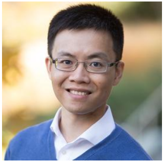
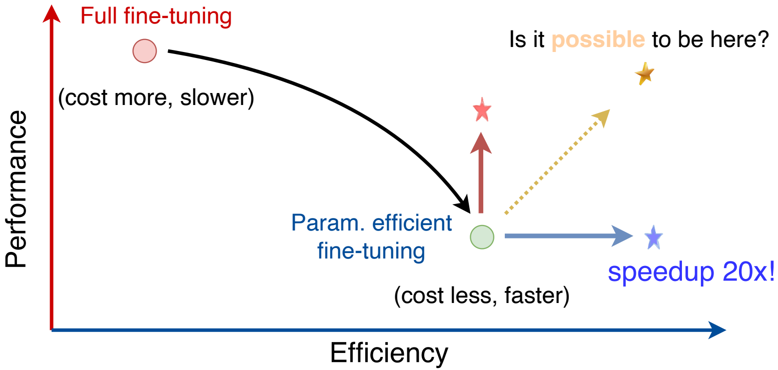
LoRA-One: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently

Yuanhe Zhang (Warwick), **Fanghui Liu** (Warwick), **Yudong Chen** (UW-Madison)



at ICML'25, Vancouver

How can theory contribute to efficiency in LLMs?

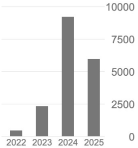


LoRA: Low-rank adaption

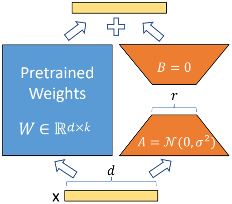
Published as a conference paper at ICLR 2022

LoRA: Low-Rank Adaptation of Large Language Models

Edward Hu* **Yelong Shen*** **Phillip Wallis** **Zeyuan Allen-Zhu**
Yuanzhi Li **Shean Wang** **Lu Wang** **Weizhu Chen**
Microsoft Corporation
`edward.hu@mila.quebec`
`{yeshe, phwallis, zeyuana, swang, luw, wzchen}@microsoft.com`
`yuanzhil@andrew.cmu.edu`



$$\boldsymbol{W}^{\mathrm{FT}} = \boldsymbol{W}^{\mathrm{pre}} + \Delta \in \mathbb{R}^{d \times k}$$



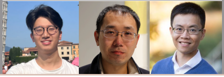
- Formulation:

$$\Delta \approx \boldsymbol{AB} \text{ with } \boldsymbol{A} \in \mathbb{R}^{d \times r} \text{ and } \boldsymbol{B} \in \mathbb{R}^{r \times k}$$

- Initialization:

$$[\boldsymbol{A}_0]_{ij} \sim \mathcal{N}(0, \alpha^2) \quad \text{and} \quad [\boldsymbol{B}_0]_{ij} = 0\,.$$

Today's talk: Improve "sub-optimal" LoRA



How can theory guide practice

- understanding: training dynamics of $(\boldsymbol{A}_t, \boldsymbol{B}_t)$
- design new algorithm -> performance improvement
- clarify some misconceptions in previous algorithm designs

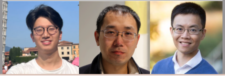
❒ Even for linear model (pre-training and fine-tuning), nonlinear dynamics...

$$\begin{bmatrix} \boldsymbol{A}_{t+1} \\ \boldsymbol{B}_{t+1}^\top \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_d & \eta \boldsymbol{G} \\ \eta \boldsymbol{G}^\top & \boldsymbol{I}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_t \\ \boldsymbol{B}_t^\top \end{bmatrix} + \text{nonlinear term} \qquad \begin{cases} [\boldsymbol{A}_0]_{ij} \sim \mathcal{N}(0, \alpha^2) \\ [\boldsymbol{B}_0]_{ij} = 0\,. \end{cases}$$

❒ One-step full gradient: $\boldsymbol{G} \in \mathbb{R}^{d \times k}$ and $\operatorname{rank}(\boldsymbol{G}) = r^*$

$$\boldsymbol{G} := -\nabla_{\boldsymbol{W}} L(\boldsymbol{W}^{\text{pre}}) = \frac{1}{N} \widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{X}} \Delta\,.$$

Today's talk: Improve "sub-optimal" LoRA



How can theory guide practice

- understanding: training dynamics of $(\boldsymbol{A}_t, \boldsymbol{B}_t)$
- design new algorithm -> performance improvement
- clarify some misconceptions in previous algorithm designs

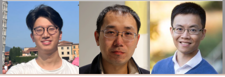
❒ Even for linear model (pre-training and fine-tuning), nonlinear dynamics...

$$\begin{bmatrix} \boldsymbol{A}_{t+1} \\ \boldsymbol{B}_{t+1}^\top \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_d & \eta \boldsymbol{G} \\ \eta \boldsymbol{G}^\top & \boldsymbol{I}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_t \\ \boldsymbol{B}_t^\top \end{bmatrix} + \text{nonlinear term} \qquad \begin{cases} [\boldsymbol{A}_0]_{ij} \sim \mathcal{N}(0, \alpha^2) \\ [\boldsymbol{B}_0]_{ij} = 0 . \end{cases}$$

❒ One-step full gradient: $\boldsymbol{G} \in \mathbb{R}^{d \times k}$ and $\text{rank}(\boldsymbol{G}) = r^*$

$$\boldsymbol{G} := -\nabla_{\boldsymbol{W}} L(\boldsymbol{W}^{\text{pre}}) = \frac{1}{N} \widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{X}} \Delta .$$

Today's talk: Improve "sub-optimal" LoRA



How can theory guide practice

- understanding: training dynamics of $(\boldsymbol{A}_t, \boldsymbol{B}_t)$
- design new algorithm -> performance improvement
- clarify some misconceptions in previous algorithm designs

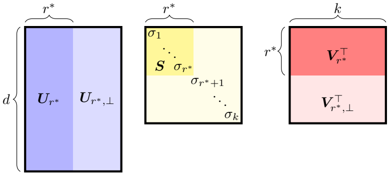
❒ Even for linear model (pre-training and fine-tuning), nonlinear dynamics...

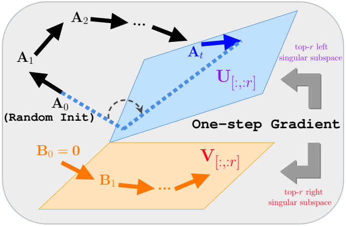
$$\begin{bmatrix} \boldsymbol{A}_{t+1} \\ \boldsymbol{B}_{t+1}^\top \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_d & \eta \boldsymbol{G} \\ \eta \boldsymbol{G}^\top & \boldsymbol{I}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_t \\ \boldsymbol{B}_t^\top \end{bmatrix} + \text{nonlinear term} \qquad \begin{cases} [\boldsymbol{A}_0]_{ij} \sim \mathcal{N}(0, \alpha^2) \\ [\boldsymbol{B}_0]_{ij} = 0\,. \end{cases}$$

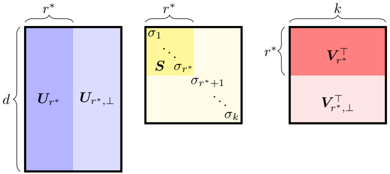
❒ One-step full gradient: $\boldsymbol{G} \in \mathbb{R}^{d \times k}$ and $\operatorname{rank}(\boldsymbol{G}) = r^*$

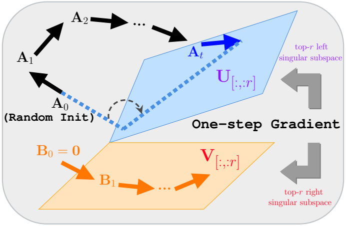
$$\boldsymbol{G} := -\nabla_{\boldsymbol{W}} L(\boldsymbol{W}^{\text{pre}}) = \frac{1}{N} \widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{X}} \Delta\,.$$

Understanding: Alignment on B_t







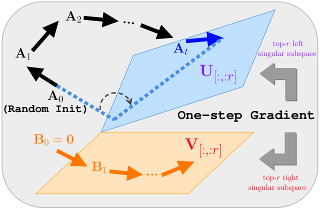


Theorem (Alignment between G and B_t, informal)

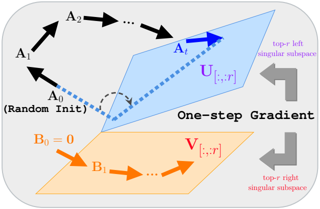
For the linear setting, LoRA via gradient descent yields

$$\angle(V_{r^*}(B_t), V_{r^*}(G)) = 0\,, \quad \forall t \in \mathbb{N}_+\,.$$

Understanding: Alignment on A_t



$$\begin{bmatrix} \boldsymbol{A}_{t+1} \\ \boldsymbol{B}_{t+1}^\top \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_d & \eta \boldsymbol{G} \\ \eta \boldsymbol{G}^\top & \boldsymbol{I}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_t \\ \boldsymbol{B}_t^\top \end{bmatrix} + \text{nonlinear term}$$

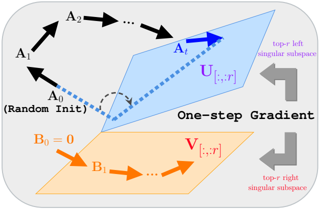


$$\begin{bmatrix} \boldsymbol{A}_{t+1} \\ \boldsymbol{B}_{t+1}^\top \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_d & \eta \boldsymbol{G} \\ \eta \boldsymbol{G}^\top & \boldsymbol{I}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_t \\ \boldsymbol{B}_t^\top \end{bmatrix} + \text{nonlinear term}$$

Theorem (Alignment between G and A_t, informal)

For small initialization over $\boldsymbol{A}_0$, after $t^* = \Theta(\ln d)$ steps, LoRA updates yield

$$\angle(\boldsymbol{U}_{r^*}(\boldsymbol{A}_{t^*}), \boldsymbol{U}_{r^*}(\boldsymbol{G})) \text{ is small, w.h.p.}$$



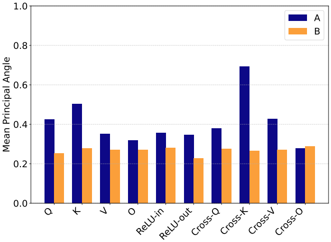


Figure 2: Principal angle of fine-tuning T5 on MRPC.

Theorem (Alignment between G and A_t, informal)

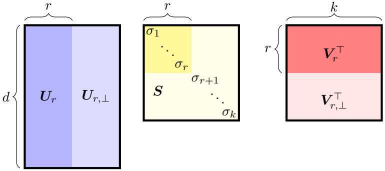
For small initialization over $\mathbf{A}_0$, after $t^* = \Theta(\ln d)$ steps, LoRA updates yield

$$\angle(U_{r^*}(A_{t^*}), U_{r^*}(G)) \text{ is small, w.h.p.}$$

Algorithm design principle

❐ SVD: $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$

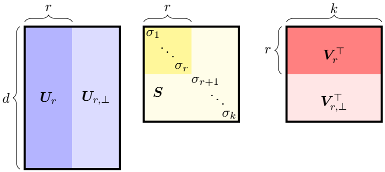
$$\boldsymbol{A}_0 = \boldsymbol{U}_{[:,1:r]}\boldsymbol{S}_{[1:r]}^{\frac{1}{2}}\,.$$
$$\boldsymbol{B}_0 = \boldsymbol{S}_{[1:r]}^{\frac{1}{2}}\boldsymbol{V}_{[:,1:r]}^\top\,. \qquad \text{(Spec-init.)}$$



Algorithm design principle

❐ SVD: $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$

$$\boldsymbol{A}_0 = \boldsymbol{U}_{[:,1:r]}\boldsymbol{S}^{\frac{1}{2}}_{[1:r]}\,, \qquad \text{(Spec-init.)}$$
$$\boldsymbol{B}_0 = \boldsymbol{S}^{\frac{1}{2}}_{[1:r]}\boldsymbol{V}^\top_{[:,1:r]}\,.$$



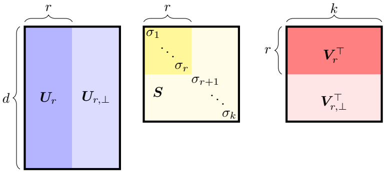
Key Message: we can "escape" the alignment stage

Under (Spec-init.), for both linear/nonlinear models, we can directly achieve the alignment at initialization.

$$\|\boldsymbol{A}_0\boldsymbol{B}_0 - \Delta\|_{\mathrm{F}} \text{ is small, } w.h.p.$$

❒ SVD: $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$

$$\boldsymbol{A}_0 = \boldsymbol{U}_{[:,1:r]}\boldsymbol{S}^{\frac{1}{2}}_{[1:r]}\,,\qquad \boldsymbol{B}_0 = \boldsymbol{S}^{\frac{1}{2}}_{[1:r]}\boldsymbol{V}^\top_{[:,1:r]}\,. \qquad \text{(Spec-init.)}$$



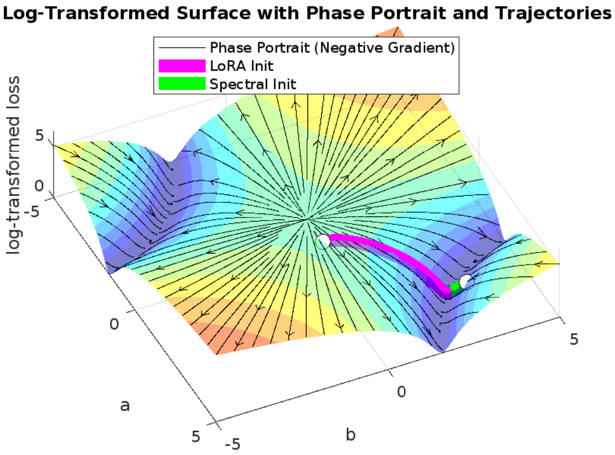
Key Message: we can "escape" the alignment stage

Under (Spec-init.), for both linear/nonlinear models, we can directly achieve the alignment at initialization.

$$\|\boldsymbol{A}_0\boldsymbol{B}_0 - \Delta\|_{\mathrm{F}} \text{ is small, } w.h.p.$$

The "best" initialization strategy!

“Best” initialization: phase portrait



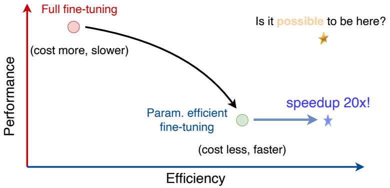
One-step gradient can suffice on small-scale datasets!

Dataset	**MNLI**	**SST-2**	**CoLA**	**QNLI**	**MRPC**
Size	393k	67k	8.5k	105k	3.7k
Pre-trained	-	89.79	59.03	49.28	63.48
Spectral init.	-	90.48	73.00	76.64	68.38
LoRA_8	$85.30_{\pm 0.04}$	$94.04_{\pm 0.09}$	$72.84_{\pm 1.25}$	$93.02_{\pm 0.07}$	$68.38_{\pm 0.01}$

One-step gradient can suffice on small-scale datasets!

Dataset	MNLI	SST-2	CoLA	QNLI	MRPC
Size	393k	67k	8.5k	105k	3.7k
Pre-trained	-	89.79	59.03	49.28	63.48
Spectral init.	-	90.48	73.00	76.64	68.38
LoRA_8	$85.30_{\pm 0.04}$	$94.04_{\pm 0.09}$	$72.84_{\pm 1.25}$	$93.02_{\pm 0.07}$	$68.38_{\pm 0.01}$

Time cost (sec.)	LoRA	Spectral init.
CoLA	47s	$<$1s
MRPC	25s	$<$1s



Results on LLaMA 2-7B (continue to run)

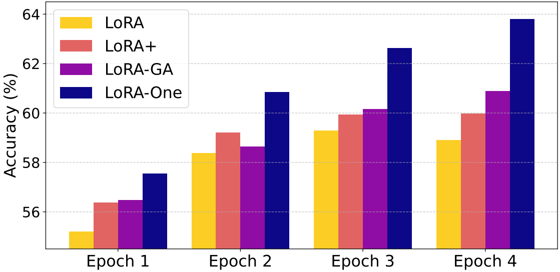


Figure 3: Accuracy comparison across different methods over epochs on GSM8K.

Time cost	LoRA: 6h 20min	+ 3min
Memory	LoRA: 21.6 GB	+ 0.1GB

Results on LLaMA 2-7B (continue to run)

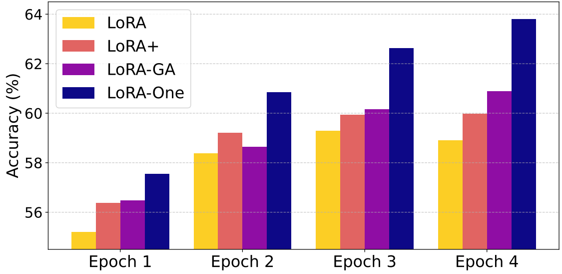
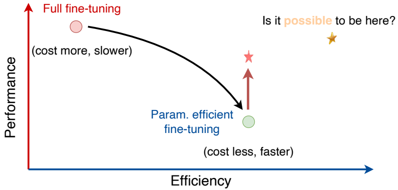


Figure 3: Accuracy comparison across different methods over epochs on GSM8K.

Time cost	LoRA: 6h 20min	+ 3min
Memory	LoRA: 21.6 GB	+ 0.1GB



Clarification on gradient alignment based work

LoRA-GA (Wang et al, 2024): make LoRA's gradients align to full fine-tuning!

Clarification on gradient alignment based work

LoRA-GA (Wang et al, 2024): make LoRA's gradients align to full fine-tuning!

❒ best $2r$ approximation

$$\begin{cases} \operatorname{rank}(\nabla_{\boldsymbol{A}} L(\boldsymbol{A}_t, \boldsymbol{B}_t)) \leq r \\ \operatorname{rank}(\nabla_{\boldsymbol{B}} L(\boldsymbol{A}_t, \boldsymbol{B}_t)) \leq r \end{cases}$$

Clarification on gradient alignment based work

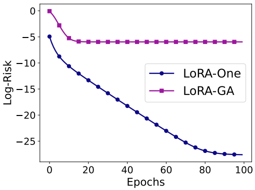
LoRA-GA (Wang et al, 2024): make LoRA's gradients align to full fine-tuning!

❒ best $2r$ approximation

$$\begin{cases} \mathrm{rank}(\nabla_{\boldsymbol{A}} L(\boldsymbol{A}_t, \boldsymbol{B}_t)) \leq r \\ \mathrm{rank}(\nabla_{\boldsymbol{B}} L(\boldsymbol{A}_t, \boldsymbol{B}_t)) \leq r \end{cases}$$

Method	Init. on $\boldsymbol{A}$	Init. on $\boldsymbol{B}$	Calibration
LoRA	$\mathcal{N}(0, \alpha^2)$	0	-
LoRA-GA	$\boldsymbol{U}_{[:,1:r]}$	$\boldsymbol{V}^\top_{[:,r+1:2r]}$	$\boldsymbol{W}^{\mathrm{pre}} - \boldsymbol{A}_0 \boldsymbol{B}_0$
LoRA-One	$\boldsymbol{U}_{[:,1:r]} \boldsymbol{S}^{1/2}_{[1:r]}$	$\boldsymbol{S}^{1/2}_{[1:r]} \boldsymbol{V}^\top_{[:,1:r]}$	-

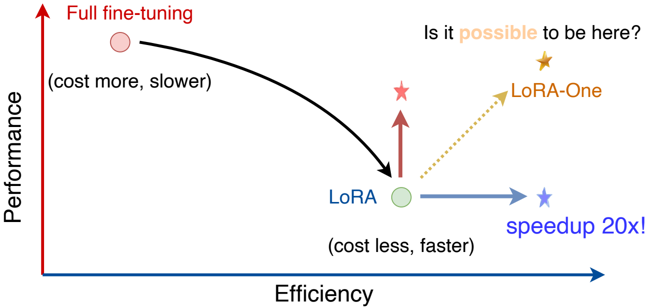
LoRA-GA (Wang et al, 2024): make LoRA's gradients align to full fine-tuning!



Method	Init. on $\boldsymbol{A}$	Init. on $\boldsymbol{B}$	Calibration
LoRA	$\mathcal{N}(0, \alpha^2)$	0	-
LoRA-GA	$\boldsymbol{U}_{[:,1:r]}$	$\boldsymbol{V}^{\top}_{[:,r+1:2r]}$	$\boldsymbol{W}^{\text{pre}} - \boldsymbol{A}_0\boldsymbol{B}_0$
LoRA-One	$\boldsymbol{U}_{[:,1:r]}\boldsymbol{S}^{1/2}_{[1:r]}$	$\boldsymbol{S}^{1/2}_{[1:r]}\boldsymbol{V}^{\top}_{[:,1:r]}$	-

Takeaway messages: speedup via spectral initialization

LoRA-One



- subspace alignment: $\boldsymbol{G}$ and $(\boldsymbol{A}_t, \boldsymbol{B}_t) \Rightarrow$ theory-grounded algorithm design
- "optimal" non-zero initialization strategy
- clarification on gradient alignment based algorithms
- spectral initialization enables feature learning...
- global convergence on nonlinear models, scaled GD...