The unexplained dimensionality of the entire proof is the sum of the unexplained dimensions of all black-box components.

Intuitively speaking, unexplained dimensionality tries to capture the degrees of freedom that we have to check via brute enumeration over black-box computations. Proofs with less unexplained dimensionality contain more mechanistic understanding, and vice versa.

M Computing approximate FLOPs

In Figure 3 and Table 1, we display approximate floating point operations. We instrument our code to execute on phantom tensors that track their shape and accumulate an approximate count of floating point operations. We compute matrix additions and multiplications in the obvious way. We take the instruction count of SVD to be the cost of verifying that the output of SVD is a valid decomposition: that we have a pair of orthonormal bases which when multiplied out give the original basis.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We present the key challenges in the field of formal verification for neural networks, and describe our proposed solution. Then we summarize our experimental setup and results.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 7.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide intuitions for our proof constructions in the main body of the paper. In the supplemental material, we lay out in full detail theorem statements and proofs. Along with this, we provide algorithms and plots that are useful for understanding how the proofs were constructed.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We directly link to a codebase with the specific implementation of our case study.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In supplementary materials, we provide the full details of our model training set up. In the main body of the paper, we describe our experimental setting and provide reasoning for why we chose this experimental setup as a very simple case study of our theoretical work.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We run our experiment on models trained with 151 different hyperparameters. For most reported computations, we provide statistical significance information. We do not need to perform explicit t-tests or such since it is not relevant to our setup.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: See Appendix A.

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: We confirm that we have read the Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This paper seeks to advance the fields of mechanistic interpretability and formal verification of machine learning systems. While there are many indirect societal consequences of our work through the impacts on these fields, we feel that none are sufficiently consequential as to be highlighted here.

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper does not involve any such assets.

Guidelines:

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We list all important Python packages used in the paper in Appendix A.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce no new assets except for the codebase needed to reproduce our experiments, which does contain appropriate documentation.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.