

# Greedy-Gnorm: A Gradient Matrix Norm-Based Approach Replacing Attention Entropy for Head Pruning

Yuxi Guo

Southwestern University of Finance and Economics

yuxiguo@udel.edu

November 6, 2024

## Abstract

We present a novel head pruning method called Greedy-Gnorm, for transformer-based models that leverages gradient matrix norms to identify and remove less essential attention heads. By employing a greedy algorithm based on the Gnorm indicator, Greedy-Gnorm overcomes limitations of Attention Entropy (AE) methods, such as underflow issues that can lead to NaN values. Experimental results across multiple transformer architectures—BERT, ALBERT, RoBERTa, and XLM-RoBERTa—demonstrate that Greedy-Gnorm achieves superior accuracy retention and stability compared to AE and other pruning approaches, especially in scenarios of progressive pruning. The proposed method selectively prunes heads with minimal impact on performance, as evidenced by a consistent pruning curve that validates its reliability and efficiency. Furthermore, random pruning experiments support Greedy-Gnorm’s effectiveness, showing that it outperforms random pruning by maintaining higher model accuracy. Greedy-Gnorm’s success suggests that it offers a practical, robust solution for model compression, enhancing deployment feasibility in resource-constrained environments.

**Keywords:** Gnorm, Attention Entropy, greedy algorithm, model compression

## 1 Introduction

In recent years, deep learning models, especially transformer-based architectures such as BERT, ALBERT and so on, have demonstrated remarkable performance across various natural language processing tasks, like GPT-3(Brown et al.) whose size of parameters can reach over 150B. However, the increasing model size presents challenges in terms of parameters redundancy(Michel et al.) and deployment efficiency, particularly in resource-constrained environments. To define the importances of parameters, we need to find the particular functions of them. Dodrio tells us that different heads play various roles, like semantic or syntactic functions to understand the sentences.(Wang et al.) Under the assumption that lower entropy signifies a more focused and crucial attention head. However, this method is not without limitations. In certain scenarios, entropy may suffer from underflow issues, which can lead to NAN value spreading phenomenon. Then this paper will give a scheme to solve this problem.

We choose four models to validate the effectiveness of new indicator: Gnorm (Norms of gradient matrix), and this research gives the comparison between Attention Entropy and Gnorm. Models are respectively BERT(Devlin et al.), ALBERT(Lan et al.), RoBERTa(Liu et al.) and XLM-RoBERTa(Conneau et al.). The datasets are also corresponding to their focused tasks(financial-sentiment-analysis, mnli, tweets-sentiment-analysis and language-identification).

We need to test a phenomenon first: after cutting off a head, the parameter gradient of other heads changes. If there is a change, indicating that the greedy algorithm cut a head to calculate a Gnorm idea is more reasonable.

The variation of the gradient matrix is visualized here and presented as a heat map. We cut the heads outside the observed object layer and the heads inside the observed object layer. After capturing two gradient matrices of pruned Q weights respectively, and the original gradient matrix is different to observe the heat maps of their differences. If heat maps do not have exactly the same color, it is obvious that the gradient changes.

For the convenience of presentation, downsampling can reduce the  $768 \times 768$  gradient matrix into  $12 \times 12$  matrix with  $64 \times 64$  pooling layer. The results show whether the pruned heads are at layers above object layer

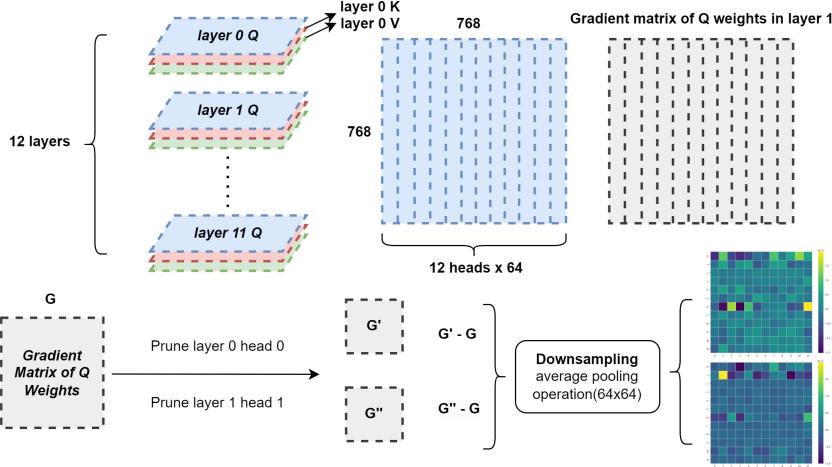


Figure 1: Sampled gradient matrix changes after pruning heads

or not. The gradient matrix always changes, which means that we have to calculate the importance scores after pruning each head of models.

The contributions of our paper are as follows:

- Give the rectification of Attention Entropy to solve the underflow problem when calculating the AE scores of heads. And we prove that the rectification will not change the pruning order with curve integral, especially when the fine-tune value is small enough to make it fall original value's neighbourhood.
- Present a novel method called Greedy-Gnorm, which can prune heads based on norms of gradient matrix with greedy algorithm. And we compare four models and conclude its effectiveness based on its convex pruning curve. We give the steps about how to find the least important head in terms of current pruning conditions and heads mask.
- Provide corresponding solutions to prune four models with heatmaps which illustrates the left heads in models. And take BERT as example, we observe the particular conditions of inner parameters in BERT.

## 2 Related Work

ATTATTR and TIS, two indicators depends on partial derivatives. Actually, when we prune one head, scores of original, Attr or TIS which rely on gradients will update values. Given that the gradients alter during pruning all the time, we need to calculate new scores after pruning each head, which causes huge computational cost.

### 2.1 Attention Entropy

After previous researches, there are some indicators that evaluate the importance of heads in Transformer. Attention Entropy is one of solutions to prune redundancy heads in BERT, but when text becomes longer, tokens will be paid less attention for average. Then Attention Entropy will bring underflow phenomenons for small values of  $a_{ij}$ , so that  $\log(0)$  will appear for computer to spread NAN value. The validation dataset is denoted as set  $\mathcal{X}$ , then the  $t_{x_k}$  means the total input tokens of the  $k^{th}$  validation sentence. Here,  $a_{ij}^{(h)}$  means the attention score that  $i^{th}$  token points towards  $j^{th}$  token in the  $h^{th}$  head:

$$E(AE(h)) = -\frac{1}{|\mathcal{X}|} \sum_{k=1}^{|\mathcal{X}|} \frac{1}{t_{x_k}} \sum_{i=1}^{t_{x_k}} \sum_{j=1}^{t_{x_k}} a_{ij}^{(h)} \cdot \log(a_{ij}^{(h)}) \quad (1)$$

### 2.2 Head Pruning through Self-Attention Attribution

Another line of research focuses on pruning attention heads within Transformers to streamline attention mechanisms without compromising model accuracy. Hao et al. introduces the ATTATTR method, which uses self-attention attribution to determine the importance of attention heads by analyzing token dependencies. This

technique not only facilitates effective head pruning but also constructs "attribution trees" that reveal hierarchical information flow. Experiments on BERT show that ATTATTR can prune less important heads with marginal performance loss, proving its versatility across tasks.

Attention Entropy doesn't attach importance to gradients. In contrast, Attr(i.e. self-attention attribution score(Hao et al.)) takes the gradients of parameters in Transformers into account:

$$Attr_h(A) = A_h \odot \int_{\alpha=0}^1 \frac{\partial F(\alpha A)}{\partial A_h} d\alpha \in R^{n \times n} \quad (2)$$

$$\widetilde{Attr_h}(A) = \frac{A_h}{m} \odot \sum_{k=1}^m \frac{\partial F\left(\frac{k}{m} A\right)}{\partial A_h} \quad (3)$$

$$I_h = E_x [\max(Attr_h(A))] \quad (4)$$

This approach can save important heads when pruning and rewrites the BERT to make attention scores able to invoke their gradients.

### 2.3 Layer Pruning in Large Language Models

K et al. mentioned that the last p% layers can be pruned to make LLM into lightweight encoders. The parameters of last layers in LLM maybe are redundant to be reduced for model compression. Overparameterization in large language models (LLMs) can be mitigated by pruning redundant layers, reducing memory and computational demands. K et al. demonstrates that pruning up to 30% of model layers allows LLMs to retain high performance in text encoding tasks. The L3Prune strategy is particularly effective, identifying layers to prune based on initial loss values and thus minimizing trial-and-error during pruning. This approach is essential for enabling resource-efficient LLM deployment.

### 2.4 TIS Approaches

The HEAT framework leverages Taylor-Expansion Importance Scores (TIS) to assess the importance of each attention head in a Multi-Head Self Attention (MHSA) setting.(Zhong and Zhou) By calculating the impact of setting a parameter to zero using first-order Taylor expansion, TIS provides a way to evaluate the contribution of each head to the model's performance. TIS is derived for each parameter in a head and accumulates across transformer blocks.(Michel et al.)

TIS(i.e. Taylor-expansion Importance Scores(Michel et al.) also relies on gradients.

$$TIS_h = E_x \left| A_h^T \frac{\partial \mathcal{L}(x)}{\partial A_h} \right| \quad (5)$$

## 3 Our method: Gnorm and AE rectification

### 3.1 Gnorm

#### 3.1.1 Definition

We present a novel method to evaluate the importances of heads during the dynamic pruning process: Greedy-Gnorm(Greedy Pruning based on Norms of gradient matrix). Transformers are renowned for attention mechanism. Q, K and V weights help models to interpret the text information like human.(Vaswani et al.) We calculated the gradients of QKV weights. Select the method of element-by-element multiplication to multiply the norm of the corresponding head QKV gradient matrix to get the Gnorm score of each head, and then use the greedy algorithm to cut, select the head with the smallest Gnorm score to cut. Independent variable n means that there are n heads pruned in models. Gnorm score in Greedy-Gnorm method is defined as following:

$$S(n) = G_{Q(n)} \odot G_{K(n)} \odot G_{V(n)} \quad (6)$$

$G_{Q(n)}$  means the expectation of norms matrix of Q weights gradient matrix after pruning n heads.

$$G_{Q(n)} = E(G_{q(n)}) \quad (7)$$

$G_{q(n)}$  is the gradient norms matrix of all heads. It is defined as following:

$$G_{q(n)} = \begin{pmatrix} \|G_{q(n)}^{(11)}\| & \|G_{q(n)}^{(12)}\| & \dots & \|G_{q(n)}^{(1H)}\| \\ \|G_{q(n)}^{(21)}\| & \|G_{q(n)}^{(22)}\| & \dots & \|G_{q(n)}^{(2H)}\| \\ \vdots & \vdots & \ddots & \vdots \\ \|G_{q(n)}^{(L1)}\| & \|G_{q(n)}^{(L2)}\| & \dots & \|G_{q(n)}^{(LH)}\| \end{pmatrix} \in \mathbb{R}^{L \times H} \quad (8)$$

Where  $G_{q(n)}^{(ij)}$  means the gradient matrix of Q weights at  $i^{th}$  layer  $j^{th}$  head. Correspondingly,  $G_{k(n)}^{(ij)}$  and  $G_{v(n)}^{(ij)}$  mean the gradient matrix of K and V weights at in  $i^{th}$  layer  $j^{th}$  head. Based on different inputs in datasets, we need to consider the expectations of the gradient matrix norms and select the head with the smallest Gnorm value. After pruning each head, we all need to find the new Gnorm stages. So S(n) tells that after pruning n heads, which head is the least important to do next pruning. Given the demand of reduce computational complexity, we choose to calculate the L2 norm and  $F_n(X)$  means outputs of the Transformer after pruning n heads:

$$\|G_{q(n)}^{(ij)}\| = \sqrt{\sum_m (\frac{\partial \|F_n(X)\|}{\partial q_m^{(ij)}})^2} \quad (9)$$

$$G_{Q(n)} = E(G_{q(n)}) = \begin{pmatrix} \frac{1}{\|X\|} \sum_{x \in X} \sqrt{\sum_m (\frac{\partial \|F_n(x)\|}{\partial q_m^{(11)}})^2} & \dots \\ \vdots & \ddots & \vdots \\ \dots & \dots & \frac{1}{\|X\|} \sum_{x \in X} \sqrt{\sum_m (\frac{\partial \|F_n(x)\|}{\partial q_m^{(LH)}})^2} \end{pmatrix} \quad (10)$$

### 3.1.2 Greedy Pruning

We give a general pruning scheme to prune the heads of model, including how to prune the heads based on Gnorm and greedy algorithms and how to address the collapsed gradient matrix after pruning some heads.

---

#### Algorithm 1 HOW TO PRUNE HEADS BASED ON GNORM

---

```

1: Input :  $G_{l \times h} = [G_{norm_{ij}}]_{l \times h}$  : Gnorm matrix of model, layer
2: Output : Accuracy scores of pruned models
3: Initialization :  $M_{l \times h} = [m_{ij}]_{l \times h}$  : Mask matrix filled with 1 ( $m_{ij} = 1$ )
4: function PRUNING( $G, M, model$ )
5:   while  $M \neq [0]_{l \times h}$  do
6:      $A \leftarrow (G_{max} - G) \odot M$ 
7:     #Gmax is full of the maximum of G and have the same shape
8:     find the  $a_{ij}$  which is equal to  $\max(A)$ 
9:      $m_{ij} \leftarrow 0$  #Prune the  $j^{th}$  head in the  $i^{th}$  layer of model
10:    end while
11:    return  $model, newM$ 
12: end function
13: function EXPANDGRADMATRIX( $Grad, M$ )
14:   #Grad: Incomplete gradient matrix; M: Masks list in one layer
15:    $E \leftarrow \emptyset$ 
16:    $cursor \leftarrow 0$ 
17:   for all masks in M do
18:     if mask==1 then
19:       put the elements of  $G$  from cursor to cursor +  $d_k$  into  $E$ 
20:        $cursor \leftarrow cursor + d_k$ 
21:     else
22:       put  $O_{1 \times d_k}$  into  $E$  # $O_{1 \times d_k}$  is a zero vector
23:     end if
24:   end for
```

---

```

25:    return  $E$  #  $E$ : Filled gradient matrix
26: end function
27:  $model, M \leftarrow \text{PRUNING}(G, M, model)$ 
28: Calculate accuracy score
29: Get new incomplete gradient matrix Grad after pruning
30: for all layers in model do
31:    $Grad_l \leftarrow \text{EXPANDGRADMATRIX}(Grad, M[layer])$ 
32: end for #concat all  $Grad_l$  into complete gradient matrix Grad for calculating next Gnorm

```

---

**Step 1** Calculate the Gnorm of QKV weights, find the useless head based on Gnorm

**Step 2** Prune the head with smallest Gnorm, the algorithm is given in Algorithm 1. We get the next gradient matrix(Grad).

**Step 3** The collapsed gradient matrix is extended so that it retains its shape, and then we go back to Step 1. With filled gradient matrix, we continue to calculate new Gnorm with gradient matrix(Grad) from **Step 2**.

**Step 4** The shear schemes at the inflection points of the accuracy curve can be taken out separately as pruning schemes for testing and production.

## 3.2 AE rectification

### 3.2.1 $\epsilon$ Fine-tune

Because of too many tokens, attention values are dispersed into small values on each tokens which causes that  $\log(0)$  often appears during calculating Attention Entropy. To solve the underflow issues, we rectify the AE as following:

$$A(a_1, a_2, \dots, a_n) = - \sum_{i=1}^n a_i \log(a_i) \quad (11)$$

$$B(a_1, a_2, \dots, a_n) = - \sum_{i=1}^n a_i \log(a_i + \epsilon) \quad (12)$$

$$C(a_1, a_2, \dots, a_n) = - \sum_{i=1}^n (a_i + \epsilon) \log(a_i + \epsilon) \quad (13)$$

$$C - A = C - B + B - A$$

Where  $\epsilon$  is an extremely small positive value( $\epsilon + a_{ij} < 1$ ). Obviously,  $C - B > 0; B - A < 0$ .  $B - A < 0$  is easy to proof so that we omit it. Given that  $\sum a_{ij} = 1$  and  $\epsilon \ll 1/2 \leq \frac{n-1}{n}$ ,  $C - B > 0$  can be considered as following:

$$\left( \prod_i (a_i + \epsilon) \right)^{\frac{1}{n}} < \frac{n\epsilon + 1}{n} < 1 \quad (14)$$

Here we choose C as rectification of AE, and the reason for this choice is that  $C > B$ . Because when B is less than A and A has underflow, logarithmic item in every term of B will no longer have underflow, but we hope that the result of entropy calculation itself will not underflow, so we choose the larger result of C.

### 3.2.2 Invariant pruning order

$F(a_1, a_2, \dots, a_n) = \sum_{i=1}^n a_i \log(a_i)$  is continuous in  $R^+$ . Given that  $\sum_{i=1}^n x_i = \sum_{i=1}^n y_i = 1$  we discuss following mathematic problems:

Let  $\delta$  be the difference between  $F(P_2)$  and  $F(P_1)$ ,  $\delta > 0$ . (i.e  $\delta = F(P_2) - F(P_1) > 0$ ), then if  $P'_1 = (x_1 + \epsilon, x_2 + \epsilon, \dots, x_n + \epsilon)$  and  $P'_2 = (y_1 + \epsilon, y_2 + \epsilon, \dots, y_n + \epsilon)$ , new difference  $\delta' = F(P'_2) - F(P'_1)$ , is  $\delta'$  also positive?

Actually, let's consider a line segment in n-dimensional space that can be represented as follows:

$$\vec{l} = (l_1, l_2, \dots, l_n) = \frac{\vec{P_1 P_2}}{\|\vec{P_1 P_2}\|} = \frac{\vec{P'_1 P'_2}}{\|\vec{P'_1 P'_2}\|} \quad (15)$$

In fact, point  $P_1$  and point  $P_2$  in n-dimensional space form a line segment in N-dimensional space, and the quality of the length of the line segment is the value of the curve integral. The density of the line is the directional gradient of the function  $F$ , so actually the directional length of the integrated line segment does not change, but the density does. The original difference can be represented as following:

$$\delta = F(P_2) - F(P_1) = \int_{P_1}^{P_2} \nabla F \vec{l} dl > 0 \quad (16)$$

Where  $\nabla F = (\log(a_1) + 1, \log(a_2) + 1, \dots, \log(a_n) + 1)$ :

$$\nabla F \vec{l} = \sum_{i=1}^n l_i + \sum_{i=1}^n l_i \log(a_i) = \sum_{i=1}^n (y_i - x_i) + \sum_{i=1}^n l_i \log(a_i) = \sum_{i=1}^n l_i \log(a_i) \quad (17)$$

Because the function is continuous, when  $\varepsilon$  is small enough, it falls in the small neighborhood and the gradient sign remains unchanged:

$$\nabla F' \vec{l} = \sum_{i=1}^n l_i \log(a_i + \varepsilon) \in N_r(\nabla F \vec{l}) \quad (18)$$

$$s.t. \delta' = F(P'_2) - F(P'_1) = \int_{P'_1}^{P'_2} \nabla F' \vec{l} dl > 0 \quad (19)$$

Here  $r$  can control the change of the integral value in a small range, so that the sign of the integral value remains unchanged. Then the size relationship of attention entropy remains unchanged, and thus ensure that our rectification can be smoothly clipped without changing the original clipping order, finally solving the underflow problem of AE.

## 4 Experiments

### 4.1 Experimental Setup

To keep the comparison as meaningful as possible, we choose four different kinds of models which are respectively BERT, ALBERT, RoBERTa and XLM-RoBERTa. Their tasks focus on financial-sentiment-analysis, mnli, tweets-sentiment-analysis and language-identification. With textpruner as our analysis tools(Yang et al.), inverse pruning is used to validate the effectiveness of Gnorm. However, ALBERT's structure is single-layer reused for 12 times so that the masks only have 12 to control 144 heads. Fig. 2 shows the example for pruning process with merely 3 mask.

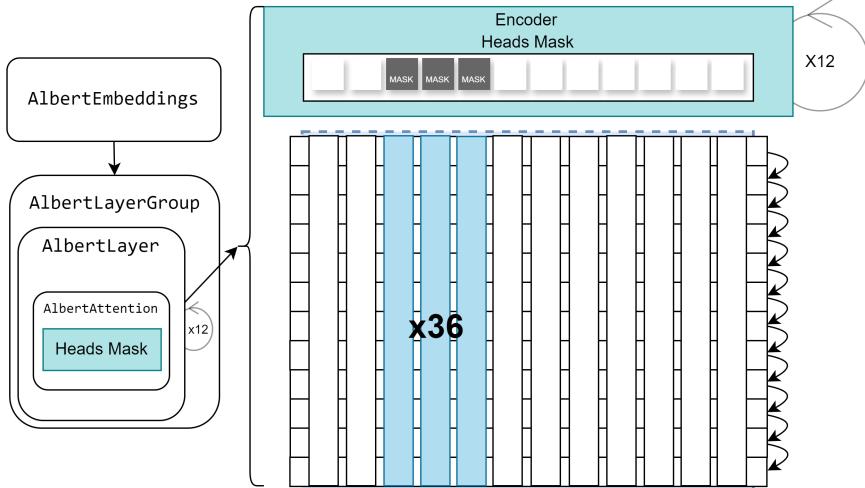


Figure 2: One example: Special structure of ALBERT during pruning based on only 3 mask

**Datasets** Applied datasets to test changes of accuracies are “hw2942/financial-news-sentiment”, “nyu-mll/glue”, “siebert/sentiment-roberta-large-english” and “papluca/language-identification”.

**Models** During doing experiments, we have to pay attention to the special parameters sharing mechanism in ALBERT. Each mask in ALBERT means pruning 12 heads in 12 layer at the same position.

Selected models includes:

- BERT “hw2942/bert-base-chinese-finetuning-financial-news-sentiment-v2”;
- ALBERT “Alireza1044/albert-base-v2-mnli”;
- RoBERTa “rahmaabusalma/tweets\_sentiment\_analysis”;
- XLM-RoBERTa “papluca/xlm-roberta-base-language-detection”.

## 4.2 Comparison Experiments

We compare the four kinds of prunings: AE Pruning, Inverse AE Pruning, Greedy Gnorm Pruning, Inverse Greedy Gnorm Pruning. Greedy-Gnorm performs excellently when pruning BERT, ALBERT and XLM-RoBERTa. Unlike BERT popular to prune(Parnami et al.), RoBERTa(larger model) has 384 heads, Fig. 3 shows that useful heads take up about 30%. After inverse pruning of Greedy Gnorm, when 150 heads are pruned, the accuracy remains constant level. For four curves of Inverse Greedy Gnorm Pruning results in four models, about 30% heads are really useful, playing an importance role for text understanding. After pruning these heads, the models can be regarded as untrained models which are disable for their tasks.

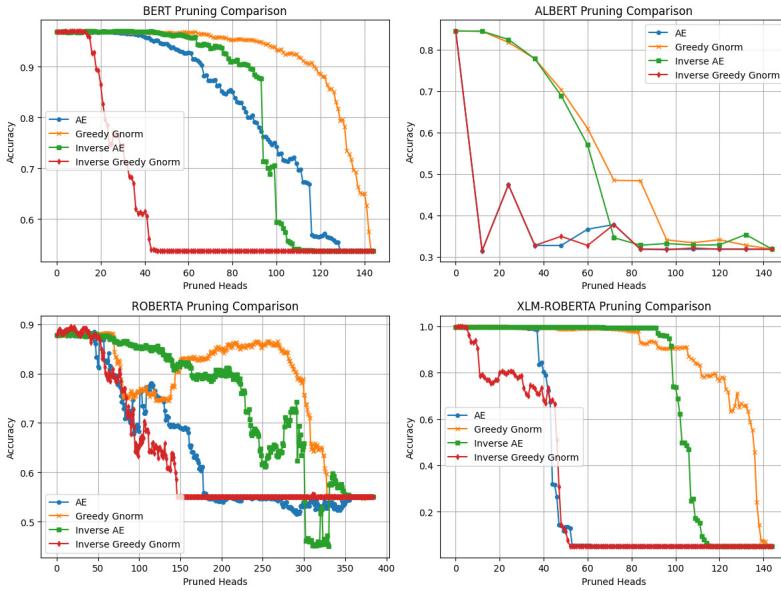


Figure 3: Greedy-Gnorm and AE Pruning Comparison

We adopted Heads Gates method to prune the heads in four models.(Shim et al.) The heads mask component is set up during concatenating the different outputs of each head. So the heads gates will be closed one by one as greedy algorithms work for pruning. The gates which first closed is the least important based on Gnorm. Left heads at last always have great considerable gradient to affect the outputs for model.

Fig. 3 shows that the Greedy-Gnorm pruning can outstand AE pruning method. Especially for BERT, despite the left heads less than 20%, the accuracy can still maintain 90%. At least, it means that Greedy-Gnorm can tell which head is really important for models.

## 4.3 Random Prunings

The Fig. 4 demonstrates the effectiveness of the Greedy Gnorm pruning strategy across various transformer models, including BERT, ALBERT, RoBERTa, and XLM-RoBERTa. In each sub-figure, the accuracy is plotted against the percentage of heads pruned, comparing the performance of Greedy Gnorm with a random pruning strategy.

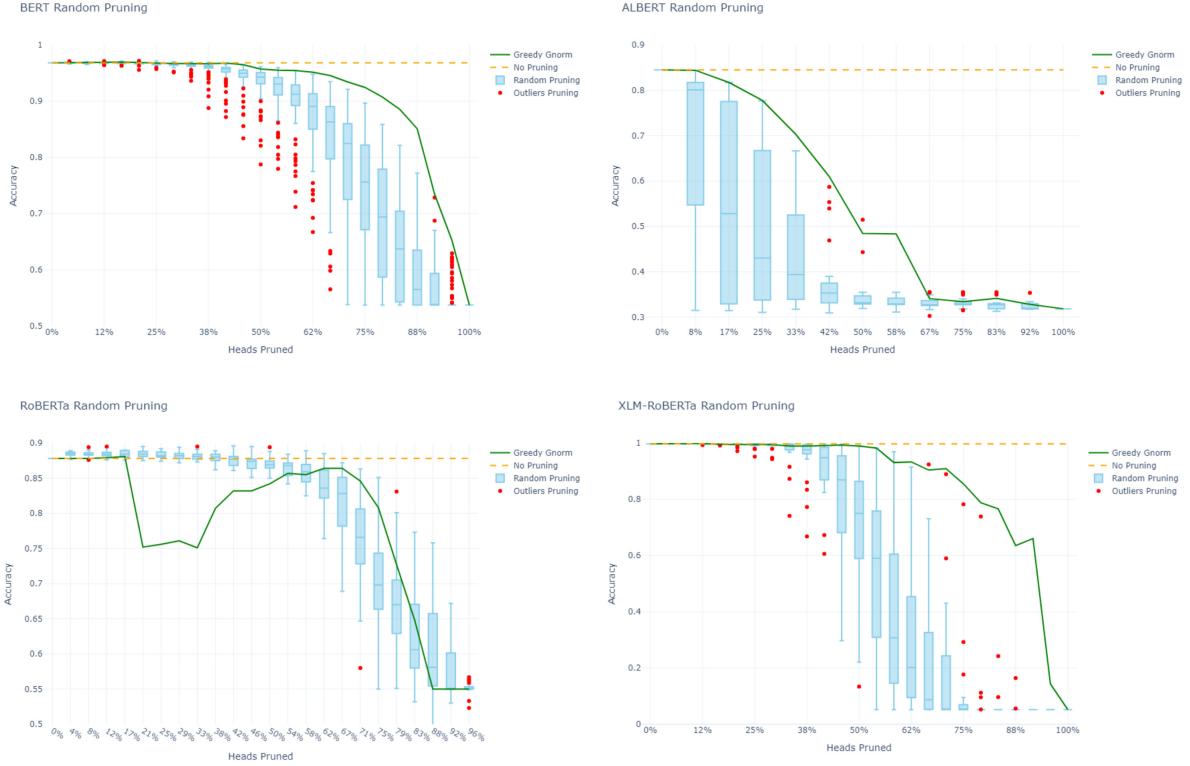


Figure 4: The effectiveness of Greedy-Gnorm among random prunings

Greedy Gnorm shows a clear advantage over random pruning, as evidenced by the more gradual decline in accuracy. The green line representing Greedy Gnorm maintains higher accuracy levels across all pruning stages compared to the random pruning box plots, which display significant variability. Notably, random pruning results in numerous outliers, indicated by the red dots, which reflect the instability and inconsistent accuracy caused by randomly removing attention heads. These outliers become increasingly prominent as the pruning percentage grows, underscoring the limitations of an unstructured pruning approach.

In contrast, Greedy Gnorm’s structured pruning approach effectively preserves critical attention heads, resulting in a controlled and steady decline in accuracy. Even as the pruning percentage approaches high levels (e.g., 75% or more), Greedy Gnorm retains a substantial portion of the model’s original accuracy, whereas random pruning leads to a rapid and often erratic drop in performance.

The robustness of Greedy Gnorm across different models confirms its generalizability and effectiveness in selectively pruning heads without severely impacting model performance. This experiment validates that Greedy Gnorm provides a reliable and efficient pruning strategy, offering a significant improvement over random pruning by preserving model accuracy under substantial parameter reduction

#### 4.4 Pruning solutions

Considering that four models have different conditions in parameters or structures, we respectively give their pruning solutions. After Step 4 in our method, we select the inflection points during four comparison experiments about Greedy-Gnorm.

**BERT** The pruning solution for BERT reveals a selective pruning strategy. The retained heads are scattered across all 12 layers, indicating that attention heads in varying depths contribute differently to model performance. This solution suggests the presence of specialized functions performed by different heads at various layers, highlighting the flexibility of BERT’s architecture.

**ALBERT** In contrast, the pruning result for ALBERT displays a uniform pattern with clear vertical bands of retained heads. This observation indicates that specific attention heads consistently hold higher importance across all layers, while others are consistently pruned. This result could be due to ALBERT’s parameter-sharing mechanism, which enforces identical attention configurations across its layers, promoting the uniformity of the

solution.

**RoBERTa** The heatmap for RoBERTa reveals a more complex pruning pattern, where retained heads are distributed irregularly across 24 layers. This irregularity suggests that RoBERTa’s attention heads serve a diverse range of functions across layers, with some layers relying on a higher number of unpruned heads. The intricate nature of this pruning pattern reflects the robustness of RoBERTa’s design, where more heads are retained to support nuanced attention mechanisms.

**XLM-RoBERTa** For XLM-RoBERTa, the heatmap indicates a selective pruning scheme that retains attention heads across layers in a dispersed manner. While the distribution of retained heads appears more balanced than in RoBERTa, there is a focus on preserving certain heads within specific layers, suggesting these heads may play crucial roles in multilingual tasks or cross-lingual transfer learning.

Fig. 5 shows our final chosen solutions for four models. After pruning heads, their accuracies are kept as following: 90.08%, 77.76%, 86.40% and 90.97%.

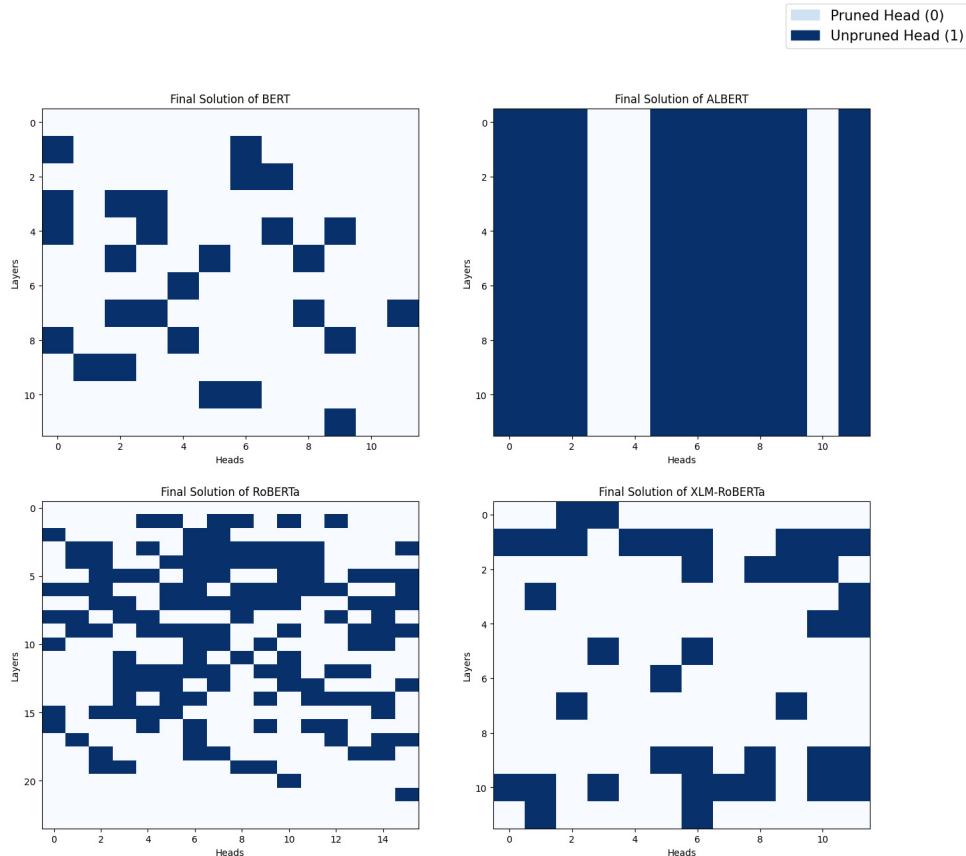


Figure 5: The final pruning solutions for four kinds of model

Table 1: The solutions of Greedy-Gnorm for four models(BP means “Before pruning”, AP means “After pruning”)

	BERT	ALBERT	RoBERTa	XLM-RoBERTa
Accuracy(BP)	96.82%	84.48%	87.80%	99.73%
Accuracy(AP)	90.08%	77.76%	86.40%	90.97%
Size(BP)	390.13	44.58	1355.60	1060.71
Size(AP)	302.29	42.33	1110.42	981.88

It seems that the amount of pruned parameters is not obvious, but actually when scale of parameters reaches several billions, Greedy-Gnorm can prune more than half of heads in LLM, which will compress the models with considerable reductions.

Here we take the BERT as example to observe the differences of all parts in model after pruning.

Table 2: The parameters condition of original BERT and pruned BERT(BP means “Before pruning”, AP means “After pruning”)

Layer Name	Params(BP)	Ratio(BP)	MB(BP)	Params(AP)	Ratio(AP)	MB(AP)
Model	102,269,955	100.00%	390.13	79,244,355	100.00%	302.29
Bert	102,267,648	100.00%	390.12	79,242,048	100.00%	302.28
Embeddings	16,622,592	16.25%	63.41	16,622,592	20.98%	63.41
Encoder	85,054,464	83.17%	324.46	62,028,864	78.28%	236.62
Pooler	590,592	0.58%	2.25	590,592	0.75%	2.25
Classifier	2,307	0.00%	0.01	2,307	0.00%	0.01
Weight	2,304	0.00%	0.01	2,304	0.00%	0.01
Bias	3	0.00%	0.00	3	0.00%	0.00
Accuracy		0.9682			0.9008	

## 5 Conclusion

In this study, we proposed Greedy-Gnorm as a new head pruning method for transformer-based models, leveraging the norms of gradient matrices to select and prune less essential attention heads. By replacing Attention Entropy with Gnorm, our approach successfully mitigates underflow issues and provides a more stable pruning mechanism. The results indicate that Greedy-Gnorm outperforms existing pruning methods in terms of maintaining accuracy, even after substantial model compression. Future work could explore extending Greedy-Gnorm to additional transformer models like Electra(Clark et al.), BART(Lewis et al.), T5(Raffel et al.) and mT5(Xue et al.), as well as investigating potential improvements in gradient-based pruning strategies. The findings suggest that Greedy-Gnorm holds promise for efficient and reliable transformer model compression, enhancing deployment feasibility in resource-constrained environments.

## Bibliography

- [1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL <https://arxiv.org/abs/2005.14165>.
- [2] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555, 2020. URL <https://arxiv.org/abs/2003.10555>.
- [3] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019. URL <http://arxiv.org/abs/1911.02116>.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL <http://arxiv.org/abs/1810.04805>.
- [5] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. *CoRR*, abs/2004.11207, 2020. URL <https://arxiv.org/abs/2004.11207>.
- [6] Thennal D K, Tim Fischer, and Chris Biemann. Large language models are overparameterized text encoders, 2024. URL <https://arxiv.org/abs/2410.14578>.

- [7] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019. URL <http://arxiv.org/abs/1909.11942>.
- [8] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019. URL <http://arxiv.org/abs/1910.13461>.
- [9] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL <http://arxiv.org/abs/1907.11692>.
- [10] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *CoRR*, abs/1905.10650, 2019. URL <http://arxiv.org/abs/1905.10650>.
- [11] Archit Parnami, Rahul Singh, and Tarun Joshi. Pruning attention heads of transformer models using a\* search: A novel approach to compress big NLP architectures. *CoRR*, abs/2110.15225, 2021. URL <https://arxiv.org/abs/2110.15225>.
- [12] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019. URL <http://arxiv.org/abs/1910.10683>.
- [13] Kyuhong Shim, Iksoo Choi, Wonyong Sung, and Jungwook Choi. Layer-wise pruning of transformer attention heads for efficient language modeling. *CoRR*, abs/2110.03252, 2021. URL <https://arxiv.org/abs/2110.03252>.
- [14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL <http://arxiv.org/abs/1706.03762>.
- [15] Zijie J. Wang, Robert Turko, and Duen Horng Chau. Dodrio: Exploring transformer models with interactive visualization. *CoRR*, abs/2103.14625, 2021. URL <https://arxiv.org/abs/2103.14625>.
- [16] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934, 2020. URL <https://arxiv.org/abs/2010.11934>.
- [17] Ziqing Yang, Yiming Cui, and Zhigang Chen. TextPruner: A model pruning toolkit for pre-trained language models. In Valerio Basile, Zornitsa Kozareva, and Sanja Stajner, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 35–43, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-demo.4. URL <https://aclanthology.org/2022.acl-demo.4>.
- [18] Yibo Zhong and Yao Zhou. Heat: Head-level parameter efficient adaptation of vision transformers with taylor-expansion importance scores, 2024. URL <https://arxiv.org/abs/2404.08894>.