# Optimised Grouped-Query Attention Mechanism for Transformers

**Anonymous Authors**[1]


## Abstract

Grouped-query attention (GQA) has been widely adopted in LLMs to mitigate the complexity of multi-head attention (MHA). To transform an MHA to a GQA, neighbour queries in MHA are evenly split into groups where each group shares the value and key layers. In this work, we propose AsymGQA, an activation-informed approach to asymmetrically grouping an MHA to a GQA for better model performance. Our AsymGQA outperforms the GQA within the same model size budget. For example, AsymGQA LLaMA-2-7B has an accuracy increase of 7.5% on MMLU compared to neighbour grouping. Our approach addresses the GQA's trade-off problem between model performance and hardware efficiency.


## 1. Introduction

Transformer-based models have achieved remarkable success on large-scale language tasks (Devlin et al., 2019; Vaswani et al., 2023; Brown et al., 2020). Multi-head attention (MHA), the core operation of the Transformer, allows the model to attend to information from different representation subspaces at different positions. However, computational and memory complexity increases quadratically with the sequence length in MHA. To mitigate this problem, researchers have introduced grouped-query attention (GQA) (Ainslie et al., 2023), which evenly splits query heads into groups, and each group shares a single key and value layer (see GQA in Figure 1).

GQA trades model quality for hardware efficiency. By operating on groups rather than individual queries, GQA reduces the computation cost in Transformer and consumes less memory. Typically, GQA models are created by design – they are originally designed and trained explicitly as GQA models. In this study, we ***investigate the challenges of converting MHA into a GQA***, essentially treating GQA as an post-training optimization for the efficient deployment of LLMs. The naive merging is then take the average of all key and value layers in a group. Such a simple merging views every head in MHA equally, but our experiments show that this can cause a significant quality degradation even after fine-tuning. To mitigate the quality degradation, we propose asymmetric GQA (AsymGQA), an activation-informed merging approach that considers similarity between layers. Specifically, our contributions are as follows:

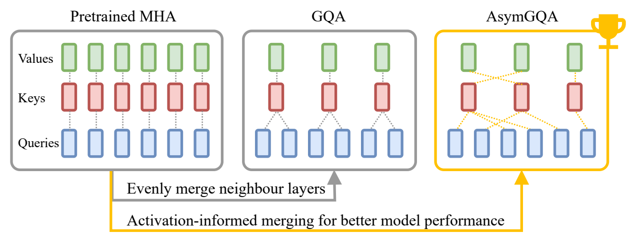



*Figure 1.* Comparison of GQA and AsymGQA. AsymGQA leverages activation-induced layer similarity to determine the attention head grouping for better model performance.

- We introduce AsymGQA, an activation-informed fusion approach for converting MHA into GQA models, which delivers superior model performance within the same computational constraints.
- Through extensive experiments, we answer two unexplored GQA questions. We first verify that our activation-induced method improves the performance for evenly grouped GQA models. Furthermore, we find that more performance gain could be achieved if asymmetric grouping (varying group size) is allowed.

AsymGQA models significantly outperform the GQA baseline. For example, LLaMA-2-7B with an average group size of 4 has an increase in accuracy of 7.2% on MMLU compared to naive MHA to GQA conversion.

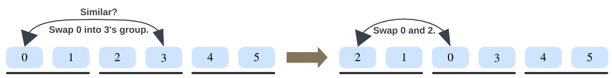
## 2. Method

Section 2.1 introduces our search-based grouping methods. Section 2.2 elaborates how we calculate the similarity of key (value) layers to guide the search.

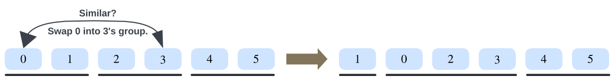
[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.


(a) Neighbour grouping (baseline).




(b) Activation-informed symmetric grouping.




(c) Activation-informed asymmetric grouping.

*Figure 2.* Naive neighbour grouping vs AsymGQA.

### 2.1. Grouping Strategies

**Neighbour grouping** We propose a naive scheme called neighbour grouping. Neighbour grouping clusters adjacent attention heads (key layers and value layers) together, while keeping all the groups equally-sized. Figure 2a provides a visualization of neighbour grouping. This serves as the baseline in this work.

**Activation-informed symmetric grouping** We apply the grouping sequentially from the initial MHA layer to the final one. In each MHA layer, the key and value layers are grouped independently. Our proposed method employs a search strategy to determine the optimal grouping of key (and value) layers based on the similarity among them within MHA, as detailed in Section 2.2.

As illustrated in Figure 2b, before the search starts, a key layer grouping $g$ of size $m$ is generated randomly. In each search iteration, $g$ may be updated so that similar heads are swapped into the same group. Specifically, in one iteration, a key layer $a$ is randomly sampled ($a$'s group is noted as $A$), then the similarity between $a$ and all the other key layers is calculated. Among the top-$k$ most similar layers not belonging to group $A$, a key layer $b$ is selected ($b$'s group noted as $B$). Another head $b'$ in group $B$ is sampled to exchange with $a$, to maintain the group size. If the swapped model has a higher accuracy, the best accuracy and grouping are updated. We also introduce random noise to the search to explore more grouping possibilities. An acceptance probability is set to update $g$ even if swapped model has a lower accuracy. Furthermore, another probability is specified that resets $g$ at the start of each iteration to prevent the search from being trapped in a local minima. The detailed algorithm is defined in Algorithm 1 in Appendix A.

**Activation-informed asymmetric grouping (AsymGQA)** Compared to symmetric grouping, asymmetric grouping allows for varied group sizes. Asymmetric grouping is particularly promising in scenarios where the relevance of information is not uniformly distributed across the input space. Asymmetric grouping also extends the search space from only containing equal-sized groups to any groupings, providing opportunities to find even better grouping configuration than symmetric grouping (Figure 2c).

Support for asymmetric grouping requires only minor changes to the search algorithm. In symmetric grouping, to maintain the idenitical group sizes, whenever an key or value layer is moved to a new group, an element in the new group must be swapped back. To allow for groups with varying sizes, another parameter $p_{preserve}$ is introduced, representing the probability of preserving the group sizes after swap. A larger $p_{preserve}$ encourages a more unbalanced grouping. A detailed description is included in Algorithm 2 in the Appendix Appendix A.

### 2.2. Activation-Informed Head Similarity

As stated in Section 2.1, our search calculates the similarity between key (value) layers at the beginning of each iteration to guide the search. We *cluster similar layers into the same group*, which is advantageous for two reasons:

- It leads to better optimization and generalization. As similar layers contribute to a focused gradient update that is consistent across the group, the error signal propagated back through the network can more effectively tune the shared parameters, enhancing the stability and efficiency of learning.
- When attention heads share similar and value layers, they are likely to encode and focus on comparable aspects of the input data. This similarity in processing enables more coherent feature extraction, as these heads reinforce each other's understanding and interpretation of specific data patterns, leading to a more nuanced and detailed representation within that particular subspace of the feature space.

We measure the similarity between pairs of key (or value) layers using two potential approaches:

1. Define the similarity using the difference between the weights of two layers.
2. Define the similarity using the difference between the output activations of two layers.

Based on our experiments (See Appendix B), we find that activation-informed similarity is a better reference to guide the search. We use consine similarity between vectors to define activation-informed similarity between two layers.

*Table 1.* Comparison between AsymGQA and GQA of LLaMA-2-7B. The column "GQA" means the standard GQA which uses neighbour grouping to merge heads. Results of full fine-tuning ("Full FT") and parameter-efficient LoRA fine-tuning ("LoRA") after grouping are both included. The highest accuracy is highlighted in **bold**. Ech cell has two values, Acc ($\Delta$), wehre Acc is accuracy, and $\Delta$ denotes the accuracy difference from fine-tuned MHA, *i.e.*, group size equals 1. We observe that AsymGQA consistently achieves higher accuracy than GQA by a clear margin and close to fine-tuned MHA.

| Task | Group size | Full FT | | LoRA | |
|---|---|---|---|---|---|
| | | GQA | AsymGQA | GQA | AsymGQA |
| SST2 | 2 | 90.9±0.5 (-2.4) | **92.9±0.2 (-0.4)** | 90.6±0.5 (-2.6) | **92.6±0.1 (-0.6)** |
| | 3 | 88.4±0.3 (-4.9) | **92.0±0.4 (-1.3)** | 88.0±0.6 (-5.2) | **91.8±0.3 (-1.4)** |
| | 4 | 87.3±0.6 (-5.8) | **90.4±0.2 (-2.9)** | 87.1±0.3 (-6.1) | **90.1±0.1 (-3.1)** |
| | 6 | 86.8±0.6 (-6.5) | **89.6±0.5 (-3.7)** | 86.2±0.3 (-7.0) | **89.5±0.1 (-3.7)** |
| QNLI | 2 | 83.7±0.2 (-6.4) | **89.5±0.6 (-0.6)** | 82.9±0.2 (-6.7) | **89.0±0.3 (-0.6)** |
| | 3 | 80.5±0.6 (-9.6) | **88.6±0.4 (-1.5)** | 80.3±0.5 (-9.3) | **88.3±0.5 (-1.3)** |
| | 4 | 74.6±0.2 (-15.5) | **86.5±0.7 (-3.6)** | 73.2±0.2 (-16.3) | **84.9±0.7 (-3.7)** |
| | 6 | 73.2±0.6 (-16.9) | **85.0±0.3 (-5.1)** | 72.0±0.2 (-17.6) | **84.5±0.6 (-5.1)** |
| MNLI | 2 | 79.7±0.4 (-1.5) | **81.0±0.1 (-0.2)** | 78.7±0.6 (-2.7) | **80.8±0.6 (-0.6)** |
| | 3 | 77.7±0.6 (-3.5) | **80.3±0.6 (-0.9)** | 77.0±0.6 (-4.4) | **80.2±0.2 (-1.2)** |
| | 4 | 75.1±0.5 (-6.2) | **78.4±0.5 (-2.8)** | 74.4±0.3 (-7.0) | **78.0±0.4 (-3.3)** |
| | 6 | 72.7±0.3 (-8.6) | **77.3±0.1 (-3.9)** | 72.0±0.4 (-9.3) | **77.2±0.2 (-4.1)** |
| MMLU | 2 | 34.5±0.5 (-5.3) | **39.3±0.7 (-0.5)** | 33.7±0.3 (-6.7) | **39.6±0.5 (-0.8)** |
| | 3 | 33.0±0.1 (-6.7) | **38.3±0.7 (-1.5)** | 32.7±0.2 (-7.7) | **39.0±0.3 (-1.4)** |
| | 4 | 29.0±0.4 (-10.8) | **36.5±0.2 (-3.3)** | 28.5±0.6 (-11.9) | **36.8±0.3 (-3.6)** |
| | 6 | 26.4±0.2 (-13.4) | **33.8±0.2 (-6.0)** | 26.3±0.1 (-14.1) | **34.7±0.3 (-5.7)** |

The similarity $\text{sim}(A, B)$ between two activation matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{n\times m}$ using row vectors is as follows.

$$\begin{aligned}\text{sim}(A,B) = \frac{1}{2}(&\sum_{i=0}^{n-1}\max_{j=0}^{n-1}\text{cosim}(A_{i,*},B_{j,*}) \\ &+\sum_{i=0}^{n-1}\max_{j=0}^{n-1}\text{cosim}(B_{i,*},A_{j,*}))\end{aligned} \quad (1)$$

where $\text{cosim}(\cdot)$ calculates the cosine similarity between two vectors $\mathbf{u}$ and $\mathbf{v}$:

$$\text{cosim}(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \quad (2)$$

# 3. Evaluation

Section 3.1 introduces our basic experiment setup. Section 3.2 presents our main results on activation-informed asymmetric grouping, which is the best grouping strategy out of the three introduced in Section 2.1. Section 3.3 includes two ablation studies to verify the efficacy of activation-informed grouping and varied group sizes.

## 3.1. Experiment Setup

**Models and datasets** We apply our methods to popular decoder-only models including OPT (Zhang et al., 2022), LLaMA (Touvron et al., 2023a) and LLaMA-2 (Touvron et al., 2023b) with the number of parameters ranging from 125 million to 7 billion. We evaluated these models on QNLI (Wang et al., 2018), MNLI (Williams et al., 2017), SST2(Socher et al., 2013), and MMLU (Hendrycks et al., 2020). The entire MMLU is evaluated under a zero-shot setting. Each experiment has three independent runs with different random seeds. The mean and standard deviation of three runs are calculated.

**Grouping** For each layer, 10 search iterations are executed with different groupings, with an acceptance probability of $0.1$, a reset probability of $0.1$ and a preservation probability of $0.2$. In each iteration, we randomly choose one of the top 3 closest heads to group.

**Fine-tuning** We fine-tune the grouped model for three more epochs to recover the model performance. We include both full fine-tuning and LoRA (Hu et al., 2021) fine-tuning in results. Besides, we use a beam search to find optimal fine-tuning hyperparameters, including batch size, learning rate and weight decay of AdamW optimizer (Loshchilov & Hutter, 2017). Detailed hyperparameters for each dataset can be found in Appendix C.

## 3.2. Remarkable Performance Gain of AsymGQA

Table 1 presents the accuracy of the grouped LLaMA-2-7B with various average group sizes. We compare AsymGQA to neighbor grouping ("NG"), which evenly groups neighbor key and value layers. Both full fine-tuning results, noted as "FT", and LoRA fine-tuning results, noted as "LoRA", are included in the table. Another result table of OPT-1.3B can be found in Table 4 in Appendix D. We have the following observations.

- AsymGQA achieves *consistently higher accuracy than the baseline by a clear margin*, across group sizes and fine-tuning methods. Among these reulsts, the maximum enhancement of accuracy is up to 12.5%.
- This margin (accuracy enhancement) is more obvious on more challenging tasks such as MMLU. For example, full fine-tuned AsymGQA has an average accuracy increase of 6.3% compared to GQA on MMLU, while on the less challenging SST2, this margin is 2.8%.

We also inspect the trade-off tuned by group size, *i.e.*, trading model quality for hardware efficiency. Figure 3 illustrates how the number of parameters and inference FLOPs decreases as the group size increases. The hardware efficiency has diminishing returns as we increase the group size. Therefore, for the grouping problem in Table 1, a group size of 2 or 3 may be ideal for real-world applications.

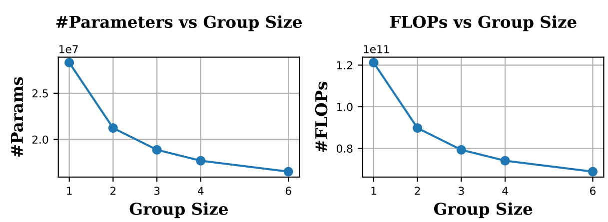



*Figure 3.* #Parameters and floating-point operations (FLOPs) vs group size of attention layer. We see a diminishing hardware efficiency return as the group size increases.

### 3.3. Ablation Study

The first ablation study, SG vs NG, shows that even with a uniform group size, our activation-informed grouping (symmetric grouping, noted as "SG") still improves the model performance compared to neighbour grouping (NG). The second ablation study shows the performance gain of asymmetric grouping (AG) compared to symmetric grouping.

**SG vs NG** Figure 4 compares activation-informed symmetric grouping (SG) to neighbour grouping (NG), indicating that the activation-induced grouping search contributes to performance gain.

**AG vs SG** Figure 5 compares asymmetric grouping (AG) to symmetric grouping (SG), highlighting that varied group size further improves model performance.

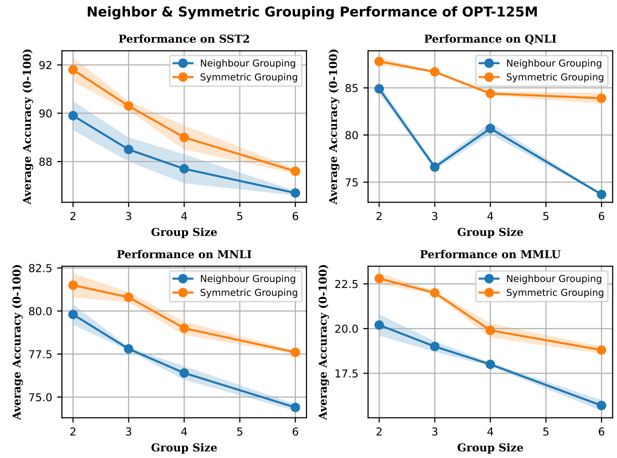



*Figure 4.* Neighbour grouping vs activation-informed symmetric grouping. Activation-induced similarty between key (value) layers improves model performance even without varied group sizes.

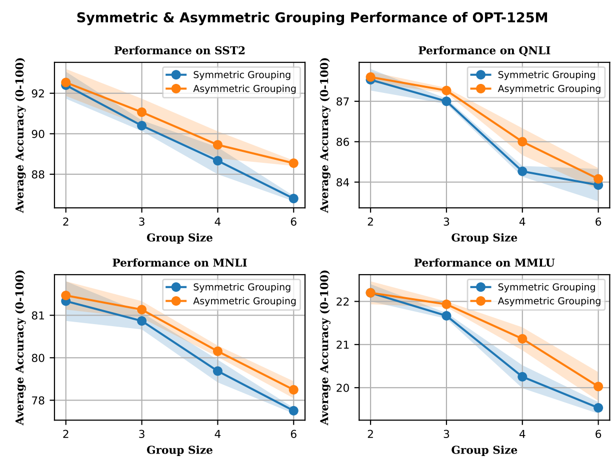



*Figure 5.* Symmetric vs Asymmetric grouping. Asymmetric further improves model performance by allowing varied group sizes.

## 4. Conclusion

We introduce AsymGQA, an activation-guided asymmetric grouping strategy for transforming a pretrained MHA model into a GQA model. AsymGQA significantly outperforms otherweight-merging baseline, and it effectively manages the trade-off between model performance and hardware efficiency in GQA.

## References


Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300, 2020. URL https://arxiv.org/abs/2009.03300.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Linzen, T., Chrupała, G., and Alishahi, A. (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., and Zettlemoyer, L. Opt: Open pre-trained transformer language models, 2022.

**Algorithm 1** Activation-Informed Symmetric Grouping Search

**Input:**
model = pretrained MHA model
$n$ = number of search iterations
$m$ = group size
$k$ = number of similar attention heads to consider when altering groupings
$p_{\text{acc}}$ = probability of accepting a worse-performing group
$p_{\text{reset}}$ = probability of resetting the current grouping at each iteration

**function** SYMMETRIC_GROUPING (model, $n$, $m$, $k$, $p_{\text{acc}}$, $p_{\text{reset}}$):
$S \leftarrow \{h | h \text{ is an attention head in model}\}$
$g \leftarrow \{P[i : i+m] | i = 0, m, 2m, \dots\}$, where $P \sim$ Uniform(permutation(S)) // A random grouping of size $m$.
best_acc $\leftarrow 0$ // Keep track of best accuracy.
best_grouping $\leftarrow$ None // Keep track of best grouping.
**for** $i = 1$ **to** $n - 1$ **do**
$x \leftarrow X \sim \text{Uniform}(0, 1)$
**if** $x < p_{\text{reset}}$ **then**
$g \leftarrow \{P[i : i+m] | i = 0, m, 2m, \dots\}$, where $P \sim$ Uniform(permutation(S)) // Reset $g$ with probability $p_{\text{reset}}$.
**end if**
$a \leftarrow A \sim \text{Uniform}(S)$ // A randomly chosen attention head.
$H \leftarrow \text{top-}k(S, a)$ // Top $k$ closest heads in a different group from the group of $a$.
$b \leftarrow B \sim \text{Uniform}(H)$ // A randomly chosen attention head from $H$.
$g' \leftarrow$ swap $a$ with an element from the group of $b$ in $g$ // Move head $a$ to be grouped with a head similar to $a$, while keeping the equal sizes of all groups.
$y \leftarrow Y \sim \text{Uniform}(0, 1)$
**if** $\text{acc}(\text{GQA}(g')) >$ best_acc **or** $y < p_{\text{acc}}$ **then**
$g' \leftarrow g$
**if** $\text{acc}(\text{GQA}(g')) >$ best_acc **then**
best_acc $\leftarrow \text{acc}(\text{GQA}(g'))$ // Update best accuracy.
best_grouping $\leftarrow g'$ // Update best grouping.
**end if**
**end if**
**end for**
**return** best_acc, best_grouping
**end function**

## A. Activation-Informed Grouping Search

Algorithm 1 and Algorithm 2 describe *activation-informed symmetric grouping* and *activation-informed asymmetric grouping* respectively.

**Algorithm 2** Activation-Informed Asymmetric Grouping Search

**Input:**
- model = pretrained MHA model
- $n$ = number of search iterations
- $m$ = group size
- $k$ = number of similar attention heads to consider when altering groupings
- $p_{\text{acc}}$ = probability of accepting a worse-performing group
- $p_{\text{reset}}$ = probability of resetting the current grouping at each iteration
- $p_{\text{preserve}}$ = probability of preserving the difference of group sizes when moving a head from one group to another

```
function SYMMETRIC_GROUPING (model, n, m, k, p_acc, p_reset, p_preserve):
  S ← {h|h is an attention head in model}
  g ← {P[i : i + m]|i = 0, m, 2m, ...}, where P ~ Uniform(permutation(S)) // A random grouping of size m.
  best_acc ← 0 // Best accuracy.
  best_grouping ← None // Keep track of the best-performing grouping during search.
  for i = 0, ..., n − 1 do
    x ← X ~ Uniform(0, 1) // A random real number from [0,1].
    if x < p_reset then
      g ← {P[i : i + m]|i = 0, m, 2m, ...}, where P ~ Uniform(permutation(S)) // Reset g with probability p_reset.
    end if
    a ← A ~ Uniform(S) // A randomly chosen attention head.
    H ← top-k(S, a) // Top k closest heads in a different group from the group of a.
    b ← B ~ Uniform(H) // A randomly chosen attention head from H.
    y ← Y ~ Uniform(0, 1)
    if y < p_preserve then
      g′ ← move a head from the group of b back to the group of a in g // Preserve the difference of sizes of two groups.
    end if
    g′ ← move a into the group of b in g
    z ← Z ~ Uniform(0, 1)
    if acc(GQA(g′)) > best_acc or z < p_acc then
      g′ ← g // Update g with g′.
      if acc(GQA(g′)) > best_acc then
        best_acc ← acc(GQA(g′)) // Update the best accuracy.
        best_grouping ← g′ // Update the best-performing grouping.
      end if
    end if
  end for
  return best_acc, best_grouping
end function
```

## B. Weight-Informed Similarity vs Activation-Informed Similarity

Before diving into Asymmetric search, we perform an experiment to compare the weight-informed layer similarity and the activation-informed similarity. As shown in Table 2, activation-informed search usually has the best performance.

*Table 2.* The model performance of brute force search, weight-informed symmetric search, and activation-informed simmetric search. The best is highlighted in bold. Model here is OPT-125M. We find that activation-informed layer similarity usually has the best performance.

| Group Size | Brute Force Search | Weight-Informed | Activation-Informed |
|---|---|---|---|
| ACC ($\Delta$) on SST2 | | | |
| 1 | 92.1±0.1 (0.0) | 92.1±0.1 (0.0) | 92.1±0.1 (0.0) |
| 2 | 91.6±0.3 (-0.5) | 91.2±0.6 (-0.9) | **91.8±0.5 (-0.3)** |
| 3 | 90.1±0.5 (-2.0) | 90.0±0.1 (-2.1) | **90.3±0.2 (-1.8)** |
| 4 | 88.9±0.1 (-3.2) | **89.1±0.4 (-3.0)** | 89.0±0.5 (-3.1) |
| 6 | 87.5±0.1 (-4.6) | **87.8±0.4 (-4.3)** | 87.6±0.1 (-4.5) |
| ACC ($\Delta$) on QNLI | | | |
| 1 | 88.1±0.4 (0.0) | 88.1±0.3 (0.0) | 88.1±0.3 (0.0) |
| 2 | 87.4±0.6 (-0.7) | 87.7±0.1 (-0.4) | **87.8±0.4 (-0.3)** |
| 3 | 86.6±0.4 (-1.5) | 85.6±0.1 (-2.5) | **86.7±0.1 (-1.4)** |
| 4 | 84.0±0.1 (-4.1) | **84.8±0.4 (-3.3)** | 84.4±0.2 (-3.7) |
| 6 | 82.5±0.2 (-5.6) | 83.4±0.1 (-4.7) | **83.9±0.6 (-4.2)** |
| ACC ($\Delta$) on MNLI | | | |
| 1 | 82.1±0.2 (0.0) | 82.1±0.1 (0.0) | 82.1±0.1 (0.0) |
| 2 | **81.4±0.2 (-0.7)** | **81.4±0.1 (-0.7)** | **81.5±0.7 (-0.6)** |
| 3 | 80.3±0.1 (-1.8) | 80.4±0.3 (-1.7) | **80.8±0.3 (-1.3)** |
| 4 | 78.5±0.5 (-3.6) | 78.4±0.2 (-3.7) | **79.0±0.4 (-3.1)** |
| 6 | 77.6±0.6 (-4.5) | **77.8±0.5 (-4.4)** | 77.6±0.1 (-4.5) |
| ACC ($\Delta$) on MMLU | | | |
| 1 | 23.1±0.3 (0.0) | 23.1±0.1 (0.0) | 23.1±0.1 (0.0) |
| 2 | 22.5±0.1 (-0.6) | **22.8±0.6 (-0.3)** | **22.8±0.3 (-0.3)** |
| 3 | 21.6±0.4 (-1.5) | 21.7±0.4 (-1.4) | **22.0±0.1 (-1.1)** |
| 4 | 19.7±0.5 (-3.4) | **20.0±0.5 (-3.1)** | 19.9±0.4 (-3.2) |
| 6 | 18.5±0.4 (-4.6) | 18.2±0.6 (-4.9) | **18.8±0.2 (-4.3)** |

## C. Hyperparameters

*Table 3.* Details of the selected hyper-parameters, including batch size, learning rate $\eta$ and weight decay $w_d$ for each set of experiments with the same dataset and fine-tuning method.

| Dataset | LoRA | Batch Size | $\eta$ | $w_d$ | Dataset | LoRA | Batch Size | $\eta$ | $w_d$ |
|---|---|---|---|---|---|---|---|---|---|
| SST2 | No | 256 | 1e-5 | 0.01 | QNLI | No | 128 | 2e-5 | 0.1 |
| | Yes | 128 | 2e-5 | 0.01 | | Yes | 128 | 2e-5 | 0.1 |
| MNLI | No | 128 | 2e-5 | 0.01 | Alpaca | No | 128 | 2e-5 | 0.01 |
| | Yes | 128 | 1e-5 | 0.01 | | Yes | 128 | 2e-5 | 0.01 |

## D. More Experiment Results

We also run the experiment to compare neighbour grouping and AsymGQA on OPT-1.3B. As shown in Table 4, AsymGQA still outperforms GQA by a clear margin.

*Table 4.* The performance of OPT-1.3B with experimentally best GQA architecture (i.e. asymmetric grouping + random-search-based approach + vector-wise cosine similarity on activations) compared with neighbour grouping. Column `NG` means neighbour grouping. Columns noted with `LoRA` are results of models fine-tuned by LoRA while other columns are from models fully fine-tuned. Grouping methods with the highest accuracies are highlighted in bold.

| | | ACC (Δ) on SST2 | | |
|---|---|---|---|---|
| **Group Size** | GQA (FT) | AsymGQA (FT) | GQA (LoRA) | AsymGQA (LoRA) |
| 1 | 93.0±0.5 (0.0) | 93.0±0.5 (0.0) | 92.8±0.6 (0.0) | 92.8±0.6 (0.0) |
| 2 | 91.3±0.5 (-1.7) | **92.8±0.7 (-0.2)** | 89.6±0.1 (-3.1) | **91.9±0.5 (-0.8)** |
| 3 | 89.2±0.5 (-3.8) | **91.9±0.3 (-1.1)** | 88.1±0.2 (-4.7) | **91.2±0.2 (-1.5)** |
| 4 | 88.5±0.6 (-4.5) | **90.6±0.2 (-2.4)** | 86.6±0.4 (-5.2) | **90.2±0.1 (-2.5)** |
| 6 | 87.7±0.2 (-5.3) | **90.5±0.5 (-2.5)** | 84.6±0.1 (-8.2) | **89.1±0.6 (-3.7)** |
| | | ACC (Δ) on QNLI | | |
| **Group Size** | GQA (FT) | AsymGQA (FT) | GQA (LoRA) | AsymGQA (LoRA) |
| 1 | 89.1±0.3 (0.0) | 89.1±0.3 (0.0) | 89.0±0.7 (0.0) | 89.0±0.7 (0.0) |
| 2 | 84.7±0.2 (-4.4) | **88.7±0.1 (-0.4)** | 84.3±0.1 (-4.7) | **88.2±0.4 (-0.8)** |
| 3 | 84.2±0.4 (-4.9) | **87.3±0.1 (-1.8)** | 82.8±0.6 (-6.2) | **87.2±0.4 (-1.8)** |
| 4 | 79.1±0.5 (-10.0) | **85.7±0.5 (-3.4)** | 80.3±0.4 (-8.7) | **85.1±0.6 (-3.9)** |
| 6 | 78.0±0.1 (-11.1) | **85.0±0.4 (-4.1)** | 77.9±0.6 (-11.1) | **84.2±0.2 (-4.8)** |
| | | ACC (Δ) on MNLI | | |
| **Group Size** | GQA (FT) | AsymGQA (FT) | GQA (LoRA) | AsymGQA (LoRA) |
| 1 | 84.2±0.6 (0.0) | 84.2±0.6 (0.0) | 83.8±0.5 (0.0) | 83.8±0.5 (0.0) |
| 2 | 81.3±0.2 (-2.9) | **83.7±0.3 (-0.5)** | 80.1±0.5 (-3.7) | **81.8±0.3 (-1.0)** |
| 3 | 79.4±0.5 (-4.8) | **82.8±0.2 (-1.4)** | 78.9±0.3 (-4.9) | **82.5±0.6 (-1.3)** |
| 4 | 79.0±0.2 (-5.2) | **81.2±0.4 (-3.0)** | 77.9±0.5 (-5.9) | **81.1±0.2 (-2.7)** |
| 6 | 76.7±0.3 (-7.5) | **80.0±0.1 (-4.2)** | 75.3±0.5 (-8.5) | **79.2±0.6 (-4.6)** |
| | | ACC (Δ) on MMLU | | |
| **Group Size** | GQA (FT) | AsymGQA (FT) | GQA (LoRA) | AsymGQA (LoRA) |
| 1 | 22.9±0.3 (0.0) | 22.9±0.3 (0.0) | 23.0±0.2 (0.0) | 23.0±0.2 (0.0) |
| 2 | 21.6±0.5 (-1.3) | **23.2±0.1 (0.3)** | 20.3±0.6 (-2.7) | **22.6±0.7 (-0.4)** |
| 3 | 19.7±0.1 (-3.2) | **22.3±0.7 (-0.6)** | 19.5±0.2 (-3.5) | **21.8±0.1 (-1.2)** |
| 4 | 17.5±0.4 (-5.4) | **20.7±0.5 (-2.2)** | 16.5±0.5 (-6.5) | **20.9±0.6 (-2.1)** |
| 6 | 16.5±0.6 (-6.4) | **19.2±0.5 (-3.6)** | 15.2±0.4 (-7.8) | **19.4±0.7 (-3.6)** |