rebuttal

> W1. Training Overhead

Train on 9900 * 10 = 99000 samples, on Llama3.1-8b-inst:

	Ours	Full parameter	LoRA (r=8, alpha=16)
Time	44 min	10 hr 8 min	46 min
VRAM	16.29 GiB	77.15 GiB	56.69 GiB

> W2. 8:2 experiments

For time reason, we only conducted the en/zh translation experiments.

PPL, With instruction						
↓translate to→	en	zh	fr	es	de	ru
en	0	1.062	0.563	0.532	0.631	**0.8**
zh	**0.996**	0	**0.926**	**0.972**	**1.043**	**1.**

PPL, With mask						
↓translate to→	en	zh	fr	es	de	ru
en	0	**0.991**	**0.523**	**0.484**	**0.599**	0.9
zh	1.006	0	0.938	0.991	1.069	1.1

ROUGE-L, With instruction						
↓translate to→	en	zh	fr	es	de	ru
en	0	0.549	0.614	0.641	0.591	**0.5**
zh	**0.63**	0	**0.531**	**0.544**	**0.497**	**0.4**

ROUGE-L, With mask						
↓translate to→	en	zh	fr	es	de	ru
en	0	**0.573**	**0.705**	**0.729**	**0.657**	0.3
zh	0.593	0	0.509	0.527	0.469	0.3

- Winogrande, on llama3.1-8b-inst（839 heads）

Instructed 5shot_gen_b36770	Instructed gen_6faab5	Mask 0-shot, direct generate the person name
66.77	62.98	**76.09**

- Hellaswag, on llama3.1-8b-inst

Instructed 10shot_gen_e42710	Instructed gen_6faab5	Instructed ppl Accuracy	mask ppl Accuracy (823 heads)	mask 0-shot, gen option (908 heads)
76.87	73.48	75.70	68.52	**80.31**

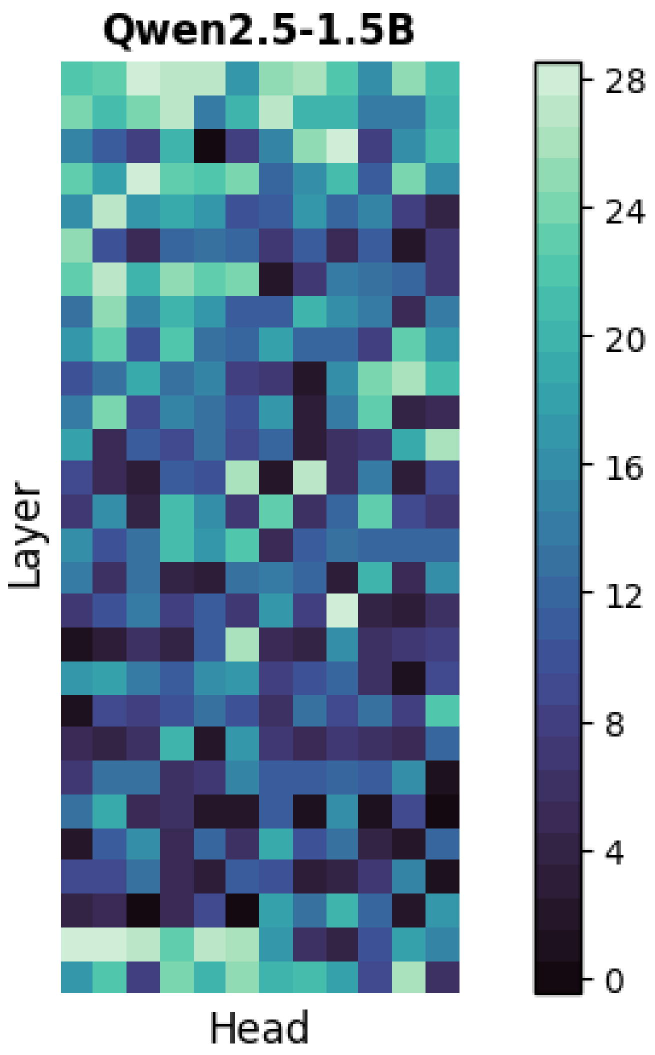
- ARC-C, on llama3.1-8b-inst（925 heads）

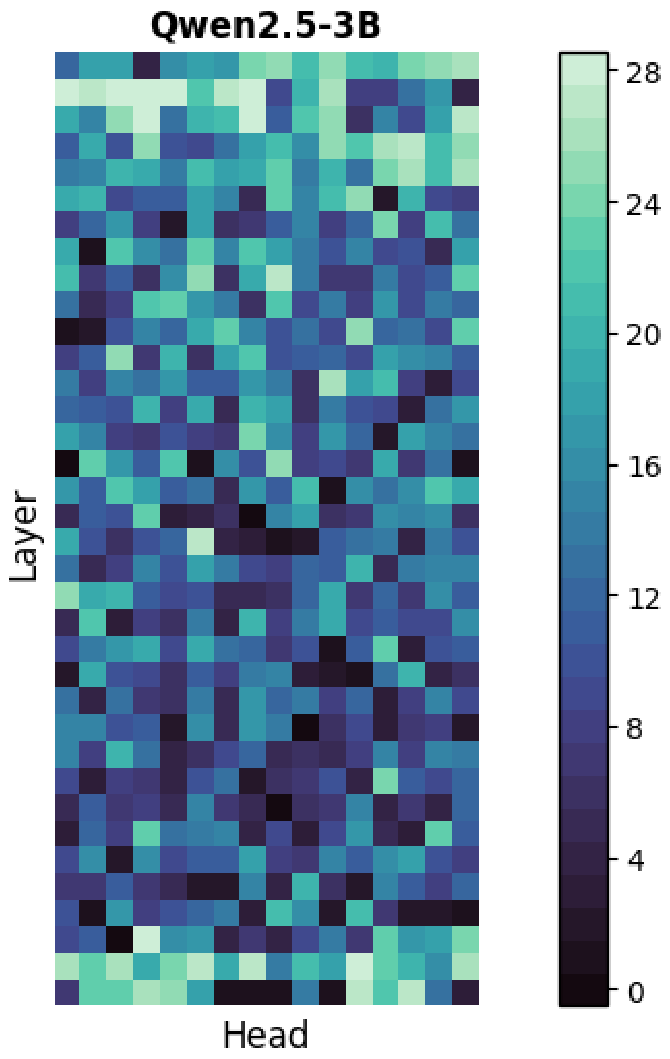
Instructed few_shot_gen_e9b043	Instructed gen_1e0de5	mask 0-shot, gen option
83.73	81.69	75.93

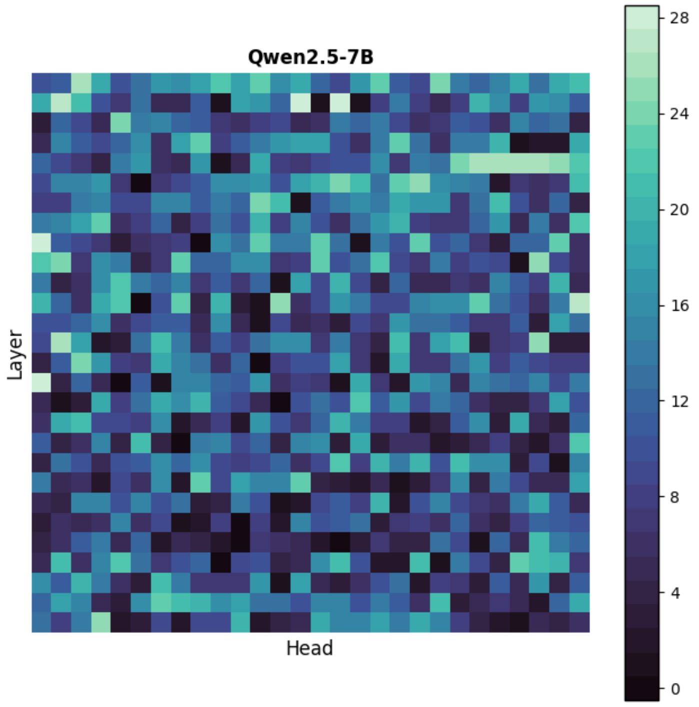
> Q1. Head pattern

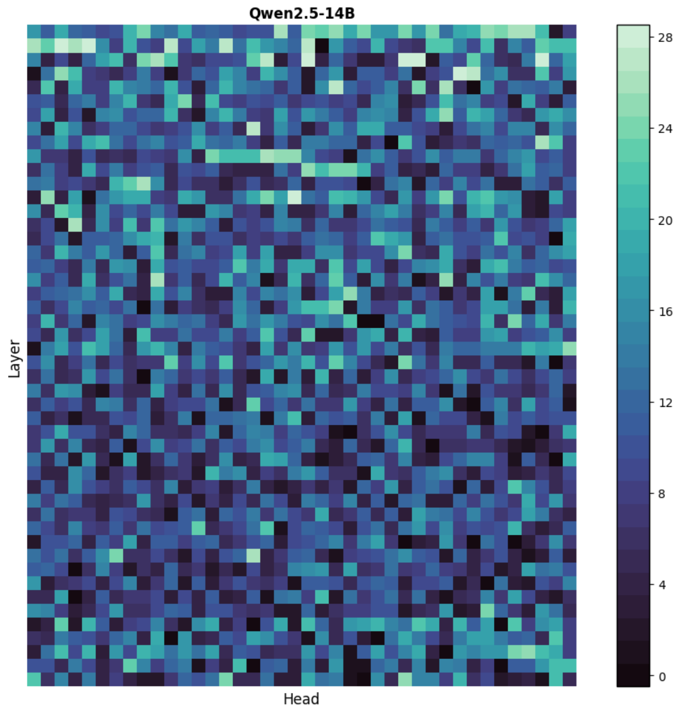
We train 28 simple tasks on Qwen2.5-1.5B/3B/7B/14B, and for each head we show how many tasks use it. Due to sparsity, as the model size increases, each attention head tends to be responsible for fewer tasks on average.

Model	1.5B	3B	7B	14B
Average tasks per head	12.97	12.94	11.07	10.61









> Q2: few-shot learning

1. When using the mask obtained from 0-shot training for 5-shot inference, the performance ranking is: instructed 5-shot > mask 0-shot > mask 5-shot > no-inst 5-shot.

2. For the tasks in Table 2, using a mask trained with 5-shot and then performing mask 5-shot inference yields better performance than instructed 5-shot.
3. We combined multiple simple task datasets into a hybrid dataset, where each sample may come from different tasks but includes 5-shot examples from its corresponding task. After training the mask on this hybrid dataset, the model was able to enhance its 5-shot context learning performance (including tasks that were not seen during training).

	antonym	capitalize_first_letter	conll2003_location	fruit_v_animal_3	product-company	sentiment	Average on all 35 tasks
Instruction, few-shot	0.67	1.00	0.72	1.00	0.84	0.97	0.84
Mask, 0-shot Trained on 0-shot	0.76	1.00	0.92	0.98	0.83	0.86	0.81
No instruction, few-shot	0.56	0.95	0.61	0.98	0.77	0.96	0.67
Mask, few-shot Trained on 0-shot	0.57	1.00	0.74	1.00	0.83	0.36	0.70
Mask, few-shot Trained on task few-shot	0.73	0.99	0.95	1.0	0.88	0.99	
Mask, few-shot Trained on hybrid few-shot	0.69	1.0	0.82	1.0	0.88	0.98	

> Q3. DuoAttention

Thank you for providing the paper, this is an interesting work. DuoAttention [1] shares a similar approach with our work, as both methods train a weight for each attention head and only optimize that weight. However, there are several key differences between our work and [1] in terms of research objectives and implementation details:

- The research focus of [1] is on long-context scenarios, where it improves inference efficiency by controlling different attention window sizes. In contrast, our work focuses on a broader range of task scenarios and the interpretability of model functionality. We treat attention heads as nodes in the information flow and study their impact on model behaviors.

- The weights we train do not operate on the attention scores (inside softmax) but instead affect the final output of the entire attention head. We discuss the results of training weights on attention scores in App. C.3.2.
- In [1], the outputs of the same attention head across different window sizes are combined through weighted mixing during training. Additionally, a constraint term is introduced to encourage the model to reduce the number of full-window attention heads. In our work, however, the output of each attention head is injected into the final output in a hard 0-1 manner from the stage of training, without applying a constraint term. (Although our code contains segments related to the constraint term, it is not actually used when calling the bash script.) This allows the model to naturally switch its behavior during the weight training process.

> Q4. Long-context Scenario

As our training tasks typically involve only one or two sentences without long contexts, we may discard attention heads related to long-context processing. In long-context translation tasks, we have observed that the model can translate within a context of 300 tokens and exhibit translation behavior within 500 tokens. However, beyond this length, it loses its ability to translate.

> Q5. Masking Ratio

Since the proportion of attention heads we choose is entirely controlled by the training process (to achieve behavior switching), we have limited flexibility in adjusting this proportion. At present, we have some following insights:

- Whether the mask is initialized to 1 or 0 (Line 312-317, left column) results in a number difference of approximately 200(translation tasks)~500(simple tasks) attention heads in the final trained mask.
- Removing attention heads sequentially based on their weights is a good approach, as it allows for rapid behavior switching while minimizing the number of heads removed. The removal process can continue before the model loses its language capability (Figure 2).