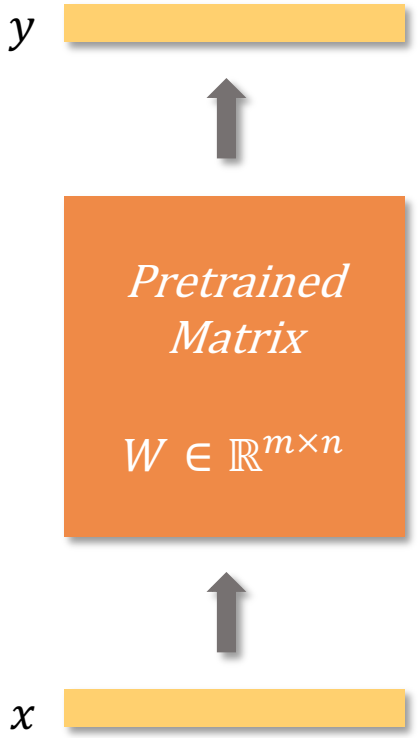
北京大学
PEKING UNIVERSITY

PiSSA: Principal Singular Values and Singular Vectors Adaptation of Large Language Models

Fanxu Meng, Zhaohui Wang, Muhan Zhang
Peking University

1. Parameter-Efficient Fine-Tuning

Full Parameters Fine-Tuning

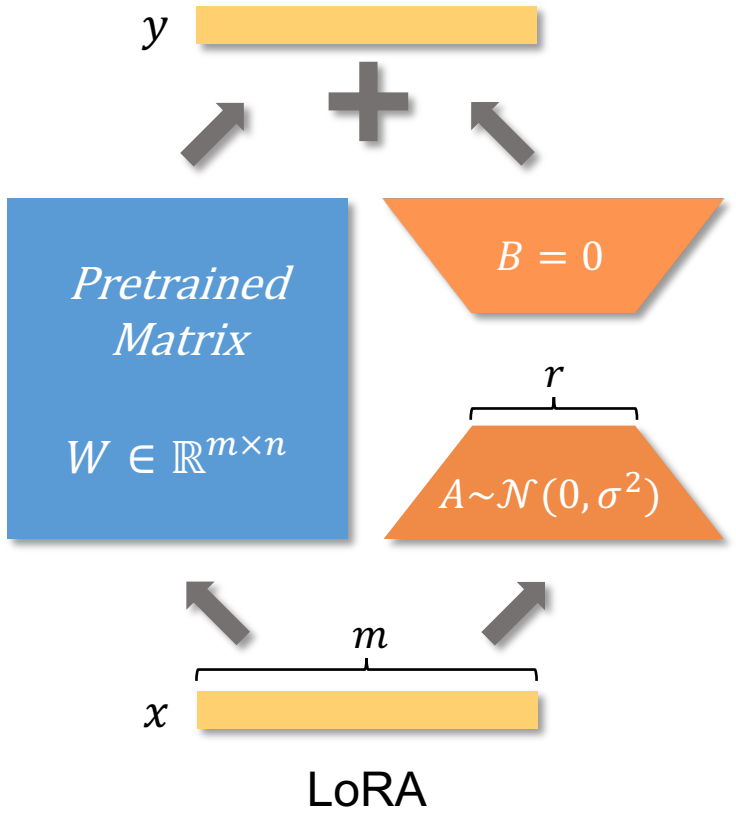


Full Parameters Fine-Tuning

- **Advantage**: straightforward and easy to use.
- **Disadvantage**: requires substantial computational resources.
  - Fine-tuning a LLaMA 65B in 16-bit requires over 780 GB of GPU memory [1].
  - The VRAM consumption for training GPT-3 175B reaches 1.2TB [2].

[1] QLoRA: Efficient Finetuning of Quantized LLMs
[2] LoRA: Low-Rank Adaptation of Large Language Models

Low-Rank Adaptation

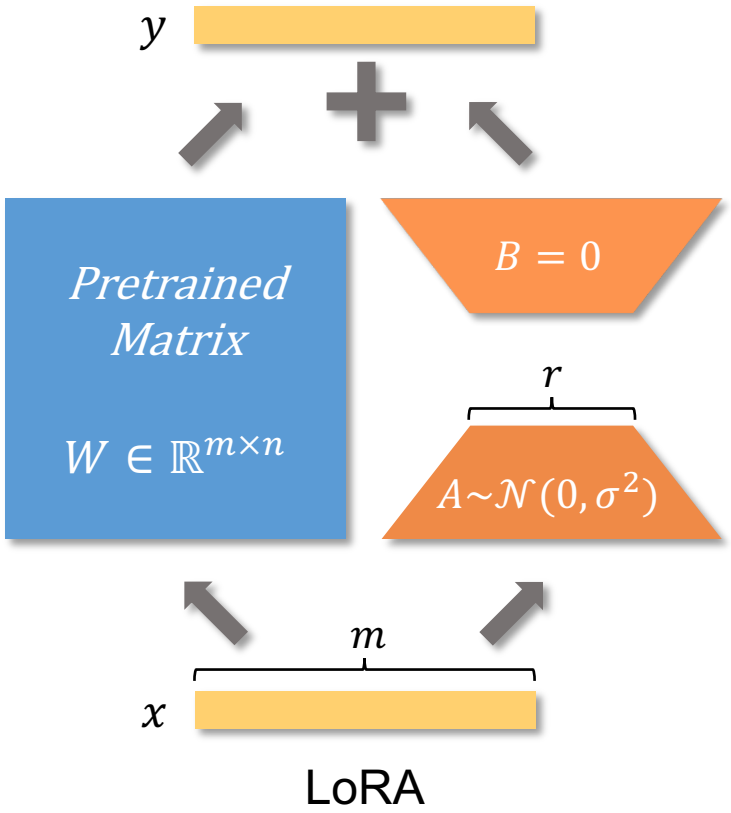


- Insert low dimension matrix A and B.
- Freezing the pre-trained matrix.
- Fine-tuning A and B.
- Reducing trainable parameters by ~100×.

北京大学
PEKING UNIVERSITY

2. Initialization & Gradient Direction

How Does LoRA Initialize A & B?



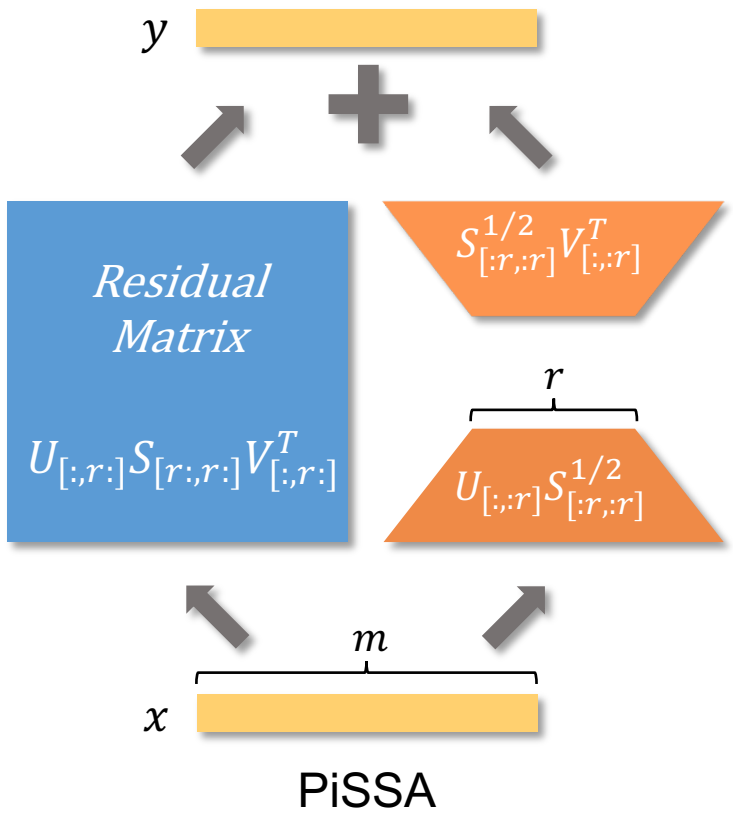
- The insertion of A&B should not disrupt the model's functionality:

$$Y = XW = X(W + \Delta W) = X(W + AB),$$

where $A\sim\mathcal{N}(0,\sigma^2) \in \mathbb{R}^{m\times r}$ and $B = 0 \in \mathbb{R}^{r\times n}$.

- $\frac{\partial L}{\partial A} = X^T \frac{\partial L}{\partial Y} B^T \rightarrow 0$,
- $\frac{\partial L}{\partial B} = A^T X^T \frac{\partial L}{\partial Y} \rightarrow$ random direction.
- LoRA finetune the noise while freezing the W, slow convergence and unsatisfactory performance.

PiSSA Initialization



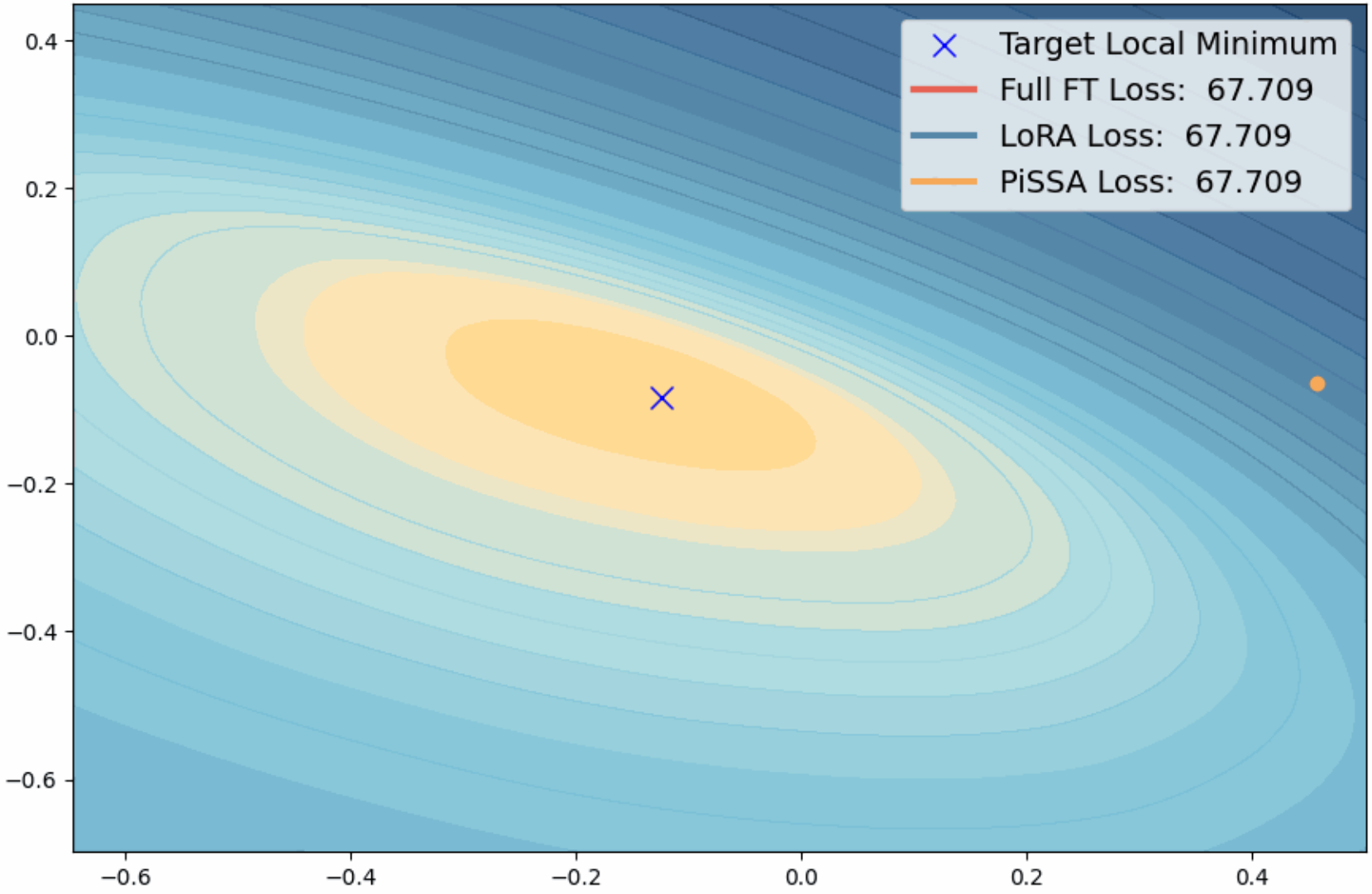
PiSSA

- Apply SVD: $W = USV^T$ and **decompose** the original matrix W into the **principal** part and the **residual** part:

$$Y = XW = X\left(W^{pri} + W^{res}\right) = X(AB + W^{res}),$$

- $A = U_{[:,:r]}\,S^{1/2}_{[:r,:r]} \in \mathbb{R}^{m\times r}$, and $B = S^{1/2}_{[:r,:r]}V^T_{[:,:r]} \in \mathbb{R}^{r\times n}$,
- $W^{res} = U_{[:,r:]}S_{[r:,r:]}V^T_{[:,r:]} = W - AB \in \mathbb{R}^{m\times n}$,
- $\frac{\partial L}{\partial A} = X^T\frac{\partial L}{\partial Y}B^T \rightarrow PiSS$, $\frac{\partial L}{\partial B} = A^TX^T\frac{\partial L}{\partial Y} \rightarrow PiSS$.
- Finetune the principal parts of the model while freezing the residual parts, fast convergence and better performance.

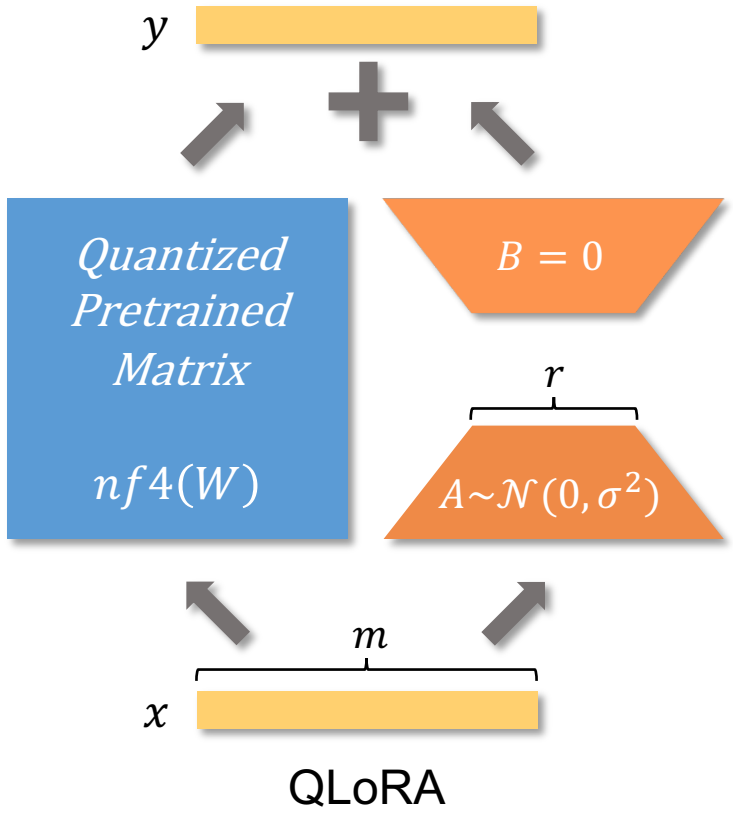
PiSSA Converge Faster

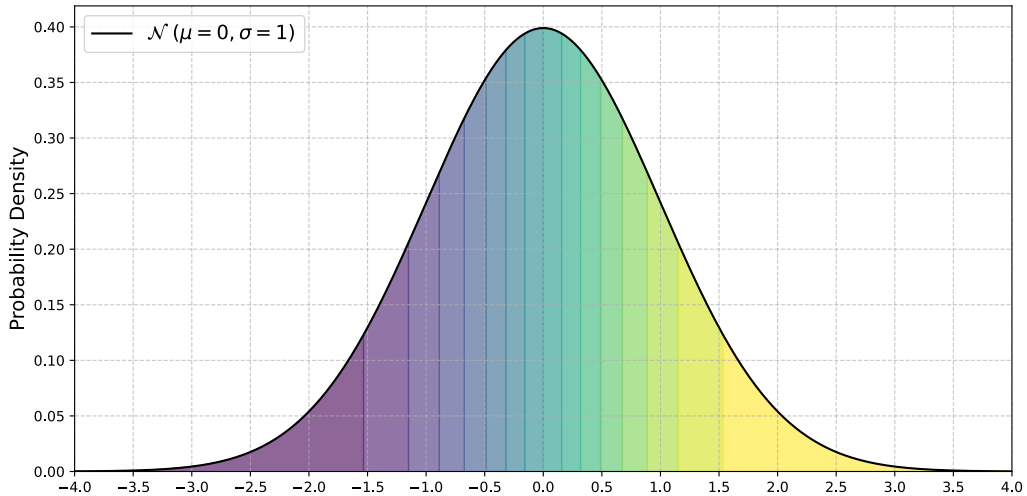


- Model: A two-layer MLP.
- Pretraining on the **odd**-numbered digits of the MNIST dataset.
- Fine-tuning on the **even**-numbered digits of the MNIST dataset.

3. Quantization Error

Quantization Error of QLoRA



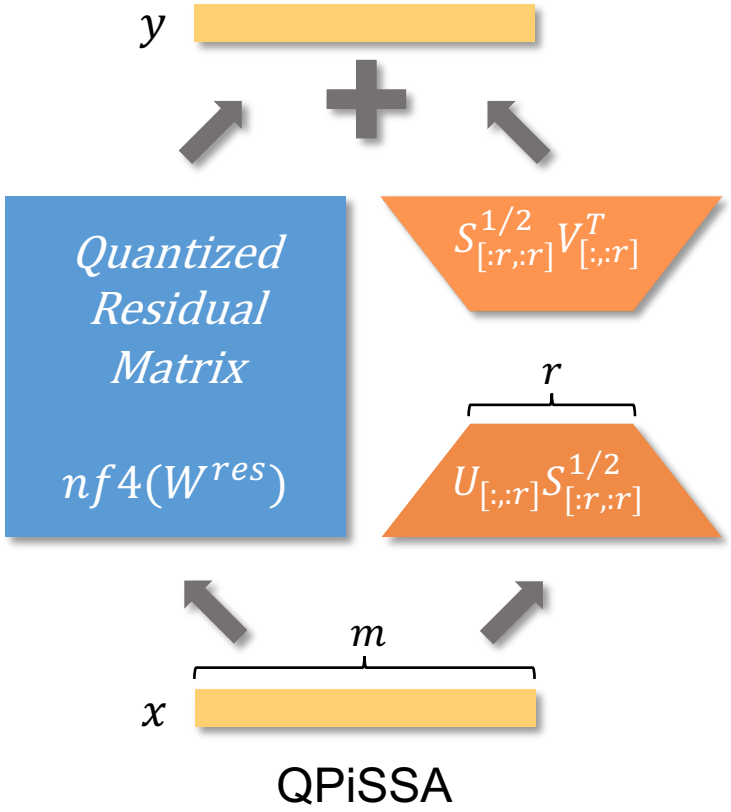


Normal Float Quantile Function

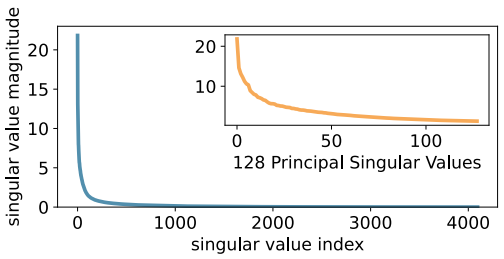
- Quantization Error $= \|W - (nf4(W) + AB)\|_* = \|W - nf4(W)\|_*$, where AB=0.
- $\frac{\partial L}{\partial A} = X^T \frac{\partial L}{\partial Y} B^T \rightarrow 0, \frac{\partial L}{\partial B} = A^T X^T \frac{\partial L}{\partial Y} \rightarrow$ random direction.

Quantization Error of QPiSSA

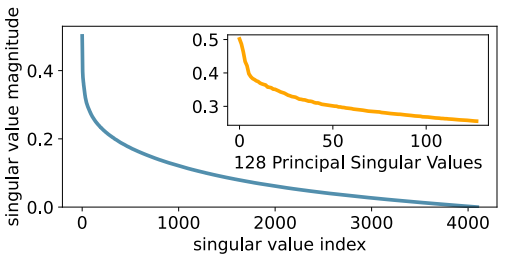
北京大学 PEKING UNIVERSITY



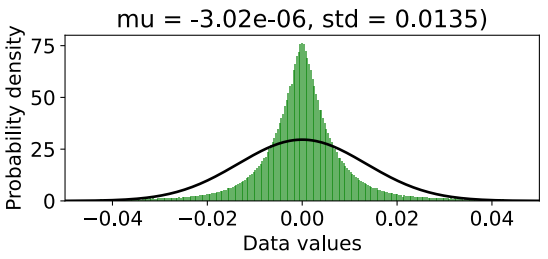
- Quantization Error $= \|W - (nf4(W^{res}) + AB)\|_*$
  $= \|W^{res} - nf4(W^{res})\|_*$, where $W - AB = W^{res}$.
- $\frac{\partial L}{\partial A} = X^T \frac{\partial L}{\partial Y} B^T \rightarrow PiSS$, $\frac{\partial L}{\partial B} = A^T X^T \frac{\partial L}{\partial Y} \rightarrow PiSS$.



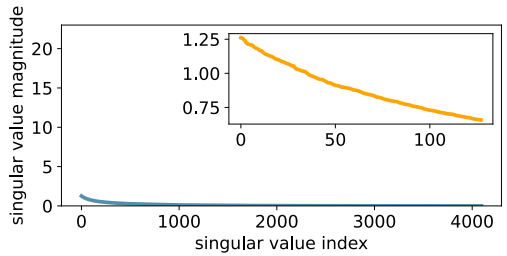
Original matrix W



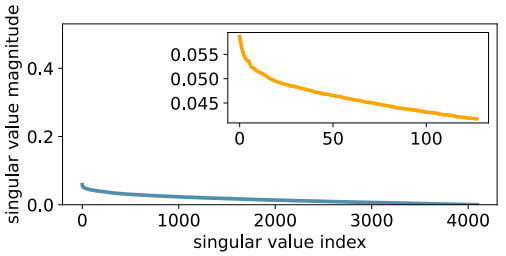
Error matrix of QLoRA



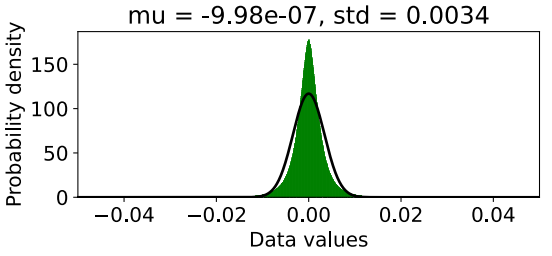
The distribution of W



Residual matrix W^{res}



Error matrix of QPiSSA



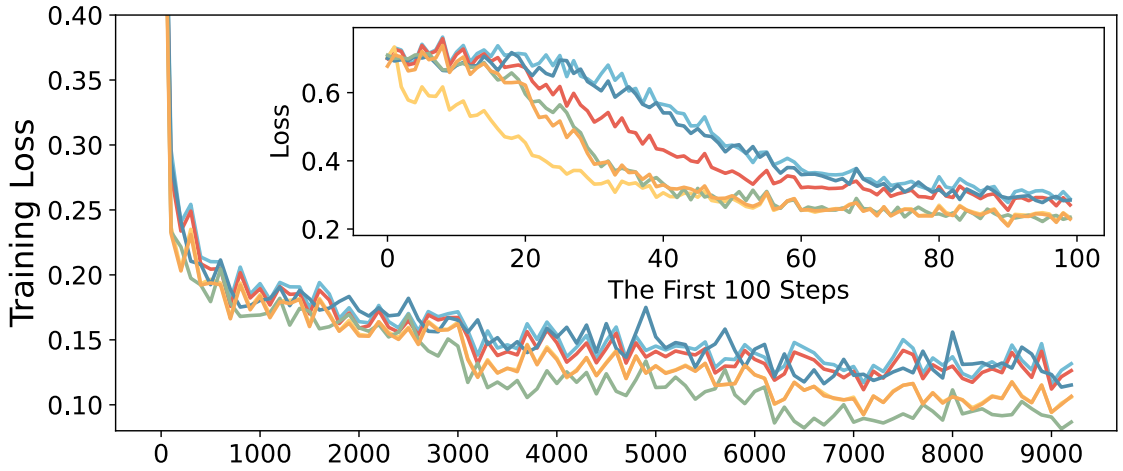
The distribution of W^{res}

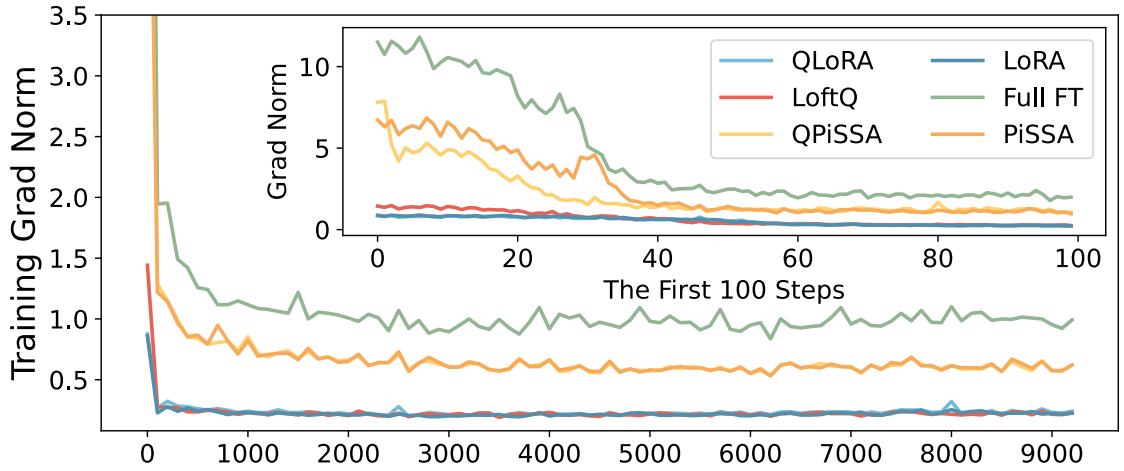
QPiSSA Reduce Quantization Error

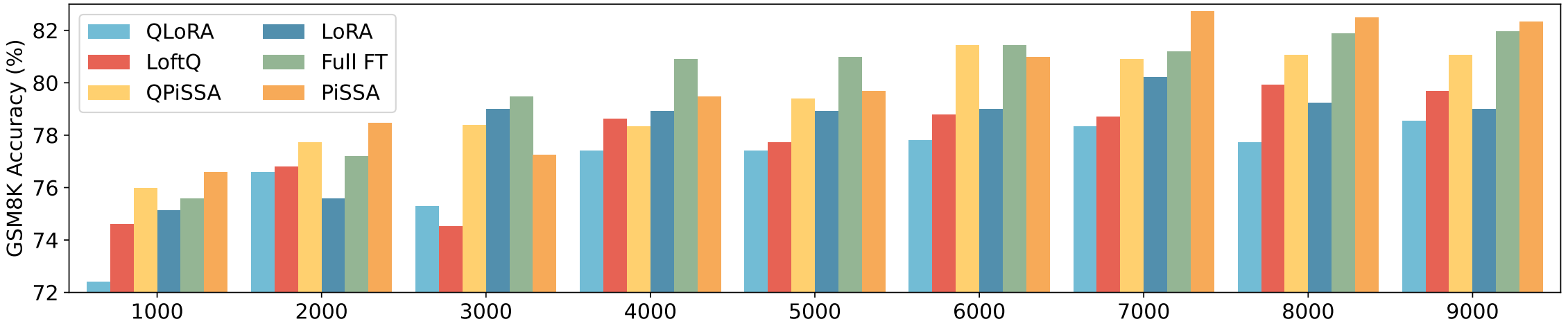
北京大学 PEKING UNIVERSITY

	Method	Rank	Q	K	V	O	Gate	Up	Down	AVG
LLaMA 2-7B	QLoRA	--	0	0	0	0	0	0	0	0
	LoftQ	128	16.5	16.5	15.9	16.0	12.4	12.4	12.3	14.6
	PiSSA	**128**	**27.9**	**27.2**	**18.7**	**18.6**	**15.8**	**13.6**	**13.6**	**19.4**
LLaMA 3-8B	QLoRA	--	0	0	0	0	0	0	0	0
	LoftQ	128	16.4	29.8	28.8	16.1	11.9	11.7	11.7	18.1
	PiSSA	**128**	**26.3**	**41.7**	**32.3**	**20.1**	**14.4**	**12.5**	**12.9**	**22.9**
LLaMA 3-70B	QLoRA	--	0	0	0	0	0	0	0	0
	LoftQ	64	6.1	17.8	17.0	6.0	4.3	4.4	4.2	8.5
	PiSSA	**64**	**15.7**	**34.2**	**18.9**	**7.5**	**6.7**	**5.7**	**4.7**	**13.4**
	PiSSA	**128**	**23.2**	**49.0**	**30.5**	**12.5**	**10.1**	**8.8**	**8.2**	**20.3**

QPiSSA Converge Faster







4. Experiments

Performance on NLG and NLU Tasks

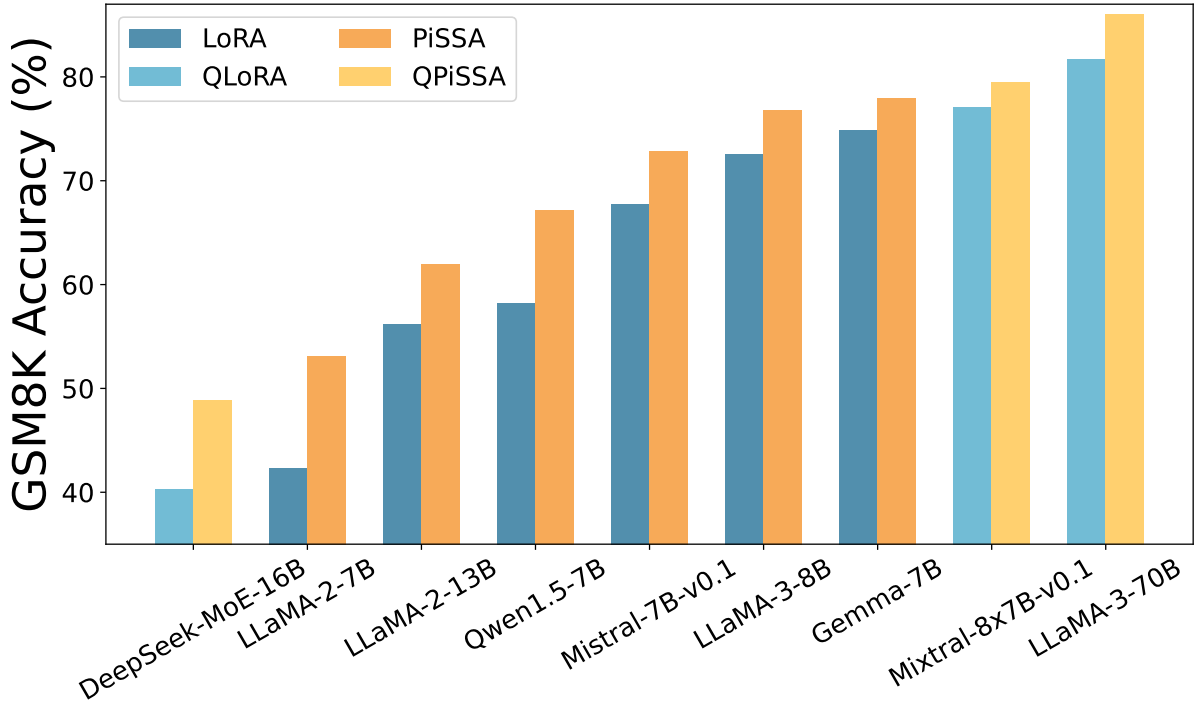
北京大学 PEKING UNIVERSITY

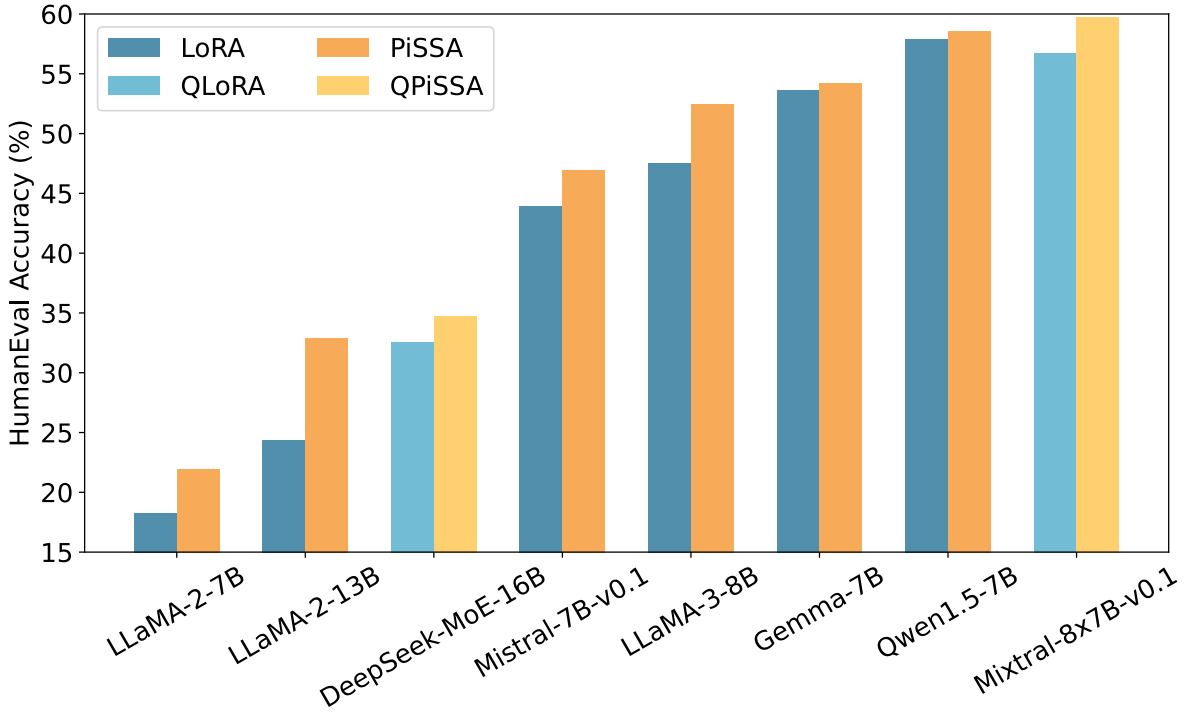
Model	Strategy	Param	GSM8K	MATH	HumanEval	MBPP	MT-Bench
LLaMA 2-7B	Full FT	6738M	49.13	7.29	21.20	35.59	**4.91**
	LoRA	320M	42.85	5.50	18.35	35.50	4.59
	PiSSA	320M	**53.22**	**7.47**	**21.92**	**37.24**	4.88
Mistral-7B	Full FT	7242M	69.91	18.64	45.31	51.46	4.95
	LoRA	168M	69.50	20.08	43.78	58.46	4.90
	PiSSA	168M	**73.31**	**23.12**	**46.88**	**62.55**	**5.34**
Gemma-7B	Full FT	8538M	72.09	22.74	47.02	55.67	5.40
	LoRA	200M	75.11	30.41	53.70	65.58	4.98
	PiSSA	200M	**77.78**	**31.33**	**54.31**	**66.17**	**5.64**

Method	Params	MNLI	SST2	MRPC	CoLA	QNLI	QQP	RTE	STSB	ALL
Full FT	184M	89.9	95.63	89.46	69.19	94.03	**92.4**	83.75	91.60	88.25
BitFit	0.1M	89.37	94.84	87.75	66.96	92.24	88.41	78.70	91.35	86.20
HAdapter	1.22M	90.13	95.53	89.95	68.64	94.11	91.91	84.48	91.48	88.28
PAdapter	1.18M	90.33	95.61	89.46	68.77	94.29	92.04	85.20	91.54	88.41
LoRA	1.33M	90.65	94.95	89.95	69.82	93.87	91.99	85.20	91.60	88.50
DoRA	1.27M	90.29	95.79	90.93	70.85	94.10	92.07	86.04	91.79	88.98
AdaLoRA	1.27M	**90.76**	96.10	90.69	71.45	**94.55**	92.23	88.09	91.84	89.46
PiSSA	1.33M	90.37	**96.22**	**91.50**	**73.12**	94.43	92.33	**88.69**	**92.00**	**89.83**

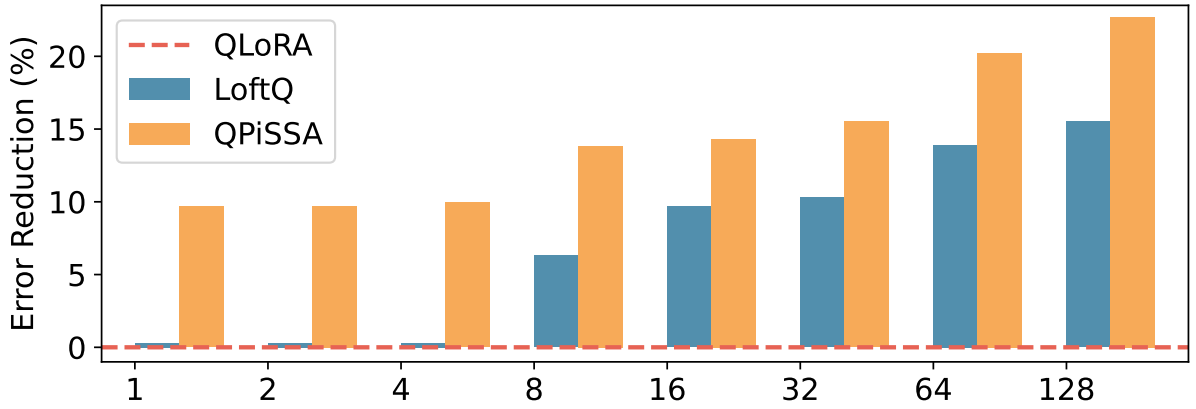
Performance on Various Models

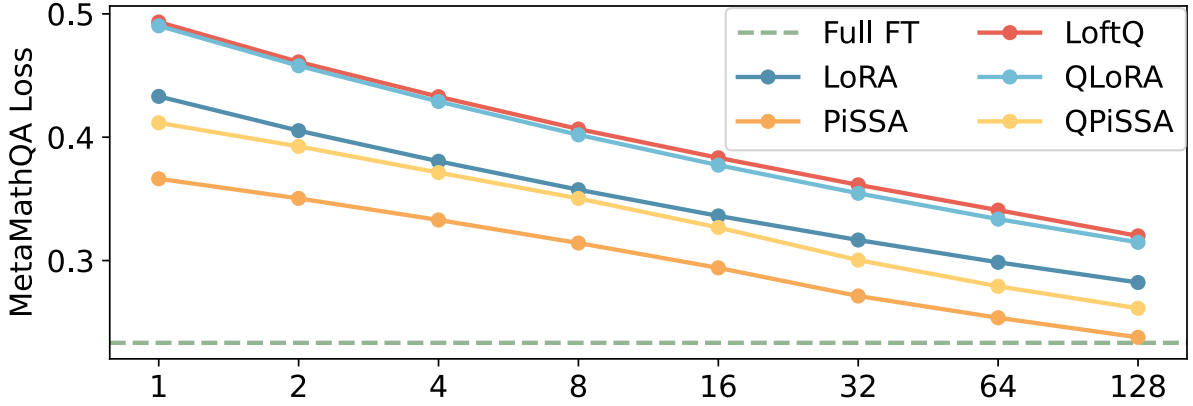
北京大学 PEKING UNIVERSITY

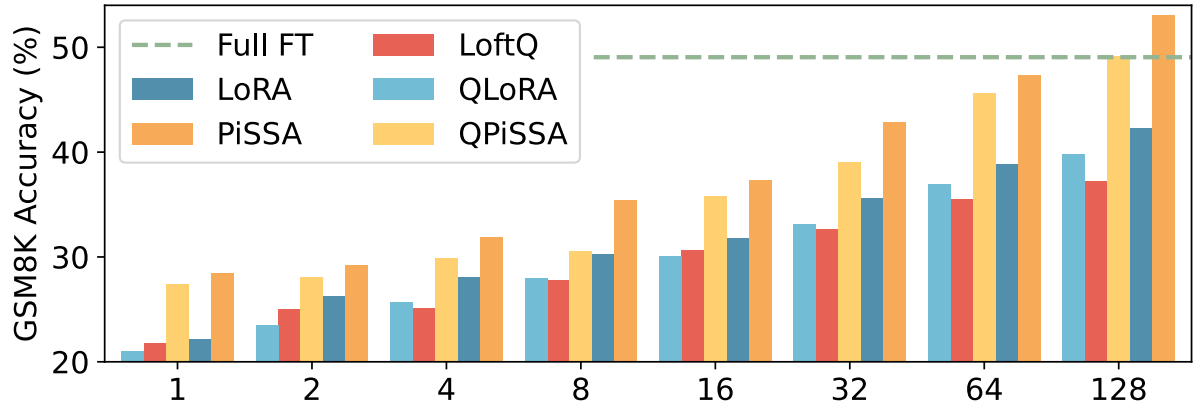


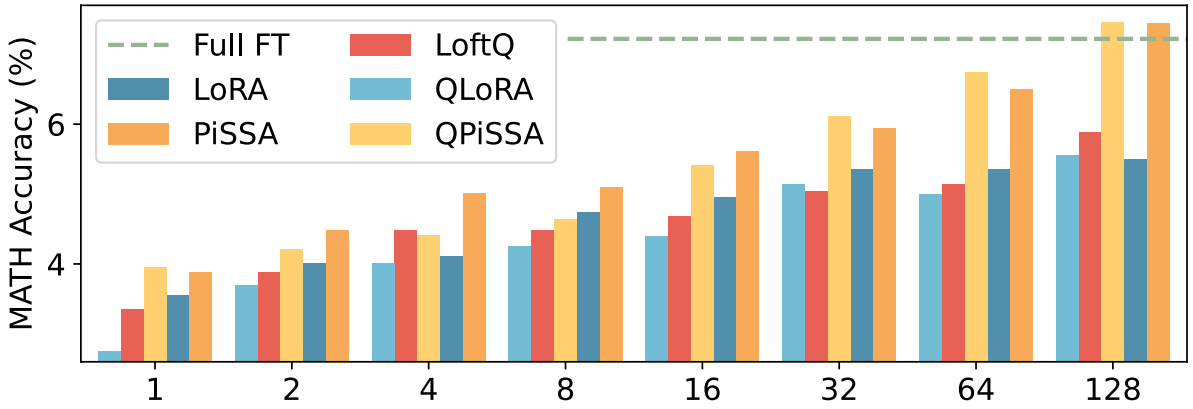


Performance on Various Ranks





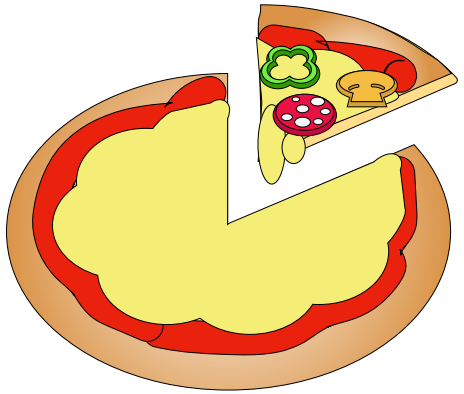




Conclusion

北京大学 PEKING UNIVERSITY

- Proposes new parameter efficient fine-tuning methods **(Q)PiSSA**
- Shares the **same architecture** as (Q)LoRA, **different initializations** of W^{res} and AB
- No need to change any code, **significant performance improvement**!



- Paper arxiv: https://arxiv.org/pdf/2404.02948.pdf
- Github link: https://github.com/GraphPKU/PiSSA
- Initialized model: https://huggingface.co/collections/fxmeng
- SR code:

Thanks for listening!