Balanced Low-Rank Adaptation: Removing Parameter Invariance to Accelerate Convergence

Valérie Castin

École Normale Supérieure PSL, Paris

Kimia Nadjahi
ENS PSL

Pierre Ablin
Apple

Gabriel Peyré
ENS PSL

Journées SMAI-MODE 2026, Nice

Full fine-tuning versus low-rank adaptation

Fine-tuning: adapting pretrained model (e.g. Transformer…) to new dataset

- **full fine-tuning:** retrain all weights starting from pretrained

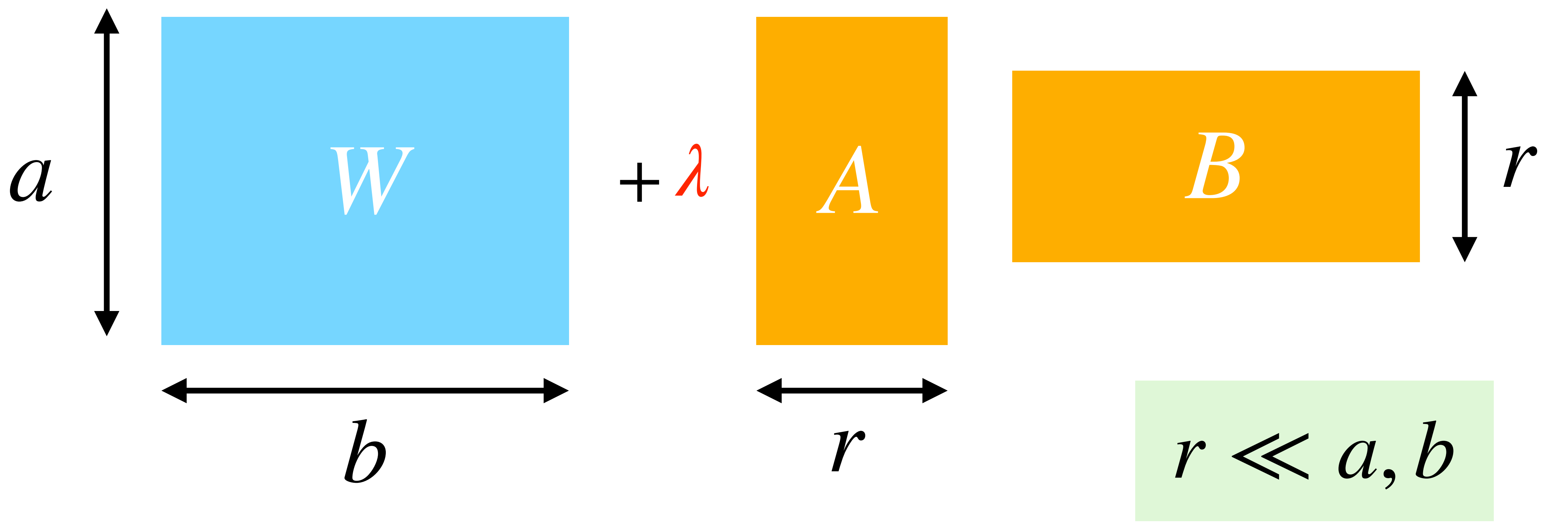
- **Low-Rank Adaptation (LoRA):** [Hu et al., 2022]

  ❄️ freeze pretrained weights W

  🔥 train low-rank update to W as

  $$\underbrace{W}_{\text{frozen}} + \lambda \underbrace{AB}_{\substack{\text{trained} \\ \text{low rank}}}$$

  with stepsize γ



$$\min_{A_1,B_1,\ldots,A_L,B_L} \ell(W_1 + A_1B_1, \ldots, W_L + A_LB_L)$$

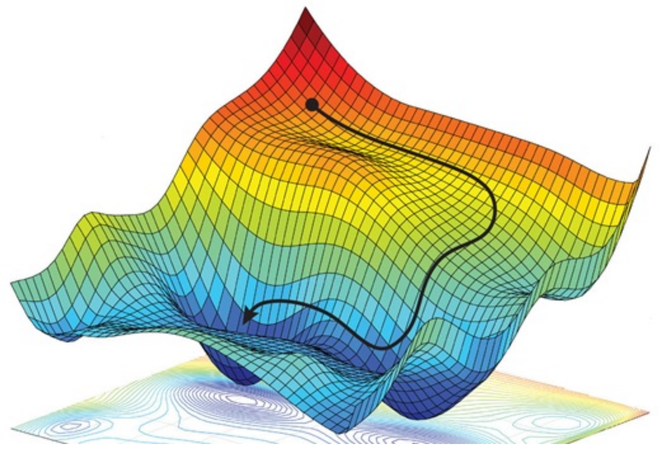
Advantages of LoRA:

✓ reduced memory and computational cost when training

✓ allows to ship easily several fine-tuned models

✓ does not hurt performance too much

Background: convergence guarantees for LoRA



LoRA on one layer: **minimizing** loss $f(A, B) = \ell(W + AB)$ with **optimizer** (GD, AdamW…)

One-layer linear network W_{frozen} with target W^* s.t. $Z := W^* - W_{frozen}$ has **rank** r

GD on $f(A, B) = \frac{1}{2}\|Z - AB\|_F^2$ (matrix factorization)

→ [Ye & Du, 2021] convergence rate to global minimum with Gaussian init

Last layer W_{frozen} **of multilayer linear network** with **no rank assumption**

GD on $f(A, B) = \frac{1}{2}\|W^* - (W_{frozen} + AB)X\|_F^2$

→ [Nguegnang et al., 2024] convergence to global minimizer

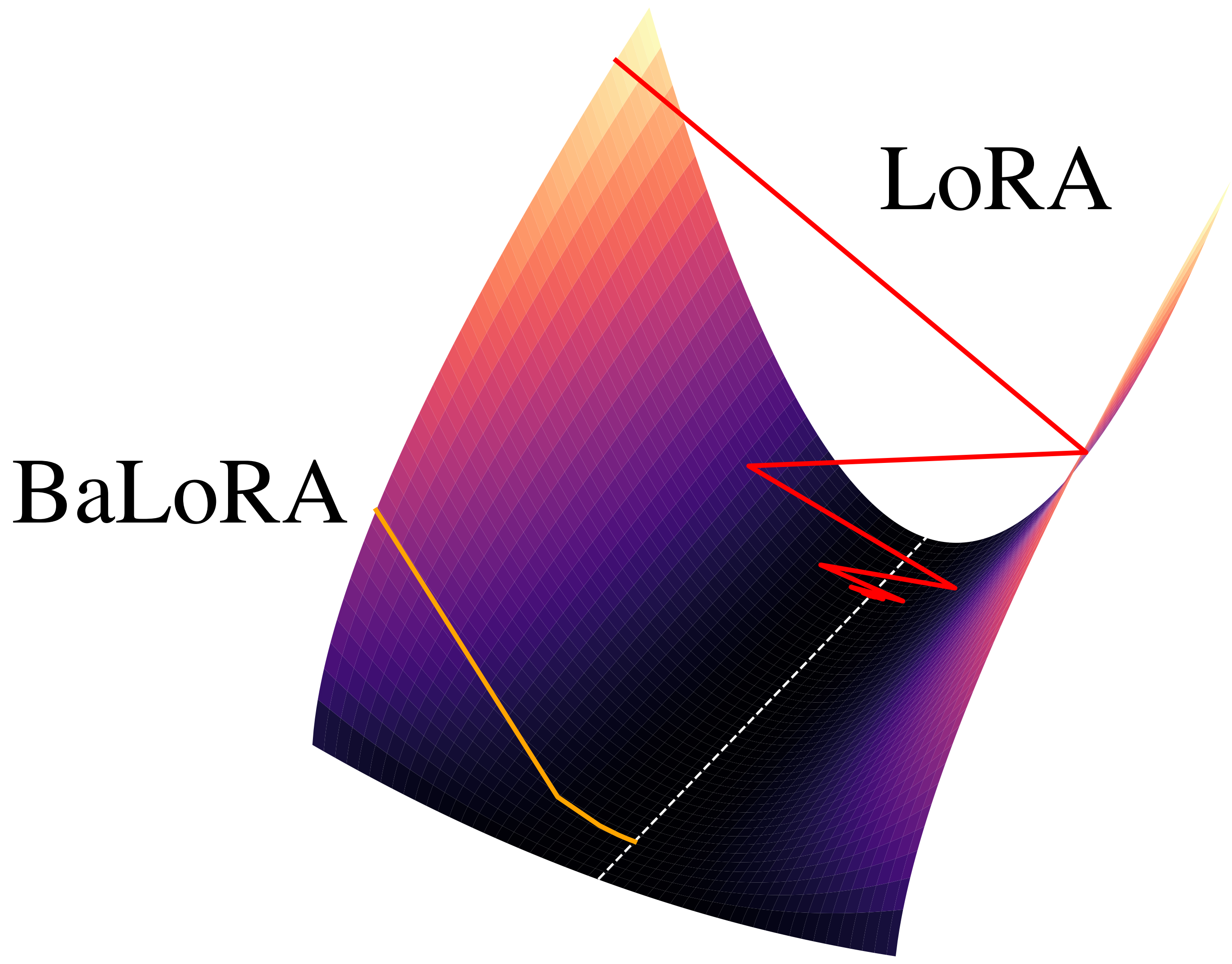
f is **overparameterized**:
$f(AR, R^{-1}B) = f(A, B)$
for R invertible $r \times r$

Question: How does the (asymptotic) rate depend on the limiting minimizer?

Outline

I - Asymptotic convergence rate and conditioning of minimizers

II - Balanced Low-Rank Adaptation (BaLoRA)

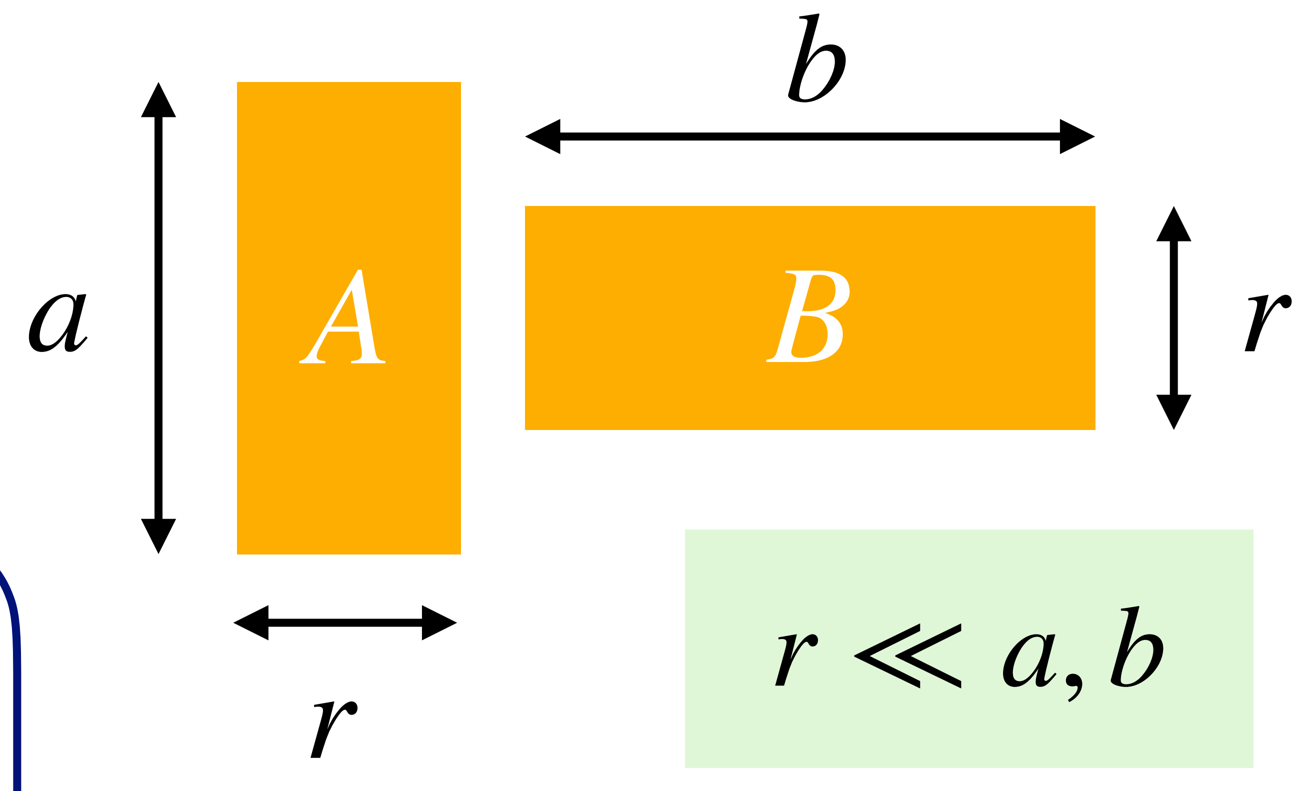


I - Asymptotic convergence rate and conditioning of loss minimizers

Conditioning of the loss and asymptotic convergence rate

Model: one-layer linear network W_{frozen} with $Z := W^* - W_{frozen}$

$$f(A, B) = \frac{1}{2}\|Z - AB\|_F^2$$



GD iterations: $\begin{cases} A_{t+1} = A_t - \gamma \nabla_A f(A_t, B_t) \\ B_{t+1} = B_t - \gamma \nabla_B f(A_t, B_t) \end{cases}$ → CV to minimizer (A^*, B^*) [Nguegnang et al., 2024]

Denote $H^* := D^2 f(A^*, B^*)$ Hessian of loss, $L := \lambda_{max}(H^*)$ and $\mu := \lambda_{min \neq 0}(H^*)$

Classical lemma: with $\kappa := L/\mu$, we have

$$\limsup_{t \to +\infty} \frac{f(A_{t+1}, B_{t+1}) - f^*}{f(A_t, B_t) - f^*} \leq \max((1 - \gamma L)^2, (1 - \gamma\mu)^2) \underbrace{\leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^2}_{\text{if } \gamma = 2/(L + \mu)}$$

Goal: understand the eigenvalues of $H = D^2 f(A, B)$ for each minimizer (A, B) to understand asymptotic convergence rate

Diagonalizing the Hessian of the loss at a minimizer [VC et al. 2025]

$$f(A,B) = \frac{1}{2}\|Z - AB\|_F^2 \qquad L := \lambda_{max}(H) \qquad \mu = \lambda_{min\neq 0}(H)$$

Lemma: Hessian of f at minimizer (A, B)

$$H = \begin{pmatrix} (BB^\top)\otimes I_a & B \otimes A \\ B^\top \otimes A^\top & I_b \otimes (A^\top A) \end{pmatrix} + \begin{pmatrix} 0 & (I_r \otimes (AB - Z))K_{r,b} \\ ((AB-Z)^\top \otimes I_r)K_{a,r} & 0 \end{pmatrix}$$

with K commutation matrices: $\mathrm{vec}(X^\top) = K\mathrm{vec}(X)$

Proposition: Matrix factorization case

$$L = \sigma_1(A)^2 + \sigma_1(B)^2$$

$$\mu = \min(\sigma_r(A)^2, \sigma_r(B)^2)$$

Proposition: General case

$$L = \sigma_1(A)^2 + \sigma_1(B)^2$$

$$\min(\sigma_r(A)^2, \sigma_r(B)^2) - \sigma_{r+1}(Z) \leq \mu \leq \min(\sigma_r(A)^2, \sigma_r(B)^2)$$

Proposition: optimal conditioning $\kappa_{opt} = \dfrac{2\sigma_1(Z)}{\sigma_r(Z) - \sigma_{r+1}(Z)}$

Balanced minimizers are optimally conditioned

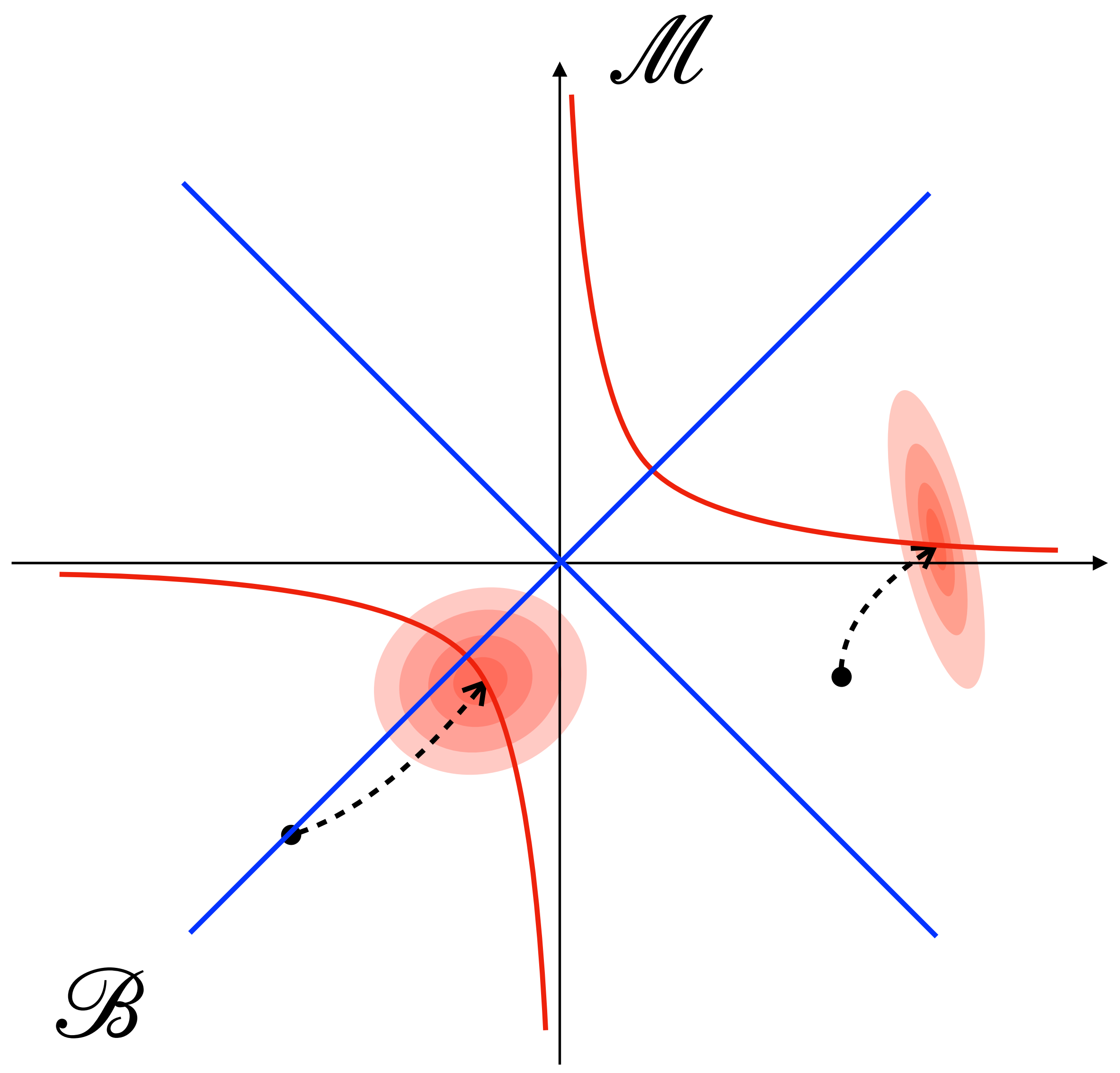
[VC et al. 2025]

Proposition:
H is optimally conditioned if (A, B) is **balanced**, i.e. $A^\top A = BB^\top$.

Balanced = same pairwise scalar products for the columns of A and $B^\top$

Appears in the literature:

- GD and gradient flow for multi-layer NNs [Du et al., 2018; Nguegnang et al. 2024]
- Deep matrix factorization [Ghosh et al., 2025]
- Conservation laws in NNs [Marcotte et al., 2023]



$$\mathscr{B} := \{(A, B) \in \mathbb{R}^{a\times r} \times \mathbb{R}^{r\times b} : A^\top A = BB^\top\}$$
$$\mathscr{M} := \{(A, B) \in \mathbb{R}^{a\times r} \times \mathbb{R}^{r\times b} : AB = LR_r(Z)\}$$

Variants of LoRA with **balanced init**: PiSSA [Meng et al., 2024], Lora-GA [Wang et al., 2024]

Generalization to the deep non-linear case

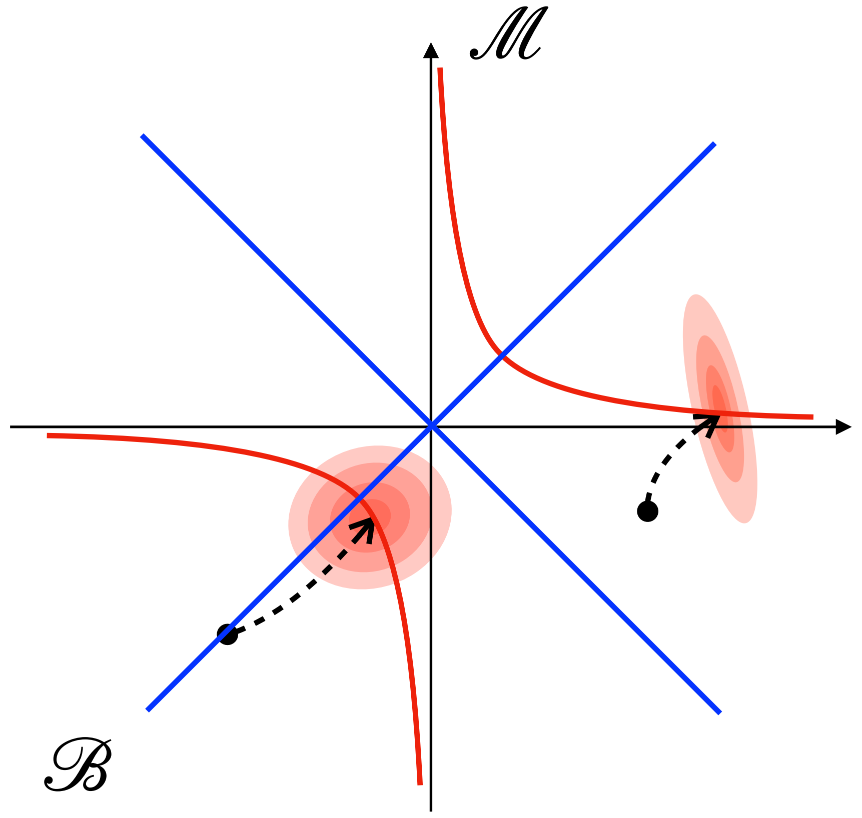
[VC et al. 2025]

Loss: $f(A, B) = \frac{1}{2}\|h(AB) - Z\|_F^2$ with h generic deep non-linear model

Example: $h(AB) = W_2 \mathrm{ReLU}((W_1 + AB)X)$

Proposition: if $h(AB) = Z$ (interpolating regime),

$$\kappa(D^2 f(A, B)) \leq \underbrace{\kappa(Dh(AB)^\top Dh(AB))^{1/2}}_{\text{additional factor}} \underbrace{\frac{\sigma_1(A)^2 + \sigma_1(B)^2}{\min(\sigma_r(A)^2, \sigma_r(B)^2)}}_{\text{minimized for balanced}}$$



Idea: enforce converging to a balanced minimizer to accelerate asymptotic convergence

II - Balanced Low-Rank Adaptation (BaLoRA)

BaLoRA: projecting on the hyperbalanced manifold [VC et al. 2025]

Idea: After each optimizer step, enforce balancing of (A, B) while preserving $f(A, B)$

$$\begin{cases} \tilde{A}_{t+1} = A_t - \gamma \nabla_A f(A_t, B_t) \\ \tilde{B}_{t+1} = B_t - \gamma \nabla_B f(A_t, B_t) \\ \underbrace{(A_{t+1}, B_{t+1})}_{\text{balanced}} = P\underbrace{(\tilde{A}_{t+1}, \tilde{B}_{t+1})}_{\text{unbalanced}} \end{cases}$$

« Projection » step: decompose $AB = USV^\top$ and apply $P(A, B) = (US^{1/2}, S^{1/2}V^\top)$

P sends (A, B) to the **hyperbalanced manifold** $\mathscr{H}$:

$$\mathscr{H} = \{(US^{1/2}, S^{1/2}V^\top) \mid U^\top U = V^\top V = I_r \text{ and } S \succ 0 \text{ diagonal}\}$$

Lemma: $(A, B) \in \mathscr{H} \mapsto AB \in \{\text{rank-}r \text{ matrices}\}$ is surjective
→ optimizing in $\mathscr{H}$ essentially removes LoRA invariances

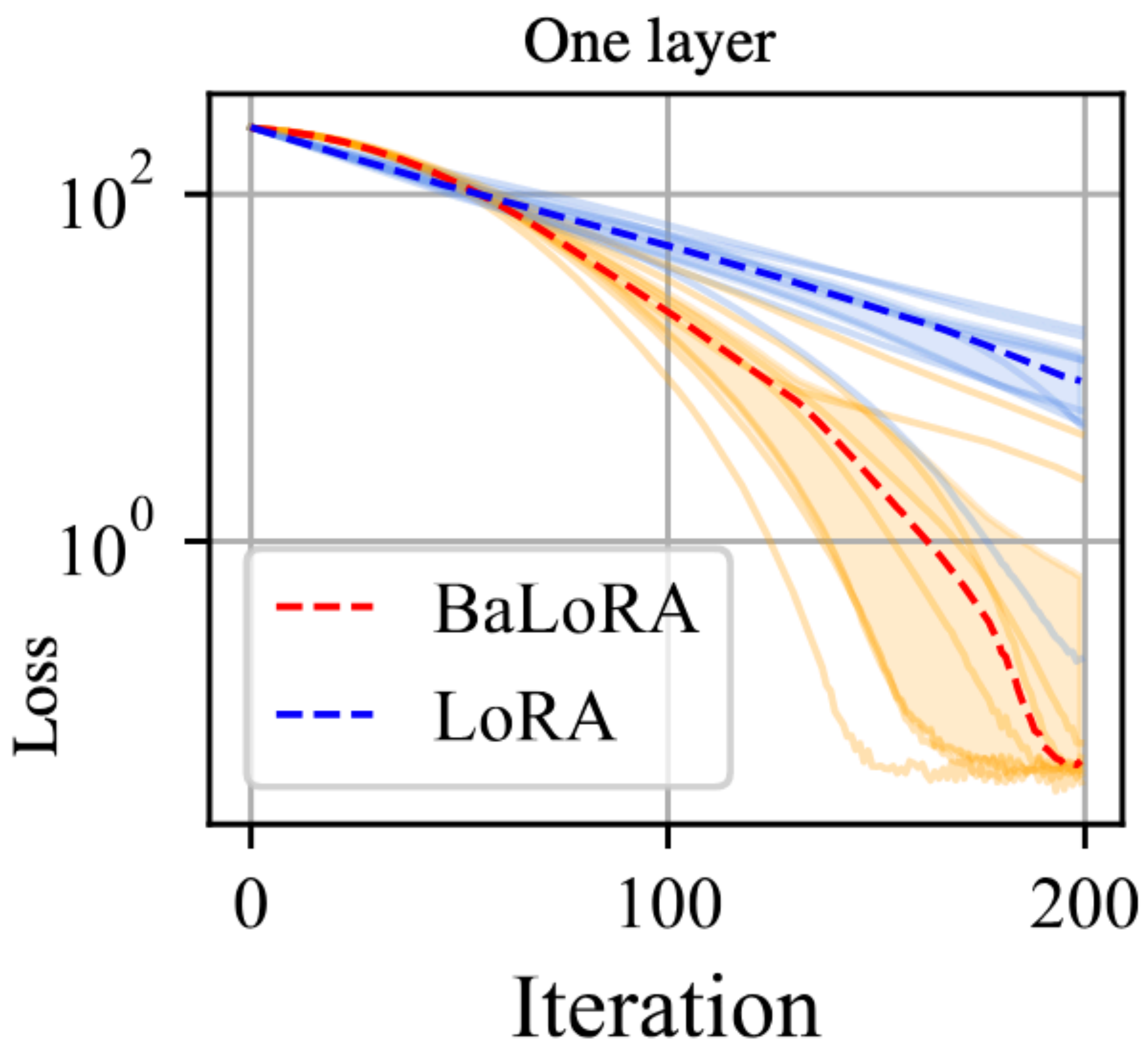
- ✓ preserves the product
- ✓ negligible computational overhead
- ✓ retraction-like properties
- ✓ can be written as an intrinsic GD on the manifold of rank-r matrices

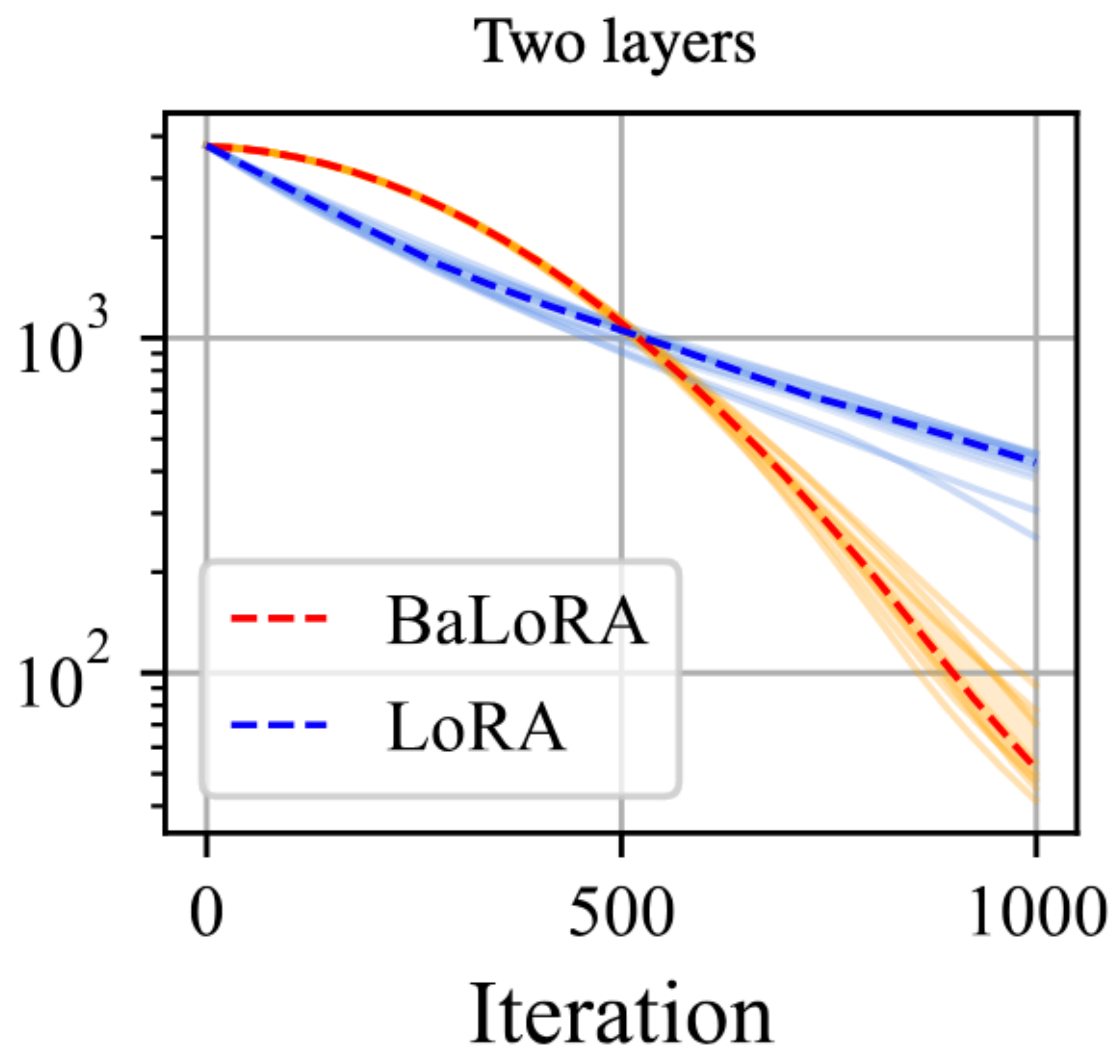
Experiments on synthetic data

[VC et al. 2025]

$$f(A,B) = \frac{1}{2}\|Z - AB\|_F^2$$

$$f((A_1,B_1),(A_2,B_2)) = \frac{1}{2}\|W^* - (W_{frozen,1} + A_1B_1)(W_{frozen,2} + A_2B_2)\|_F^2$$

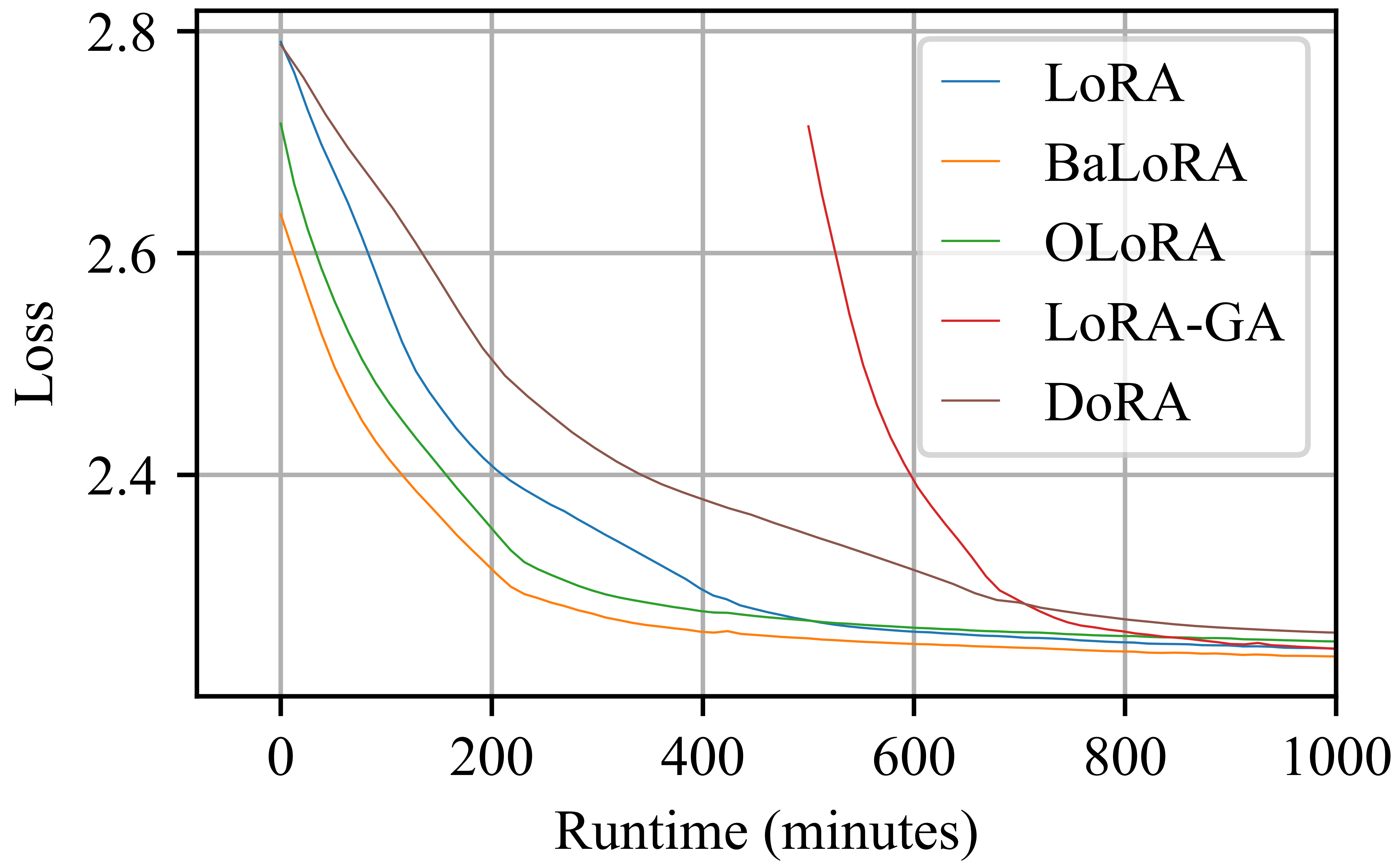


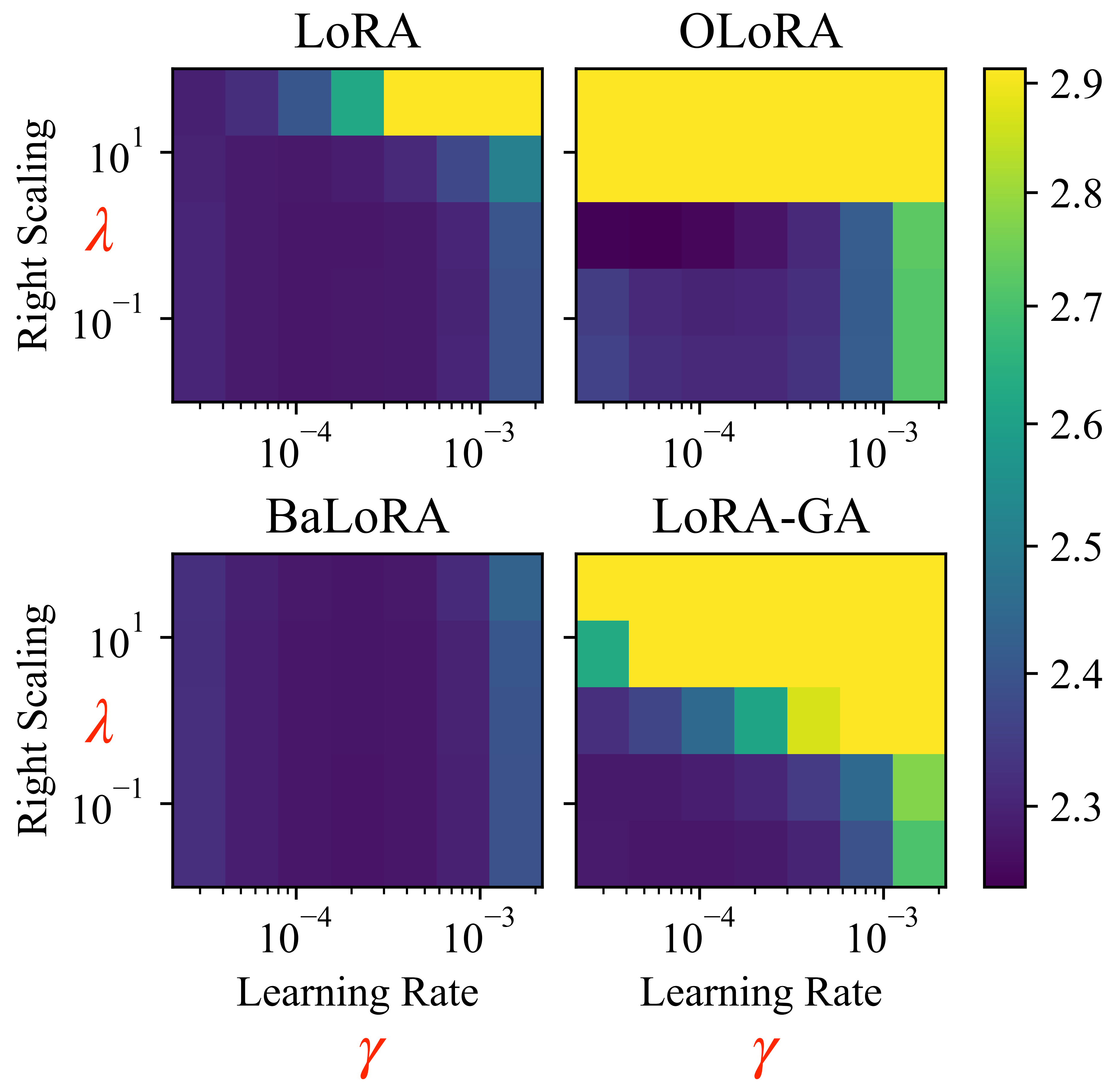


Experiments with LLMs on real data

[VC et al. 2025]

Fine-tuning Llama-3.2-3B on Wikitext-2-raw-v1:



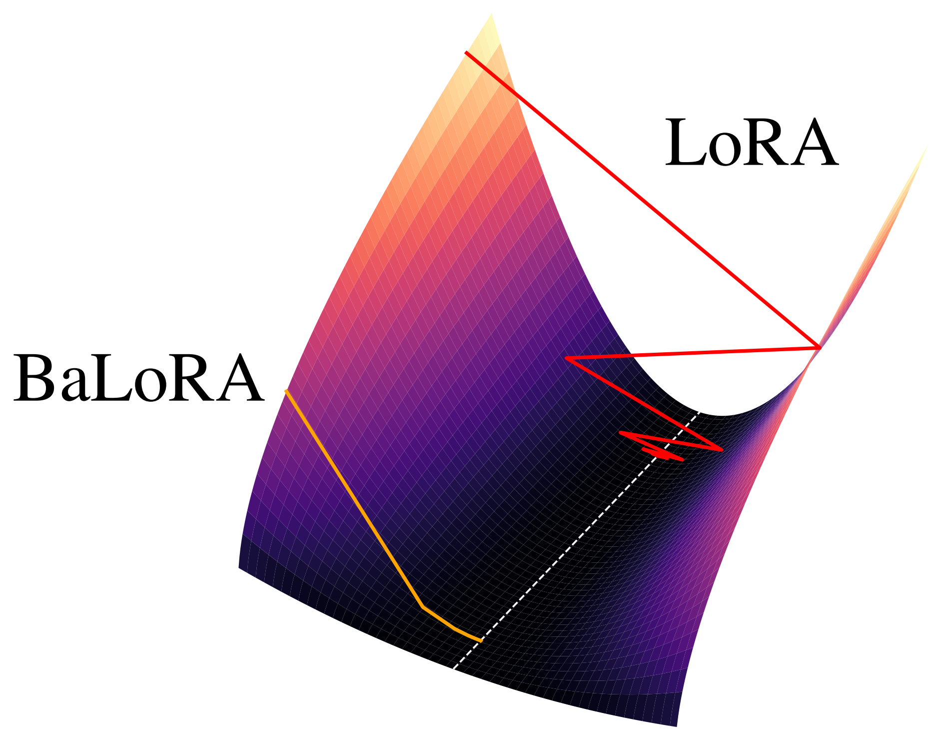


Conclusion

- LoRA is **overparameterized**, and different minimizers of the loss can have different condition numbers
- **Balanced** minimizers $A^\top A = BB^\top$ are **optimally conditioned**
- We build on this observation to introduce **Balanced Low-Rank Adaptation** (BaLoRA) → enforce balancedness of (A, B) after each optimizer step
- BaLoRA performs on par with or outperforms several baselines (LoRA, DoRA, OLoRA, LoRA-GA) on several datasets (1k to 500k samples) for GPT2, Llama-3.2-3B and Qwen2.5-3B

Perspectives:

- Extend theory in the deep non-linear case?
- Beyond asymptotic CV rate?
- Practical improvement on which datasets?



Thank you!

References:

[Hu et al., 2022] Low-rank adaptation of large language models

[Ye & Du, 2021] Global convergence of gradient descent for asymmetric low-rank matrix factorization

[Nguegnang et al., 2024] Convergence of gradient descent for learning linear neural networks

[Du et al., 2018] Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced

[Marcotte et al., 2023] Abide by the law and follow the flow: Conservation laws for gradient flows

[Ghosh et al., 2025] Learning dynamics of deep linear networks beyond the edge of stability

[Castin et al., 2025] Balanced low-rank adaptation: removing parameter invariance to accelerate convergence