
Limitations of Normalization in Attention Mechanism

Timur Mudarisov

University of Luxembourg
Luxembourg
timur.mudarisov@uni.lu

Mikhail Burtsev

London Institute for
Mathematical Sciences
mb@lims.ac.uk

Tatiana Petrova

University of Luxembourg
Luxembourg
tatiana.petrova@uni.lu

Radu State

University of Luxembourg
Luxembourg
radu.state@uni.lu

Abstract

This paper investigates the limitations of the normalization in attention mechanisms. We begin with a theoretical framework that enables the identification of the model’s selective ability and the geometric separation involved in token selection. Our analysis includes explicit bounds on distances and separation criteria for token vectors under softmax scaling. Through experiments with pre-trained GPT-2 model, we empirically validate our theoretical results and analyze key behaviors of the attention mechanism. Notably, we demonstrate that as the number of selected tokens increases, the model’s ability to distinguish informative tokens declines, often converging toward a uniform selection pattern. We also show that gradient sensitivity under softmax normalization presents challenges during training, especially at low temperature settings. These findings advance current understanding of softmax-based attention mechanism and motivate the need for more robust normalization and selection strategies in future attention architectures.

1 Introduction

The attention mechanism [1, 7, 17, 8] has become a fundamental component of modern deep learning. Since its popularisation in the Transformer [18], attention has powered state-of-the-art systems in machine translation, text generation [3], and multimodal reasoning [11]. Yet the same softmax rule that enables differentiable “focus” also introduces a chronic failure mode: as the context length L grows, attention weights collapse toward $1/L$, a phenomenon we call *vanishing attention*. The resulting gradients are too small for effective learning, especially in long-context settings [4, 21]. Architectural work-arounds – sparse windows, locality-sensitive hashing, or compressed memories—reduce compute but do not eliminate the collapse, and may even worsen it [5, 16, 2, 10]. A principled understanding of *why* softmax fails in this regime is still missing.

Our view: attention as a capacity-limited retriever (selector). We revisit normalisation in attention from first principles and show that softmax, and indeed *any* length-independent normaliser, possesses an intrinsic capacity limit. Our contributions are:

1. **Distance bound.** We derive a non-asymptotic upper bound on the representation distance between selected and non-selected tokens (Theorem 1). The bound proves that once the top- N set grows proportionally to L , the distance *must* collapse, formalising the “softmax bottleneck” [20].

2. **Geometric bound.** Under mild spherical assumptions we show that no more than $\approx 80\%$ of the top- N tokens can be simultaneously distinguished in Euclidean space (Theorem 2). This quantifies a hard limit on what a single head can represent.
3. **Gradient bound.** We bound the Jacobian norm of a general normaliser (Lemma 2); specialised to softmax it recovers the classic $1/(4T)$ instability and shows why aggressive temperature scaling trades separability for optimisation difficulty.
4. **Empirical validation.** Experiments on GPT-2 [15] confirm all three predictions: distance collapse, separability saturation, and $1/T$ gradient growth.

Our analysis frames attention as a *selector with finite resolution*: it works well while the active set is a small fraction of the context, then degrades predictably. This viewpoint explains the empirical success of recent alternatives—Sparsemax, Scalable-Softmax, Self-Adjusted Softmax—and suggests concrete design rules we distil in the discussion. More broadly, our theoretical tools provide diagnostics for deciding when a head has reached its intrinsic limit and when architectural or normalisation changes are warranted.

By bridging closed-form theory with large-scale experiments, we provide both a deeper understanding of normalisation in attention and practical guidelines for building robust, long-context Transformer models.

2 Theoretical Analysis

Consider a sequence of token embeddings $\mathbb{X} = \{x_i\}_{i=1}^L$, where each embedding $x_i \in \mathbb{R}^d$ is a d -dimensional vector. We begin by reviewing the classical self-attention mechanism introduced by Vaswani et al. [18]:

$$\mathbf{q}_m = f_q(x_m, m), \quad \mathbf{k}_n = f_k(x_n, n), \quad \mathbf{v}_n = f_v(x_n, n), \quad (1)$$

where $\mathbf{q}_m$, $\mathbf{k}_n$, and $\mathbf{v}_n$ denote the query, key, and value vectors, respectively. The attention weights are computed using these query and key vectors as follows:

$$a_{m,n} = \exp(\mathbf{q}_m^\top \mathbf{k}_n / T) / \sum_{j=1}^L \exp(\mathbf{q}_m^\top \mathbf{k}_j / T), \quad (2)$$

where the parameter T is known as the temperature, typically set to $T = \sqrt{d}$ (as recommended in [18]). To extend the scope of our analysis beyond the standard softmax normalization, we introduce a more general normalization framework:

$$a_{m,n} = F(\mathbf{q}_m^\top \mathbf{k}_n, \theta) / \sum_{j=1}^L F(\mathbf{q}_m^\top \mathbf{k}_j, \theta), \quad (3)$$

where $F : \mathbb{R}^{1+c} \rightarrow \mathbb{R}$ is a smooth positive function parameterized by θ , which can include parameters such as temperature or the number of tokens. For convenience, we denote the inner product $\mathbf{q}_m^\top \mathbf{k}_n$ as $l_{m,n}$, referring to it as the *logit* associated with the token pair (m, n) .

In this work, we provide a detailed theoretical examination of the general normalization framework introduced in Equation (3). We focus on several critical aspects: (1) general limitations associated with softmax-type normalization; (2) the influence of normalization on token separation; (3) geometric insights into token separation; (4) connections between normalization and training dynamics.

We begin by addressing the general limitations of softmax normalization. As highlighted by [19], one critical limitation is the phenomenon of *vanishing attention weights*. Intuitively, when the number of tokens L increases, the normalization procedure (e.g., softmax) distributes attention weight across many tokens, causing the weights for individual tokens to become extremely small. This issue hampers the model's ability to clearly differentiate between relevant and irrelevant tokens.

Formally, consider a set of logits $\{l_1, \dots, l_L\}$ and their corresponding attention weights $\{\alpha_1, \dots, \alpha_L\}$, computed from the logits using a normalization function. We present the following general results regarding these normalized attention weights (see Appendix A for full proofs):

Lemma 1. Consider the normalization scheme defined by Equation (3) with a smooth function $F(l_i, \theta)$ that does not explicitly depend on the number of tokens L (i.e., $L \notin \theta$). Assume also that the logits are bounded, $l_i \in [-a, a]$. Then, the normalized attention weights satisfy:

$$\frac{C_1}{L} \leq \alpha_i \leq \frac{C_2}{L}, \quad (4)$$

where the constants C_1 and C_2 do not depend on L .

This lemma implies that for any normalization of function independent of token count, the attention weights inevitably become uniformly small (on the order of $1/L$) as the context size grows. As a result, the mechanism loses the ability to effectively highlight specific important tokens when processing long sequences. The proof relies on the boundedness and continuity of the function $F(l_i, \theta)$ on the compact interval $[-a, a]$. Due to these properties, both the numerator and denominator are bounded by constants independent of L . Hence, when normalized by the summation over all L tokens, each weight naturally scales as $1/L$. Full mathematical details are provided in Appendix A.

Corollary 1. In the special case of softmax normalization (Equation (2)), with temperature parameter T , the attention weights satisfy:

$$\frac{1}{L} \exp\left(-\frac{2\Delta}{T}\right) \leq \alpha_i \leq \frac{1}{L} \exp\left(\frac{2\Delta}{T}\right), \quad (5)$$

where $\Delta = \|\mathbf{q}\|_2 \|\mathbf{k}\|_2$.

This corollary highlights how the magnitude of the query and key vectors directly impacts the distribution of attention weights. Specifically, the bounds show that unless the vectors have sufficiently large magnitudes (relative to temperature T), the attention weights remain close to uniform. Thus, increasing vector magnitudes can partially mitigate—but not eliminate—the vanishing attention problem.

These results demonstrate a fundamental limitation of softmax-type normalization: as the sequence length L grows, each attention weight α_i shrinks toward $O(1/L)$. In turn, the model struggles to assign appreciably larger weights to genuinely informative tokens, blurring the distinction between relevant and irrelevant embeddings. This loss of focus impairs the model’s ability to exploit salient information within long input sequences and ultimately hampers effective learning.

2.1 Distance analysis

We next study how normalization influences the *total deviation*—the discrepancy between a learned representation and the underlying conditional distribution. Yang et al. [20] describe the *softmax bottleneck*: the low rank of the logit matrix limits the model’s ability to represent the true conditional distribution, thereby increasing total deviation. This deviation is not a consequence of insufficient model capacity or sub-optimal optimization, but rather of restrictions imposed by the final softmax projection layer.

To quantify the role of normalization in separating informative from non-informative tokens, let $\{\alpha_1, \dots, \alpha_L\}$ and $\mathbb{X} = \{x_1, \dots, x_L\}$, $x_i \in \mathbb{R}^d$ denote the attention weights and their corresponding token embeddings, respectively. We focus on the top- N tokens with the largest weights; their indices are collected in $I_N = \{i_1, \dots, i_N\}$. The *context vector* derived from these tokens is

$$s = \sum_{i \in I_N} \alpha_i x_i. \quad (6)$$

Our quantity of interest is the cumulative distance between this context vector s and all *non-selected* embeddings:

$$\tilde{d} = d(\mathbb{X} \setminus \mathbb{X}_{I_N}, s) = \sum_{i \in I \setminus I_N} \|\alpha_i x_i - s\|_2, \quad (7)$$

where $I = \{1, \dots, L\}$, $\|\cdot\|_2$ is the Euclidean norm, and $\mathbb{X}_{I_N} = \{x_{i_1}, \dots, x_{i_N}\}$ is the set of selected embeddings. Intuitively, a smaller value of $\tilde{d}$ indicates that many low-weight tokens lie close to the high-weight aggregate s , signalling reduced separability and a greater likelihood of attention “dilution.”

Next, we distinguish two settings for the weight vector $\{\alpha_i\}$. In the first, the weights are treated as *fixed*; in the second, we analyse a *random-selection* scenario in which the index set I_N is drawn uniformly at random from all subsets of size N : $I_N \sim U(I, N)$, $I = \{1, \dots, L\}$. In this random case, the attention weights on the non-selected tokens vary with the draw of I_N . We seek the expected cumulative distance

$$E = \mathbb{E}_{I_N}[\tilde{d}], \quad (8)$$

where the expectation is taken over the uniform choice of I_N .

We now adopt a classical metric-learning perspective: the above quantities quantify the Euclidean “separation margin” between selected and non-selected token embeddings. The next theorem formalises these observations by providing explicit upper bounds on the distance $\tilde{d}$ in both the fixed and random top- N settings.

Theorem 1. *For the representation distance defined in (7) the following bounds hold.*

1. **Fixed top- N set.** *If I_N is fixed,*

$$\tilde{d} \leq (1 - \bar{\alpha}_N) d_1 + \max_{j \in I_N} \|x_j\|_2 [\bar{\alpha}_N(L - N) - (1 - \bar{\alpha}_N)],$$

$$\text{where } d_1 = \max_{i \notin I_N, j \in I_N} \|x_i - x_j\|_2 \text{ and } \bar{\alpha}_N = \sum_{i \in I_N} \alpha_i.$$

2. **Uniformly random top- N set.** *When I_N is sampled uniformly from all $\binom{L}{N}$ subsets of size N ,*

$$E = \frac{L - N}{L} \sum_{i=1}^L \left\| \left(\alpha_i + \frac{N}{L-1} \right) x_i - \bar{x} \right\|_2 + \varepsilon,$$

$$\text{where } \bar{x} = \sum_{i=1}^L \alpha_i x_i \text{ and } \varepsilon \leq \frac{1}{2} \left(1 - \frac{N}{L} \right) \sum_{i=1}^L \frac{N}{L-1} \frac{\sum_{j \neq i} \alpha_j^2 \|x_j\|_2^2}{\left\| \alpha_i x_i - \frac{N}{L-1} \sum_{j \neq i} \alpha_j x_j \right\|_2}.$$

Intuitively, for a *fixed* top- N set, the first term scales the largest out-of-set distance d_1 by the total weight $(1 - \bar{\alpha}_N)$ carried by the remaining $L - N$ tokens, while the second term accounts for how much those low-weight tokens can still perturb the context vector, proportional to the norm of the largest selected embedding. When the top- N indices are chosen *uniformly at random*, each token is excluded with probability $(L - N)/L$. Replacing the random indicator variables by their expectations yields the main sum; the residual ε captures the Jensen gap and is small whenever the individual weights α_i are not too concentrated. Full derivations are provided in Appendix A.

Theorem 1 relates the distance $\tilde{d}$ to key parameters (N , L , and the weights α_i). To see how the bound behaves in two extreme regimes, we state the following corollary.

Corollary 2. (i) **Fixed $N \ll L$.** *When N is held constant and L grows,*

$$E \approx \sum_{i=1}^L \|\alpha_i x_i\|_2.$$

(ii) **Fixed L , $N \rightarrow L$.** *If L is fixed and the top set expands to the full sequence, i.e. $N \rightarrow L$,*

$$E \rightarrow 0. \quad (9)$$

When N is small relative to the sequence length, most tokens are excluded and the expected distance is dominated by the individual contributions $\|\alpha_i x_i\|_2$ of those low-weight tokens. In contrast, as N approaches L , the context vector s eventually incorporates *all* embeddings, so the distance between s and the (now empty) set of non-selected tokens vanishes.

2.2 Geometrical interpretation

We now recast the analysis in geometric terms, focusing on the spatial arrangement of the token embeddings x_i . When these vectors lie close together, the model struggles to separate informative from non-informative tokens, hindering training. To quantify how many tokens can be reliably distinguished, we work under two standard geometric assumptions:

Assumption 1 (Uniform spherical distribution). Each embedding lies uniformly on a d -dimensional sphere of radius M :

$$x_i \sim U(\mathbb{S}^{d-1}(M)), \quad i = 1, \dots, L,$$

where $\mathbb{S}^{d-1}(M)$ denotes the $(d-1)$ -sphere of radius M . In practice, we *normalize* embeddings so that they lie on this sphere.

Assumption 2 (Minimum pairwise separation). There exists a fixed lower bound on the distance between any two embeddings:

$$\min_{i \neq j} \|x_i - x_j\|_2 = \delta > 0,$$

ensuring that no two tokens collapse onto one another. In experiments, we set δ to the empirical minimum pairwise distance.

Under these assumptions, we can bound the number of embeddings that fall within a specified neighbourhood of the context vector, thereby quantifying the model’s effective resolution.

Let $I_N = \{i_1, \dots, i_N\}$ denote the indices of the N selected tokens and recall the context vector

$$s = \sum_{i \in I_N} \alpha_i x_i.$$

Fix a tolerance radius $r > 0$ and consider the closed Euclidean ball $B_r(s)$ centred at s . We say that a selected embedding x_i is *geometrically distinguishable* if its rescaled version $\alpha_i x_i$ remains within $B_r(s)$ while every non-selected embedding satisfies $\alpha_j x_j \notin B_r(s)$ for $j \notin I_N$. The count of such distinguishable embeddings is

$$N_s = \#\{i \in I_N : \|\alpha_i x_i - s\|_2 \leq r\}, \quad (10)$$

and the ratio N_s/N measures the fraction of “non-noise” vectors the model can reliably separate (see Fig. 1 for an illustration).

Our objective is to bound the expectation $\mathbb{E}[N_s]$ and hence provide explicit upper and lower limits on the model’s effective resolution as a function of r , N , and L under Assumptions 1–2.

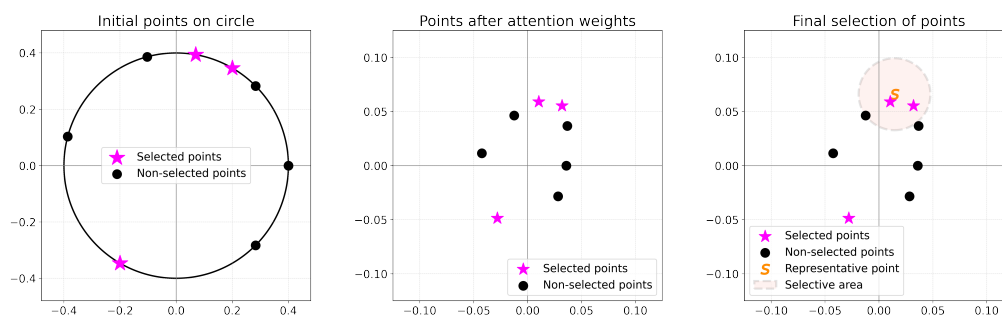


Figure 1: Illustrative example of the geometric separation. **Left:** Token embeddings lie on a circle. **Middle:** After scaling by their attention weights α_i , both attended (magenta stars) and non-attended (black dots) points move toward the origin. **Right:** Only the selected tokens that remain inside the ball $B_r(s)$ (shaded) are deemed distinguishable.

In the previous section, we examined separation in terms of Euclidean distances. The geometric approach introduced here, by contrast, focuses on directional separability and reframes the problem as one of metric learning. Now, we present the main result (for the proof, see Appendix A):

Theorem 2. Under Assumptions 1–2, the fraction of geometrically distinguishable embeddings satisfies

$$1 - \frac{1}{rN} \sum_{i \in I_N} \xi_i \leq \frac{\mathbb{E}[N_s]}{N} \leq \frac{1}{N} \sum_{i \in I_N} \exp\left[-\frac{(r-\xi_i)^2}{16M^2}\right], \quad (11)$$

where

$$\xi_i^2 = M^2 \sum_{\substack{j \in I_N \\ j \neq i}} \alpha_j^2 + \left(M^2 - \frac{\delta^2}{2}\right) \sum_{\substack{j, k \in I_N \\ j \neq k \neq i}} \alpha_j \alpha_k. \quad (12)$$

The quantity ξ_i measures how widely the other $N - 1$ selected embeddings, rescaled by their weights, are spread around x_i . If ξ_i is small, most selected points cluster near the context vector s , so many of them fall inside the ball $B_r(s)$ and become distinguishable. The lower bound in (11) subtracts from one a penalty proportional to the cumulative spread $\sum_i \xi_i$; the tighter the cluster (smaller ξ_i), the closer the ratio $\mathbb{E}[N_s]/N$ is to 1. Conversely, the upper bound shows that once r is smaller than ξ_i for many i , the exponential term decays rapidly, implying very few embeddings remain separable under the chosen radius.

2.3 Gradient Sensitivity of Attention

The results above show that a language model must sharply distinguish informative from non-informative tokens; in other words, the attention weight distribution should be as selective as possible. Refining that distribution, however, exposes a second difficulty: *gradient sensitivity* during training.

Consider two nearly identical logit vectors

$$\ell^{(1)} = (0, \dots, 0, a, a + \varepsilon), \quad \ell^{(2)} = (0, \dots, 0, a + 2\varepsilon, a),$$

so that their Euclidean distance satisfies $\|\ell^{(1)} - \ell^{(2)}\|_2 = \sqrt{5}\varepsilon$. Let $\alpha^{(1)}, \alpha^{(2)}$ be the corresponding softmax weights (Equation (2)) and denote by $\nabla_\ell \alpha$ the Jacobian of the softmax map. A first-order expansion gives

$$\|\alpha^{(1)} - \alpha^{(2)}\|_2 \approx \|\nabla_\ell \alpha^{(1)} (\ell^{(1)} - \ell^{(2)})\|_2 \sim \sqrt{2} \frac{\varepsilon}{T},$$

because the largest two logits swap order and the associated softmax gradient scales as $1/T$. Hence, even though the logits differ by only $O(\varepsilon)$, the output distribution can change by $O(\varepsilon/T)$. For a sufficiently small temperature T (a common tactic for sharpening attention) this factor becomes large, making the gradient step volatile and potentially destabilising optimisation.

This example highlights a fundamental trade-off: stronger normalisation (smaller T , or any variant that steepens the softmax) improves token separability but simultaneously amplifies gradient variance, complicating training of deep transformers.

Let have a closer look at how normalisation affects gradient descent. For the general scheme in (3), let $\{l_i\}_{i=1}^L$ be the logits and $\{\alpha_i\}_{i=1}^L$ the resulting attention weights. We characterise the magnitude of the Jacobian $\nabla_l \alpha$.

Lemma 2. *With the notation above,*

$$\|\nabla_l \alpha\|_2 \leq \min \left\{ \|F'\|_2 \left(\frac{1}{L \min_j F(l_j, \theta)} + \frac{\|F\|_2}{L^2 \min_j F^2(l_j, \theta)} \right), \sqrt{2} \right\} \quad (13)$$

where $\|F\|_2 = \max_i |F(l_i, \theta)|$ and $\|F'\|_2 = \max_i |F'(l_i, \theta)|$.

The upper bound grows when any logit l_j pushes $F(l_j, \theta)$ toward zero, because both terms inside the parentheses scale inversely with $\min_j F(l_j, \theta)$. Thus, a sharply peaked normalisation—softmax with a small temperature, for instance, induces large Jacobian norms, signalling high gradient sensitivity during training.

Corollary 3. *For the softmax normalisation in (2),*

$$\|\nabla_l \alpha\|_2 \leq \min \left\{ \frac{1}{4T}, \sqrt{2} \right\}.$$

A smaller temperature T sharpens the softmax but simultaneously inflates the Jacobian norm (scaling as $1/T$), making the attention distribution highly sensitive to even tiny logit perturbations and thus harder to train stably.

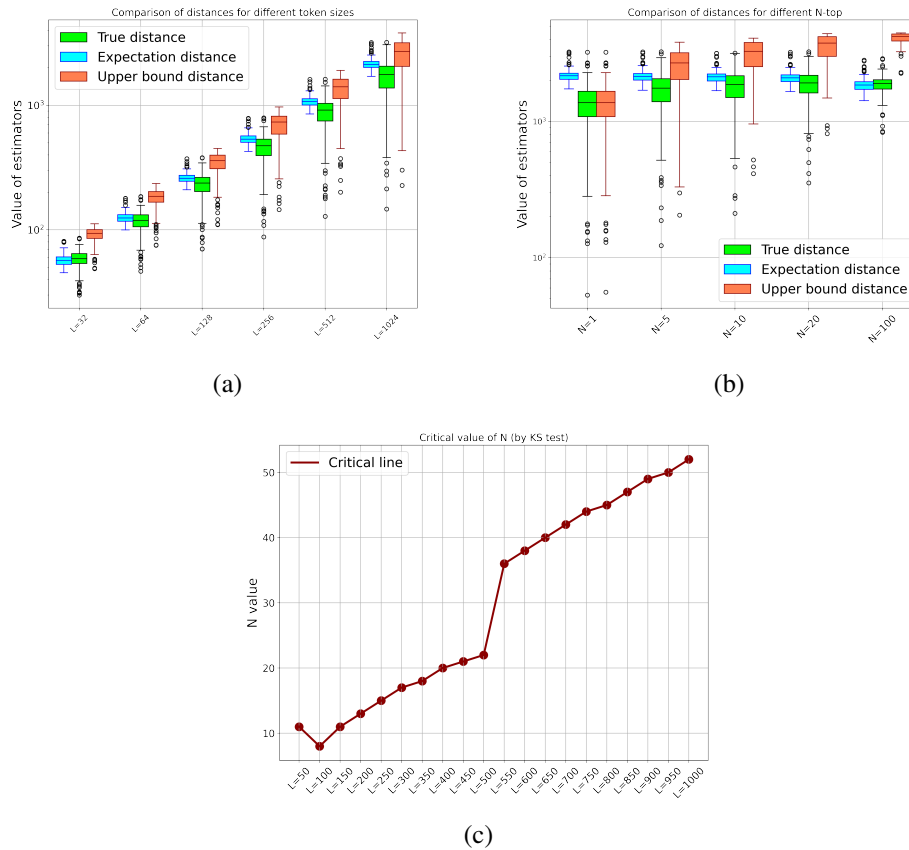


Figure 2: **Distance statistics validate Theorem 1.** (a) With $N = 5$, both the true distance (green) and its expectation (blue) grow roughly linearly in L ; the red upper bound is safe but conservative. (b) With $L = 1024$, increasing N beyond 20 yields diminishing returns: the distance plateaus while the bound tightens. (c) Critical top- N obtained by a KS test ($\alpha = 0.01$); fewer than 6 % of the tokens need to be selected before the empirical and expected distances become statistically indistinguishable.

3 Experiments

We evaluate our theoretical findings on the publicly available GPT-2 model family¹ [15]. All text is tokenised with byte-pair encoding (BPE) [6] as implemented in the Hugging Face transformers library. Unless otherwise stated, the input consists of consecutive chapters from *War and Peace* by Leo Tolstoy (public domain), providing long-form prose well beyond the model’s context window. For every layer and attention head, we extract the full attention matrix $A \in \mathbb{R}^{L \times L}$ and the associated query, key, and value tensors, enabling direct comparison with our distance and geometry metrics. Implementation details, hyperparameters, and reproducibility scripts are included in Appendix B.

3.1 Distance analysis

We now test the non-asymptotic bounds of Theorem 1. Implementation details appear in Appendix B. Two complementary experiments are performed:

1. *Scaling with sequence length.* Fix $N = 5$ and vary $L \in \{32, \dots, 1024\}$.
2. *Scaling with top- N .* Fix $L = 1024$ and vary $N \in \{1, 5, 10, 20, 100\}$.

¹We report results for the 124 M parameter version; qualitatively identical trends were observed for larger variants.

For each configuration we compute, across all 144 GPT-2 heads/layers, (i) the true distance $\tilde{d}$ (7), (ii) the expectation term of Theorem 1, and (iii) the analytic upper bound. In addition, we estimate a *critical* top- N value: the smallest N for which the empirical and expected distance distributions are indistinguishable under a two-sample Kolmogorov–Smirnov test ($\alpha = 0.01$).

Results summarised in Fig. 2 support following key observations. **(i)** For $N \ll L$ the distance scales linearly with sequence length, exactly as Corollary 2(i) predicts. **(ii)** As N approaches 100, the true and expected distances converge while the upper bound tightens—evidence for the $N \rightarrow L$ collapse of Eq. (9). **(iii)** The critical- N curve grows sub-linearly ($\approx 0.06L$), confirming that only a small subset of tokens can be separated before attention behaves as if weights were uniform.

These empirical trends corroborate the theoretical claim that softmax normalisation retains discriminative power only when the active set is a small fraction of the context; larger N values chiefly add noise.

3.2 Geometric Separability

We now quantify how many of the N highest-weight tokens remain *geometrically distinguishable* according to Theorem 2. For each sequence we set $r = \min_{i \notin I_N} \|s - \alpha_i x_i\|_2$, so that every non-selected token lies outside the ball $B_r(s)$. Using GPT-2 embeddings normalised as in Assumptions 1–2, we compute the empirical ratio N_s/N (Definition (10)) across all heads and layers and compare it with the analytic bounds of Theorem 2; results are summarised in Fig. 3.

Across the full model, the proportion of distinguishable tokens declines up to $N \approx 16$ and then saturates between 0.7 and 0.85. The exponential *upper* bound tracks the empirical maxima closely, showing the theorem is tight on the high side, whereas the *lower* bound is intentionally conservative. Thus—even with idealised spherical embeddings—softmax attention cannot cleanly separate more than about four-fifths of the tokens it selects; adding further tokens mainly dilutes the representation without improving geometric resolution.

3.3 Gradient sensitivity

We next explore how the softmax temperature T and logit perturbation scale ε affect the stability of the attention map. For each head–layer pair we evaluate the finite-difference Jacobian norm

$$g(T, \varepsilon) = \frac{1}{\varepsilon} \left\| \alpha_{\ell+\varepsilon\Delta\ell} - \alpha_{\ell} \right\|_2, \quad \|\Delta\ell\|_2 = 1,$$

which approximates $\|\nabla_{\ell} \alpha\|_2$. Full implementation details are provided in Appendix B. Figure 4 shows the *maximum* value of $g(T, \varepsilon)$ across all 144 heads/layers of GPT-2 for $\varepsilon \in \{10^{-3}, 10^{-1}, 10\}$.

For $T < 0.1$ the empirical curves follow the theoretical $1/T$ trend of Corollary 3; smaller ε values yield larger gradients because the sharpest logits dominate the variation. Once $T \geq 1$ all curves collapse and drop by two orders of magnitude, indicating much improved robustness to logit perturbations but a concomitant loss of selectivity. These findings confirm the trade-off already highlighted by our theory: *sharper softmax improves token separability yet*

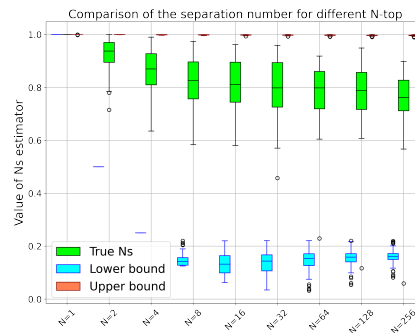


Figure 3: Geometric separability saturates at 70–85%. For increasing top- N , the empirical fraction of distinguishable embeddings N_s/N (green boxes) quickly plateaus; roughly one-fifth of selected tokens remain outside $B_r(s)$.

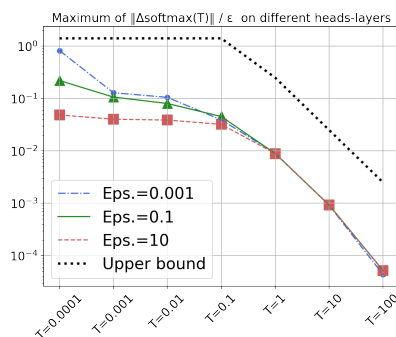


Figure 4: Gradient sensitivity decays as $1/T$. Maximum finite-difference Jacobian norm $g(T, \varepsilon)$ for three perturbation magnitudes (coloured curves, log–log scale). The dashed black curve is the theoretical bound $\min\{1/(4T), \sqrt{2}\}$ from Corollary 3.

inflates gradient variance, whereas higher temperatures stabilise training at the cost of blurrier attention.

4 Discussion

Softmax remains the default normalisation in modern Transformers because it is simple, differentiable and lends a probabilistic interpretation to the weights [18]. Its limitations, however, are now well established: it produces dense distributions, its peak value drops as the context grows (*vanishing-attention* effect), and its Jacobian explodes when the temperature is driven low, yielding unstable gradients. Three main research lines have emerged to address these issues.

1. *Sparsity-inducing rules* such as Sparsemax and α -Entmax replace the softmax exponential with projections onto a simplex, producing exact zeros and lowering entropy [12].
2. *Length-aware rescales* such as Scalable-Softmax scale logits by $\log L$ so that the maximum attention weight stays roughly constant as the sequence grows, alleviating the vanishing-attention problem [13].
3. *Gradient-controlled variants* (e.g. Self-Adjusted Softmax) re-weight logits according to the layer-wise dynamic range, keeping the Jacobian spectrum in a healthy band [22].

Our work complements these efforts by *explaining, in closed form, why the above modifications are necessary*. Where previous papers introduced a new rule and showed empirical gains, we derived non-asymptotic bounds (Theorems 1–2) and gradient limits (Lemma 2) that *apply to any normalisation function* $F(\cdot, \theta)$. In particular:

- The **distance bound** shows that when N grows proportionally to L , the representation distance $\tilde{d}$ necessarily collapses to zero, formalising the empirical softmax bottleneck noted by [20]. Our GPT-2 experiment (Fig. 2, middle) confirms the predicted plateau and justifies why sparse or length-aware rules can improve long-context performance.
- The **geometric bound** states that even under optimistic spherical assumptions no more than $\approx 80\%$ of the top- N tokens can stay inside the selective ball (Fig. 3). This explains why empirical studies report that multiple heads are required to cover distinct parts of the context: a single head cannot make every “important” token simultaneously salient. The reason behind this idea is pretty simple. Assuming the independence of the heads as attention mechanism parts, we can conclude that for given separability level p , H heads can cover up to $1 - (1 - p)^H$ of the top- N tokens. Under our analysis, we see that for $p = 0.8$ it is needed approximately $H = 3$ to cover up the 99% of information.
- The **gradient bound** recovers the familiar $1/(4T)$ law but also shows how any normaliser with a vanishing minimum value will inherit the same instability. Figure 4 demonstrates that GPT-2 operates close to the theoretical limit when $T < 10^{-1}$, validating the usefulness of gradient-controlled variants such as SA-Softmax.

The combined theory–experiment picture suggests three practical guidelines.

1. **Keep the active set small.** The critical- N curve in Fig. 2 grows roughly like $0.06L$; selecting more tokens yields vanishing returns and erodes separability. Top- k or sparse attention should be preferred when $L \gg k$.
2. **Monitor attention entropy.** A rising entropy or a drop in the empirical N_s/N ratio is an early sign that a head has saturated its geometric capacity; adding additional heads or switching to a length-aware normaliser can restore separability.
3. **Avoid overly sharp softmax.** Lowering T below 10^{-1} increases Jacobian norms without increasing separability (Figs. 3, 4). Practitioners should instead use normalisers that decouple selectivity from gradient health (e.g. Sparsemax, SS-Max, or SA-Softmax).

Our analysis assumes embeddings are *a priori* L2-normalised and roughly isotropic. Real models may violate these assumptions, and future work should extend the geometric bound to non-spherical distributions. Another promising direction is to design a length-adaptive, gradient-controlled normaliser that inherits the best properties of Sparsemax (sparsity), SS-Max (length awareness) and SA-Softmax (stable Jacobians) while admitting proofs analogous to Theorems 1–2.

5 Conclusions

This work provides theoretical and empirical analysis of normalisation in attention mechanisms beyond the classical softmax. We derived two non-asymptotic bounds (Theorems 1 and 2) that link token separability to sequence length L , selection size N , and the embedding geometry, and we established a general Jacobian bound (Lemma 2) that explains the well-known trade-off between sharpness and gradient stability. Experiments on GPT-2 confirmed all three predictions:

- the representation distance collapses once N grows proportionally to L ;
- no more than $\approx 80\%$ of the selected tokens can be geometrically distinguished, even under ideal spherical embeddings;
- the empirical Jacobian norm tracks the theoretical $1/(4T)$ law, saturating at low temperature.

Taken together, these results recast softmax attention as a *selective but capacity-limited aggregator*: it discriminates well only while the active set is a small fraction of the context. The analysis also clarifies why recently proposed normalisers—Sparsemax, Scalable-Softmax, and Self-Adjusted Softmax—offer complementary benefits: they relax one or more of the intrinsic limits quantified here.

Practical takeaways. (1) limit the top- k set to a sub-linear function of the context length; (2) monitor attention entropy or the N_s/N ratio during training; and (3) prefer length-aware or sparsity-inducing normalisers over aggressive temperature scaling.

Overall, our experiments shows several limitations providing the practical diagnostic idea:

1. When the level of geometrical separability drops to 70 – 80%, this signifies that the head has saturated its geometric capacity.
2. The temperature limitations shows that making the distribution sharp makes the Jacobian norm exploding. The teoretical and practical analysis shows that it's better to avoid using the $T \leq 0.1$.

Future directions. Immediate extensions include (i) relaxing the spherical-embedding assumption, (ii) analysing multi-query and multi-head interactions, and (iii) designing a single normalisation rule that is simultaneously length-adaptive, sparse, and gradient-stable. We hope the quantitative framework introduced here will serve as a benchmark for such developments and, more broadly, for principled improvements to long- context transformers.

Overall, the present study provides theoretical footing for the growing body of work that modifies softmax, and furnishes quantitative tools for diagnosing when a given attention head has reached its intrinsic limit.

References

- [1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [2] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer, 2020.
- [3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.
- [4] R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers, 2019.
- [5] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In A. Korhonen, D. Traum, and L. Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics.

- [6] P. Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, Feb. 1994.
- [7] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines, 2014.
- [8] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.
- [9] Joblib Development Team. Joblib: running python functions as pipeline jobs, 2020.
- [10] N. Kitaev, Łukasz Kaiser, and A. Levskaya. Reformer: The efficient transformer, 2020.
- [11] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, 2019.
- [12] A. F. T. Martins and R. F. Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification, 2016.
- [13] K. M. Nakanishi. Scalable-softmax is superior for attention, 2025.
- [14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.
- [15] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
- [16] J. W. Rae, A. Potapenko, S. M. Jayakumar, and T. P. Lillicrap. Compressive transformers for long-range sequence modelling, 2019.
- [17] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. *Advances in neural information processing systems*, 28, 2015.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [19] P. Veličković, C. Perivolaropoulos, F. Barbero, and R. Pascanu. softmax is not enough (for sharp out-of-distribution), 2024.
- [20] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen. Breaking the softmax bottleneck: A high-rank rnn language model, 2018.
- [21] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. Big bird: Transformers for longer sequences, 2021.
- [22] C. Zheng, Y. Gao, G. Chen, H. Shi, J. Xiong, X. Ren, C. Huang, X. Jiang, Z. Li, and Y. Li. Self-adjust softmax, 2025.

A Theory justification

In the given appendix, we provide the proofs for our theory part.

Lemma 1. Consider the α_i term:

$$\alpha_i = \frac{F(l_i, \theta)}{\sum_{j=1}^L F(l_j, \theta)} \leq \frac{C_2'(\theta)}{\sum_{j=1}^L C_1'(\theta)} \leq \frac{C_2(\theta)}{L}, \quad (14)$$

since the function is continuous on a compact space. The lower bound works the same.

For the proof of the corollary, notice that $l_i = \mathbf{q}^\top \mathbf{k}_i$, therefore $|l_i|$ is bounded by norm of $\|\mathbf{q}\| \|\mathbf{k}\|$. Hence, we have the following bound. $\square$

Theorem 1. 1. Here we have fixed $I_N = \{i_1, \dots, i_N\} \subset [1, \dots, L]$. Therefore, we have:

$$\tilde{d} = \sum_{i \in I \setminus I_N} \left\| \alpha_i x_i - \sum_{j \in I_N} \alpha_j x_j \right\| \leq \sum_{i \in I \setminus I_N} \sum_{j \in I_N} \alpha_j \left\| \frac{\alpha_i}{\bar{\alpha}_N} x_i - x_j \right\| = \quad (15)$$

$$\sum_{i \in I \setminus I_N} \sum_{j \in I_N} \alpha_j \left\| \frac{\alpha_i}{\bar{\alpha}_N} x_i - \frac{\alpha_i}{\bar{\alpha}_N} x_j + \frac{\alpha_i}{\bar{\alpha}_N} x_j - x_j \right\| \leq \quad (16)$$

$$\sum_{i \in I \setminus I_N} \sum_{j \in I_N} \alpha_j \left(\frac{\alpha_i}{\bar{\alpha}_N} d_1 + \|x_j\| \left| 1 - \frac{\alpha_i}{\bar{\alpha}_N} \right| \right) = \quad (17)$$

$$(1 - \bar{\alpha}_N) d_1 + \max_{j \in I_N} \|x_j\| [\bar{\alpha}_N (L - N) - (1 - \bar{\alpha}_N)], \quad (18)$$

where $d_1 = \max_{i \notin I_N, j \in I_N} \|x_i - x_j\|$.

2. For the probability part:

$$E = \mathbb{E} \left[\sum_{i \notin I_N} \|\alpha_i x_i - s\| \right] = \mathbb{E} \left[\sum_{i=1}^L 1(i \notin I_N) \|\alpha_i x_i - s\| \right] \approx \quad (19)$$

$$\sum_{i=1}^L \mathbb{P}(i \notin I_N) \mathbb{E} [\|\alpha_i x_i - s\| | i \notin I_N]. \quad (20)$$

We can estimate both the expectation of norm and probability terms as follows:

$$\mathbb{P}(i \notin I_N) = \frac{L - N}{N}, \quad (21)$$

$$\mathbb{E} [\|\alpha_i x_i - s\| | i \notin I_N] = \mathbb{E} \left[\left\| \alpha_i x_i - \sum_{j \neq i} 1(j \in I_N) \alpha_j x_j \right\| \middle| i \notin I_N \right] \approx \quad (22)$$

$$\left\| \alpha_i x_i - \frac{N}{L-1} \sum_{j \neq i} \alpha_j x_j \right\|. \quad (23)$$

As a result, we obtain:

$$E = \frac{L - N}{L} \sum_{i=1}^L \left\| \alpha_i \left(1 + \frac{N}{L-1} \right) x_i - \frac{N}{L-1} \bar{x} \right\| + \varepsilon \quad (24)$$

The error term between the approximation and the true value can be estimated using Jensen's gap bound:

$$\varepsilon_i = \mathbb{E}[\|\alpha_i x_i - s\| | i \notin I_N] - \|\alpha_i x_i - \mathbb{E}[s | i \notin I_N]\| \leq \frac{1}{2} \frac{N(L - N - 1) \sum_{j \neq i} \alpha_j^2 \|x_j\|^2}{\|\alpha_i x_i - s_i\|} \quad (25)$$

Therefore:

$$\varepsilon \leq \sum_{i=1}^L \frac{L - N}{L} \varepsilon_i \quad (26)$$

Now let's move to the corollary section.

1. Assuming $N \ll L$ and L grows, we have:

$$E \approx \sum_{i=1}^L \alpha_i \|x_i\| \quad (27)$$

2. When $N \rightarrow L$, we have $E \rightarrow 0$, since number of outer elements goes to zero.

□

Theorem 2. We have:

$$\mathbb{E}[N_s] = \sum_{i \in I_N} 1(\|s - \alpha_i x_i\| \leq r) = \sum_{i \in I_N} \mathbb{P}(\|s - \alpha_i x_i\| \leq r) \quad (28)$$

Hence, we need to estimate the probability of the $\alpha_i x_i$ being in the sphere.

Notice that $\|\alpha_i x_i - s\|$ is bounded random variable. We can estimate it as $\|\alpha_i x_i - s\| \in [0, 2M]$. Hence, we have a Hoeffding-type inequality:

$$\mathbb{P}(X_i \leq r) \leq \inf_t [e^{-rt} \mathbb{E} e^{X_i t}] \leq \exp \left[\inf_t (-rt + t \mathbb{E} X_i + 4M^2 t^2) \right] \leq \exp \left[-\frac{(r - \mathbb{E} X_i)^2}{16M^2} \right], \quad (29)$$

where expected value of $X_i = \|s - \alpha_i x_i\|$ can be estimated as follows:

$$\mathbb{E} X_i \leq \sqrt{\mathbb{E} X_i^2}, \quad (30)$$

and for the squared norm, we have:

$$\|X_i\|^2 = \left\| \alpha_i x_i - \sum_{j \in I_N} \alpha_j x_j \right\|^2 = \left\| \sum_{j \in I_N, j \neq i} \alpha_j x_j \right\|^2 = \sum_{\substack{j, k \in I_N \\ j \neq i \\ k \neq i}} \alpha_j \alpha_k \langle x_i, x_j \rangle \leq \quad (31)$$

$$M^2 \sum_{\substack{j \in I_N \\ j \neq i}} \alpha_j^2 + \left(M^2 - \frac{\delta^2}{2} \right) \sum_{\substack{j, k \in I_N \\ j \neq k \neq i}} \alpha_j \alpha_k \equiv \xi_i^2, \quad (32)$$

where the last bound caused by condition $\|x_i - x_j\|^2 \geq \delta^2$.

Therefore, we have:

$$\mathbb{E}[N_s] \leq \sum_{i \in I_N} \exp \left[-\frac{(r - \xi_i)^2}{16M^2} \right] \quad (33)$$

The lower bound is easier. Using Markov inequality and Cauchy-Schwarz (30):

$$\mathbb{P}(X_i \leq r) = 1 - \mathbb{P}(X_i > r) \leq 1 - \frac{\mathbb{E}X_i}{r} \geq 1 - \frac{\xi_i}{r} \quad (34)$$

Hence, we have:

$$N - \frac{1}{r} \sum_{i \in I_N} \xi_i \leq \mathbb{E}[N_s] \leq \sum_{i \in I_N} \exp \left[-\frac{(r - \xi_i)^2}{16M^2} \right] \quad (35)$$

□

B Experiments

In the appendix, we provide details of the whole experiment setup and give the pseudocode we implemented for each figure.

System parameters

For the given research, we used the Apple M1 Pro chip with a 10-core CPU and 16GB of unified memory, based on ARM architecture.

Software framework

The models were implemented and examined using PyTorch [14], running on the Apple M1 Pro's ARM-based CPU architecture to ensure efficient computation.

For the parallelization procedure, we used Joblib library [9].

Distance analysis

Algorithm 1 Distance Analysis. Different L and fixed N .

Require: Text input, list of token lengths L_{values} , fixed $N = 5$

Ensure: Averaged distance statistics across layers and heads

```

1: for  $L$  in  $L_{\text{values}}$  do
2:   Encode text using GPT-2 and extract attention matrices for all heads and layers
3:   for each (head, layer) in GPT-2 do
4:     for each token index  $t$  in  $1, \dots, L$  do
5:       Compute true distance  $\tilde{d}$  via Eq. (7)
6:       Compute upper bound  $d_{\text{max}}$  via Eq. (9)
7:       Compute expectation  $E$  via Eq. (10)
8:       Store  $(\tilde{d}, d_{\text{max}}, E)$ 
9:     end for
10:    Average distances across tokens
11:    Store result for (head, layer)
12:  end for
13:  Store all results for current  $L$ 
14: end for
15: return All distance statistics

```

Algorithm 2 Distance Analysis. Different top- N and fixed L

Require: Text input, list of top- N values N_{values} , fixed $L = 1024$

Ensure: Averaged distance statistics across layers and heads

```
1: Encode text using GPT-2 and extract attention matrices for all heads and layers
2: for  $N$  in  $N_{\text{values}}$  do
3:   for each (head, layer) in GPT-2 do
4:     for each token index  $t$  in  $1, \dots, L$  do
5:       Compute true distance  $\tilde{d}$  via Eq. (7)
6:       Compute upper bound  $d_{\text{max}}$  via Eq. (9)
7:       Compute expectation  $E$  via Eq. (10)
8:       Store  $(\tilde{d}, d_{\text{max}}, E)$ 
9:     end for
10:    Average distances across tokens
11:    Store result for (head, layer)
12:  end for
13:  Store all results for current  $N$ 
14: end for
15: return All distance statistics
```

Algorithm 3 Distance Analysis. Critical Top- N detection.

Require: Text input, L , top- N values N_{values} , significance level $\alpha = 0.05$

Ensure: First N for which expected and true distances are statistically close

```
1: Encode text using GPT-2 and extract attention matrices
2: for  $N$  in  $N_{\text{values}}$  do
3:   Initialize list of relative errors
4:   for each (head, layer) in GPT-2 do
5:     for each token  $t$  in  $1, \dots, L$  do
6:       Compute true distance  $\tilde{d}$  via Eq. (7)
7:       Compute expected distance  $E$  via Eq. (10)
8:     end for
9:     Compute mean true distance  $\bar{d}$ 
10:    Compute mean expected distance  $\bar{E}$ 
11:    Store the  $\bar{d}$  and  $\bar{E}$ 
12:  end for
13:  if Kolmogorov-Smirnov( $\bar{d}, \bar{E}$ , significance level  $\alpha$ ) is true then
14:    return  $N$ 
15:  end if
16: end for
17: return  $-1$  ▷ No  $N$  meets the condition
```

Geometrical analysis

Algorithm 4 Geometrical analysis. Separation Ratio and Bounds for Top-N Attention Tokens.

Require: Text input, sequence length L , top- N values N_{values}

Ensure: Box plots of N_s/N and its theoretical bounds

```
1: Encode text using GPT-2 and extract attention matrices
2: Extract and normalize token embeddings
3: for  $N$  in  $N_{\text{values}}$  do
4:   for each (head, layer) in GPT-2 do
5:     for each token  $t$  in  $1, \dots, L$  do
6:       Compute  $N_s/N$  via direct counting
7:       Compute lower and upper bounds from Theorem 2
8:     end for
9:     Average  $N_s/N$ , upper bound, and lower bound over all tokens
10:    Store results for (head, layer)
11:  end for
12:  Store aggregated results for  $N$ 
13: end for
14: Generate box plots comparing true and theoretical values
```

Gradient analysis

Algorithm 5 Gradient Sensitivity Analysis

Require: Text, temperature values T_{values} , shift values $\varepsilon_{\text{values}}$

Ensure: Sensitivity statistics across temperatures and shifts

```
1: Convert text to logits matrices for all (head, layer) pairs
2: for  $T$  in  $T_{\text{values}}$  do
3:   for  $\varepsilon$  in  $\varepsilon_{\text{values}}$  do
4:     for each (head, layer) do
5:       for each token  $t \in \{1, \dots, L\}$  do
6:         Sample unit vector  $v$  with  $\|v\|_2 = 1$ 
7:         Compute shifted logits  $l' = l + \varepsilon v$ 
8:         Compute softmax distributions  $\alpha = \text{softmax}(l/T)$ ,  $\alpha' = \text{softmax}(l'/T)$ 
9:         Compute sensitivity:  $\delta = \|\alpha' - \alpha\|_2 / \varepsilon$ 
10:      end for
11:      Average  $\delta$  over tokens
12:    end for
13:    Store maximum average  $\delta$  across (head, layer) for current  $\varepsilon$ 
14:  end for
15:  Store results for temperature  $T$ 
16: end for
17: return Sensitivity statistics
```

Time resources

Here, we provide the time execution for all algorithms:

Algorithm	Time of execution	Parallelization
Alg. 1	24 min.	No
Alg. 2	37 min.	No
Alg. 3	17 h. 4min.	Yes
Alg. 4	7 min.	No
Alg. 5	1 min.	No

Table 1: Comparison of algorithm execution times. The table references algorithms defined in Algorithms 1–5, highlighting their respective performance durations and whether they used parallelization.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

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

Answer: [Yes]

Justification: The abstract reflects the general idea of the paper. All additional details are provided in the paper itself.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

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

Answer: [Yes]

Justification: We discuss the theoretical limitations for Theorem 2. While the other results were provided without any additional assumptions.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: For all theoretical results, we provide the proper formulation. All objects of interest are stated in the corresponding sections. All proofs are provided in Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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: [\[Yes\]](#)

Justification: For each experimental result in the section "Experiments", we demonstrate the corresponding algorithm in the Appendix B. After the blind review process, we will share the corresponding scripts.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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 use the open-source model (GPT-2) with proper reference to the model. Also, we directly stated the dataset we used and provided more explanation in the Appendix B.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

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

Answer: [\[Yes\]](#)

Justification: We provided the algorithms and data parameters we used to replicate the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

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 apply the statistical tests in several experiments and reflect them in the corresponding algorithm. Also, we state the significance level and the test we use in the text directly.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: We provide the given information in the Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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 follow 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: There is no societal impact of the work performed. The paper is a theoretical examination of the well-known model.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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: We provide all the details about the data and the model we use.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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 provide the references to the data and models we use.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

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

Answer: [NA]

Justification: We don't provide new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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 work doesn't satisfy the given claim.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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 work doesn't satisfy the given claim.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The given paper was written by a group of authors. We followed the LLM usage guidelines and didn't violate them.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.