RaSA: Rank-Sharing Low-Rank Adaptation

Zhiwei He, Zhaopeng Tu, Xing Wang, Xingyu Chen, Zhijie Wang, Jiahao Xu, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Rui Wang

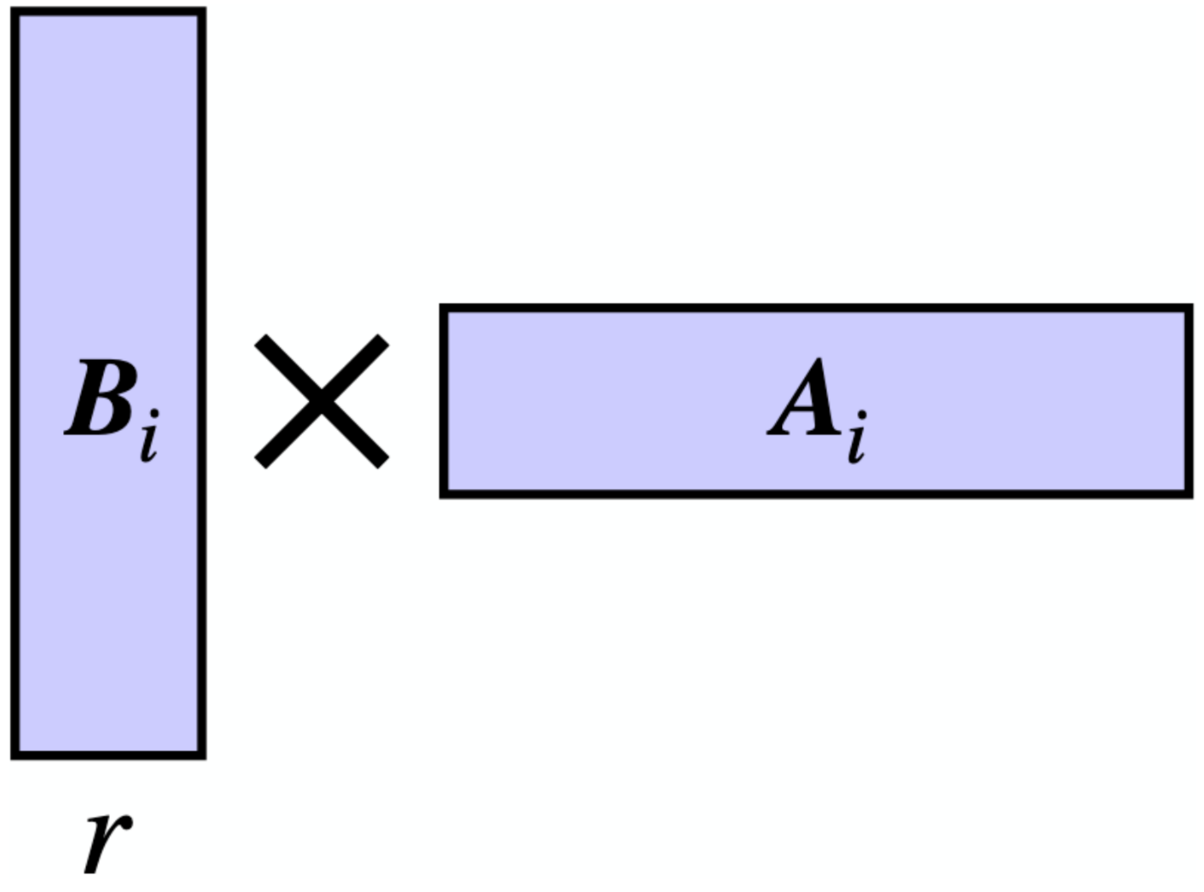
Presenter: Zhiwei He

Low-Rank Adaptation (LoRA)

Low-rank update

- Parameter Update:

$$\boldsymbol{W}_i + \underbrace{\Delta \boldsymbol{W}_i}_{\text{low-rank}} =$$

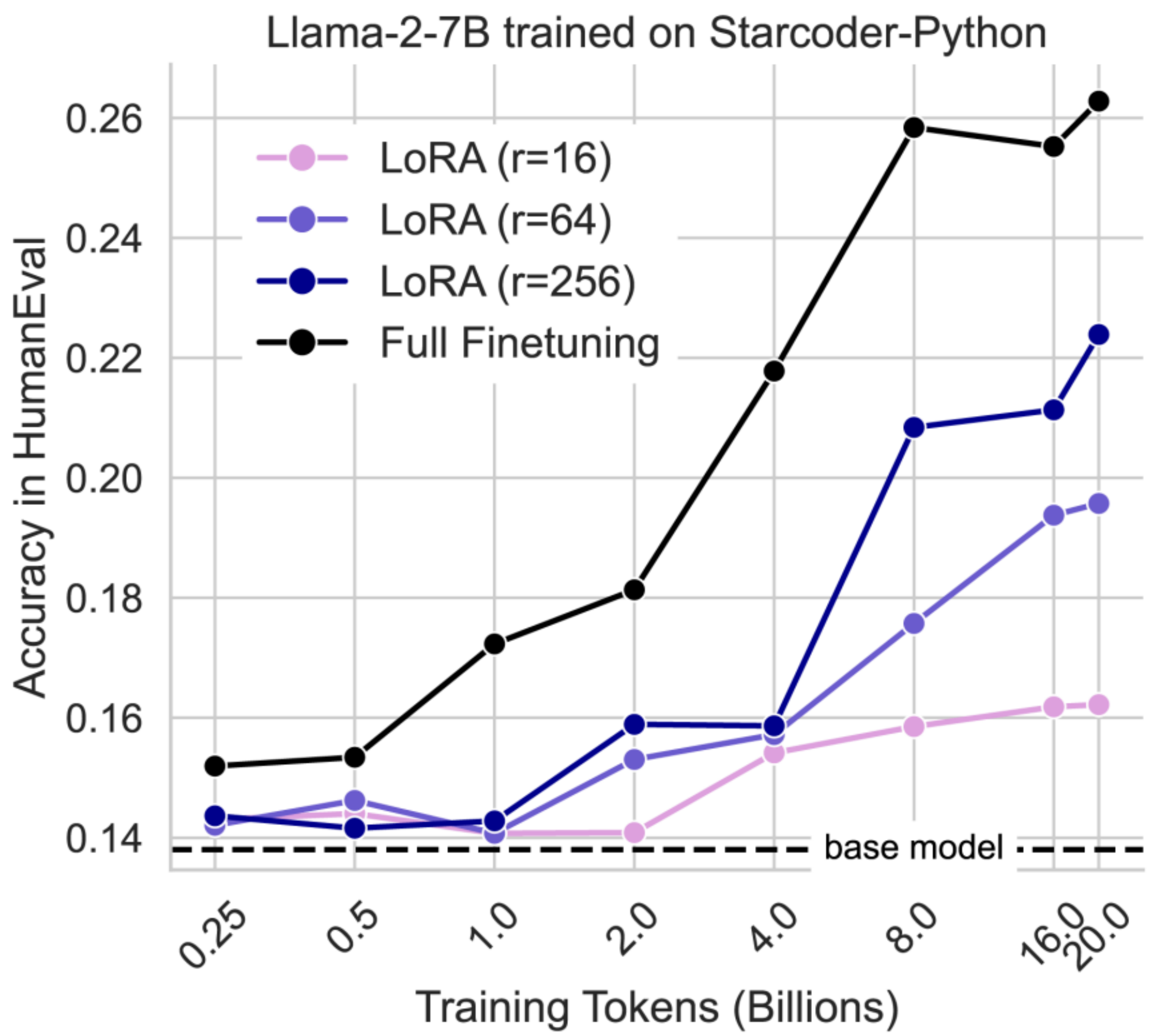


$$\boldsymbol{B}_i \in \mathbb{R}^{b \times r}, \boldsymbol{A}_i \in \mathbb{R}^{r \times a}$$

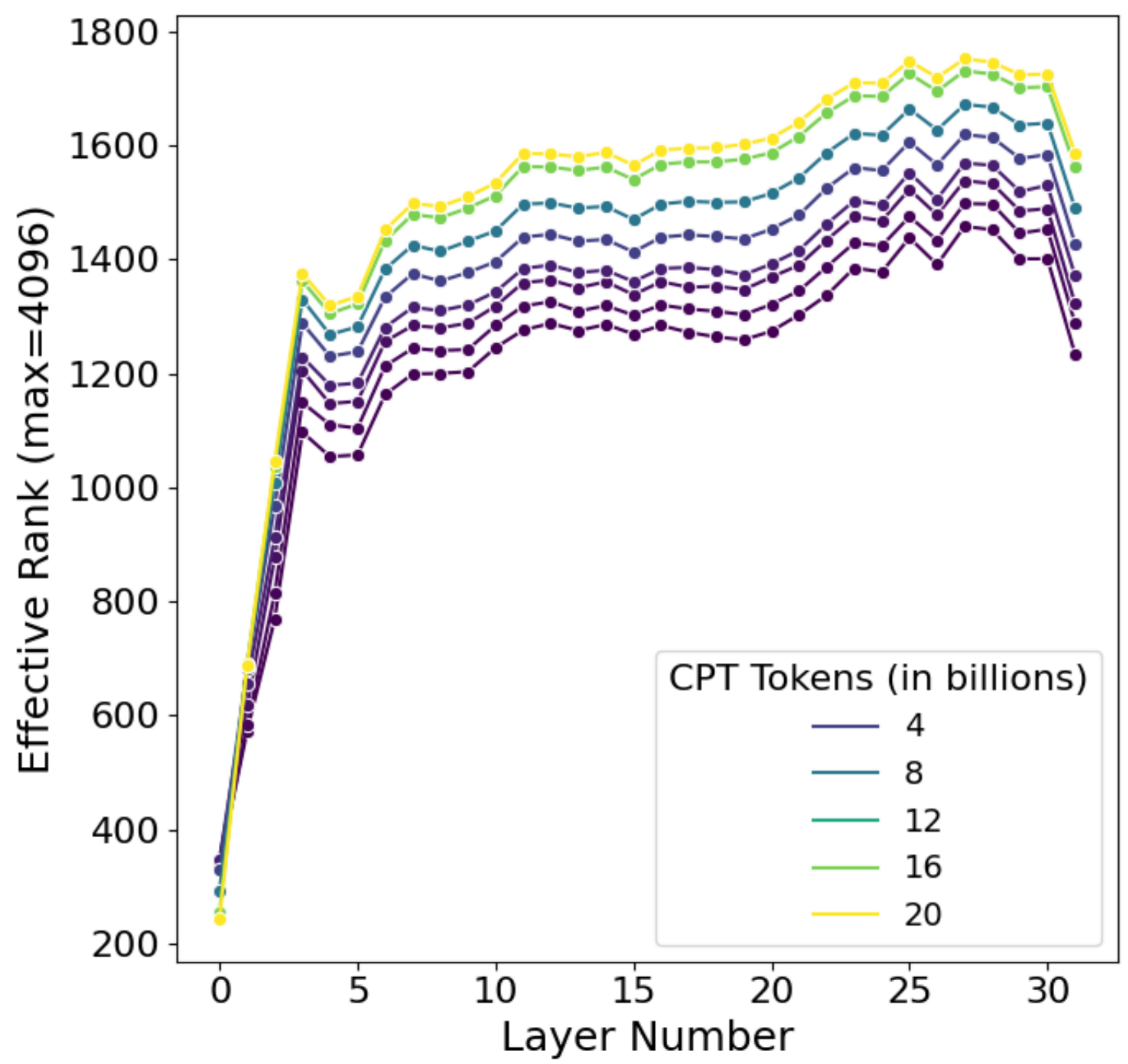
$$r \ll \min(b, a)$$

Low-Rank Adaptation (LoRA)

Low-rank constraint limits the expressive capacity of LoRA



LoRA still lags behind full fine-tuning (FFT)



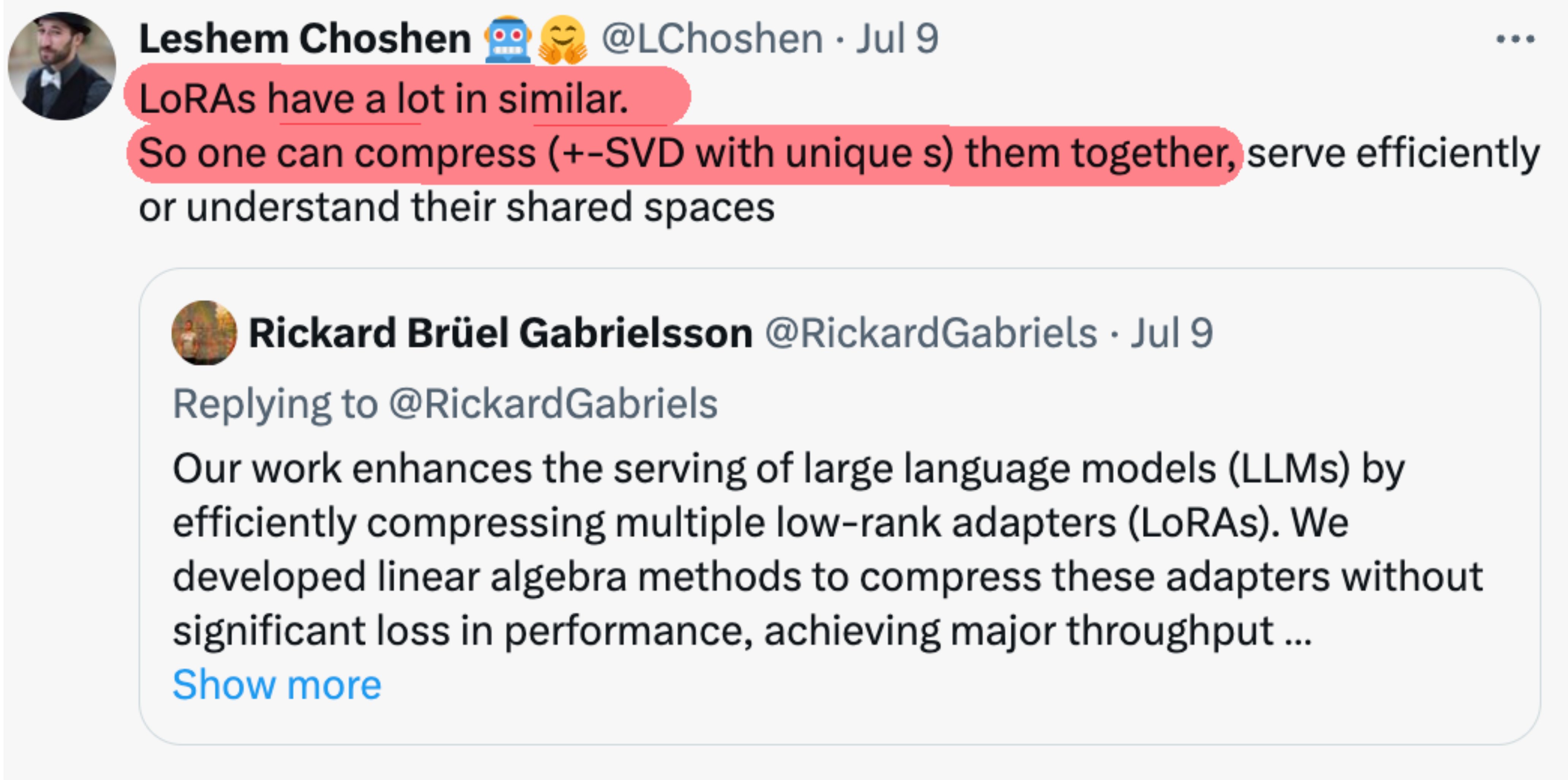
Effective rank of FFT is large

[1] Biderman, Dan, et al. "Lora learns less and forgets less." TMLR

Low-Rank Adaptation (LoRA)

LoRA can be further compressed

- Many related works demonstrate that LoRA can be further compressed by
  - [2] 7.76%
  - [3] 12.5%
  - [4] 50%
  - [5] 3%
- without performance loss.



Compressing 100 LoRAs by sharing their subspaces won't compromise performance.

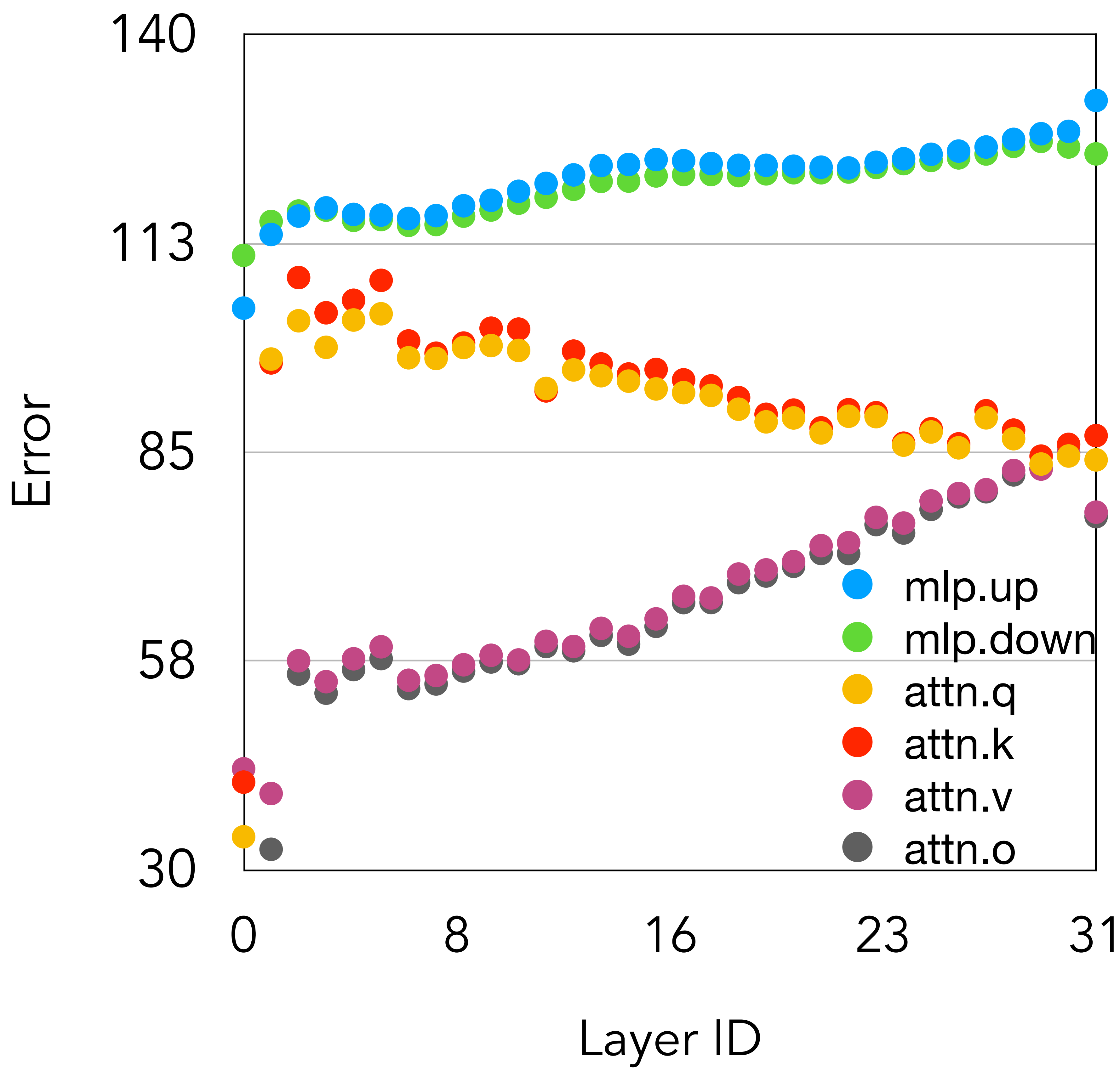
[2] Kopiczko, Dawid J., Tijmen Blankevoort, and Yuki M. Asano. "Vera: Vector-based random matrix adaptation." ICLR 2023
[3] Renduchintala, Adithya, Tugrul Konuk, and Oleksii Kuchaiev. "Tied-LoRA: Enhancing parameter efficiency of LoRA with weight tying." NAACL 2024
[4] Song, Yurun, et al. "ShareLoRA: Parameter Efficient and Robust Large Language Model Fine-tuning via Shared Low-Rank Adaptation." arXiv 2024
[5] Li, Yang, Shaobo Han, and Shihao Ji. "VB-LoRA: extreme parameter efficient fine-tuning with vector banks." NeurIPS 2024

LoRA is underutilized.

Low-Rank Adaptation (LoRA)

Why LoRA is under-utilized?

- Different components and layers require different levels of expressive capability.
- LoRA adopts an average allocation strategy.



Reconstruction error: ||LoRA - FFT||

Rank-Sharing Low-Rank Adaptation (RaSA)

Partial rank-sharing across layers

- Split the matrices $\boldsymbol{B}_i$ and $\boldsymbol{A}_i$ into layer-specific parts $(\tilde{\boldsymbol{B}}_i, \tilde{\boldsymbol{A}}_i)$ and layer-shared parts $(\hat{\boldsymbol{B}}_i, \hat{\boldsymbol{A}}_i)$

$$\boldsymbol{B}_i = [\ \underbrace{\tilde{\boldsymbol{B}}_i}_{\mathbb{R}^{b\times(r-k)}} \quad \underbrace{\hat{\boldsymbol{B}}_i}_{\mathbb{R}^{b\times k}}], \quad \boldsymbol{A}_i = [\ \underbrace{\tilde{\boldsymbol{A}}_i^T}_{\mathbb{R}^{a\times(r-k)}} \quad \underbrace{\hat{\boldsymbol{A}}_i^T}_{\mathbb{R}^{a\times k}}]^T$$

- Concatenate all layer-shared parts across layers to form shared rank pools $(\boldsymbol{B}_S \text{ and } \boldsymbol{A}_S)$

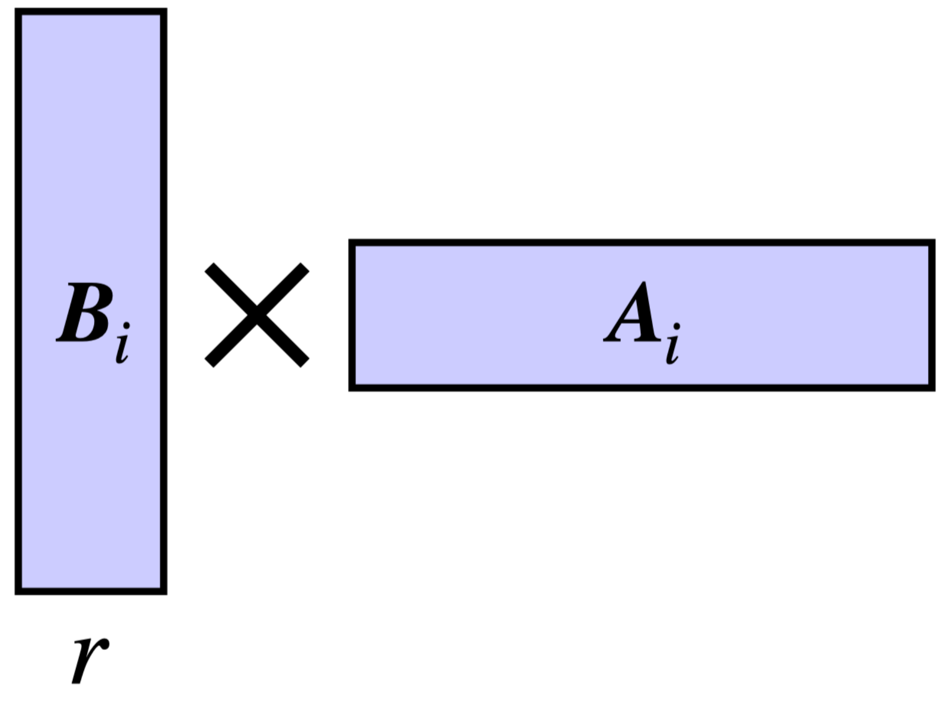
$$\boldsymbol{B}_S = [\hat{\boldsymbol{B}}_1 \quad \hat{\boldsymbol{B}}_2 \quad \cdots \quad \hat{\boldsymbol{B}}_L] \in \mathbb{R}^{b\times(L\times k)}, \quad \boldsymbol{A}_S = [\hat{\boldsymbol{A}}_1^T \quad \hat{\boldsymbol{A}}_2^T \quad \cdots \quad \hat{\boldsymbol{A}}_L^T]^T \in \mathbb{R}^{(L\times k)\times a}$$

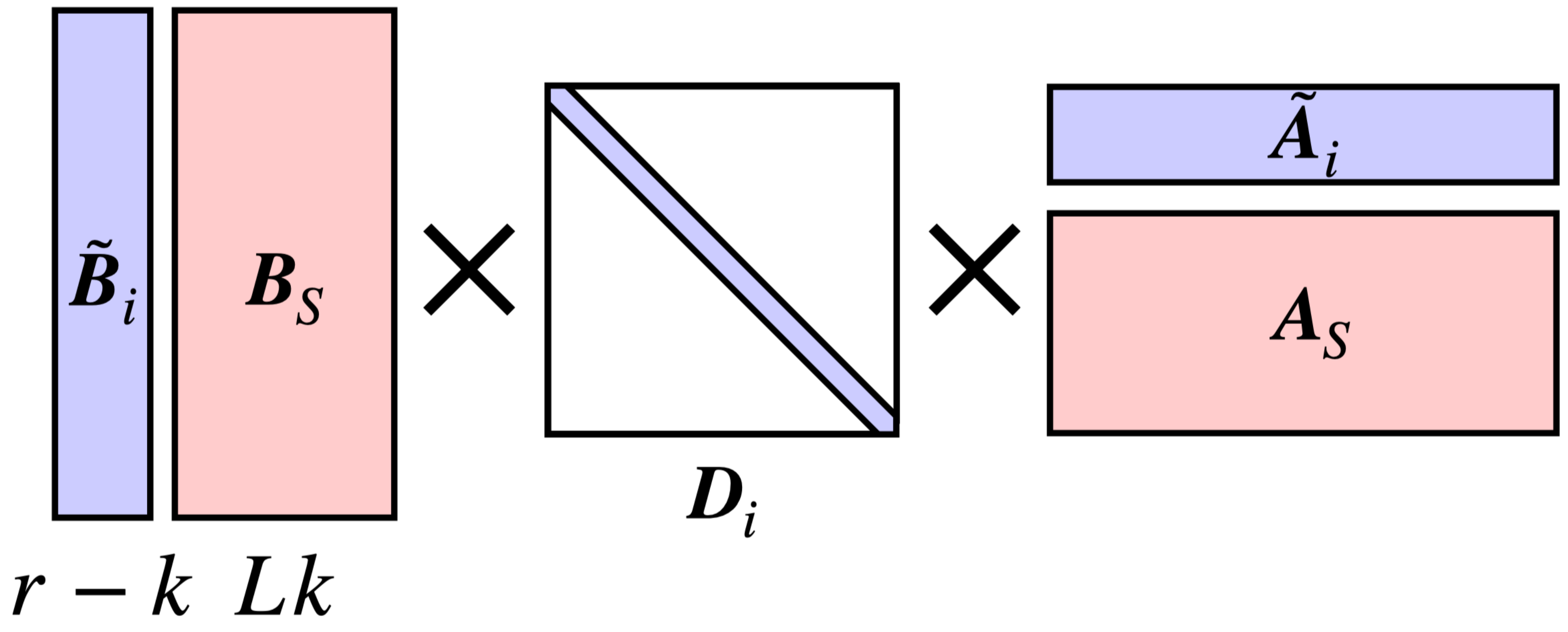
- Update of layer-i

$$\begin{aligned} \boldsymbol{W}_i + \Delta\boldsymbol{W}_i &= \boldsymbol{W}_i + \frac{\alpha}{r}(\tilde{\boldsymbol{B}}_i\tilde{\boldsymbol{A}}_i + \boldsymbol{B}_S\boldsymbol{A}_S) \\ &= \boldsymbol{W}_i + [\tilde{\boldsymbol{B}}_i \quad \boldsymbol{B}_S]\,\mathrm{diag}(\frac{\alpha}{r})\begin{bmatrix}\tilde{\boldsymbol{A}}_i \\ \boldsymbol{A}_S\end{bmatrix} \end{aligned}$$

Rank-Sharing Low-Rank Adaptation (RaSA)

Comparison between LoRA and RaSA





$$\boldsymbol{W}_i + \Delta\boldsymbol{W}_i = \boldsymbol{W}_i + \frac{\alpha}{r}\boldsymbol{B}_i\boldsymbol{A}_i \quad (\boldsymbol{B}_i \in \mathbb{R}^{b\times r}, \boldsymbol{A}_i \in \mathbb{R}^{r\times a})$$

LoRA

$$\boldsymbol{W}_i + \Delta\boldsymbol{W}_i = \boldsymbol{W}_i + \underbrace{\begin{bmatrix}\tilde{\boldsymbol{B}}_i & \boldsymbol{B}_S\end{bmatrix}}_{\mathbb{R}^{b\times(r-k+Lk)}} \boldsymbol{D}_i \underbrace{\begin{bmatrix}\tilde{\boldsymbol{A}}_i \\ \boldsymbol{A}_S\end{bmatrix}}_{\mathbb{R}^{(r-k+Lk)\times a}}$$

RaSA

- r => r-k+Lk
- extra parameters (0.01‰)

Reconstruction Error Analysis

Minimum Reconstruction Error

- We compare their abilities to reconstruct a set of high-rank matrices $\{\boldsymbol{M}_i\}_{i\in[L]}$, $\mathrm{rank}(\boldsymbol{M}_i) = R > r$

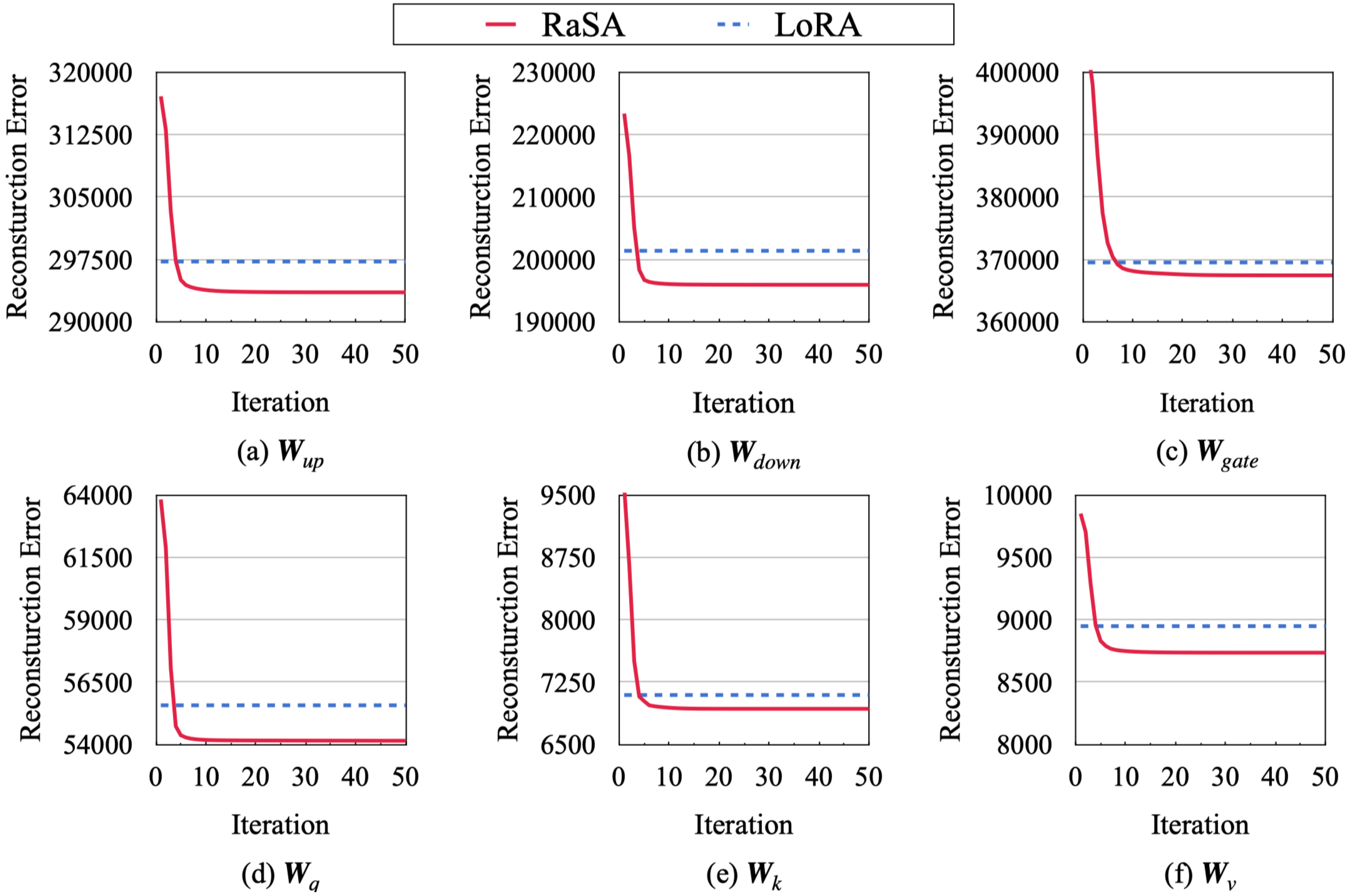
$$e_{\text{lora}} = \min_{\boldsymbol{B}_i, \boldsymbol{A}_i} \sum_{i=1}^{L} \|\boldsymbol{M}_i - \boldsymbol{B}_i \boldsymbol{A}_i\|_F^2$$

$$e_{\text{rasa}(k)} = \min_{\tilde{\boldsymbol{B}}_i, \tilde{\boldsymbol{A}}_i, \boldsymbol{B}_S, \boldsymbol{A}_S, \boldsymbol{D}_i} \sum_{i=1}^{L} \|\boldsymbol{M}_i - (\tilde{\boldsymbol{B}}_i \tilde{\boldsymbol{A}}_i + \boldsymbol{B}_S \boldsymbol{D}_i \boldsymbol{A}_S)\|_F^2$$

- We prove $e_{\text{rasa}(k)} \leq e_{\text{lora}}$ (Theorem 3.1)

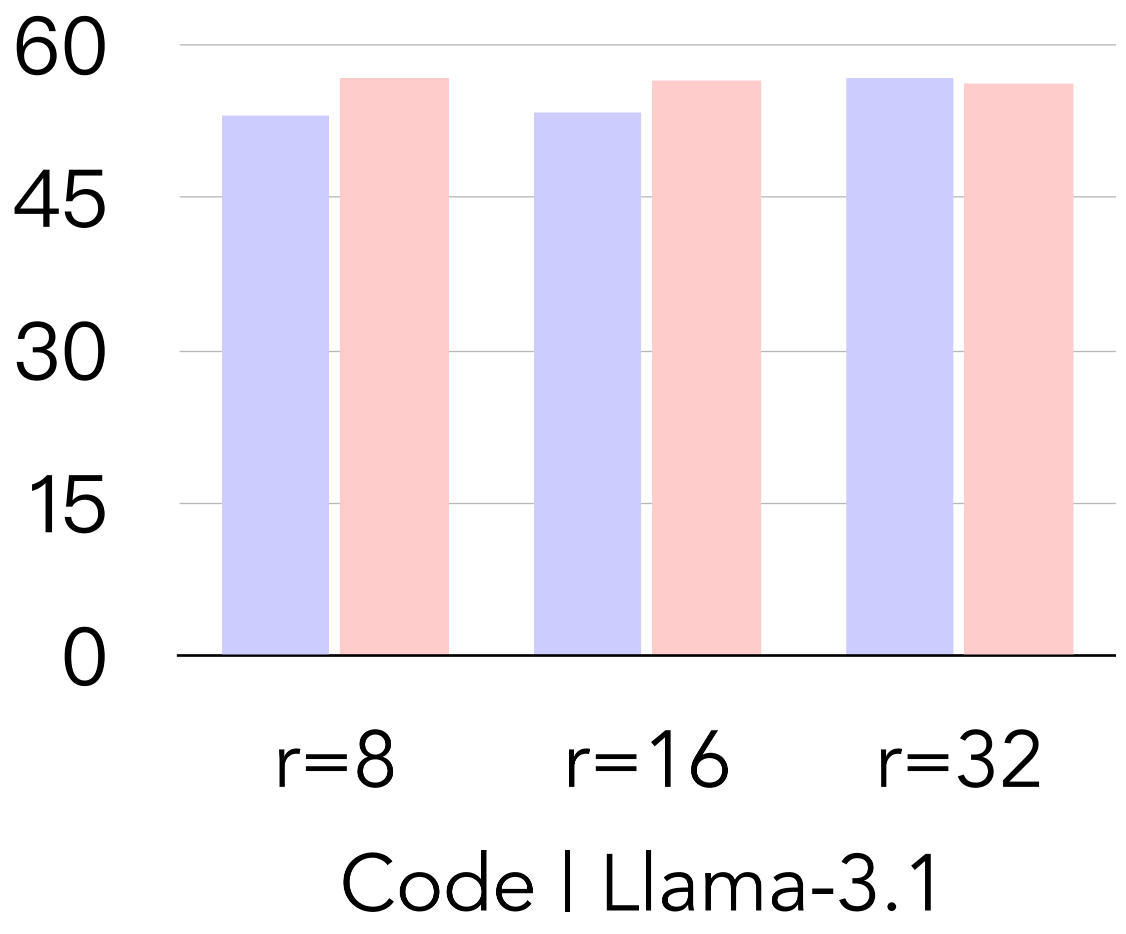
Reconstruction Error Analysis

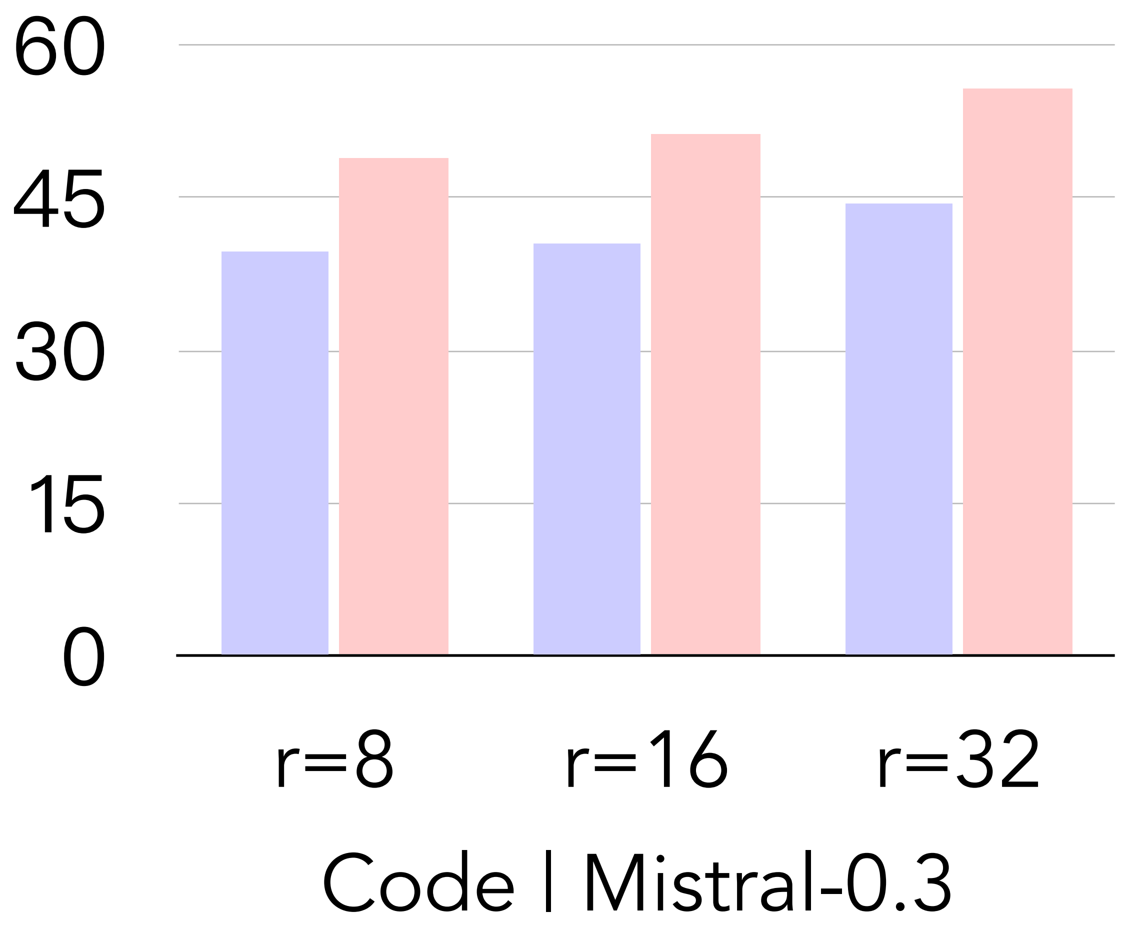
Empirical Analysis

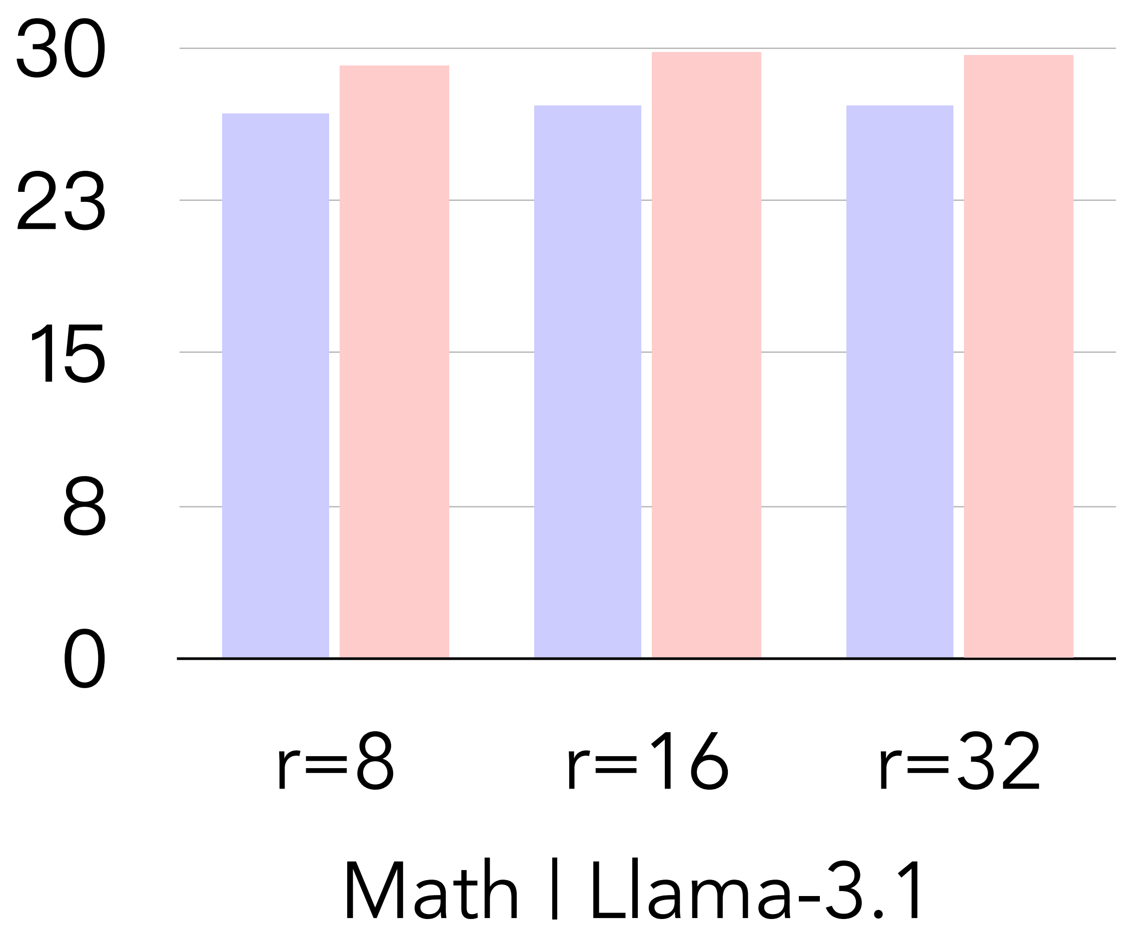


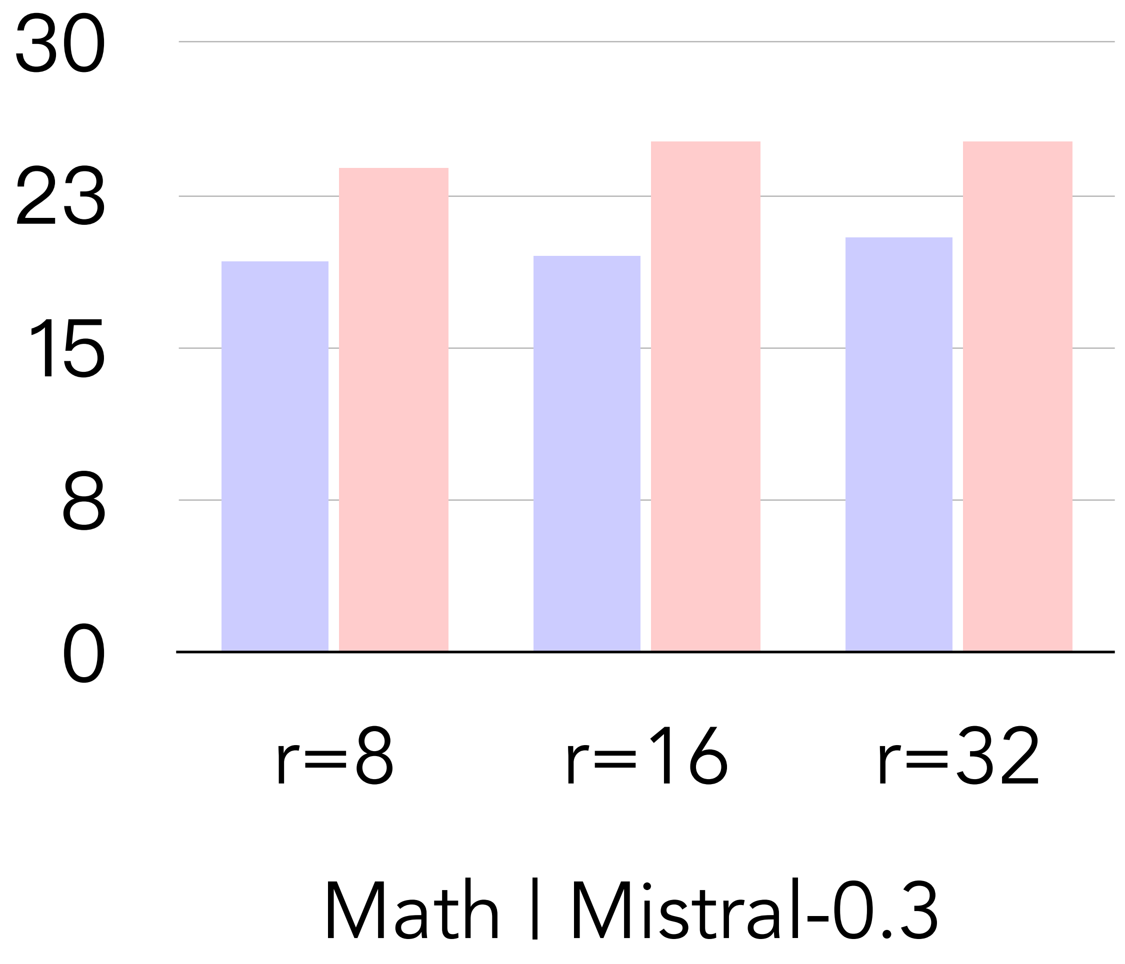
RaSA requires ~10 iterations to achieve a significantly lower reconstruction error than LoRA's minimum.

Main Results

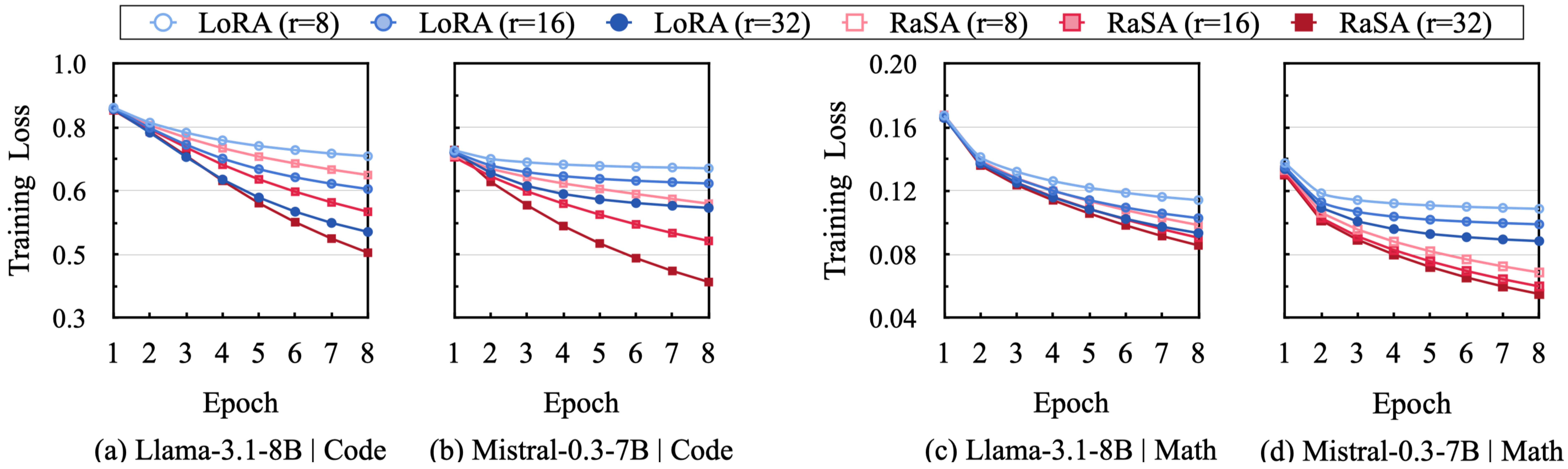






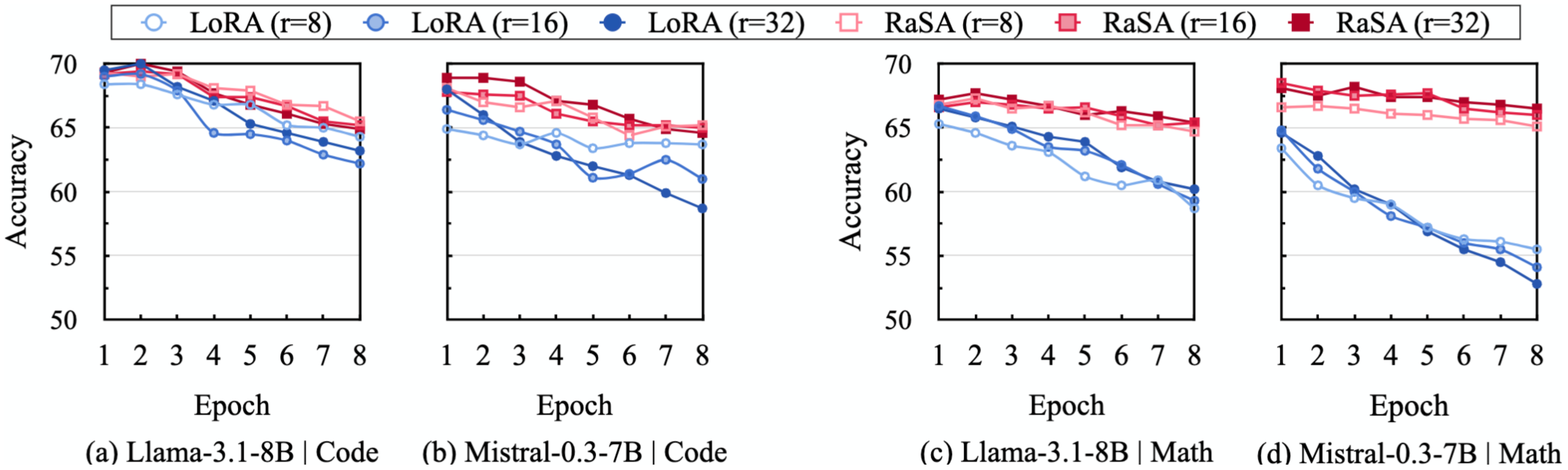


RaSA learns more and faster than LoRA



Training curves of LoRA and RaSA with different ranks.

RaSA forgets less than LoRA



Y-axis shows the average of prediction accuracy on three benchmarks to evaluate model's forgetting. Higher prediction accuracy denotes less forgetting.

Summary

- We propose RaSA, an extension of LoRA by by allowing partial rank sharing across layers, which significantly improves the efficiency and expressiveness.
- We provide a comprehensive analysis – both theoretical and empirical – showcasing RaSA's superior capacity for matrix reconstruction and its resultant improved performance on downstream tasks.

Thank You