NEURAL INFORMATION PROCESSING SYSTEMS
18th Annual Conference. 2024

中国科学院大学
University of Chinese Academy of Sciences

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

Bai du 百度

DHA: Learning Decoupled-Head Attention from Transformer Checkpoints via Adaptive Heads Fusion

Yilong Chen[1,2,*], Linhao Zhang[3*], Junyuan Shang[3‡], Zhenyu Zhang[3],
Tingwen Liu[1,2†], Shuohuan Wang[3], Yu Sun[3]

1 Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. Parameter Fusion
2 School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China.
3 Baidu Inc., Beijing, China

Presenter: Yilong Chen
E-mail: chenyilong@iie.ac.cn
11/11/2024

Paper Link

Outline

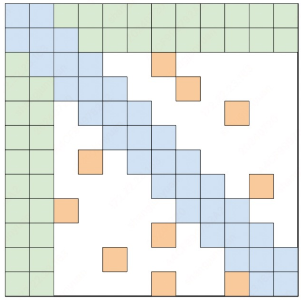
- **Background**
- **Motivation**
- **Method**
- **Experiments**
- **Summary**

- **Challenge: Large KV Cache with Long Context**
  - **KV Cache:** During decoding phase, the key and value hidden states of all previous tokens in Attention block need to be stored to avoid re-computation.

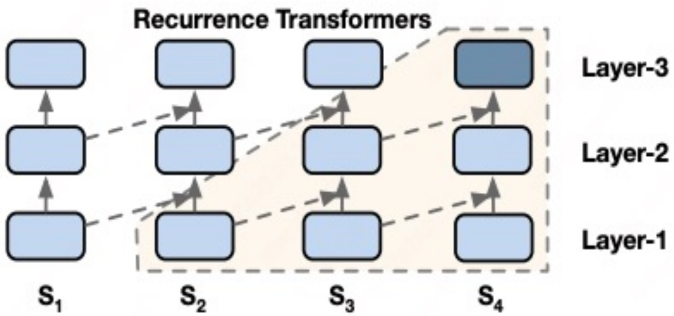
Length * Batch-Size * Num-Layers * Num-Heads * Head-Dim * 2 * 2bytes

KV Cache Memory Consumption (bf16)

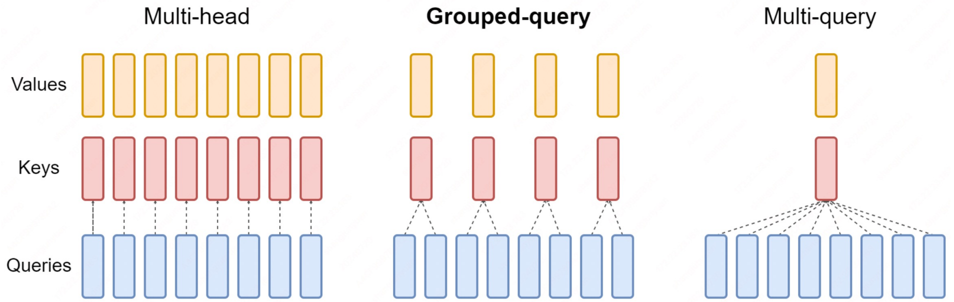
- **Difficulties in Efficient Transformer Re-training**
  - Sparse Attention /Recurrence /Head Sharing



BigBird[1]



ERNIE-Doc[2]



GQA[3] & MQA

[1] Big Bird: Transformers for Longer Sequences
[2] ERNIE-DOC: A Retrospective Long-Document Modeling Transformer
[3] GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints

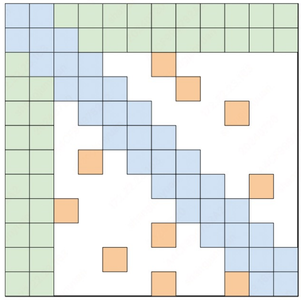
- **Challenge: Large KV Cache with Long Context**
  - **KV Cache:** During decoding phase, the key and value hidden states of all previous tokens in Attention block need to be stored to avoid re-computation.

Length * Batch-Size * Num-Layers * Num-Heads * Head-Dim * 2 * 2bytes

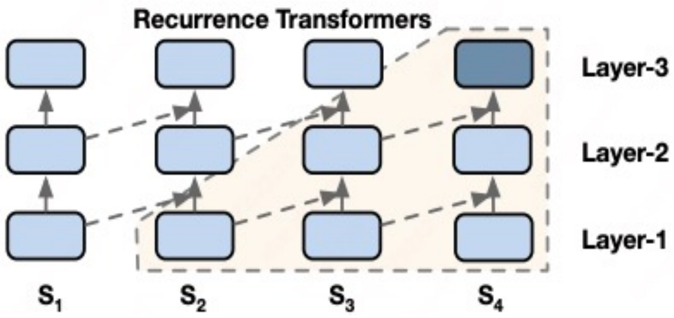
KV Cache Memory Consumption (bf16)

- **Difficulties in Efficient Transformer Re-training**
  - Sparse Attention /Recurrence /Head Sharing

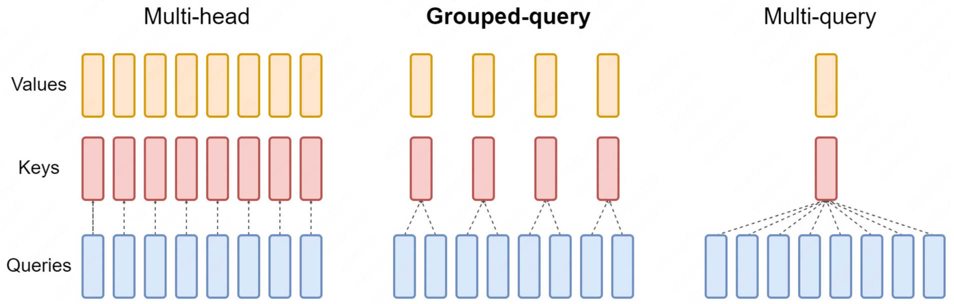
😭**Resource-Intensive Re-training**
😭**Performance Degradation**



BigBird[1]



ERNIE-Doc[2]



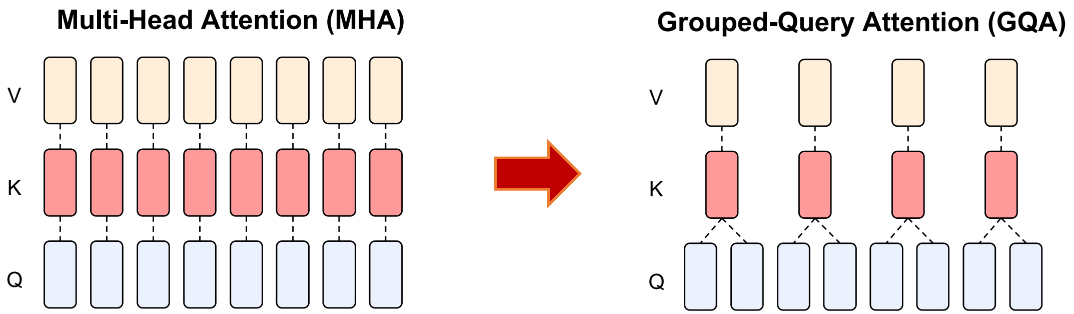
GQA[3] & MQA

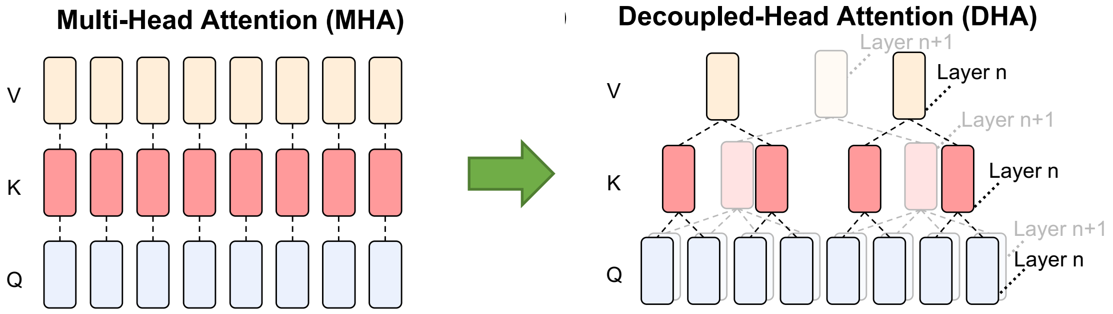
[1] Big Bird: Transformers for Longer Sequences
[2] ERNIE-DOC: A Retrospective Long-Document Modeling Transformer
[3] GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints

- **Challenge: Large KV Cache with Long Context**
  - **KV Cache:** During decoding phase, the key and value hidden states of all previous tokens in Attention block need to be stored to avoid re-computation.

$$\underbrace{\textbf{Length * Batch-Size * Num-Layers * Num-Heads * Head-Dim * 2 * 2bytes}}_{\textbf{KV Cache Memory Consumption (bf16)}}$$

- **Difficulties in Efficient Transformer Re-training**

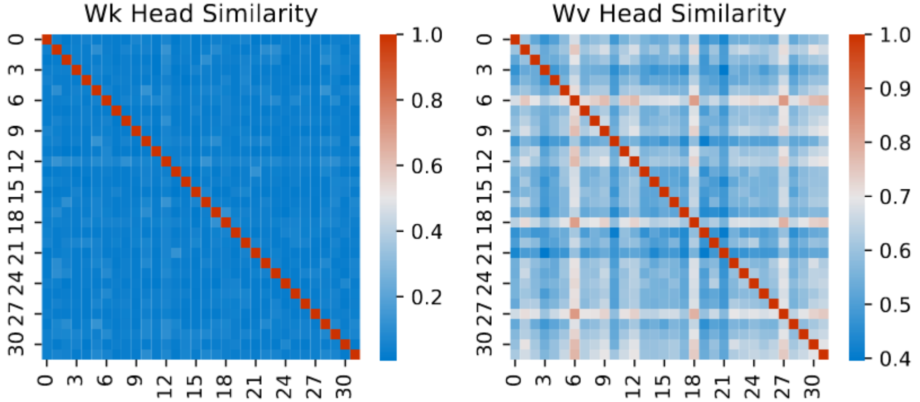




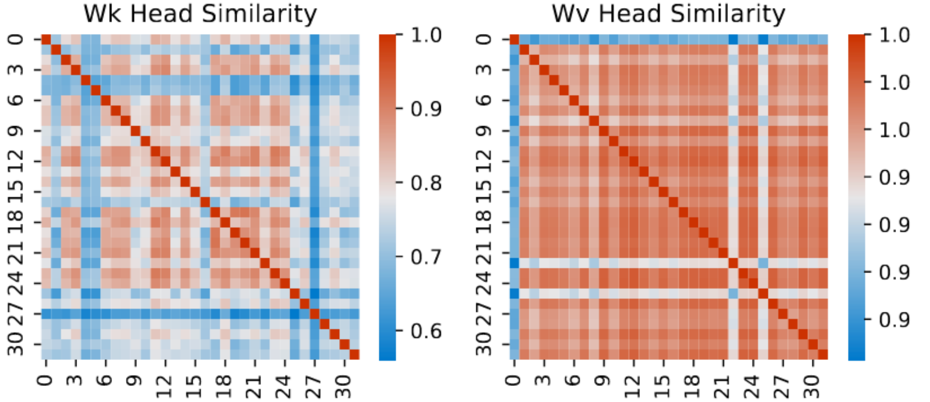
😭**Resource-Intensive Re-training**
😭**Performance Degradation**

😄**Resource-Efficient Re-training**
😄**Performance Maintenance**

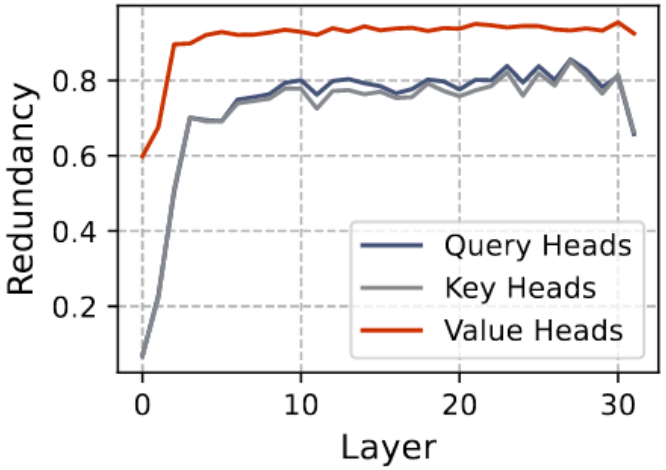
- **Heterogeneity of Head Similarity in Attention**



(a) Head Weight Similarity in 0th Layer



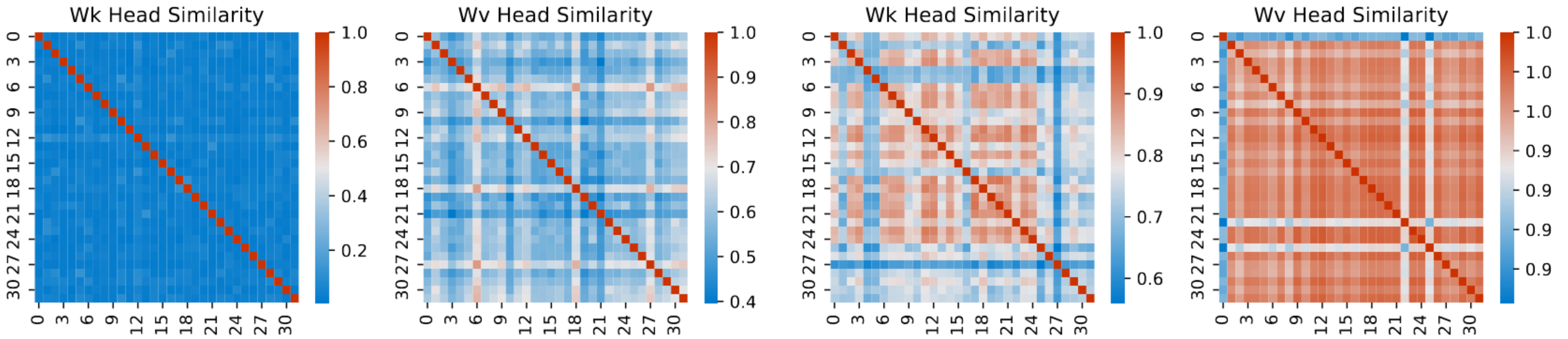
(b) Head Weight Similarity in 21st Layer



- **Head Similarity Observation Experiments**

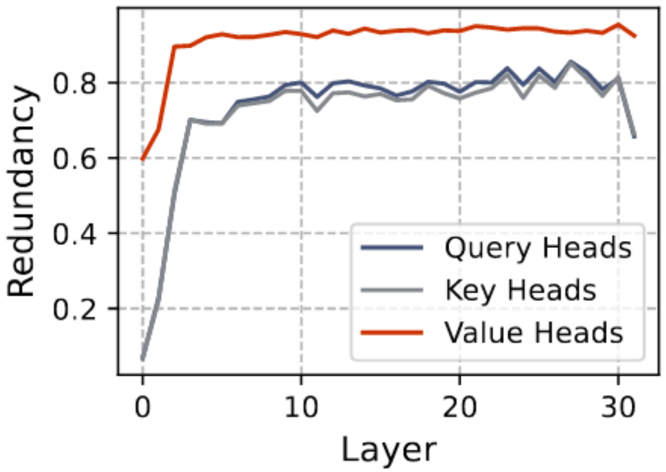
- The distribution of head **similarity varies significantly** across layers: the **initial** layers are relatively **sparse**, while the **later** layers are **more redundant**.
- The redundancy of **Values is higher than that of Keys.**

- **Heterogeneity of Head Similarity in Attention**



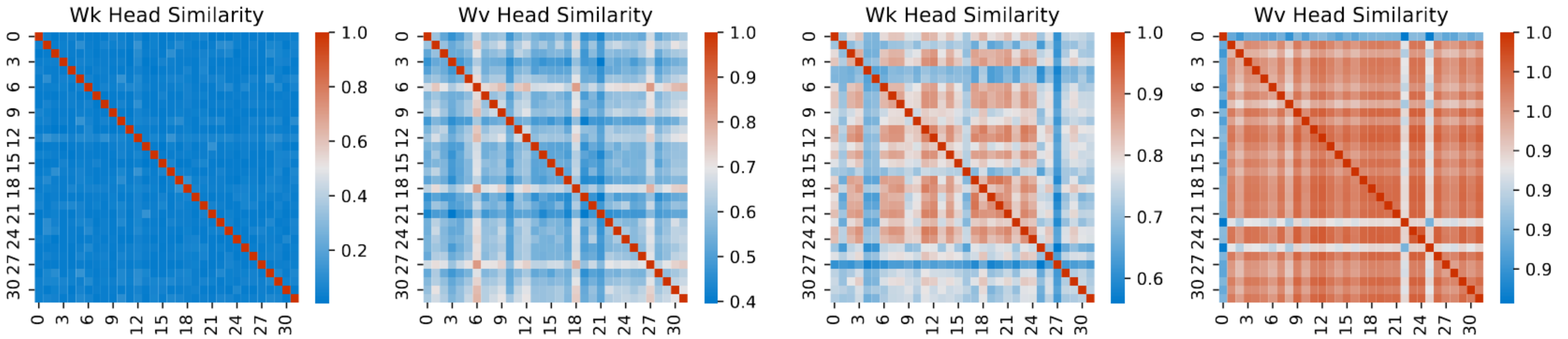
(a) Head Weight Similarity in 0th Layer

(b) Head Weight Similarity in 21st Layer



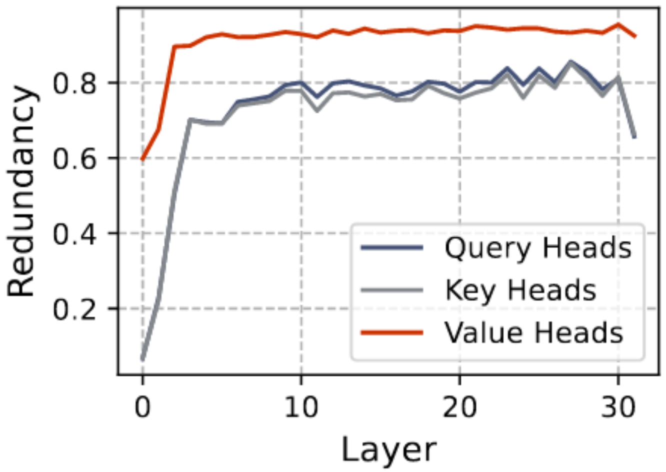
- **Head Similarity Observation Experiments**
- The distribution of head **similarity varies significantly** across layers: the **initial** layers are relatively **sparse**, while the **later** layers are **more redundant**.
- The redundancy of **Values is higher than that of Keys**.

- **Heterogeneity of Head Similarity in Attention**



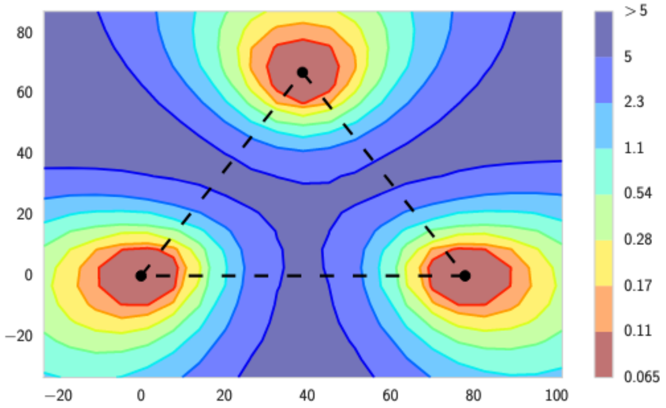
(a) Head Weight Similarity in 0th Layer

(b) Head Weight Similarity in 21st Layer

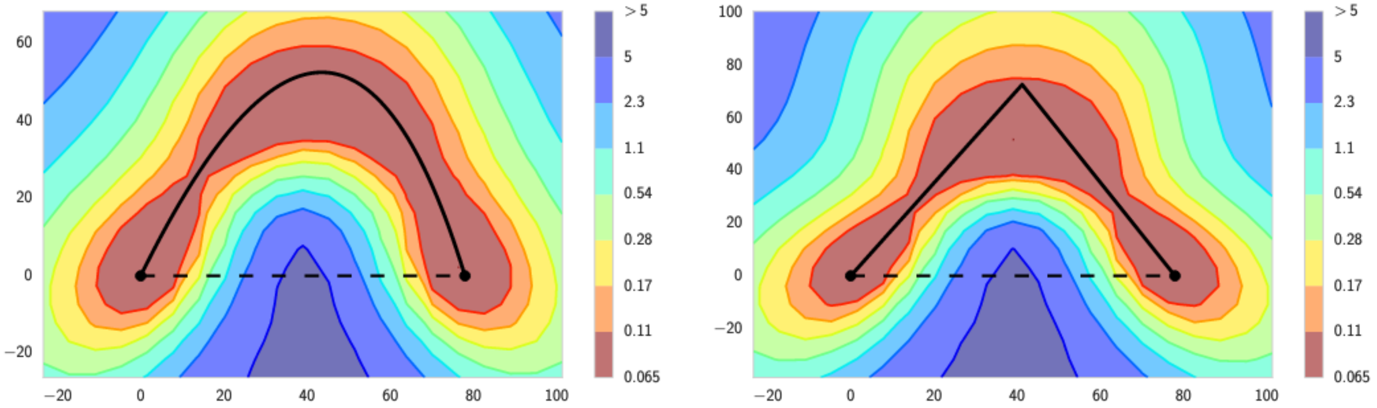


- **Motivation 1**
- By gradually **decoupling and reallocating the Head Budget across layers**, more heads can be assigned to layers with lower redundancy and specialized functions, while compressing layers with higher redundancy. This approach not only reduces model parameters but also enhances its performance.

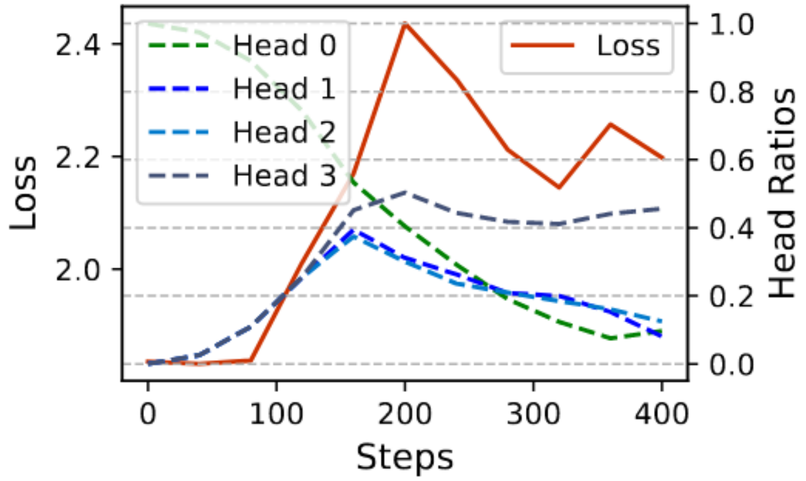
- **Connectivity of head parameters**



Independent DNNs



Connectable path between optimal points in loss landscape[12]

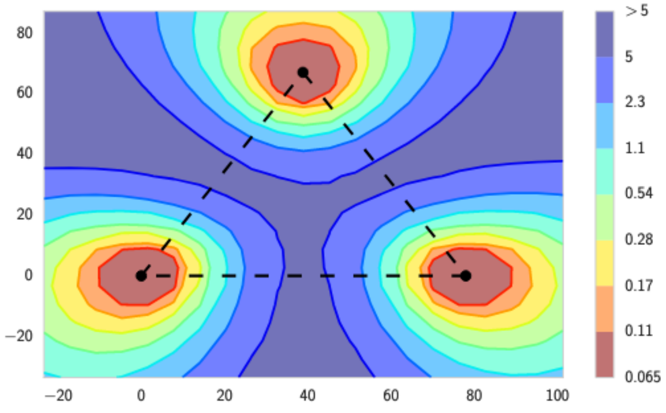


- **Head Fusion Observation Experiments**

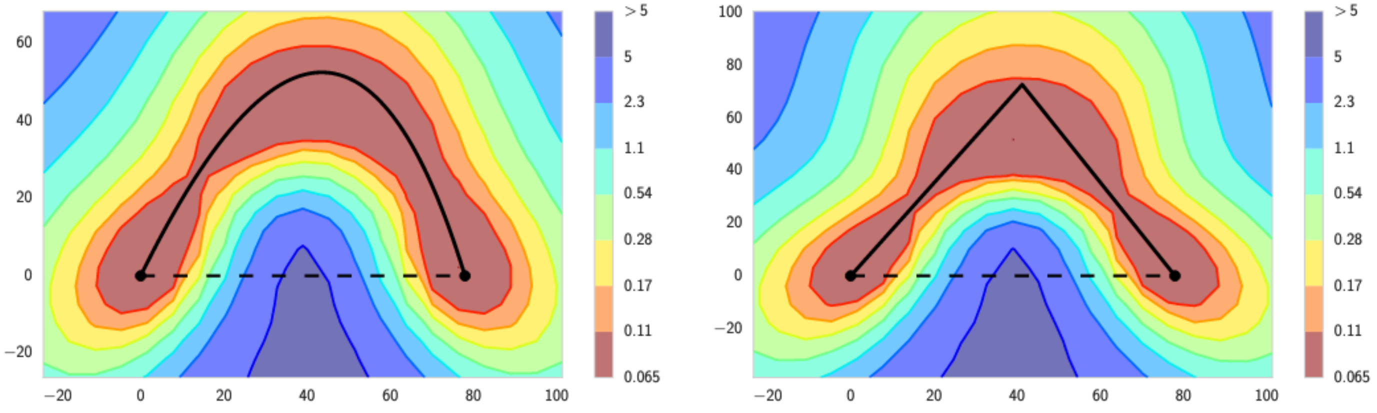
$$\mathbf{W}_{k/v}^{d^{\mathrm{K/V}}(h,l)} = \sum_{j=1}^{g^{\mathrm{K/V}}} \omega_{hj} \mathbf{W}_{k/v}^{j}$$

1.Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs.
2. Exploring Mode Connectivity for Pre-trained Language Models

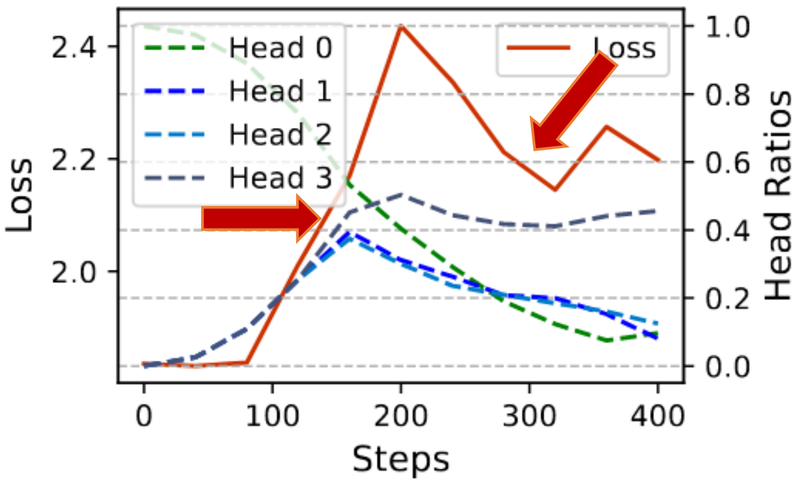
- **Connectivity of head parameters**



Independent DNNs



Connectable path between optimal points in loss landscape

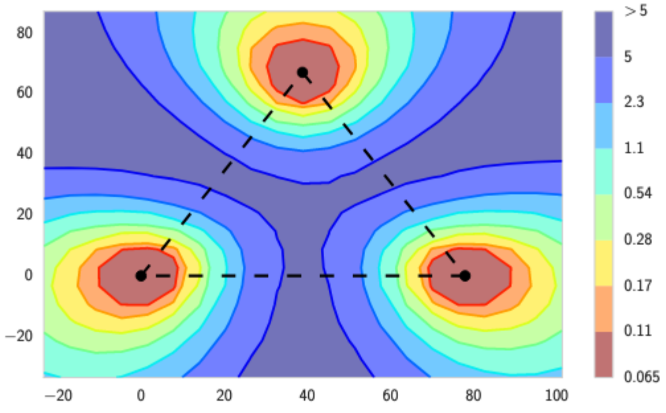


- **Head Fusion Observation Experiments**

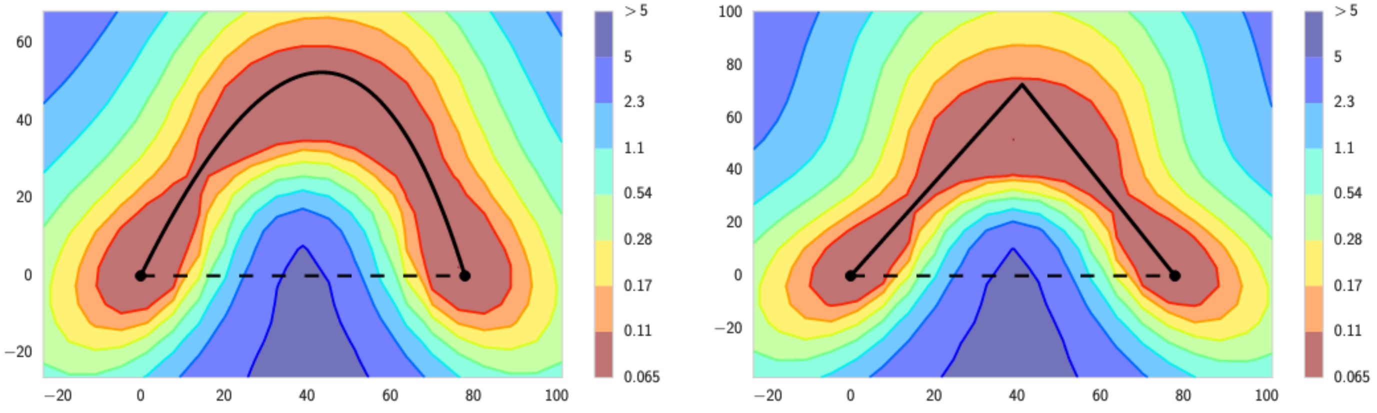
$$\mathbf{W}_{k/v}^{d^{\mathrm{K/V}}(h,l)} = \sum_{j=1}^{g^{\mathrm{K/V}}} \omega_{hj} \mathbf{W}_{k/v}^{j}$$

The loss **increases** when the head parameter **ratio approaches 0.5** but **decreases and stabilizes** toward the end.

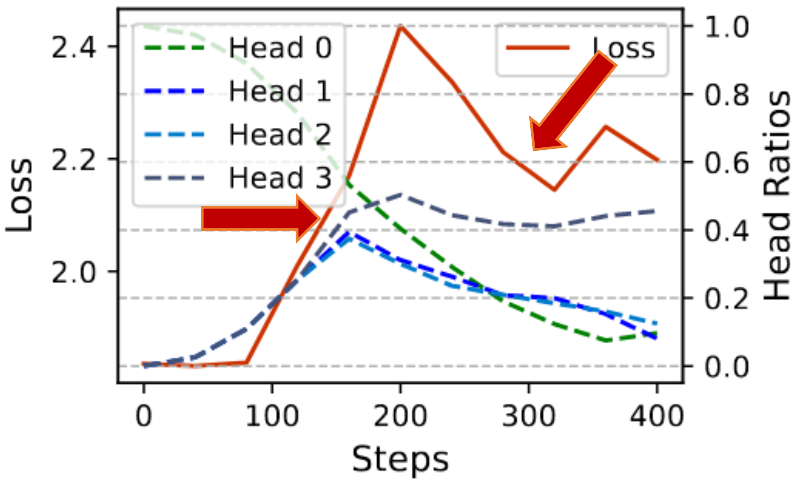
- **Connectivity of head parameters**



Independent DNNs



Connectable path between optimal points in loss landscape

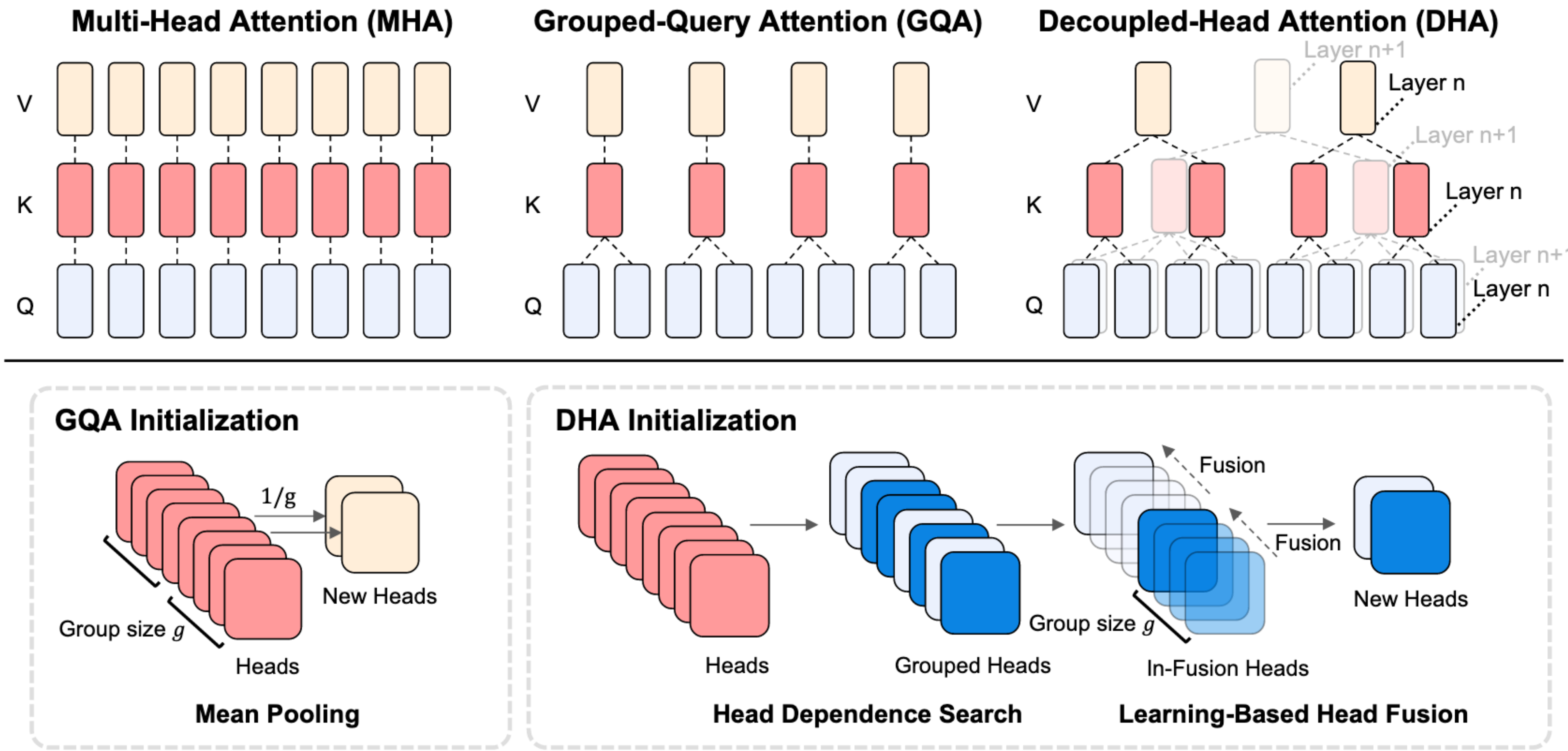


- **Head Fusion Observation Experiments**

The loss **increases** when the head parameter **ratio approaches 0.5** but **decreases and stabilizes** toward the end.

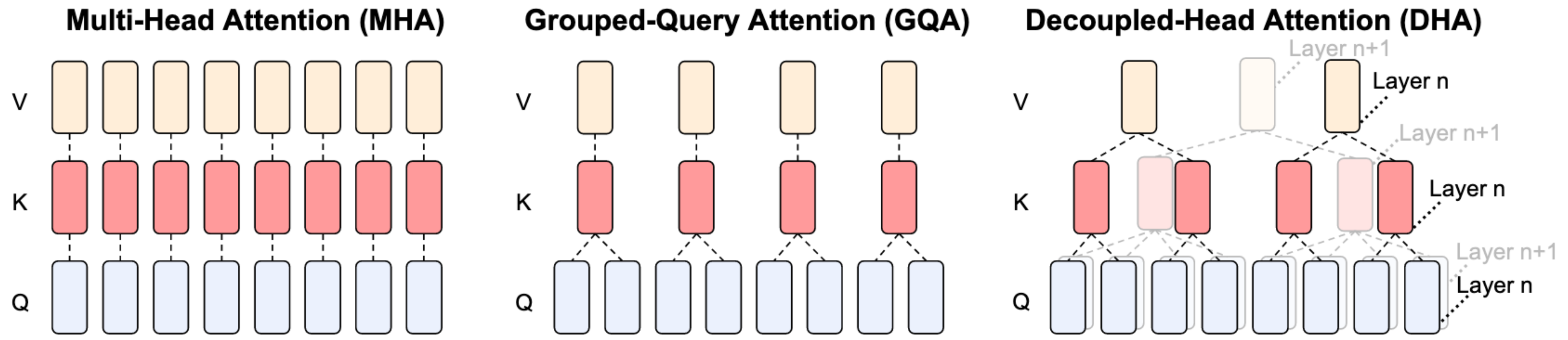
- **Motivation 2**

Parameter fusion can **reconstruct the functionality** of the original parameters while **reducing the number of heads**



How can we construct **a more efficient model** while **keeping costs as low as possible**?

- **Definition**



- **Multi-Head Attention (MHA)**

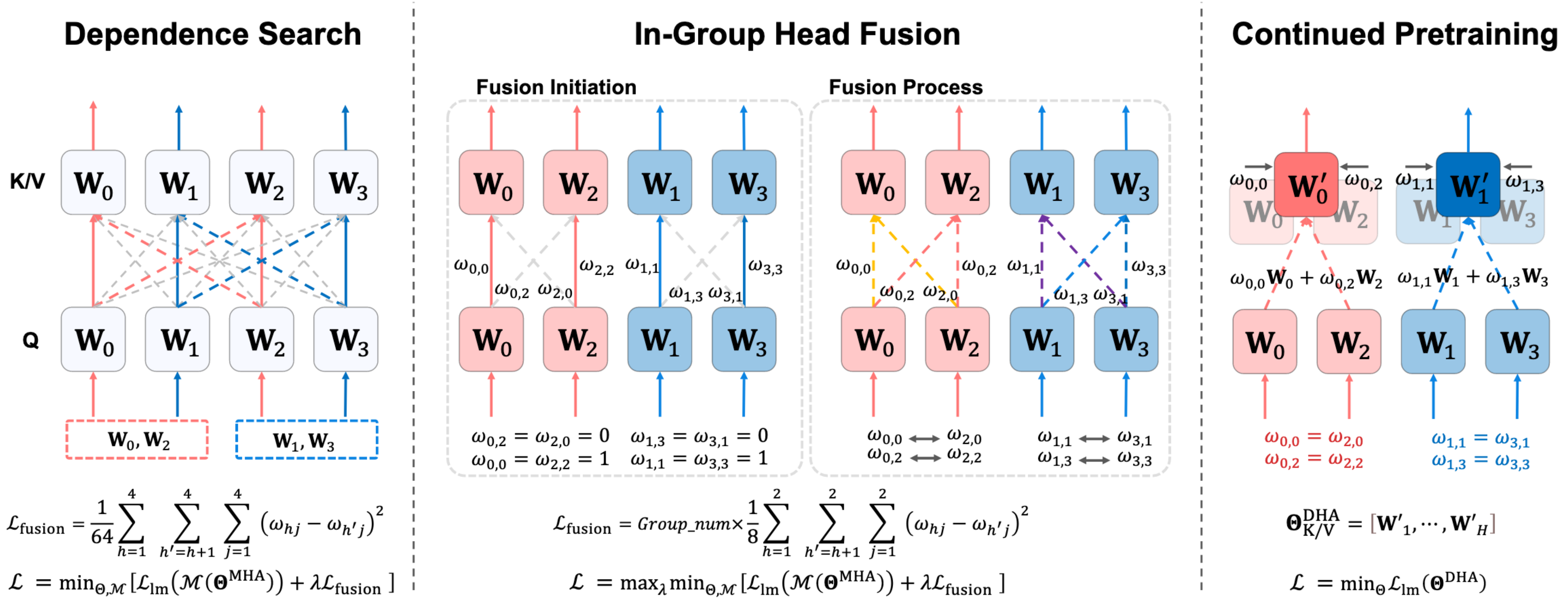
$$\text{MHA} = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_{\text{H}}\right) \mathrm{W}_O, \text{where head}_h = \sigma\left(\mathbf{X}\mathbf{W}_q^h(\mathbf{X}\mathbf{W}_k^h)^T \cdot \frac{1}{\sqrt{d_k}}\right)\mathbf{X}\mathbf{W}_v^h \qquad (1)$$

- **Decoupled-Head Attention (DHA)**

$$\text{head}_{h,l} = \sigma\left(\mathbf{X}\mathbf{W}_q^h(\mathbf{X}\mathbf{W}_k^{d^{\mathrm{K}}(h,l)})^T \cdot \frac{1}{\sqrt{d_k}}\right)\mathbf{X}\mathbf{W}_v^{d^{\mathrm{V}}(h,l)} \qquad (3)$$

DHA shares key and value heads in multi-query attention based **on independently mapped functions** across different layers.

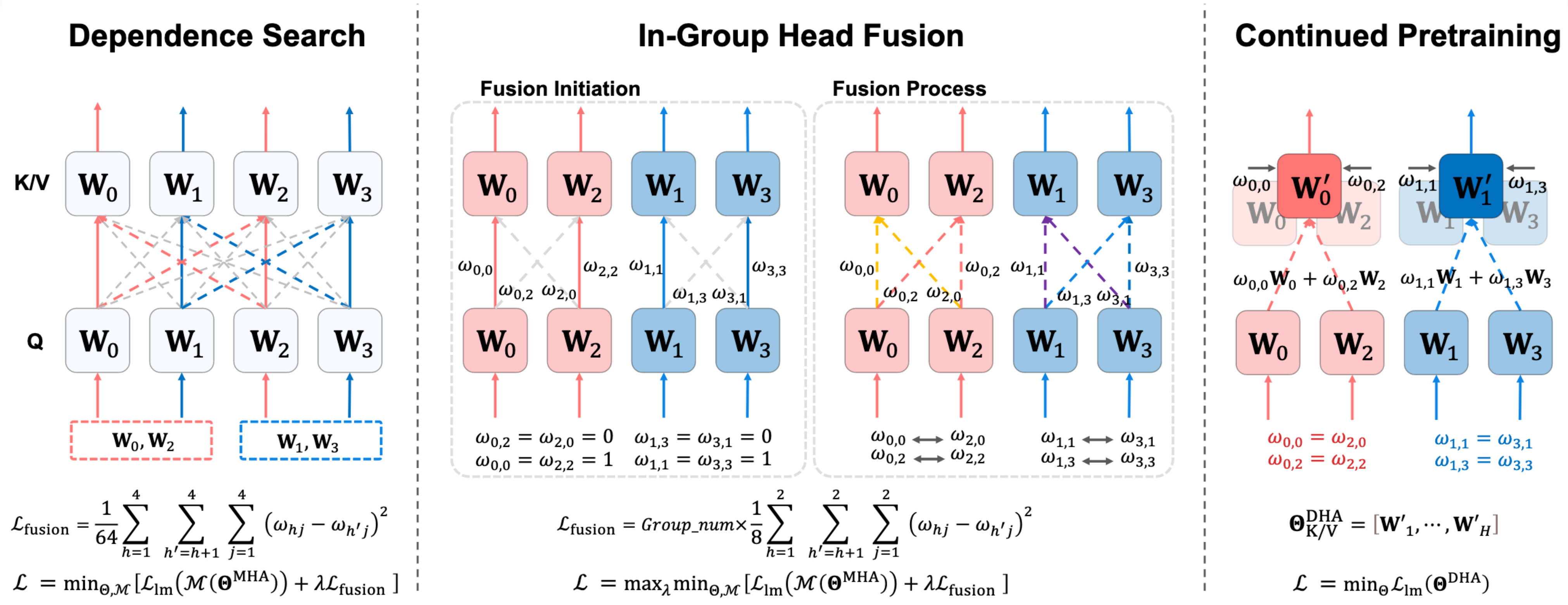
DHA consists of $H = H^{\mathrm{Q}} + \sum_{l=1}^{L} H_l^{\mathrm{K}} + \sum_{l=1}^{L} H_l^{\mathrm{V}}$ heads in total.



- **Goal**

$$\arg\min_{\Theta,\mathcal{M}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \left[\mathcal{L}_{\text{lm}}\left(\boldsymbol{x}; \mathcal{M}(\Theta^{\text{MHA}})\right) + \lambda\mathcal{L}_{\text{fusion}}\left(\boldsymbol{x}; \mathcal{M}(\Theta^{\text{MHA}}), \Theta^{\text{DHA}}\right)\right] \quad (4)$$

By **progressively merging head parameters**, we **reduce the number of heads** while **retaining the knowledge** of the original model, thus decreasing training costs and enhancing performance.



- **Fusion Operator**

$$\text{head}_{h,l} = \sigma\left(\mathbf{X}\mathrm{W}_q^h(\mathbf{X}\mathrm{W}_k^{d^{\mathrm{K}}(h,l)})^T \cdot \frac{1}{\sqrt{d_k}}\right)\mathbf{X}\mathrm{W}_v^{d^{\mathrm{V}}(h,l)}, \text{ where } \mathrm{W}_{k/v}^{d^{\mathrm{K/V}}(h,l)} = \sum_{j=1}^{g^{\mathrm{K/V}}} \omega_{hj}\mathrm{W}_{k/v}^{j} \tag{5}$$

During DHA initialization, the fusion operator constructs new heads based on **a linear combination of original key and value heads** within each group. The initial forward of DHA are **fully equivalent** to those of MHA.

- **Optimization**

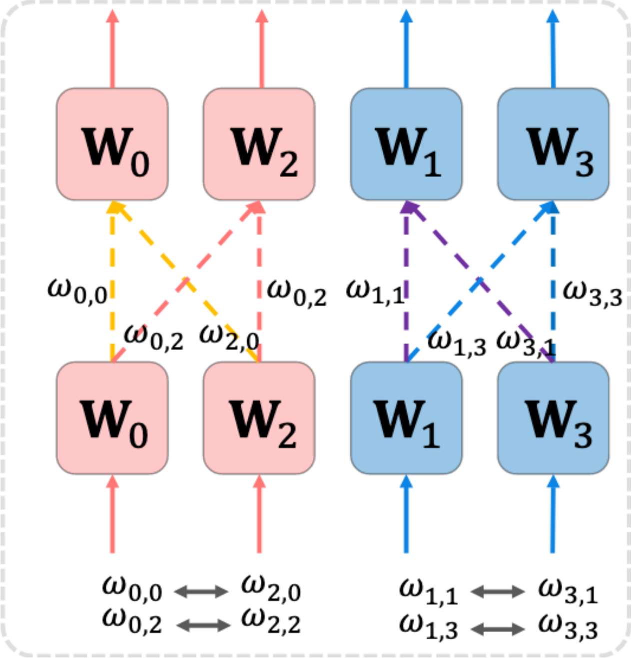
The goal is to enable **a single fused key or value head to be shared across multiple query heads** in DHA. We design a fusion loss to **optimize** the initial mapping **functions to a single unified mapping function**.

$$\mathcal{L}_{\text{head}_l^n}(h, h') = \frac{1}{g}\left\|\sum_{j=1}^{g}\omega_{hj}\mathbf{W}^{j}_{k/v} - \sum_{j=1}^{g}\omega_{h'j}\mathbf{W}^{j}_{k/v}\right\|^2 = \frac{1}{g}\left(\sum_{j=1}^{g}(\omega_{hj}-\omega_{h'j})\mathbf{W}^{j}_{k/v,ij}\right)^2 \tag{6}$$

Since W can be considered a scalar, we **only need to optimize the fusion variable ω.**

$$\mathcal{L}_{\text{fusion}} = \sum_{l=1}^{L}\sum_{n=1}^{N}\sum_{h=1}^{g}\sum_{h'=h+1}^{g}\mathcal{L}_{\text{head}_l^n}(h, h'), \text{subject to } \mathcal{L}_{\text{head}_l^n}(h, h') = \frac{1}{g}\sum_{h=1}^{g}\sum_{j=1}^{g}(\omega_{hj}-\omega_{h'j})^2 \tag{7}$$

Fusion Process



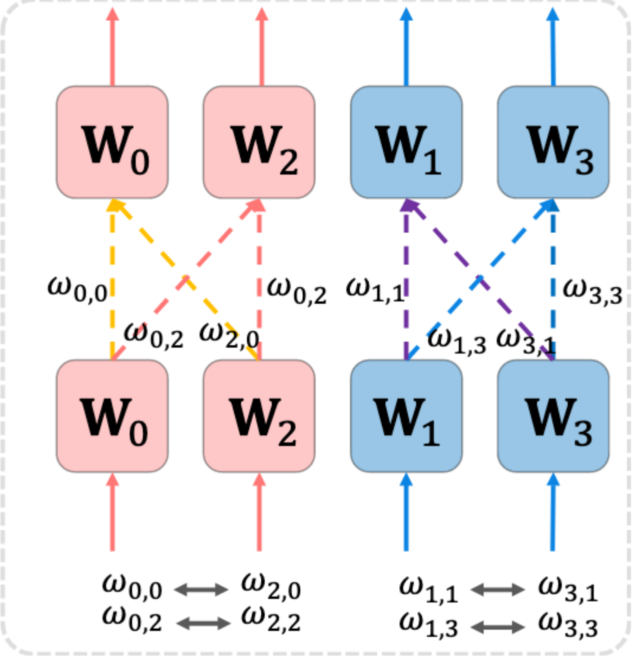
Challenge: We must **optimize the fusion loss to a near-zero minimum**, enabling effective **sharing** of the new DHA key-value head parameters **across queries within the group**.

- **Optimization**

The goal is to enable **a single fused key or value head to be shared across multiple query heads** in DHA. We design a fusion loss to **optimize** the initial mapping **functions to a single unified mapping function**.

$$\mathcal{L}_{\text{head}_l^n}(h, h') = \frac{1}{g}\left\|\sum_{j=1}^{g}\omega_{hj}\mathbf{W}_{k/v}^{j} - \sum_{j=1}^{g}\omega_{h'j}\mathbf{W}_{k/v}^{j}\right\|^2 = \frac{1}{g}\left(\sum_{j=1}^{g}(\omega_{hj} - \omega_{h'j})\mathbf{W}_{k/v,ij}^{j}\right)^2 \quad (6)$$

Fusion Process



- **Augmented Lagrangian Approach**

In the **early stages of training**, We encourage the model to **tolerate differences** among parameters to promote exploration. As training progresses, the **algorithm gradually enforces stricter reduction** of these differences, **improving parameter alignment** within each group.

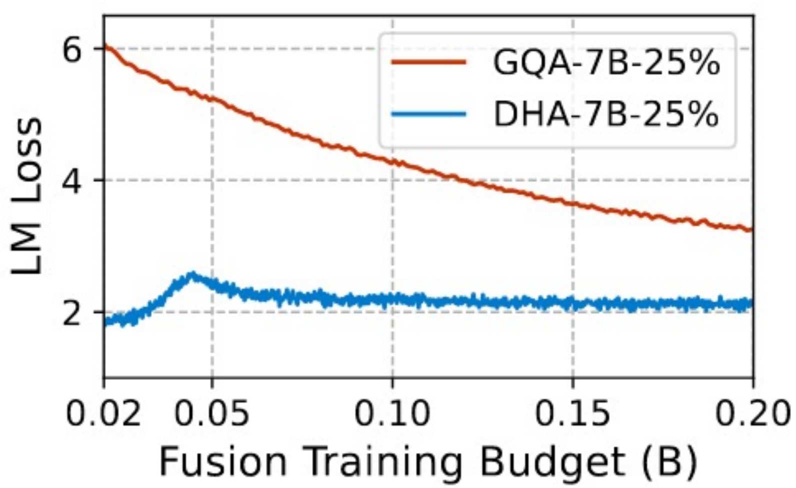
$$\max_{\lambda}\min_{\Theta,\mathcal{M}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\mathcal{L}_{\text{lm}}\left(\boldsymbol{x}; \mathcal{M}(\Theta^{\text{MHA}})\right) + \lambda\max\left(\mathcal{L}_{\text{fusion}} - t, 0\right)\right], \text{where } t = \max\left(0, b^{s}\left(1 - \frac{s}{k}\right)\right) \quad (8)$$

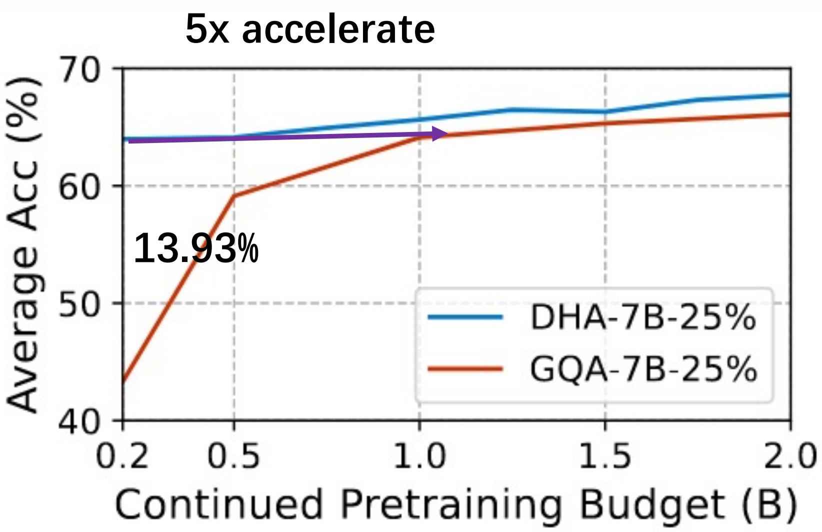
t as the target loss, b as the base decay factor, s as the current global step, k as warm-up step

Table 1: Comprehensive assessment of model's fundamental capabilities, in which DHA models demonstrate competitive performance while requiring significantly fewer training resources. Models with † use MHA.

		Commonsense & Comprehension						Continued		LM	
Model	**Budget**	**SciQ**	**PIQA**	**Wino.**	**ARC-E**	**ARC-C**	**HellaS.**	**LogiQA**	**BoolQ**	**LAMB.**	**Average**
LLaMA2-7B†	2T	94.1	78.1	69.1	76.3	49.7	58.9	25.7	80.8	74.1	67.4
DHA-7B-50%	50B	93.4	78.5	69.1	73.8	45.9	58.6	22.5	79.1	71.1	65.8
DHA-7B-25%	50B	92.4	78.5	68.6	72.9	43.9	57.6	22.4	76.7	70.2	64.8
GQA-7B-50%	1B	90.7	**76.8**	66.5	**71.3**	41.9	53.6	**22.4**	70.5	67.0	62.3
DHA-7B-50%	1B	**90.8**	76.5	**66.7**	**71.3**	**44.6**	**55.1**	**22.4**	**74.8**	**67.2**	**63.3**
GQA-7B-25%	1B	86.5	74.3	59.1	67.6	37.5	49.2	**24.1**	65.8	58.3	58.0
DHA-7B-25%	1B	**90.0**	**75.2**	**63.8**	**70.4**	**39.3**	**52.2**	21.1	**72.3**	**62.9**	**60.7**
S.-LLaMA-2.7B†	2T	91.2	76.1	64.9	67.3	38.8	52.2	22.1	74.4	68.3	61.7
GQA-2.7B-50%	1B	86.7	74.8	59.0	64.0	34.2	48.2	**23.8**	64.9	60.3	57.3
DHA-2.7B-50%	1B	**86.8**	**75.1**	**59.5**	**64.6**	**35.1**	**48.7**	22.4	**66.4**	**61.7**	**57.8**
GQA-2.7B-25%	1B	82.0	72.8	54.9	58.4	31.0	42.9	**21.7**	58.5	49.6	52.4
DHA-2.7B-25%	1B	**85.6**	**74.1**	**57.6**	**61.5**	**32.4**	**45.9**	**21.7**	**63.1**	**56.9**	**55.4**
S.-LLaMA-1.3B†	2T	87.0	73.6	58.2	60.9	29.5	45.4	21.8	65.5	61.3	55.9
GQA-1.3B-50%	1B	84.3	**72.3**	**55.8**	57.5	28.2	41.8	20.7	62.9	52.9	52.9
DHA-1.3B-50%	1B	**84.5**	72.0	55.2	**58.1**	**28.7**	**42.6**	**21.5**	**63.7**	**55.4**	**53.6**
GQA-1.3B-25%	1B	76.6	70.0	52.9	51.9	23.5	37.6	21.0	**59.9**	41.0	48.3
DHA-1.3B-25%	1B	**82.8**	**71.1**	**54.0**	**55.4**	**25.8**	**40.5**	**21.5**	57.6	**48.6**	**50.8**

- Under the **same training budget**, DHA **surpasses** GQA.
- **Higher compression rates** lead to **greater relative performance gains** for DHA.
- Achieves 97.5% performance with **just 0.05% of the training budget**.





- DHA initial loss is only **slightly above** the original model loss.
- DHA retains original model knowledge, achieving **a higher performance.**
- **0.1B data is enough** to get a good DHA initial point.

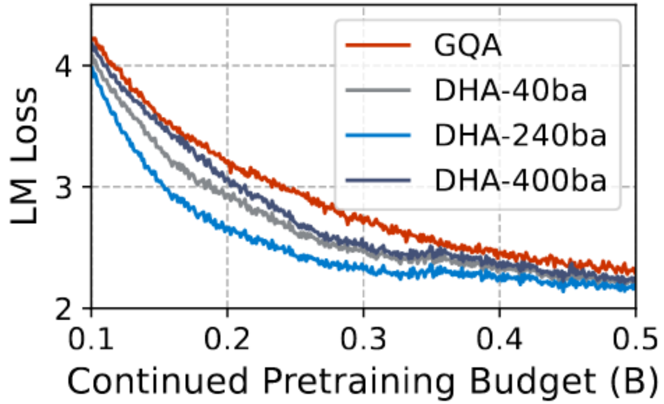
Table 2: Ablation Results of DHA *w.o.* Linear Heads Fusion and Adaptvie Transformation. Experiments are conducted with LLaMA2-7B with 25% heads budget and 0.5B & 1B training budget on 0-shot Evaluation.

Models	SciQ	PiQA	Wino.	ARC-E.	ARC-C.	LogiQA	LAMB.	Average	Diff
DHA-7B-25% (0.5B)	88.6	75.9	61.3	68.2	36.1	23.8	63.2	59.6	–
w.o. Linear Heads Fusion	83.4	73.7	57.3	63.6	29.4	22.0	51.9	54.5	−5.1
w.o. Adaptvie Transformation	87.9	74.1	60.1	69.4	34.7	19.5	62.1	58.3	−0.4
DHA-7B-25% (1B)	90.0	75.2	63.8	70.4	37.5	21.1	62.9	60.1	–
w.o. Linear Heads Fusion	87.5	74.5	60.7	67.3	32.8	21.7	58.3	57.5	−2.6
w.o. Adaptvie Transformation	89.5	74.6	62.8	69.1	36.3	21.6	62.4	59.5	−0.6
DHA-7B-25% (5B)	91.7	76.8	64.4	70.9	42.8	21.8	68.4	62.4	–
GQA-7B-25% (5B)	91.5	76.6	63.9	70.5	42.3	22.1	67.8	62.1	−0.4

Table 3: Data budget allocation to fusion and continued pre-training(CT) and 0-shot Task Average Accuracy (%) in DHA-1.3B.

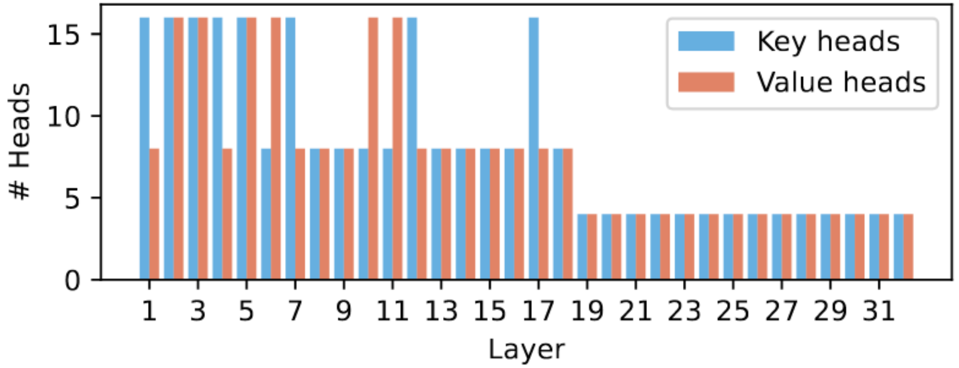
Fusion		CT	
Tokens	Avg.Acc	Tokens	Avg.Acc
0.05B	33.74	4.95B	59.08
0.10B	**38.32**	**4.90B**	**59.53**
0.15B	48.26	4.85B	59.46
0.20B	52.54	4.80B	59.16

- **Is the DHA architecture truly efficient?**



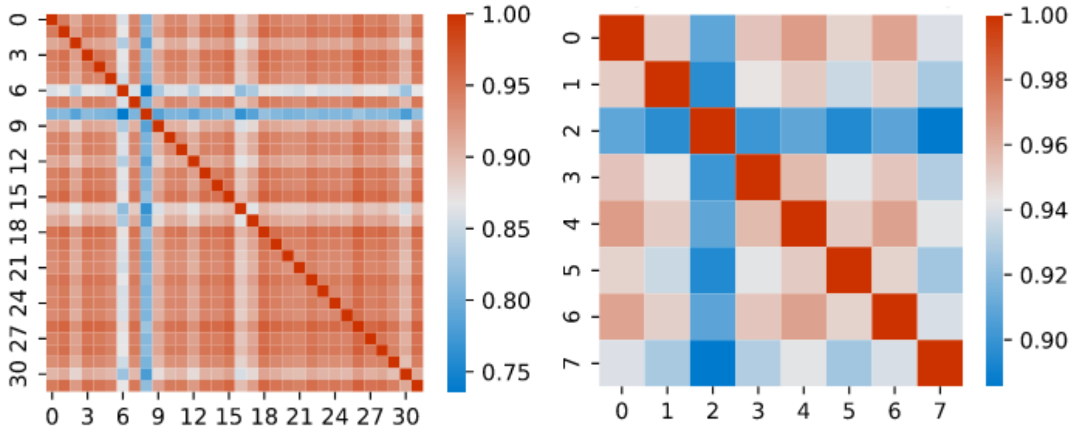
Training from scratch using **the DHA-searched architecture** achieves **faster** training speeds and **better** performance than GQA.

- **How does DHA allocate the head budget?**



DHA allocates **more parameters to critical layers.**
DHA generally **preserves parameters** in the **early layers.**
DHA **compresses parameters** in the **later layers.**

- **What is the head similarity distribution before and after DHA fusion?**



- DHA merges multiple heads within **each cluster into a single head** while preserving **inter-cluster relationships**.
- Maintains the **same overall distribution** trend as MHA.
- Effectively **reduces head parameter redundancy**.

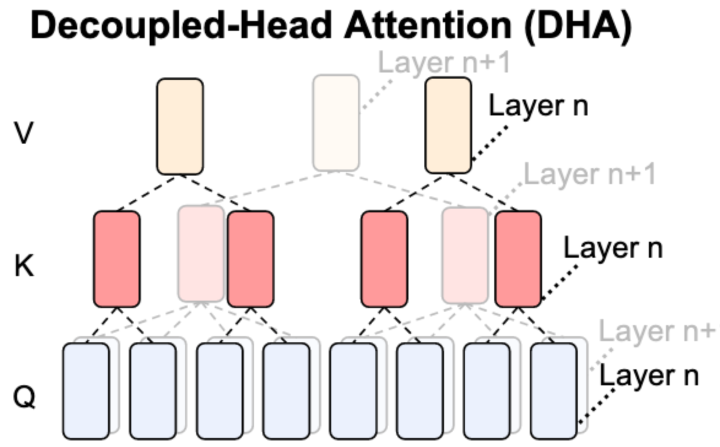
- **Performance of the Instruction Tuned DHA model**

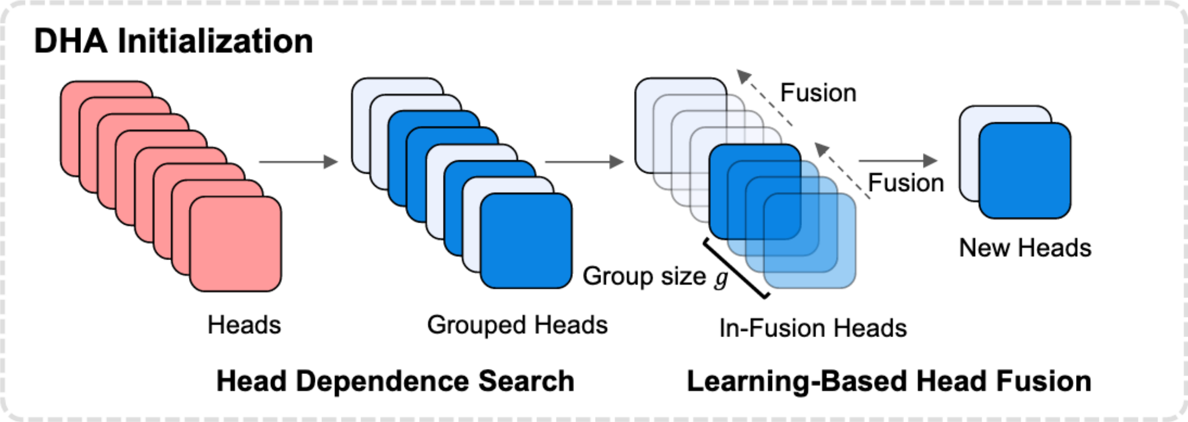
Figure 10: In model scale of 7B, 3B, and 1.3B, DHA significantly outperforms GQA and achieves comparable performance with MHA after instruction tuning .

Heterogeneous Attention Efficient Architecture
- Increases training speed
- Enhances the capability of key components
- Compresses parameters of redundant components

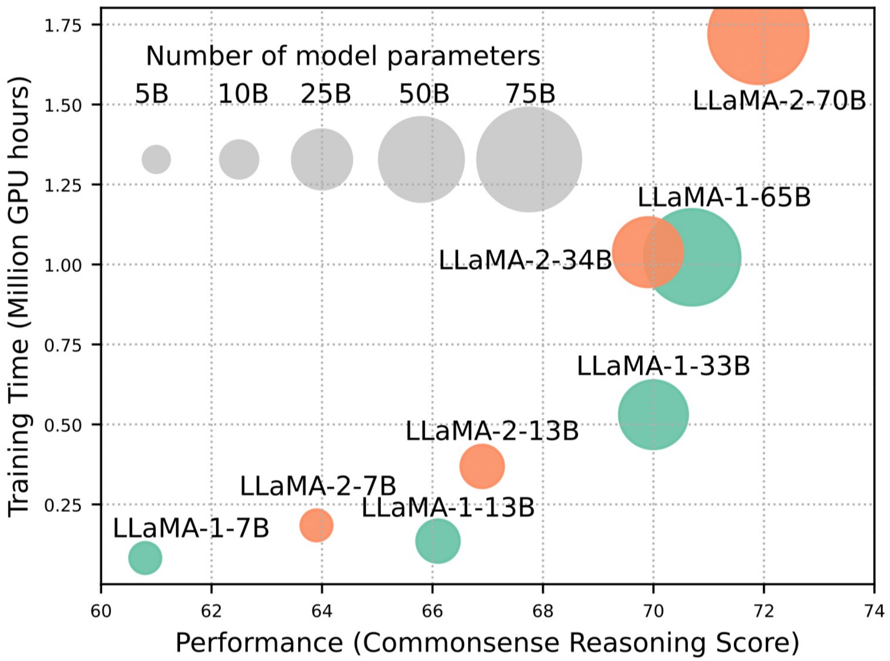
Progressive Head Parameter Fusion
- Significantly boosts training speed
- Achieves 5x training acceleration
- Reconstructs model functionality

Stronger and More Efficient Model
- 13.93% improvement with 0.01% budget
- 4% improvement with 0.05% budget
- 75% KVCache compression

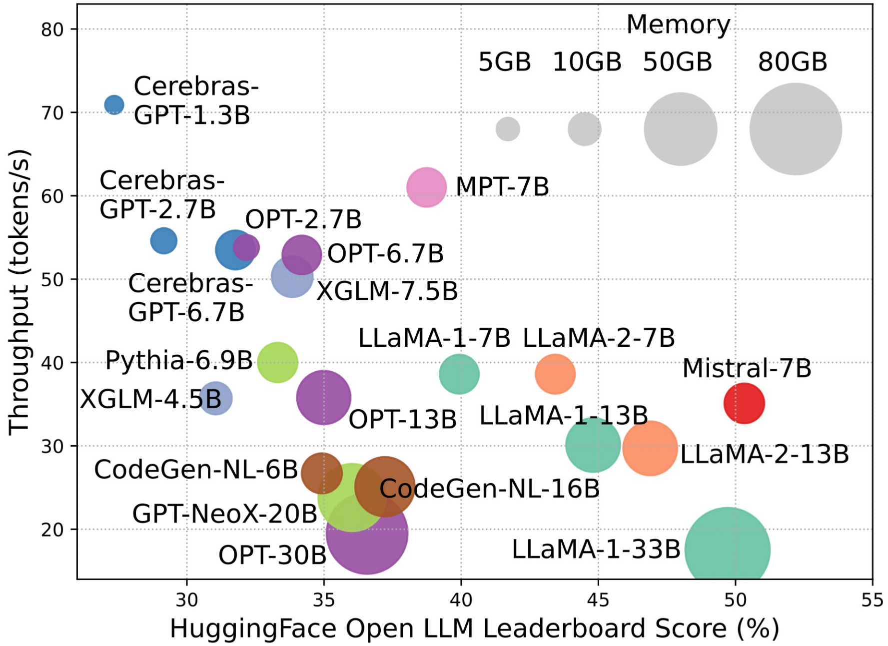




- **Considering that the training and deployment of large-scale LMs require a large amount of computing resources, Efficient-LMs are more cost-effective in actual production environments.**



Larger models are powerful but have exponential training costs[1]



Larger models use more memory and are slower at inference [1]

[1] Efficient Large Language Models: A Survey