

1 A Theory justification

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

3 *Lemma 1.* Consider the α_i term:

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

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

5 For the proof of the corollary, notice that $l_i = \mathbf{q}^\top \mathbf{k}_i$, therefore $|l_i|$ is bounded by norm of $\|\mathbf{q}\| \|\mathbf{k}\|$.

6 Hence, we have the following bound. \square

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

$$\tilde{d} = \sum_{i \in I \setminus I_N} \left\| \alpha_i x_i - \sum_{j \in I_N} \alpha_j x_j \right\| \leq \sum_{i \in I \setminus I_N} \sum_{j \in I_N} \alpha_j \left\| \frac{\alpha_i}{\bar{\alpha}_N} x_i - x_j \right\| = \quad (2)$$

8

$$\sum_{i \in I \setminus I_N} \sum_{j \in I_N} \alpha_j \left\| \frac{\alpha_i}{\bar{\alpha}_N} x_i - \frac{\alpha_i}{\bar{\alpha}_N} x_j + \frac{\alpha_i}{\bar{\alpha}_N} x_j - x_j \right\| \leq \quad (3)$$

$$\sum_{i \in I \setminus I_N} \sum_{j \in I_N} \alpha_j \left(\frac{\alpha_i}{\bar{\alpha}_N} d_1 + \|x_j\| \left| 1 - \frac{\alpha_i}{\bar{\alpha}_N} \right| \right) = \quad (4)$$

$$(1 - \bar{\alpha}_N) d_1 + \max_{j \in I_N} \|x_j\| [\bar{\alpha}_N (L - N) - (1 - \bar{\alpha}_N)], \quad (5)$$

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

10 2. For the probability part:

$$E = \mathbb{E} \left[\sum_{i \notin I_N} \|\alpha_i x_i - s\| \right] = \mathbb{E} \left[\sum_{i=1}^L 1(i \notin I_N) \|\alpha_i x_i - s\| \right] \approx \quad (6)$$

$$\sum_{i=1}^L \mathbb{P}(i \notin I_N) \mathbb{E} [\|\alpha_i x_i - s\| | i \notin I_N]. \quad (7)$$

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

$$\mathbb{P}(i \notin I_N) = \frac{L - N}{N}, \quad (8)$$

12

$$\mathbb{E} [\|\alpha_i x_i - s\| | i \notin I_N] = \mathbb{E} \left[\left\| \alpha_i x_i - \sum_{j \neq i} 1(j \in I_N) \alpha_j x_j \right\| \middle| i \notin I_N \right] \approx \quad (9)$$

13

$$\left\| \alpha_i x_i - \frac{N}{L-1} \sum_{j \neq i} \alpha_j x_j \right\|. \quad (10)$$

14 As a result, we obtain:

$$E = \frac{L - N}{L} \sum_{i=1}^L \left\| \alpha_i \left(1 + \frac{N}{L-1} \right) x_i - \frac{N}{L-1} \bar{x} \right\| + \varepsilon \quad (11)$$

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

$$\varepsilon_i = \mathbb{E}[\|\alpha_i x_i - s\| | i \notin I_N] - \|\alpha_i x_i - \mathbb{E}[s | i \notin I_N]\| \leq \frac{1}{2} \frac{N(L - N - 1) \sum_{j \neq i} \alpha_j^2 \|x_j\|^2}{\|\alpha_i x_i - s_i\|} \quad (12)$$

17 Therefore:

$$\varepsilon \leq \sum_{i=1}^L \frac{L - N}{L} \varepsilon_i \quad (13)$$

18 Now let's move to the corollary section.

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

$$E \approx \sum_{i=1}^L \alpha_i \|x_i\| \quad (14)$$

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

21 □

22 *Theorem 2.* We have:

$$\mathbb{E}[N_s] = \sum_{i \in I_N} \mathbb{1}(\|s - \alpha_i x_i\| \leq r) = \sum_{i \in I_N} \mathbb{P}(\|s - \alpha_i x_i\| \leq r) \quad (15)$$

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

24 Notice that $\|\alpha_i x_i - s\|$ is bounded random variable. We can estimate it as $\|\alpha_i x_i - s\| \in [0, 2M]$.

25 Hence, we have a Hoeffding-type inequality:

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

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

$$\mathbb{E} X_i \leq \sqrt{\mathbb{E} X_i^2}, \quad (17)$$

27 and for the squared norm, we have:

$$\|X_i\|^2 = \left\| \alpha_i x_i - \sum_{j \in I_N} \alpha_j x_j \right\|^2 = \left\| \sum_{j \in I_N, j \neq i} \alpha_j x_j \right\|^2 = \sum_{\substack{j, k \in I_N \\ j \neq i \\ k \neq i}} \alpha_j \alpha_k \langle x_i, x_j \rangle \leq \quad (18)$$

$$M^2 \sum_{\substack{j \in I_N \\ j \neq i}} \alpha_j^2 + \left(M^2 - \frac{\delta^2}{2} \right) \sum_{\substack{j, k \in I_N \\ j \neq k \neq i}} \alpha_j \alpha_k \equiv \xi_i^2, \quad (19)$$

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

29 Therefore, we have:

$$\mathbb{E}[N_s] \leq \sum_{i \in I_N} \exp \left[-\frac{(r - \xi_i)^2}{16M^2} \right] \quad (20)$$

30 The lower bound is easier. Using Markov inequality and Cauchy-Schwarz (17):

$$\mathbb{P}(X_i \leq r) = 1 - \mathbb{P}(X_i > r) \leq 1 - \frac{\mathbb{E}X_i}{r} \geq 1 - \frac{\xi_i}{r} \quad (21)$$

31 Hence, we have:

$$N - \frac{1}{r} \sum_{i \in I_N} \xi_i \leq \mathbb{E}[N_s] \leq \sum_{i \in I_N} \exp \left[-\frac{(r - \xi_i)^2}{16M^2} \right] \quad (22)$$

32 □

33 B Experiments

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

36 System parameters

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

39 Software framework

40 The models were implemented and examined using PyTorch [2], running on the Apple M1 Pro’s
41 ARM-based CPU architecture to ensure efficient computation.

42 For the parallelization procedure, we used Joblib library [1].

43 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
```

44 **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_{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
-

45 **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_{values} **do**
 - 3: **for** ε 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 δ over tokens
 - 12: **end for**
 - 13: Store maximum average δ across (head, layer) for current ε
 - 14: **end for**
 - 15: Store results for temperature T
 - 16: **end for**
 - 17: **return** Sensitivity statistics
-

46 Time resources

47 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.

48 References

- 49 [1] Joblib Development Team. Joblib: running python functions as pipeline jobs, 2020.
- 50 [2] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshein,
51 L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In
52 *Advances in Neural Information Processing Systems*, 2019.