北京大學
PEKING UNIVERSITY

THE SUPER WEIGHT IN LARGE LANGUAGE MODELS

Apple

THE SUPER WEIGHT IN LARGE LANGUAGE MODELS

超权重（super weight）

当LLMs的大小达到一定规模时，一部分中间特征包含了异常大的离群值，被称作异常值。其中，有一部分微小但非常重要的异常值，被称作超权重。

若修剪去超权重，会对模型性能造成毁灭性的影响：

- 完全摧毁文本生成的能力
- Zero-shot accuracy也会降至0
- Stopword的生成概率大大增加

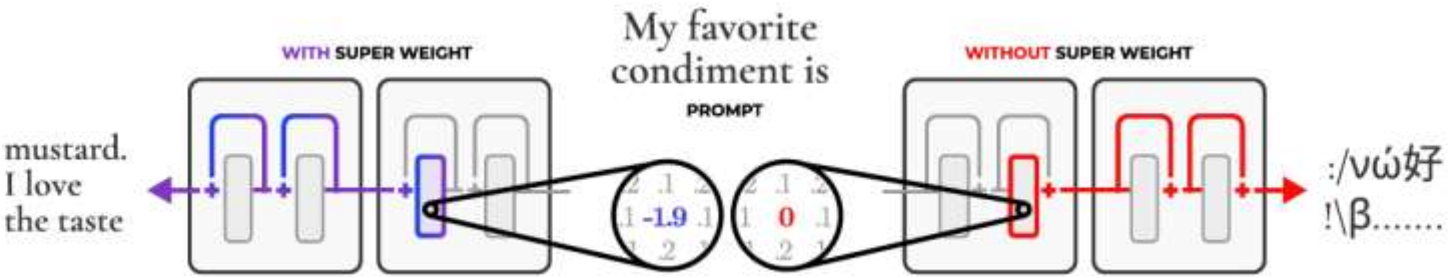


Figure 1: Super Weight Phenemenon. We discover that pruning a single, special scalar, which we call the *super weight*, can completely destroy a Large Language Model's ability to generate text. On the left, the original Llama-7B, which contains a super weight, produces a reasonable completion. On the right, after pruning the super weight, Llama-7B generates complete gibberish. As we show below, this qualitative observation has quantitative impact too: zero-shot accuracy drops to guessing and perplexity increases by orders of magnitude.

THE SUPER WEIGHT IN LARGE LANGUAGE MODELS

北京大學 PEKING UNIVERSITY

超激活（super activation）

超权重能够放大输入的激活内点，生成很大的激活值，这个现象被称作超激活。

而且无论prompt是什么且如何微调，这个超激活的量级与位置都不会改变，且会在多个层中持续存在。

同时超激活也会抑制stopwords的生成。

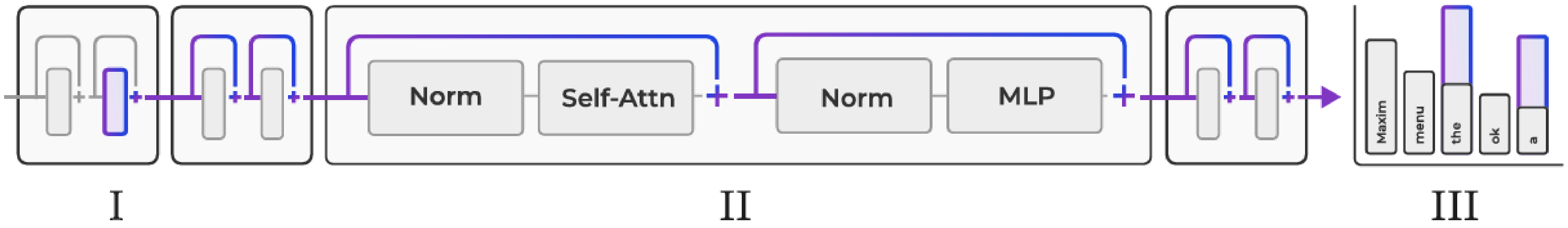


Figure 2: How Super Weights behave. *I:* Super weights are often found in an early layer's down projection, indicated with a blue-purple box. The super weight immediately creates an incredibly large-magnitude super activation. *II:* Super activations are propagated through skip connections, indicated with blue-purple lines. *III:* This has a net effect of suppressing stopword likelihoods in the final logits. Removing the super weight causes stopword likelihood skyrocket, indicated with the blue-purple stacked bars. See Appendix A.3.

THE SUPER WEIGHT IN LARGE LANGUAGE MODELS

如何识别超权重

超权重还具有另一个特征：它通常出现于mlp.down_proj中，且位于浅层。

在输入down_proj之前，gate与up_proj的乘积已产生了较大的激活值，而down_proj蕴含的超权重进一步放大且创造了超激活。

有公式$Y = XW^T$，而X_{ik}与W_{jk}都是偏离很大的异常值，因此产生了超激活值Y_{ij}。

因此可以通过检查层之间的down_proj的输入输出分布的峰值来定位超权重。

因为超权重的位置并不会随着prompt而改变，因此只需要一个prompt就可以成功定位。

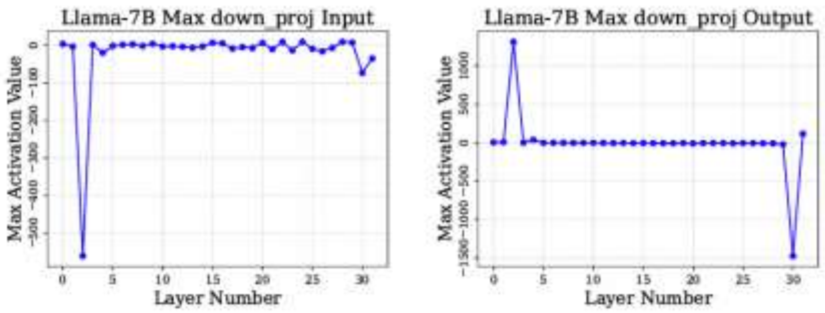


Figure 3: How to identify the Super Weight for Llama-7B. down_proj input features a large maximum-magnitude activation only in Layer 2, where the super activation first appeared. The value's channel index, e.g., 7003, tells the row of SW. down_proj output likewise features a large maximum-magnitude activation at Layer 2. This value's channel index, e.g., 3968, gives us the column of the SW.

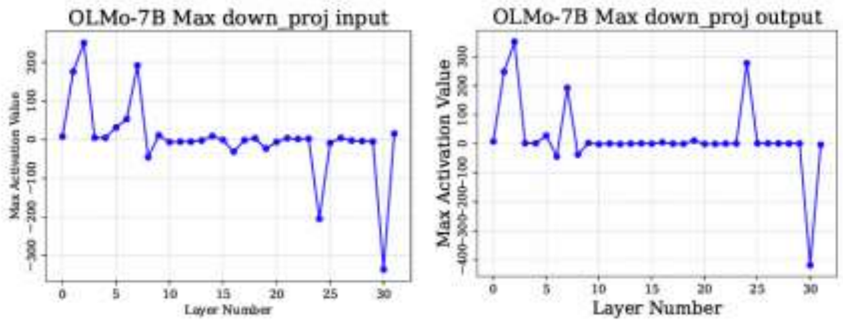
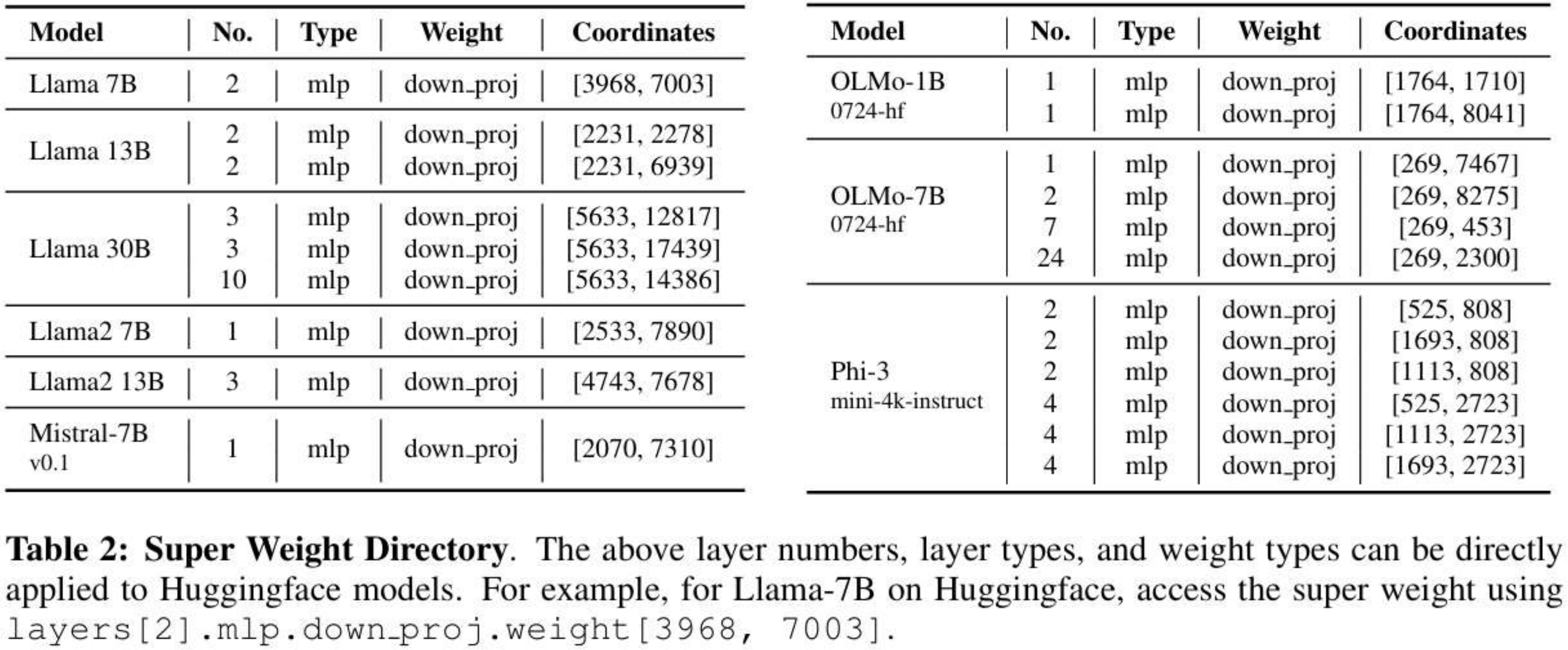


Figure 10: Maximum-magnitude activation of down_proj across all transformer layers of OLMo-7B.

THE SUPER WEIGHT IN LARGE LANGUAGE MODELS



Model	No.	Type	Weight	Coordinates
Llama 7B	2	mlp	down_proj	[3968, 7003]
Llama 13B	2 2	mlp mlp	down_proj down_proj	[2231, 2278] [2231, 6939]
Llama 30B	3 3 10	mlp mlp mlp	down_proj down_proj down_proj	[5633, 12817] [5633, 17439] [5633, 14386]
Llama2 7B	1	mlp	down_proj	[2533, 7890]
Llama2 13B	3	mlp	down_proj	[4743, 7678]
Mistral-7B v0.1	1	mlp	down_proj	[2070, 7310]

Model	No.	Type	Weight	Coordinates
OLMo-1B 0724-hf	1 1	mlp mlp	down_proj down_proj	[1764, 1710] [1764, 8041]
OLMo-7B 0724-hf	1 2 7 24	mlp mlp mlp mlp	down_proj down_proj down_proj down_proj	[269, 7467] [269, 8275] [269, 453] [269, 2300]
Phi-3 mini-4k-instruct	2 2 2 4 4 4	mlp mlp mlp mlp mlp mlp	down_proj down_proj down_proj down_proj down_proj down_proj	[525, 808] [1693, 808] [1113, 808] [525, 2723] [1113, 2723] [1693, 2723]

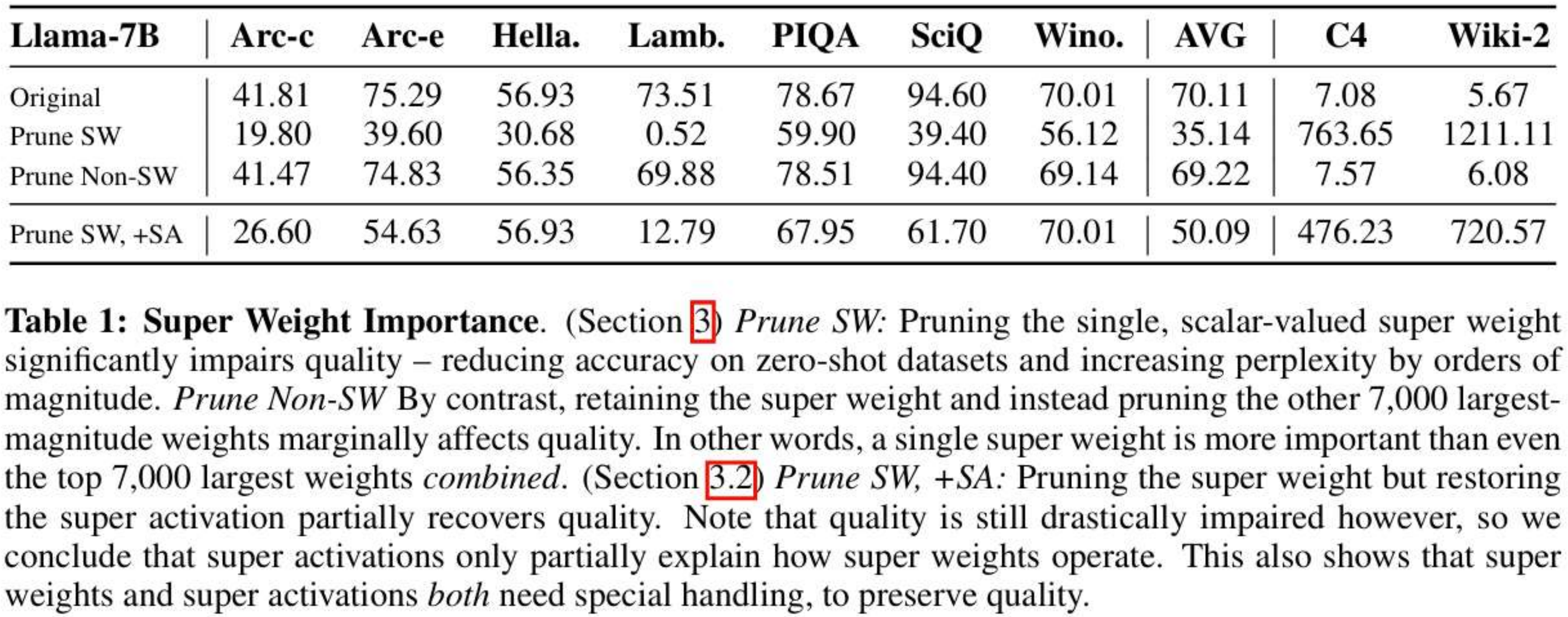
Table 2: Super Weight Directory. The above layer numbers, layer types, and weight types can be directly applied to Huggingface models. For example, for Llama-7B on Huggingface, access the super weight using `layers[2].mlp.down_proj.weight[3968, 7003]`.

同时还通过实验可得，改变prompt或进行指令微调都不会影响超权重的位置。

THE SUPER WEIGHT IN LARGE LANGUAGE MODELS

超权重发挥作用的方式

- 产生超激活
- 改变输出分布



Llama-7B	Arc-c	Arc-e	Hella.	Lamb.	PIQA	SciQ	Wino.	AVG	C4	Wiki-2
Original	41.81	75.29	56.93	73.51	78.67	94.60	70.01	70.11	7.08	5.67
Prune SW	19.80	39.60	30.68	0.52	59.90	39.40	56.12	35.14	763.65	1211.11
Prune Non-SW	41.47	74.83	56.35	69.88	78.51	94.40	69.14	69.22	7.57	6.08
Prune SW, +SA	26.60	54.63	56.93	12.79	67.95	61.70	70.01	50.09	476.23	720.57

Table 1: Super Weight Importance. (Section 3) *Prune SW:* Pruning the single, scalar-valued super weight significantly impairs quality – reducing accuracy on zero-shot datasets and increasing perplexity by orders of magnitude. *Prune Non-SW* By contrast, retaining the super weight and instead pruning the other 7,000 largest-magnitude weights marginally affects quality. In other words, a single super weight is more important than even the top 7,000 largest weights *combined*. (Section 3.2) *Prune SW, +SA:* Pruning the super weight but restoring the super activation partially recovers quality. Note that quality is still drastically impaired however, so we conclude that super activations only partially explain how super weights operate. This also shows that super weights and super activations *both* need special handling, to preserve quality.

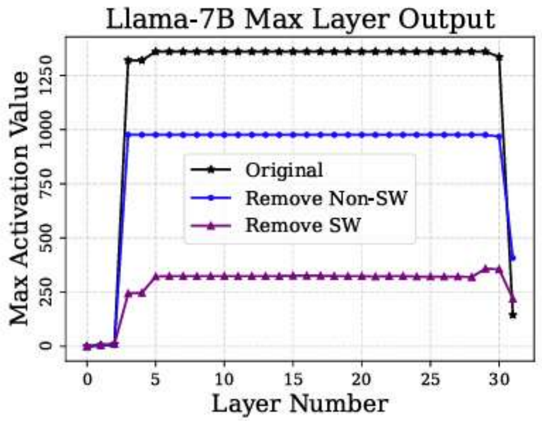


Figure 4: The super activation persists throughout the entire model, at exactly the same magnitude, starting after Layer 2. Pruning the super weight decreases the super activation's magnitude by 75%.

超权重发挥作用的方式

- 产生超激活
- 改变输出分布

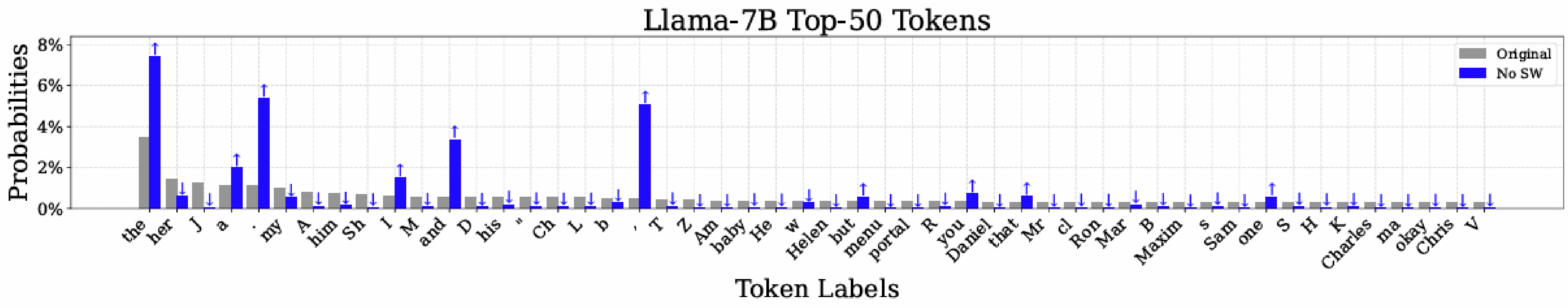


Figure 11: Output token distribution before and after removing the super weight on Llama-7B.

THE SUPER WEIGHT IN LARGE LANGUAGE MODELS

超权重的作用

适当调大超权重能够提升模型准确性

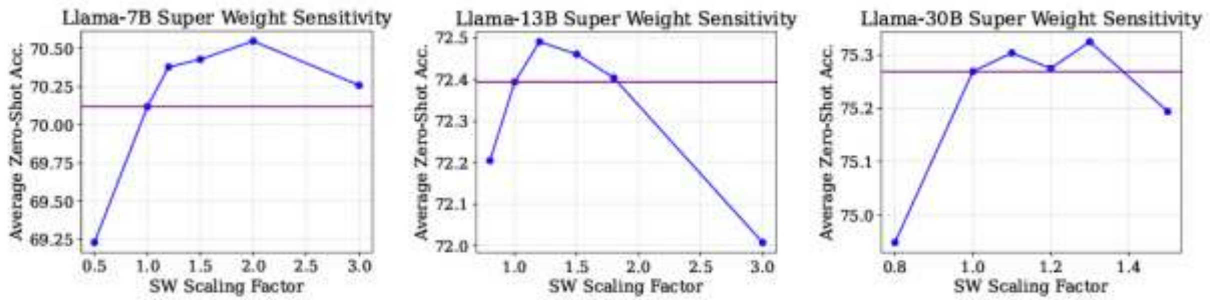


Figure 6: Amplifying super weight improves quality. Across model sizes, we consistently observe that there exists *some* scaling where quality is improved. Although the quality improvement is miniscule, a consistent and noticeable trend is surprising, given we're changing only one scalar out of billions. The purple line is the original model's zero-shot average accuracy.

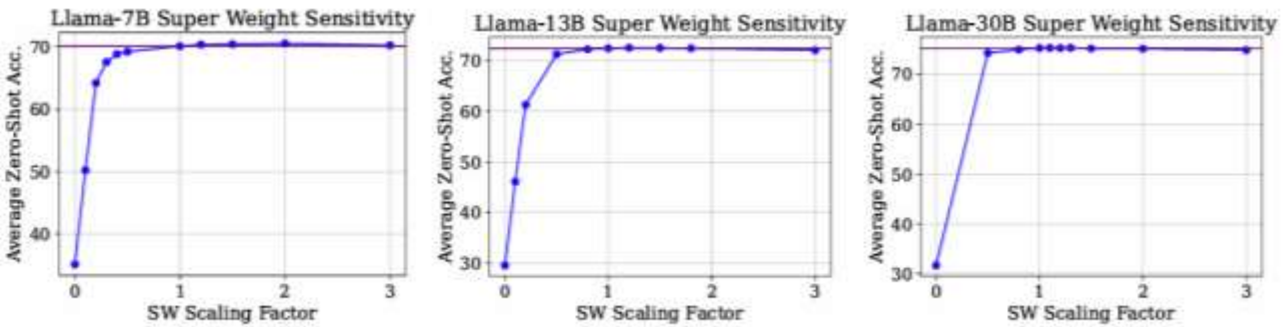


Figure 8: Amplifying super weight improves quality. Full results for scaling super weight from 0 to 3.

超异常值感知量化

$$Q(\mathbf{X}) = \text{Round}\left(\frac{\mathbf{X} - \textsc{min}(\mathbf{X})}{\Delta}\right), Q^{-1}(\hat{\mathbf{X}}) = \Delta \cdot \hat{\mathbf{X}} + \textsc{min}(\mathbf{X})$$

$$\Delta = \frac{\textsc{max}(\mathbf{X}) - \textsc{min}(\mathbf{X})}{2^{N-1} - 1}$$

一些异常值会大大降低量化质量，这些异常值被称作超异常值。

超异常值对模型质量的重要性是不成比例的，无法与其他参数/激活值按照同一个策略进行放缩。因此提出了一种在量化过程中保留超异常值的量化方法。

具体方法为：在量化时保留超异常值，在去量化后恢复超异常值。

$$\hat{A} = \textsc{Restore}(Q^{-1}(Q(\textsc{Replace}(A)))$$

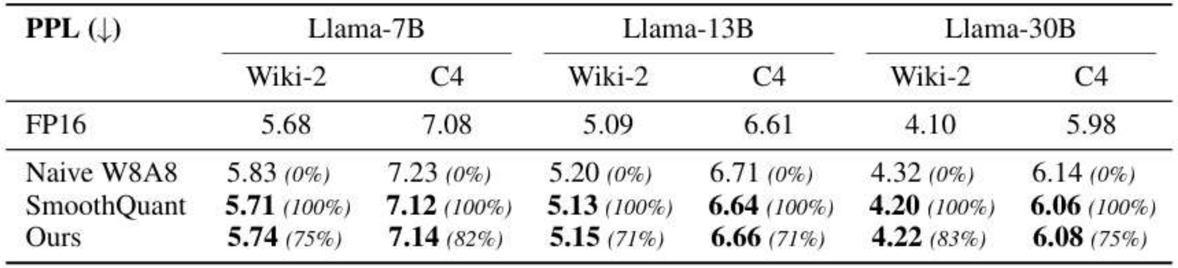
激活量化

$$\hat{W} = \textsc{Restore}(Q^{-1}(Q(\textsc{Clip}_z(W)))$$

权重量化

THE SUPER WEIGHT IN LARGE LANGUAGE MODELS

超异常值感知量化



PPL (↓)	Llama-7B		Llama-13B		Llama-30B	
	Wiki-2	C4	Wiki-2	C4	Wiki-2	C4
FP16	5.68	7.08	5.09	6.61	4.10	5.98
Naive W8A8	5.83 *(0%)*	7.23 *(0%)*	5.20 *(0%)*	6.71 *(0%)*	4.32 *(0%)*	6.14 *(0%)*
SmoothQuant	**5.71** *(100%)*	**7.12** *(100%)*	**5.13** *(100%)*	**6.64** *(100%)*	**4.20** *(100%)*	**6.06** *(100%)*
Ours	**5.74** *(75%)*	**7.14** *(82%)*	**5.15** *(71%)*	**6.66** *(71%)*	**4.22** *(83%)*	**6.08** *(75%)*

Table 3: Round-to-nearest with super-activation handling is competitive. *W8A8* is the baseline 8-bit weight and activation quantization, and the small italicized, parenthesized percentages denote what percentage of SmoothQuant's quality improvement is retained. We observe that a naive round-to-nearest, while handling a single scalar super activation per tensor, is competitive with SmoothQuant. Note that SmoothQuant uses calibration data to compute scales, whereas our method does not require data.

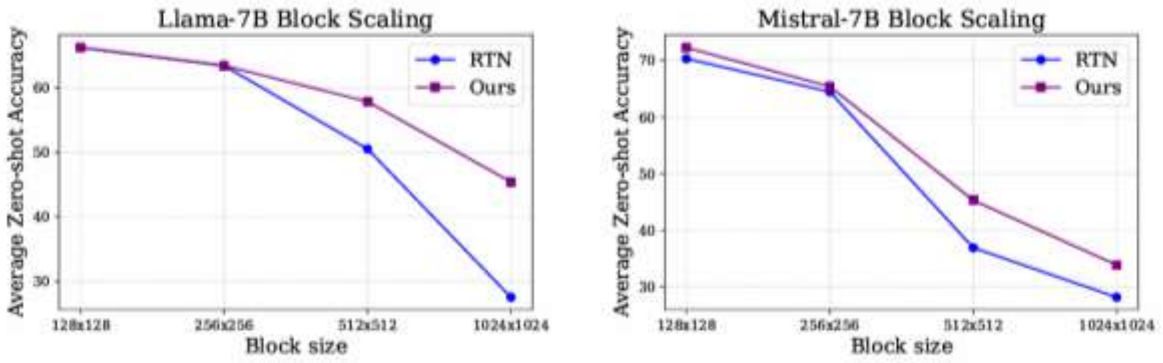


Figure 7: Restoring super weight improves block scaling. Smaller block sizes are often used to handle outliers implicitly. We note that block sizes can scale slightly more gracefully by just handling the single scalar-valued super weight.

北京大學
PEKING UNIVERSITY

Sparse Matrix in Large Language Model Fine-tuning

Sparse Matrix in Large Language Model Fine-tuning

LoRA的局限性：和FT之间性能存在差异，并且随着r（可训练参数）的增加会经历性能平台期甚至还会下降。

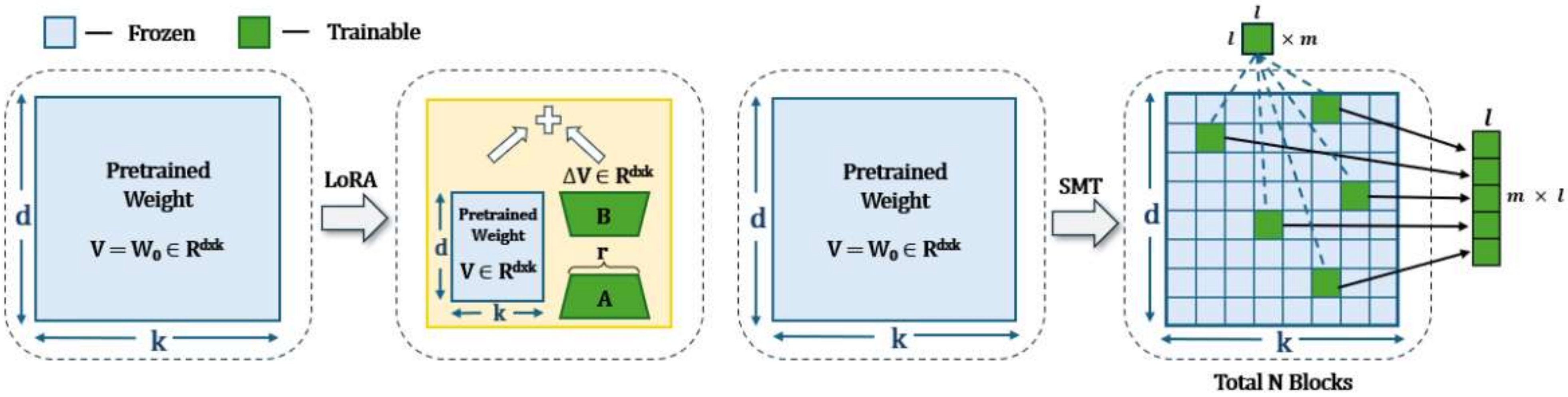


Figure 1: Differences between low-rank adaption method LoRA and SMT. Upper picture dedicates adaption approach in LoRA and lower picture represents the sub-matrices sparsity approach in SMT.

核心思想：找到与下游任务性能最相关的子矩阵，只对这些子矩阵进行微调

Sparse Matrix in Large Language Model Fine-tuning

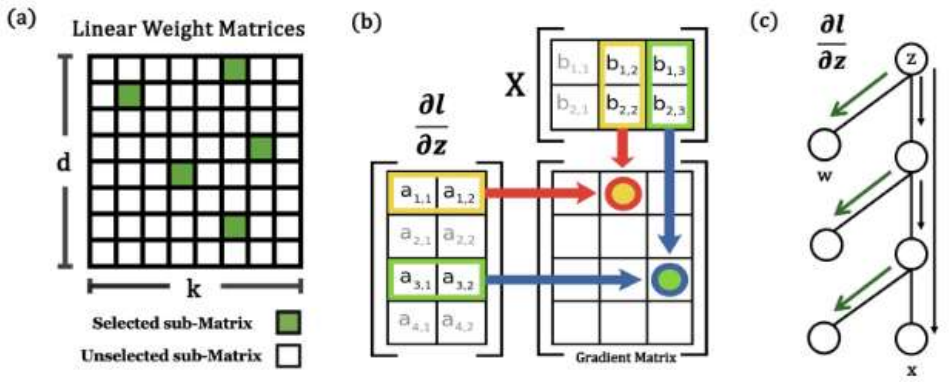
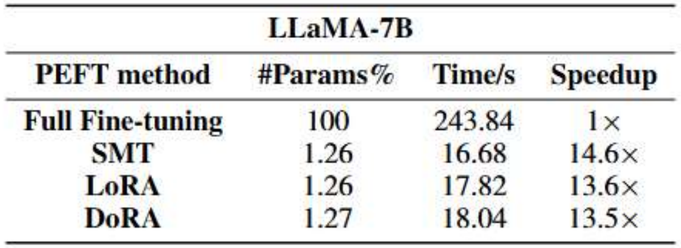


Figure 2: (a) A sparse weight matrix W. The green sub-matrices with significant gradients can be updated. (b) Backward propagation calculation for partial gradient for weight matrix w. (c) Computation graph in auto-differential systems.

Table 1: The experiments involved Full Fine-Tuning, SMT, LoRA, and DoRA on 4× A100 40GB GPUs using data parallel, with a batch size of 16. Communication between the GPU and CPU was facilitated via PCIe-G4.



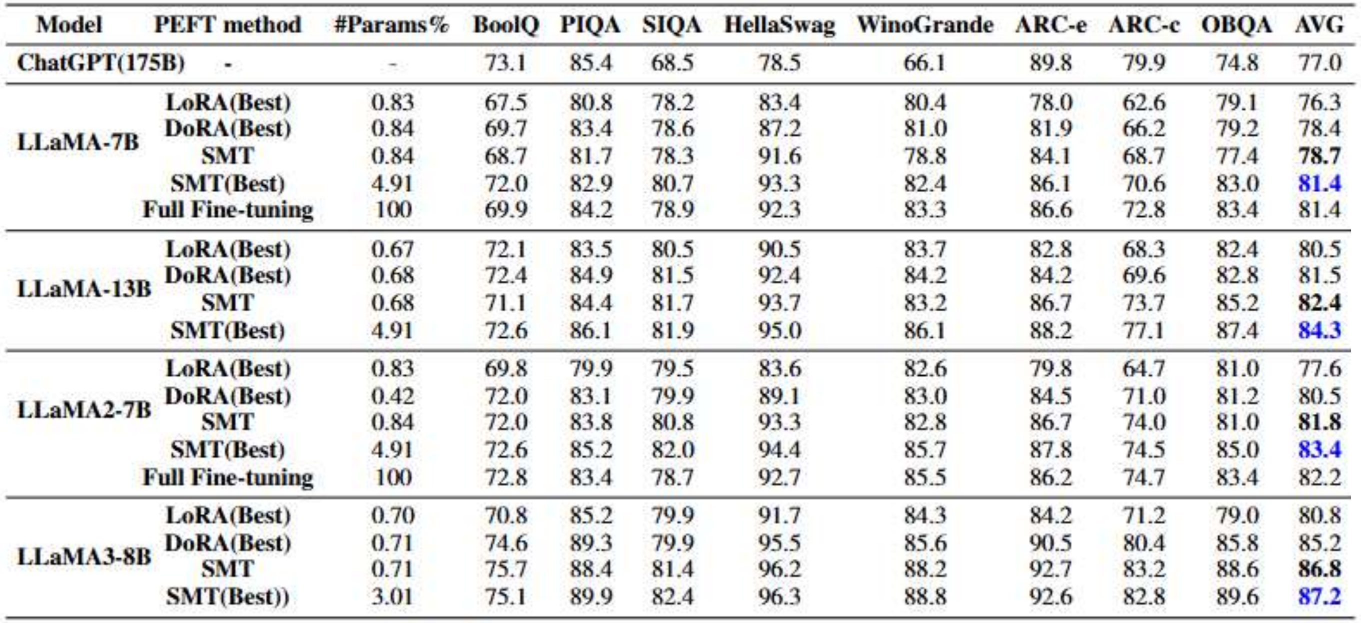
LLaMA-7B			
PEFT method	**#Params%**	**Time/s**	**Speedup**
Full Fine-tuning	100	243.84	1×
SMT	1.26	16.68	14.6×
LoRA	1.26	17.82	13.6×
DoRA	1.27	18.04	13.5×

$$\nabla_x f(x) = \frac{\partial l}{\partial Z} \cdot W; \qquad \nabla_W f(x) = \frac{\partial l}{\partial Z} \cdot x$$

权重梯度计算公式：只需选择相应的数据进行计算，从而减少了其他不必要的开销，将计算成本降低至0.5%

热身阶段：进行100次迭代，得到梯度矩阵，对子矩阵求平均值，选择梯度平均值最大的子矩阵

Sparse Matrix in Large Language Model Fine-tuning



Model	PEFT method	#Params%	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	AVG
ChatGPT(175B)	-	-	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA-7B	LoRA(Best)	0.83	67.5	80.8	78.2	83.4	80.4	78.0	62.6	79.1	76.3
	DoRA(Best)	0.84	69.7	83.4	78.6	87.2	81.0	81.9	66.2	79.2	78.4
	SMT	0.84	68.7	81.7	78.3	91.6	78.8	84.1	68.7	77.4	**78.7**
	SMT(Best)	4.91	72.0	82.9	80.7	93.3	82.4	86.1	70.6	83.0	**81.4**
	Full Fine-tuning	100	69.9	84.2	78.9	92.3	83.3	86.6	72.8	83.4	81.4
LLaMA-13B	LoRA(Best)	0.67	72.1	83.5	80.5	90.5	83.7	82.8	68.3	82.4	80.5
	DoRA(Best)	0.68	72.4	84.9	81.5	92.4	84.2	84.2	69.6	82.8	81.5
	SMT	0.68	71.1	84.4	81.7	93.7	83.2	86.7	73.7	85.2	**82.4**
	SMT(Best)	4.91	72.6	86.1	81.9	95.0	86.1	88.2	77.1	87.4	**84.3**
LLaMA2-7B	LoRA(Best)	0.83	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
	DoRA(Best)	0.42	72.0	83.1	79.9	89.1	83.0	84.5	71.0	81.2	80.5
	SMT	0.84	72.0	83.8	80.8	93.3	82.8	86.7	74.0	81.0	**81.8**
	SMT(Best)	4.91	72.6	85.2	82.0	94.4	85.7	87.8	74.5	85.0	**83.4**
	Full Fine-tuning	100	72.8	83.4	78.7	92.7	85.5	86.2	74.7	83.4	82.2
LLaMA3-8B	LoRA(Best)	0.70	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	DoRA(Best)	0.71	74.6	89.3	79.9	95.5	85.6	90.5	80.4	85.8	85.2
	SMT	0.71	75.7	88.4	81.4	96.2	88.2	92.7	83.2	88.6	**86.8**
	SMT(Best))	3.01	75.1	89.9	82.4	96.3	88.8	92.6	82.8	89.6	**87.2**

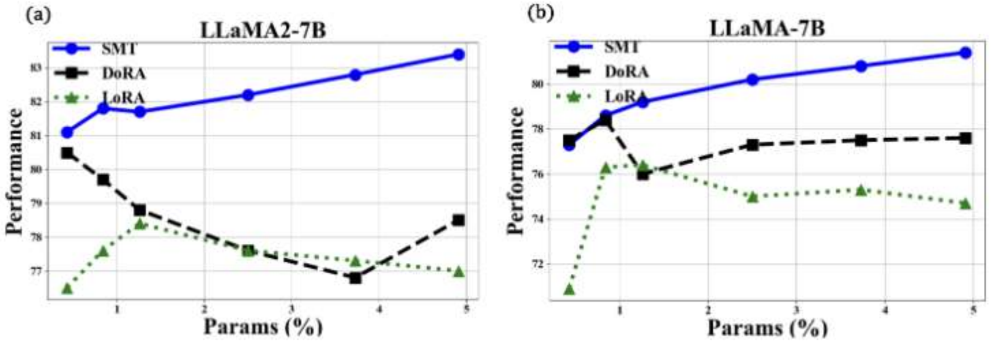


Figure 3: Accuracy comparison of LoRA, DoRA, and SMT under different scaling of trainable parameters on commonsense reasoning datasets.

Sparse Matrix in Large Language Model Fine-tuning

通过实验发现SMT所选中的子矩阵大多位于注意力机制的QKV上，而非MLP上。

而在QKV中，大多位于V中。也通过消融实验证实了这一点。一方面说明LLM的记忆部分主要位于V中，一方面也证明了SMT能够有效选择包含关键内存部分的子矩阵。

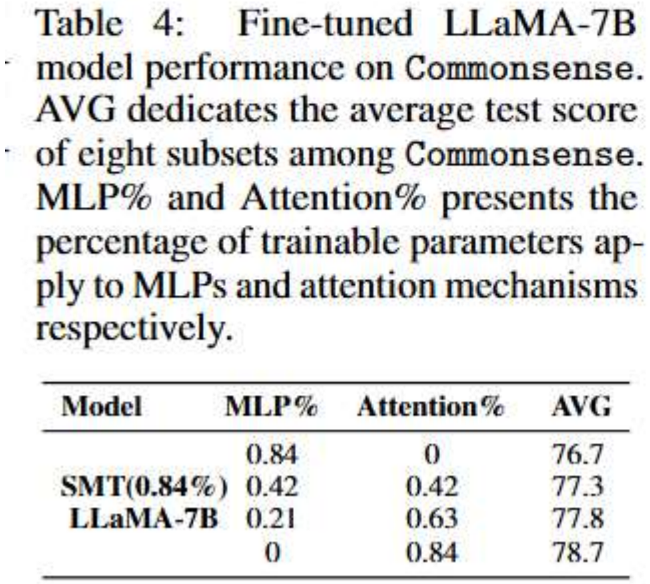
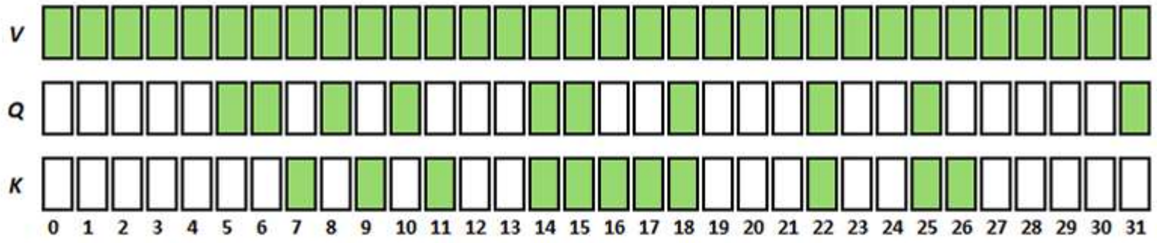


Table 4: Fine-tuned LLaMA-7B model performance on Commonsense. AVG dedicates the average test score of eight subsets among Commonsense. MLP% and Attention% presents the percentage of trainable parameters apply to MLPs and attention mechanisms respectively.

Model	MLP%	Attention%	AVG
SMT(0.84%) LLaMA-7B	0.84	0	76.7
	0.42	0.42	77.3
	0.21	0.63	77.8
	0	0.84	78.7

Figure 4: A visualization of trainable Q, K, V layers when fine-tuning 0.86% trainable parameters on LLaMA-7B. LLaMA-7B has 32 layers of MLPs, each contains a Q vector, a K vector, and a V vector. White layers are frozen and green layers contain trainable parameters.



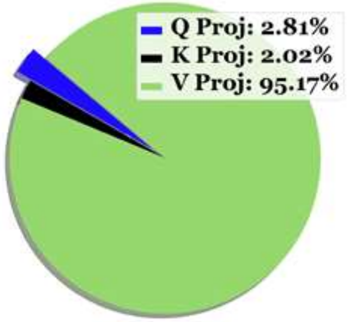


Figure 5: Distribution of trainable parameters among Q, K, V.

北京大學
PEKING UNIVERSITY

DeepSpeed-MoE

Microsoft

PR-MoE

论点1：浅层（靠输入端）学习共同特征；深层（靠输出端）学习个性特征。

每层所需的expert数是否一致? → 提出金字塔MoE（Pyramid MoE）结构

论点2：使用多位专家(k)会使gating算法的开支大幅增加；而增加专家数会增大内存开销。

是否可以有一个fixed的expert加上一个额外的可训练的expert? → 提出残差MoE（Residual MoE）

PR-MoE

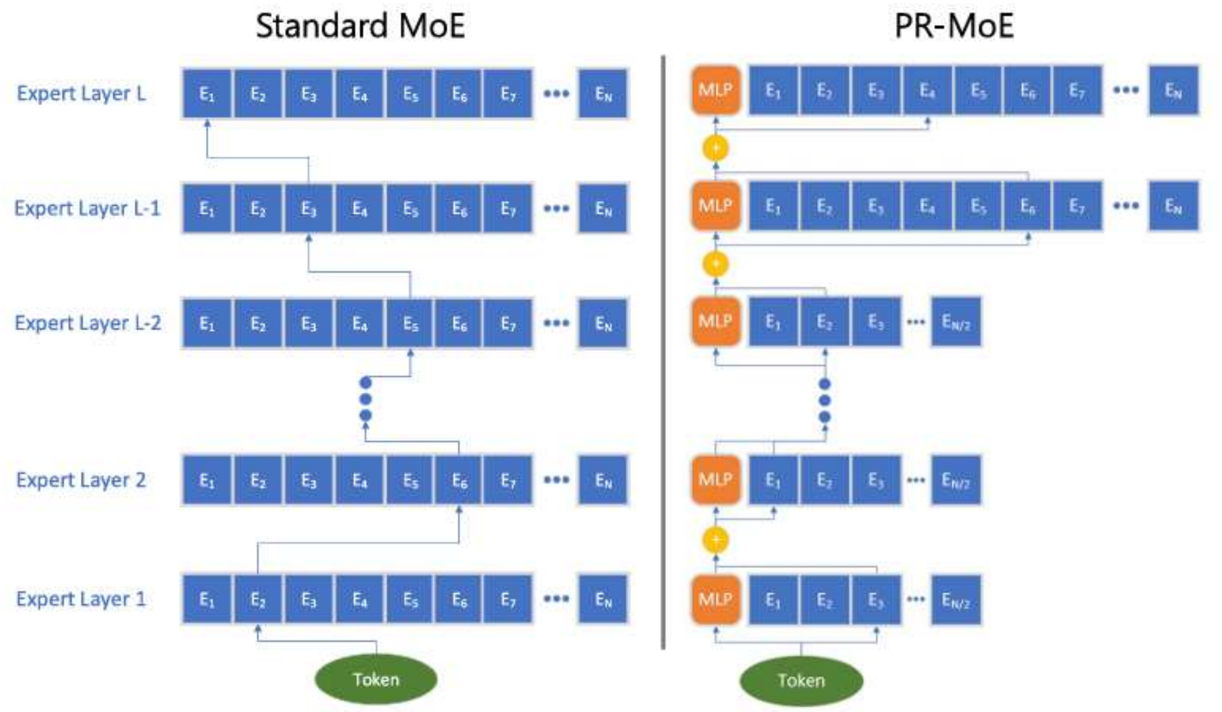
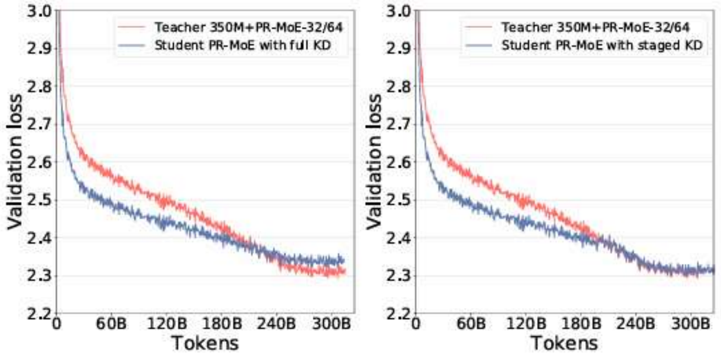


Figure 3: The illustration of standard MoE (left) and PR-MoE (right).

MoS



(a) (b)

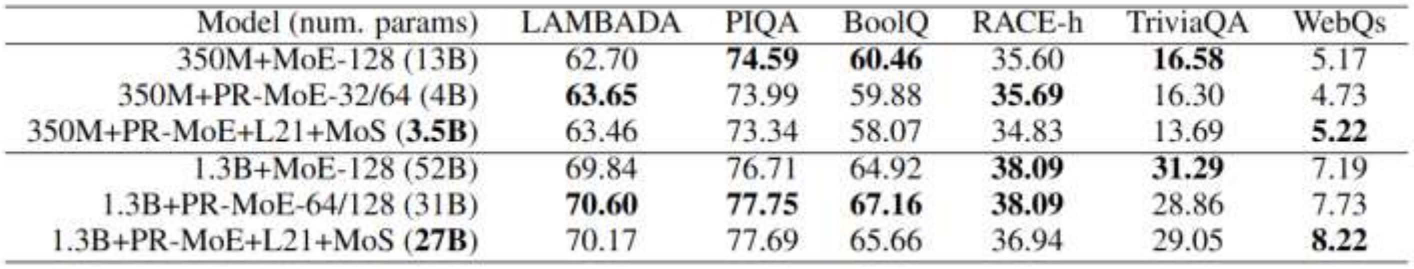
Figure 4: (a) The validation curves of training without distillation from scratch vs. performing knowledge distillation for the entire pre-training process. In the figure the student PR-MoE is trained with 21-layer. KD helps initially but starts to hurt accuracy towards the end of training. (b) The validation curves of training the student PR-MoE with staged knowledge distillation obtains almost the same validation loss as the teacher PR-MoE on 350M+PR-MoE.

对PR-MoE进行蒸馏

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}[\mathcal{L}(x;\theta) + \alpha \mathcal{L}_{KD}(x';\theta)], \qquad (1)$$

它采用了staged KD，该实验在400K steps时停止蒸馏

Table 2: Zero-shot evaluation comparison between standard MoE, PR-MoE, MoS.



Model (num. params)	LAMBADA	PIQA	BoolQ	RACE-h	TriviaQA	WebQs
350M+MoE-128 (13B)	62.70	**74.59**	**60.46**	35.60	**16.58**	5.17
350M+PR-MoE-32/64 (4B)	**63.65**	73.99	59.88	**35.69**	16.30	4.73
350M+PR-MoE+L21+MoS (**3.5B**)	63.46	73.34	58.07	34.83	13.69	**5.22**
1.3B+MoE-128 (52B)	69.84	76.71	64.92	**38.09**	**31.29**	7.19
1.3B+PR-MoE-64/128 (31B)	**70.60**	**77.75**	**67.16**	**38.09**	28.86	7.73
1.3B+PR-MoE+L21+MoS (**27B**)	70.17	77.69	65.66	36.94	29.05	**8.22**

LW2G

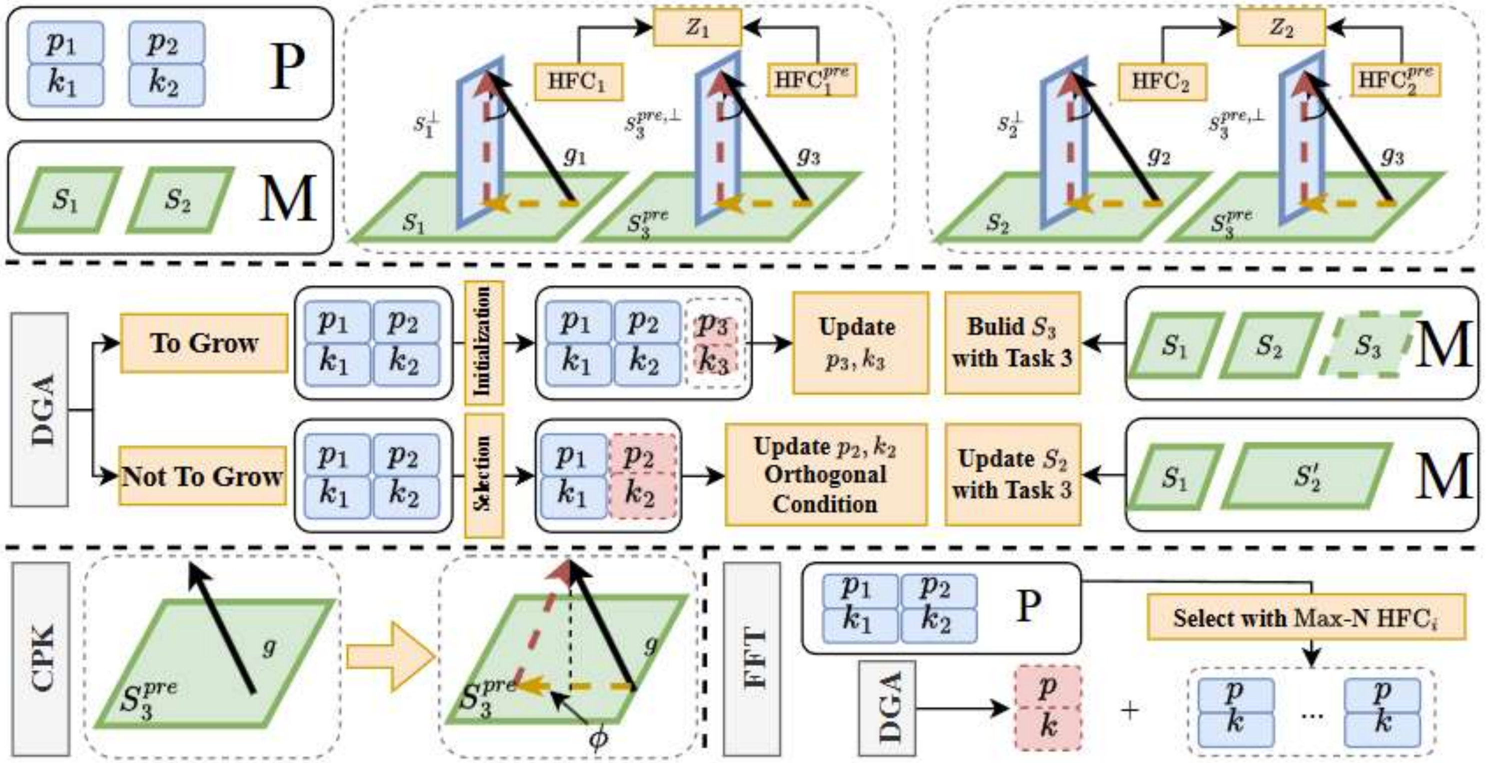


Figure 2: Illustration of three components in LW2G. Before learning task 3, assume there are two sets in $\mathcal{P} = \{(p_1, k_1), (p_2, k_2)\}$. In $\mathcal{P}$, blue represents frozen and unlearnable sets of prompts, whereas red represents learnable sets.

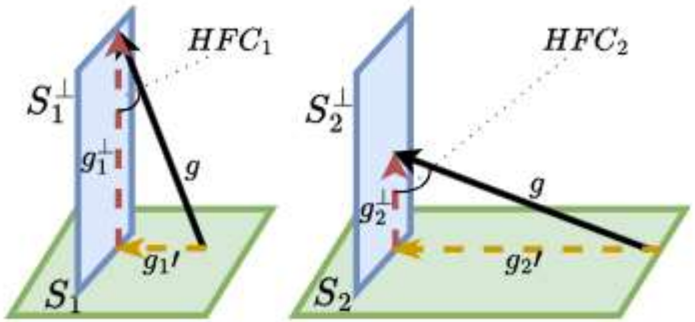


Figure 1: Illustration of HFC. $\mathcal{S}_i$ represents the feature space spanned by the old task i, while $\mathcal{S}_i^\perp$ denotes the orthogonal complement to $\mathcal{S}_i$. Then, $\text{HFC}(g, g_i^\perp)$ is denoted as HFC_i.

$$g_1 = \nabla_{(p_1,k_1)}\mathcal{L}_3(\mathcal{D}^3_{\text{sub}}).$$

$$\text{HFC}_1 = \text{HFC}(g_1, \text{Proj}_{\mathcal{S}_1^\perp}(g_1)).$$

$$\text{HFC}_1^{\text{pre}} = \text{HFC}(g_3, \text{Proj}_{\mathcal{S}_3^{\text{pre},\perp}}(g_3)),$$