LoRA-Pro: Are Low-Rank Adapters Properly Optimized?

Zhengbo Wang [1,2] Jian Liang [2,3] † Ran He [2,3] Zilei Wang [1] Tieniu Tan [2,4]

[1] University of Science and Technology of China

[2] NLPR & MAIS, Institute of Automation, Chinese Academy of Sciences (CASIA)

[3] School of Artificial Intelligence, University of Chinese Academy of Sciences

[4] Nanjing University

Proceedings of the International Conference on Learning Representations (ICLR), 2025

Presenter: Minwoo Jang (POSTECH GSAI), Donghyun Lim (POSTECH EE)

TL;DR

- Low-Rank Adaptation (LoRA) methods often fail to faithfully mimic full-parameter fine-tuning, causing LoRA-adapted foundation models to converge to suboptimal solutions.

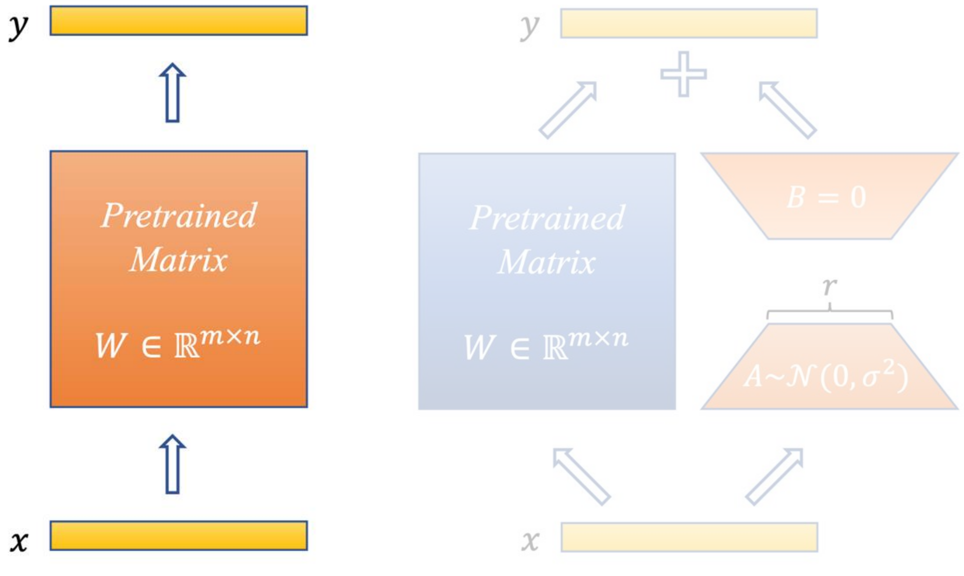
- LoRA-Pro attributes this gap to the way gradients are computed and proposes a principled correction of the LoRA gradients.

Contents

Contents

Full Fine-Tuning Vs. Low-Rank Adaptation (LoRA)

- Assume a pre-trained weight matrix $W_0 \in \mathbb{R}^{m \times n}$.
- Full Fine-Tuning: $W \leftarrow W - \eta \cdot \Delta W$

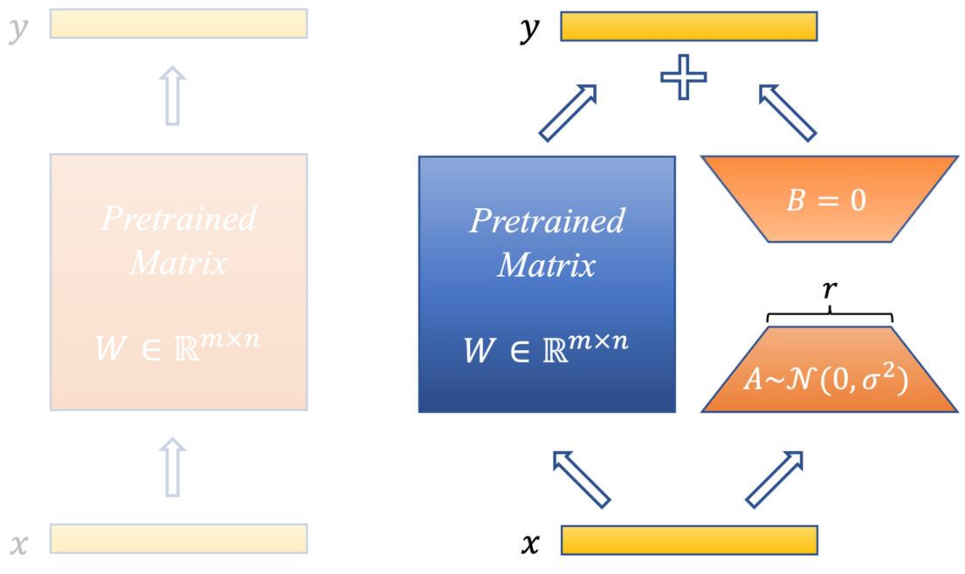


(a) Full Fine-tuning (b) LoRA

[NeurIPS 2024] PiSSA: Principal Singular Values and Singular Vectors Adaptation of Large Language Models (Meng et al.)

Full Fine-Tuning Vs. Low-Rank Adaptation (LoRA)

- Assume a pre-trained weight matrix $W_0 \in \mathbb{R}^{m \times n}$.
- LoRA: $W = W_0 + \Delta W = W_0 + \frac{\alpha}{r} \cdot BA$
  - $B \in \mathbb{R}^{m \times r}$, $A \in \mathbb{R}^{r \times n}$ with $r \ll \min\{m, n\}$



(a) Full Fine-tuning (b) LoRA

[NeurIPS 2024] PiSSA: Principal Singular Values and Singular Vectors Adaptation of Large Language Models (Meng et al.)

Low-Rank Adaptation (LoRA)

- Assume a pre-trained weight matrix $W_0 \in \mathbb{R}^{m \times n}$.
- LoRA: $W = W_0 + \Delta W = W_0 + \frac{\alpha}{r} \cdot BA$
  - $B \in \mathbb{R}^{m \times r}$, $A \in \mathbb{R}^{r \times n}$ with $r \ll \min\{m, n\}$
  - Here, α denotes the scaling factor.
    - In practice, when comparing different LoRA ranks r, practitioners often keep the learning rate fixed and adjust α instead, typically choosing it proportional to r so that the effective scaling $\frac{\alpha}{r}$ remains roughly constant.
    - From now on, we denote $W = W_0 + s \cdot BA$, where $s \coloneqq \frac{\alpha}{r}$.

Contents

Back-propagation (Full Fine-Tuning)

- Let $W \in \mathbb{R}^{m \times n}$ be the (pre-trained) weight matrix.

- The first-order change in the loss L is

$$dL = \left\langle \frac{\partial L}{\partial W}, dW \right\rangle_F ,$$

  where $L(W + dW) \approx L(W) + dL$

- For simplicity, we omit the learning rate and consider a GD step

$$dW = -\frac{\partial L}{\partial W} .$$

Back-propagation (Full Fine-Tuning)

- Define $g \coloneqq \frac{\partial L}{\partial W}$. Then, we obtain

$$dL = \left\langle \frac{\partial L}{\partial W}, dW \right\rangle_F = \left\langle \frac{\partial L}{\partial W}, -\frac{\partial L}{\partial W} \right\rangle_F = \langle g, -g \rangle_F = -\|g\|_F^2 \leq 0$$

since the squared Frobenius norm is always non-negative.

Back-propagation (LoRA)

- We parameterize the weight as $W = W_0 + s \cdot BA$.
  - Here, $B \in \mathbb{R}^{m \times r}$, $A \in \mathbb{R}^{r \times n}$ with $r \ll \min\{m, n\}$.

- Taking differentials, we have $dW = sB(dA) + s(dB)A$.

- Let $g := \frac{\partial L}{\partial W}$, then the first-order change in the loss is

$$
\begin{aligned}
dL = \langle g, dW \rangle_F &= \langle g, sB(dA) + s(dB)A \rangle_F \\
&= s\langle g, B(dA) \rangle_F + s\langle g, (dB)A \rangle_F \\
&= \langle sB^T g, dA \rangle_F + \langle sgA^T, dB \rangle_F \,.
\end{aligned}
$$

Back-propagation (LoRA)

$$dL = \langle sB^T g, dA \rangle_F + \langle sgA^T, dB \rangle_F$$

- By the definition of gradient, we also have

$$dL = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle_F + \left\langle \frac{\partial L}{\partial B}, dB \right\rangle_F ,$$

  which implies that $\frac{\partial L}{\partial A} = sB^T g$ and $\frac{\partial L}{\partial B} = sgA^T$.

  - These two identities will be used later, not now.

Back-propagation (LoRA)

$$dL = \langle sB^T g, dA \rangle_F + \langle sgA^T, dB \rangle_F$$

- By the definition of gradient, we also have

$$dL = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle_F + \left\langle \frac{\partial L}{\partial B}, dB \right\rangle_F .$$

- For simplicity, we omit the learning rate and consider a GD step

$$dA = -\frac{\partial L}{\partial A}, \qquad dB = -\frac{\partial L}{\partial B}$$

Back-propagation (LoRA)

- Define $g_{LoRA}^{B} \coloneqq \frac{\partial L}{\partial B}$ and $g_{LoRA}^{A} \coloneqq \frac{\partial L}{\partial A}$. Then, we obtain

$$
\begin{aligned}
dL &= \left\langle \frac{\partial L}{\partial A}, dA \right\rangle_F + \left\langle \frac{\partial L}{\partial B}, dB \right\rangle_F = \left\langle \frac{\partial L}{\partial A}, -\frac{\partial L}{\partial A} \right\rangle_F + \left\langle \frac{\partial L}{\partial B}, -\frac{\partial L}{\partial B} \right\rangle_F \\
&= \left\langle g_{LoRA}^{A}, -g_{LoRA}^{A} \right\rangle_F + \left\langle g_{LoRA}^{B}, -g_{LoRA}^{B} \right\rangle_F \\
&= -\left\| g_{LoRA}^{A} \right\|_F^2 - \left\| g_{LoRA}^{B} \right\|_F^2 \leq 0
\end{aligned}
$$

since the squared Frobenius norm is always non-negative.

Connection between Full Fine-Tuning and LoRA

- What we've calculated before:

$$g_{LoRA}^{B} \coloneqq \frac{\partial L}{\partial B} = sgA^T, \qquad g_{LoRA}^{A} \coloneqq \frac{\partial L}{\partial A} = sB^T g, \qquad g \coloneqq \frac{\partial L}{\partial W}$$

$$dA = -\frac{\partial L}{\partial A} = -g_{LoRA}^{A}, \qquad dB = -\frac{\partial L}{\partial B} = -g_{LoRA}^{B}$$

- Now, again noting that $W = W_0 + s \cdot BA$,

$$dW = sB(dA) + s(dB)A = s(-g_{LoRA}^{B})A + sB(-g_{LoRA}^{A})$$
$$= -(s g_{LoRA}^{B} A + sB g_{LoRA}^{A})$$

Connection between Full Fine-Tuning and LoRA

$$dW = \frac{\partial W}{\partial A} dA + \frac{\partial W}{\partial B} dB = -(s \cdot B g^A + s \cdot g^B A)$$

- Changes in A and B are inherently linked to changes in matrix W:
  - LoRA optimization, i.e., updating B with g^B and A with g^A, respectively, is equivalent to the full fine-tuning with $\tilde{g} \coloneqq s \cdot g^B A + s \cdot B g^A$.

Connection between Full Fine-Tuning and LoRA

$$dW = \frac{\partial W}{\partial A} dA + \frac{\partial W}{\partial B} dB = -(s \cdot B g^A + s \cdot g^B A)$$

- Changes in A and B are inherently linked to changes in matrix W:
  - LoRA optimization, i.e., updating B with g^B and A with g^A, respectively, is equivalent to the full fine-tuning with $\tilde{g} \coloneqq s \cdot g^B A + s \cdot B g^A$.

Definition 1 (Equivalent Gradient). *In the context of LoRA optimization, we define the equivalent gradient as,*

$$\tilde{g} \triangleq sBg^A + sg^B A,$$

where s is the scaling factor, and g^A and g^B are gradients with respect to A and B, respectively.

Q) When using $\tilde{g}$, how much information is lost?

Lemma. *Assume $B \in \mathbb{R}^{m \times r}$, $A \in \mathbb{R}^{r \times n}$ and $g^B \in \mathbb{R}^{m \times r}$, $g^A \in \mathbb{R}^{r \times n}$ represent matrices and their corresponding gradients in LoRA optimization. We demonstrate that the equivalent gradient:*

$$\tilde{g} = \underbrace{\boxed{sg^B A}}_{\text{at most } r} + \underbrace{\boxed{sBg^A}}_{\text{at most } r}, \tag{17}$$

where $s > 0$ is the scaling factor, has matrix rank at most $2r$.

- Note:
  - In this Lemma, "rank" stands for the matrix rank dealt with in Linear Algebra, not the rank r defined for the parameter size of LoRA.
  - The full gradient g can have at most $\min\{m, n\} \gg r$ matrix rank.
  - Takeaways: Equivalent gradient has low rank.

Q) When using $\tilde{g}$, how much information is lost?

Lemma. *Assume $B \in \mathbb{R}^{m\times r}$, $A \in \mathbb{R}^{r\times n}$ and $g^B \in \mathbb{R}^{m\times r}$, $g^A \in \mathbb{R}^{r\times n}$ represent matrices and their corresponding gradients in LoRA optimization. We demonstrate that the equivalent gradient:*

$$\tilde{g} = \overset{\text{at most } r}{\boxed{sg^B A}} + \overset{\text{at most } r}{\boxed{sBg^A}}, \tag{17}$$

where $s > 0$ is the scaling factor, has matrix rank at most $2r$.

- Proof) Note that for any two matrices X and Y such that the product and sum are well-defined, the following holds:
  - $\operatorname{rank}(X+Y) \le \operatorname{rank}(X) + \operatorname{rank}(Y)$
  - $\operatorname{rank}(XY) \le \min\{\operatorname{rank}(X), \operatorname{rank}(Y)\}$

Q) When using $\tilde{g}$, how much information is lost?

Lemma. *Assume $B \in \mathbb{R}^{m\times r}$, $A \in \mathbb{R}^{r\times n}$ and $g^B \in \mathbb{R}^{m\times r}$, $g^A \in \mathbb{R}^{r\times n}$ represent matrices and their corresponding gradients in LoRA optimization. We demonstrate that the equivalent gradient:*

$$\tilde{g} = \underbrace{\boxed{sg^B A}}_{\text{at most } r} + \underbrace{\boxed{sBg^A}}_{\text{at most } r}, \tag{17}$$

where $s > 0$ is the scaling factor, has matrix rank at most $2r$.

- Proof) Hence, using $r \ll \min\{m, n\}$, we can conclude that

$$\begin{aligned}\operatorname{rank}(\tilde{g}) = \operatorname{rank}(sg^B A + sBg^A) &\le \operatorname{rank}(g^B A) + \operatorname{rank}(Bg^A)\\ &\le \min\{\operatorname{rank}(g^B), \operatorname{rank}(A)\} + \min\{\operatorname{rank}(B), \operatorname{rank}(g^A)\}\\ &\le r + r = 2r. \quad \blacksquare\end{aligned}$$

Contents

Goal: To minimize $||\tilde{g} - g||_F^2$

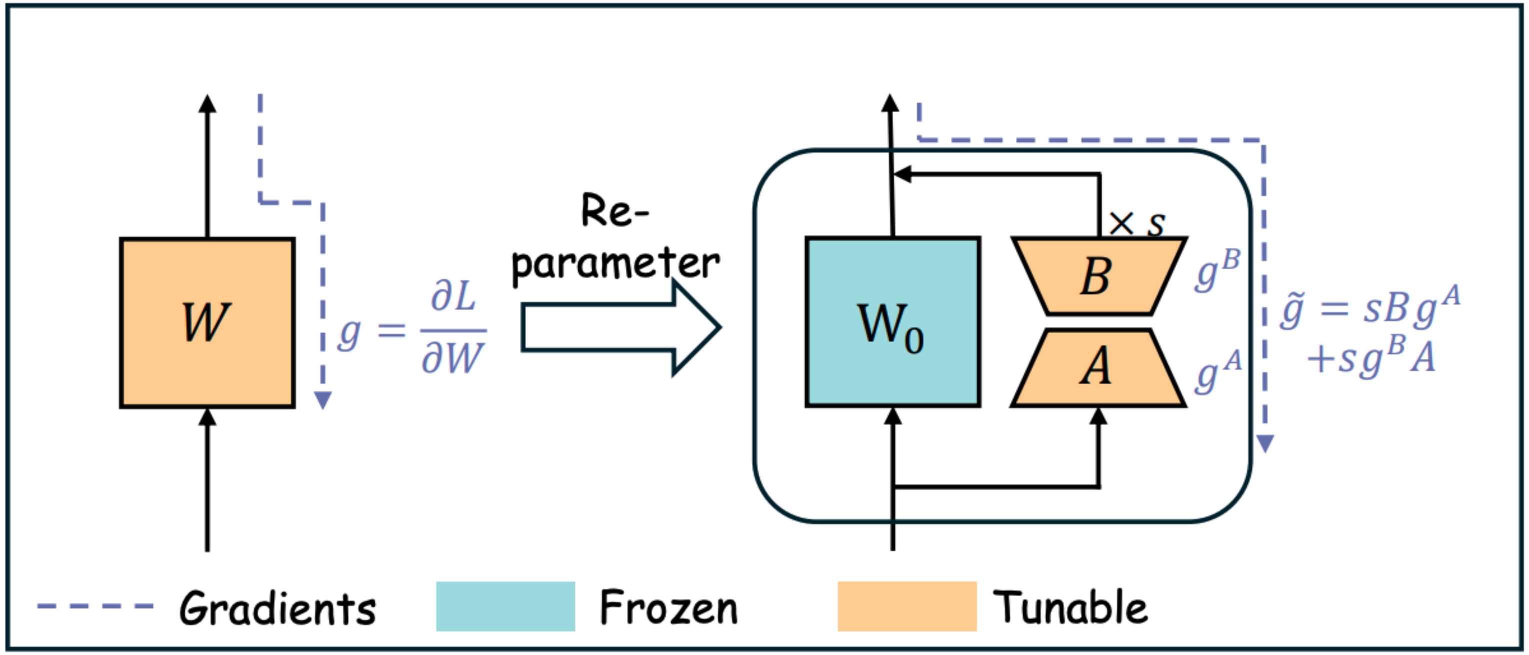
- What LoRA is supposed to do:
  - Update W directly with the full gradient g, as the same with full fine-tuning.

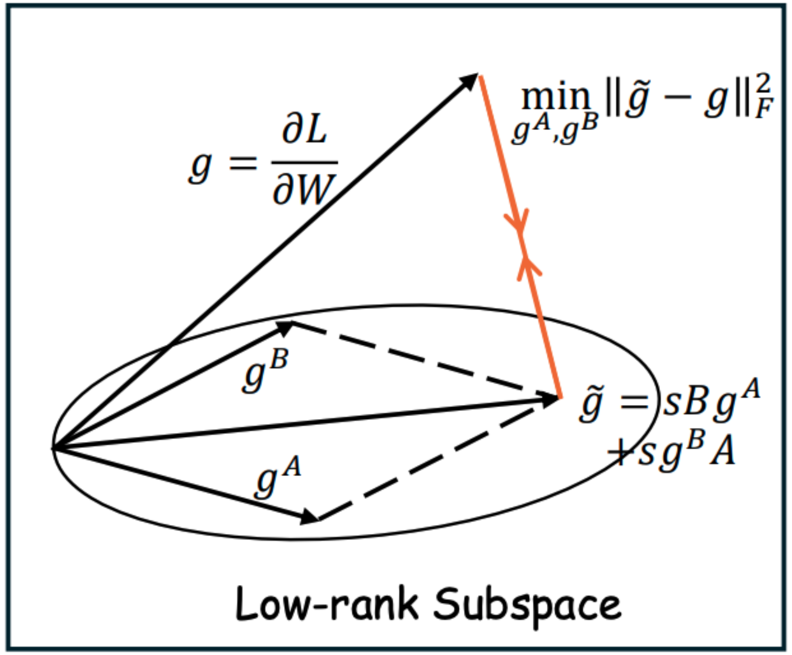
- What LoRA actually does:
  - Update B and A with the equivalent gradient $\tilde{g} \coloneqq s \cdot g^B A + s \cdot B g^A$, which may lose some information contained in g. (See Lemma: $\mathrm{rank}(\tilde{g}) \leq 2r$.)

- Goal of LoRA-Pro: Treat g^B and g^A as design variables, so that

$$\min_{\{g^A, g^B\}} ||\tilde{g} - g||_F^2 \quad \text{s.t.} \quad dL \leq 0$$

c.f.) Calculating g^B and g^A ≡ Projection





Q) How to solve the optimization problem?

Theorem 2.1. *Assume matrices $B \in \mathbb{R}^{m \times r}$, $A \in \mathbb{R}^{r \times n}$ are both full rank. For the objective* $\min_{g^A, g^B} \|\tilde{g} - g\|_F^2$, *the optimal solutions are given by:*

$$g^A = \frac{1}{s}(B^T B)^{-1} B^T g + XA = \frac{1}{s^2}(B^T B)^{-1} g^A_{lora} + XA, \tag{8}$$

$$g^B = \frac{1}{s}[I - B(B^T B)^{-1} B^T] g A^T (AA^T)^{-1} - BX \tag{9}$$

$$= \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] g^B_{lora} (AA^T)^{-1} - BX. \tag{10}$$

Here, $X \in \mathbb{R}^{r \times r}$ represents an arbitrary matrix.

Proof. See Appendix B.2. □

- Takeaways: There exists an optimal closed-form solution!

Proof of Theorem 2.1.

- Define $L \coloneqq \|\tilde{g} - g\|_F^2 = \|sg^B A + sBg^A - g\|_F^2$. Then, it suffices to check the points where $\frac{\partial L}{\partial g^A} = 0$ and $\frac{\partial L}{\partial g^B} = 0$.
  - $\frac{\partial L}{\partial g^A} = 2sB^T(sBg^A + sg^B A - g)$
  - $\frac{\partial L}{\partial g^B} = 2(sBg^A + sg^B A - g)sA^T$

Proof of Theorem 2.1.

- $\frac{\partial L}{\partial g^A} = 2sB^T(sBg^A + sg^B A - g) = 2s(sB^T B g^A + sB^T g^B A - B^T g) = 0$

- $sB^T B g^A \ = B^T g - sB^T g^B A$

- $B^T B g^A \ = \frac{1}{s} B^T g - B^T g^B A$

- $\therefore\ g^A = \frac{1}{s}(B^T B)^{-1} B^T g - (B^T B)^{-1} B^T g^B A$
  - Since B is full-rank, $B^T B$ is invertible.

Proof of Theorem 2.1.

- $\frac{\partial L}{\partial g^B} = 2(sBg^A + sg^B A - g)sA^T = 2s(sBg^A A^T + sg^B AA^T - gA^T) = 0$
- $sg^B AA^T = gA^T - sBg^A A^T$
- $g^B AA^T = \frac{1}{s} gA^T - Bg^A A^T$
- $\therefore\ g^B = \frac{1}{s} gA^T (AA^T)^{-1} - Bg^A A^T (AA^T)^{-1}$
  - Since A is full-rank, AA^T is invertible.

Proof of Theorem 2.1.

- Define $L \coloneqq \|\tilde{g} - g\|_F^2 = \|sg^B A + sBg^A - g\|_F^2$. Then, it suffices to check the points where $\frac{\partial L}{\partial g^A} = 0$ and $\frac{\partial L}{\partial g^B} = 0$.
  - $\frac{\partial L}{\partial g^A} = 2sB^T(sBg^A + sg^B A - g) \quad \Rightarrow \quad g^A = \frac{1}{s}(B^T B)^{-1} B^T g - (B^T B)^{-1} B^T g^B A$
  - $\frac{\partial L}{\partial g^B} = 2(sBg^A + sg^B A - g)sA^T \quad \Rightarrow \quad g^B = \frac{1}{s} g A^T (AA^T)^{-1} - Bg^A A^T (AA^T)^{-1}$

Proof of Theorem 2.1.

- Define $L \coloneqq \|\tilde{g} - g\|_F^2 = \|sg^B A + sBg^A - g\|_F^2$. Then, it suffices to check the points where $\frac{\partial L}{\partial g^A} = 0$ and $\frac{\partial L}{\partial g^B} = 0$.
  - $\frac{\partial L}{\partial g^A} = 2sB^T(sBg^A + sg^B A - g) \quad \Rightarrow \quad g^A = \frac{1}{s}(B^T B)^{-1} B^T g - (B^T B)^{-1} B^T g^B A$
  - $\frac{\partial L}{\partial g^B} = 2(sBg^A + sg^B A - g)sA^T \quad \Rightarrow \quad g^B = \frac{1}{s} g A^T (AA^T)^{-1} - Bg^A A^T (AA^T)^{-1}$

- Combining these two, we obtain

$$
\begin{aligned}
g^A &= \frac{1}{s}(B^T B)^{-1} B^T g - (B^T B)^{-1} B^T \left[\frac{1}{s} g A^T (AA^T)^{-1} - Bg^A A^T (AA^T)^{-1} \right] A \\
&= \frac{1}{s}(B^T B)^{-1} B^T g - \frac{1}{s}(B^T B)^{-1} B^T g A^T (AA^T)^{-1} A + g^A A^T (AA^T)^{-1} A \,.
\end{aligned}
$$

Proof of Theorem 2.1.

$$g^A = \frac{1}{s}(B^T B)^{-1} B^T g - \frac{1}{s}(B^T B)^{-1} B^T g A^T (A A^T)^{-1} A + g^A A^T (A A^T)^{-1} A$$

$$g^A - g^A A^T (A A^T)^{-1} A = \frac{1}{s}(B^T B)^{-1} B^T g - \frac{1}{s}(B^T B)^{-1} B^T g A^T (A A^T)^{-1} A$$

$$g^A [I - A^T (A A^T)^{-1} A] = \frac{1}{s}(B^T B)^{-1} B^T g [I - A^T (A A^T)^{-1} A]$$

Proof of Theorem 2.1.

$$g^A = \frac{1}{s}(B^T B)^{-1} B^T g - \frac{1}{s}(B^T B)^{-1} B^T g A^T (AA^T)^{-1} A + g^A A^T (AA^T)^{-1} A$$

$$g^A - g^A A^T (AA^T)^{-1} A = \frac{1}{s}(B^T B)^{-1} B^T g - \frac{1}{s}(B^T B)^{-1} B^T g A^T (AA^T)^{-1} A$$

$$g^A [I - A^T (AA^T)^{-1} A] = \frac{1}{s}(B^T B)^{-1} B^T g [I - A^T (AA^T)^{-1} A]$$

Proof of Theorem 2.1.

$$g^A = \frac{1}{s}(B^T B)^{-1} B^T g - \frac{1}{s}(B^T B)^{-1} B^T g A^T (AA^T)^{-1} A + g^A A^T (AA^T)^{-1} A$$

$$g^A - g^A A^T (AA^T)^{-1} A = \frac{1}{s}(B^T B)^{-1} B^T g - \frac{1}{s}(B^T B)^{-1} B^T g A^T (AA^T)^{-1} A$$

$$g^A [I - A^T (AA^T)^{-1} A] = \frac{1}{s}(B^T B)^{-1} B^T g [I - A^T (AA^T)^{-1} A]$$

- Since $P_A \coloneqq I - A^T (AA^T)^{-1} A$ is a projection matrix with rank $(n - r)$ and $AP_A = O$, the general solution for this equation is

$$g^A = \frac{1}{s}(B^T B)^{-1} B^T g + XA, \qquad X \in \mathbb{R}^{r \times r} .$$

Proof of Theorem 2.1.

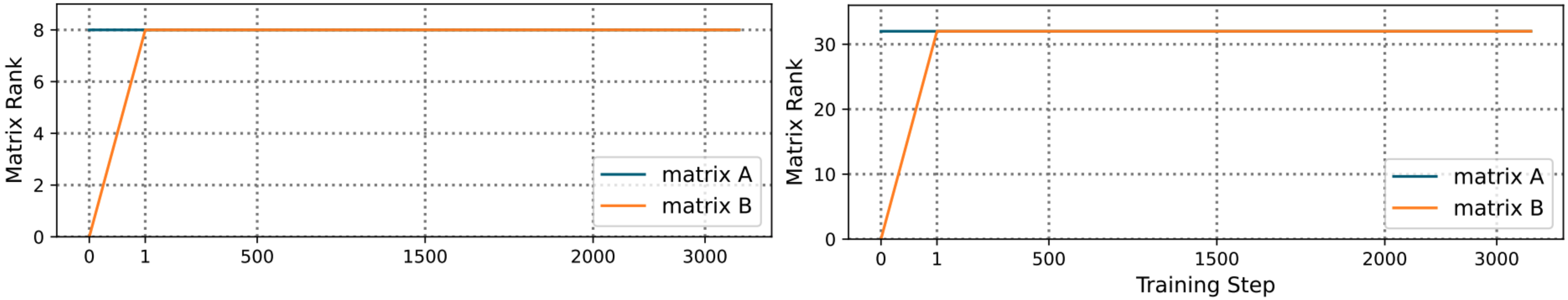
$$g^A = \frac{1}{s}(B^TB)^{-1}B^Tg + XA, \qquad X \in \mathbb{R}^{r\times r}$$

$$\begin{aligned}
g^B &= \frac{1}{s}gA^T(AA^T)^{-1} - Bg^AA^T(AA^T)^{-1} \\
&= \frac{1}{s}gA^T(AA^T)^{-1} - B\left[\frac{1}{s}(B^TB)^{-1}B^Tg + XA\right]A^T(AA^T)^{-1} \\
&= \frac{1}{s}[I - B(B^TB)^{-1}B^T]gA^T(AA^T)^{-1} - BXAA^T(AA^T)^{-1}
\end{aligned}$$

- Finally, substituting $g^B_{LoRA} \coloneqq \frac{\partial L}{\partial B} = sgA^T$ and $g^A_{LoRA} \coloneqq \frac{\partial L}{\partial A} = sB^Tg$,
  - $g^A = \frac{1}{s^2}(B^TB)^{-1}\, g^A_{LoRA} + XA$
  - $g^B = \frac{1}{s^2}[I - B(B^TB)^{-1}B^T]g^B_{LoRA}(AA^T)^{-1} - BX$

c.f.) Justification for the Full-Rank Assumption

- Theorem 2.1. assumes that both $B \in \mathbb{R}^{m \times r}$ and $A \in \mathbb{R}^{r \times n}$ are full-rank.
- Moreover, the Lemma implies that if g is not full-rank, the information loss might not that be significant.



- Note: B is initialized as O.

Q) How to solve the optimization problem?

Theorem 2.2. *When updating matrices A and B using the closed-form solution from Theorem 2.1, we proceed as follows:*

$$A \leftarrow A - \gamma g^A \tag{11}$$

$$B \leftarrow B - \gamma g^B, \tag{12}$$

where $\gamma \geq 0$ denotes the learning rate. Our method ensures a decrease in the loss, akin to the standard gradient descent algorithm, expressed by:

$$\mathrm{d}L = -\gamma \{ \langle g^A_{lora}, \frac{1}{s^2}(B^T B)^{-1} g^A_{lora} \rangle_F + \langle g^B_{lora}, \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] g^B_{lora} (AA^T)^{-1} \rangle_F \} \leq 0. \tag{13}$$

Proof. See Appendix B.3. □

- Takeways: The solution from LoRA-Pro guarantees $dL \leq 0$! Great!

Q) How to solve the optimization problem?

Theorem 2.2. *When updating matrices A and B using the closed-form solution from Theorem 2.1, we proceed as follows:*

$$A \leftarrow A - \gamma g^A \tag{11}$$

$$B \leftarrow B - \gamma g^B, \tag{12}$$

where $\gamma \geq 0$ denotes the learning rate. Our method ensures a decrease in the loss, akin to the standard gradient descent algorithm, expressed by:

$$\boxed{\mathrm{d}L = -\gamma\{\langle g^A_{lora}, \frac{1}{s^2}(B^TB)^{-1}g^A_{lora}\rangle_F + \langle g^B_{lora}, \frac{1}{s^2}[I - B(B^TB)^{-1}B^T]g^B_{lora}(AA^T)^{-1}\rangle_F\}} \leq 0. \tag{13}$$

Proof. See Appendix B.3. □

- First, we will show that dL is in the following form:

Proof of Theorem 2.2.

- Note that $dA = -\gamma g^A$, $dB = -\gamma g^B$, $\frac{\partial L}{\partial A} = g_{LoRA}^A$ and $\frac{\partial L}{\partial B} = g_{LoRA}^B$.
- Moreover, from Theorem 2.1., we've found the followings:
  - $g^A = \frac{1}{s^2}(B^T B)^{-1} g_{LoRA}^A + XA$
  - $g^B = \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] g_{LoRA}^B (AA^T)^{-1} - BX$
- By plugging g^A and g^B into the following equation

$$dL = \left\langle \frac{\partial L}{\partial A}, dA \right\rangle_F + \left\langle \frac{\partial L}{\partial B}, dB \right\rangle_F = -\gamma \langle g_{LoRA}^A, g^A \rangle_F - \gamma \langle g_{LoRA}^B, g^B \rangle_F,$$

Proof of Theorem 2.2.

$$
\begin{aligned}
dL &= -\gamma\left\langle g^A_{LoRA}, g^A\right\rangle_F - \gamma\left\langle g^B_{LoRA}, g^B\right\rangle_F \\
&= -\gamma\left\langle g^A_{LoRA}, \frac{1}{s^2}(B^T B)^{-1}\, g^A_{LoRA}\right\rangle_F \\
&\quad -\gamma\left\langle g^A_{LoRA}, XA\right\rangle_F \\
&\quad -\gamma\left\langle g^B_{LoRA}, \frac{1}{s^2}[I - B(B^T B)^{-1}B^T] g^B_{LoRA}(AA^T)^{-1}\right\rangle_F \\
&\quad +\gamma\left\langle g^B_{LoRA}, BX\right\rangle_F
\end{aligned}
$$

Proof of Theorem 2.2.

$$dL = -\gamma \left\langle g_{LoRA}^{A}, g^{A} \right\rangle_F - \gamma \left\langle g_{LoRA}^{B}, g^{B} \right\rangle_F$$

$$= -\gamma \left\langle g_{LoRA}^{A}, \frac{1}{s^2} (B^T B)^{-1} \, g_{LoRA}^{A} \right\rangle_F$$

$$-\gamma \left\langle g_{LoRA}^{A}, XA \right\rangle_F$$

$$-\gamma \left\langle g_{LoRA}^{B}, \frac{1}{s^2} [I - B(B^T B)^{-1} B^T] g_{LoRA}^{B} (AA^T)^{-1} \right\rangle_F$$

$$+\gamma \left\langle g_{LoRA}^{B}, BX \right\rangle_F$$

Proof of Theorem 2.2.

- Since

$$\begin{aligned}
\gamma\langle g_{LoRA}^{B}, BX\rangle_F - \gamma\langle g_{LoRA}^{A}, XA\rangle_F &= \gamma\left(\langle g_{LoRA}^{B}, BX\rangle_F - \langle g_{LoRA}^{A}, XA\rangle_F\right) \\
&= \gamma\left(\langle B^T g_{LoRA}^{B}, X\rangle_F - \langle g_{LoRA}^{A} A^T, X\rangle_F\right) \\
&= \gamma\langle B^T g_{LoRA}^{B} - g_{LoRA}^{A} A^T, X\rangle_F \\
&= \gamma\langle B^T sgA^T - sB^T gA^T, X\rangle_F \\
&= \gamma s\langle B^T gA^T - B^T gA^T, X\rangle_F = 0\,,
\end{aligned}$$

we can conclude that

$$dL = -\gamma\left[\left\langle g_{LoRA}^{A}, \frac{1}{s^2}(B^T B)^{-1}\, g_{LoRA}^{A}\right\rangle_F + \left\langle g_{LoRA}^{B}, \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] g_{LoRA}^{B} (AA^T)^{-1}\right\rangle_F\right].$$

Proof of Theorem 2.2.

$$dL = -\gamma \left[\left\langle g_{LoRA}^{A}, \frac{1}{s^2} (B^T B)^{-1} \, g_{LoRA}^{A} \right\rangle_F + \left\langle g_{LoRA}^{B}, \frac{1}{s^2} [I - B(B^T B)^{-1} B^T] g_{LoRA}^{B} (AA^T)^{-1} \right\rangle_F \right]$$

- Next, we will show that $dL \leq 0$.
  - Part 1: Both $(B^T B)^{-1}$ and $(AA^T)^{-1}$ are positive definite.
  - Part 2: $[I - B(B^T B)^{-1} B^T]$ is positive semi-definite.
  - Part 3: $\left\langle g_{LoRA}^{A}, \frac{1}{s^2} (B^T B)^{-1} \, g_{LoRA}^{A} \right\rangle_F \geq 0$
  - Part 4: $\left\langle g_{LoRA}^{B}, \frac{1}{s^2} [I - B(B^T B)^{-1} B^T] g_{LoRA}^{B} (AA^T)^{-1} \right\rangle_F \geq 0$

Proof of Theorem 2.2.

- Part 1: Both $(B^T B)^{-1}$ and $(AA^T)^{-1}$ are positive definite (PD).

- Consider any non-zero vector $x \in \mathbb{R}^r$. Then, since B is full-rank,
$$\langle x, B^T B x \rangle = \langle Bx, Bx \rangle = \|Bx\|^2 > 0\,.$$

- Since $B^T B$ is PD, so is $(B^T B)^{-1}$.

Proof of Theorem 2.2.

- Part 1: Both $(B^T B)^{-1}$ and $(AA^T)^{-1}$ are positive definite (PD).

- Consider any non-zero vector $x \in \mathbb{R}^r$. Then, since B is full-rank,
$$\langle x, B^T B x \rangle = \langle Bx, Bx \rangle = \|Bx\|^2 > 0\,.$$

- Since $B^T B$ is PD, so is $(B^T B)^{-1}$.

- In a similar way, we can say that $(AA^T)^{-1}$ is PD, because
$$\langle x, AA^T x \rangle = \langle A^T x, A^T x \rangle = \|A^T x\|^2 > 0\,.$$

Proof of Theorem 2.2.

- Part 2: $[I - B(B^T B)^{-1} B^T]$ is positive semi-definite (PSD).

- Claim: $P \coloneqq B(B^T B)^{-1} B^T$ is a projection matrix.
  - [Symmetricity] $(B(B^T B)^{-1} B^T)^T = B\left((B^T B)^{-1}\right)^T B^T = B(B^T B)^{-1} B^T$
  - [Idempotence] $(B(B^T B)^{-1} B^T)^2 = (B(B^T B)^{-1} B^T)^T B(B^T B)^{-1} B^T$

$$= B(B^T B)^{-1} B^T B (B^T B)^{-1} B^T$$

$$= B(B^T B)^{-1} B^T$$

Proof of Theorem 2.2.

- Part 2: $[I - B(B^TB)^{-1}B^T]$ is positive semi-definite (PSD).

- Consider any non-zero vector $x \in \mathbb{R}^m$. Then, since $P^T = P$ and $P^2 = P$,

$$\begin{aligned} \langle x, (I - B(B^TB)^{-1}B^T)x\rangle &= \langle x, (I-P)x\rangle = \langle Px + (I-P)x, (I-P)x\rangle \\ &= \langle Px, (I-P)x\rangle + \langle (I-P)x, (I-P)x\rangle \\ &= x^TP^T(I-P)x + \|(I-P)x\|_F^2 \\ &= x^TP(I-P)x + \|(I-P)x\|_F^2 \\ &= x^T(P-P^2)x + \|(I-P)x\|_F^2 \\ &= \|(I-P)x\|_F^2 \geq 0\ . \end{aligned}$$

Proof of Theorem 2.2.

- Part 3: $\left\langle g_{LoRA}^{A}, \frac{1}{s^2}(B^T B)^{-1}\, g_{LoRA}^{A} \right\rangle_F \geq 0$

- Since $(B^T B)^{-1}$ is PD, there exists an invertible matrix U which satisfies $(B^T B)^{-1} = UU^T$ by the Cholesky Decomposition. Therefore,

$$\begin{aligned}\left\langle g_{LoRA}^{A}, \frac{1}{s^2}(B^T B)^{-1}\, g_{LoRA}^{A} \right\rangle_F &= \frac{1}{s^2}\left\langle g_{LoRA}^{A}, UU^T g_{LoRA}^{A} \right\rangle_F \\ &= \frac{1}{s^2}\left\langle U^T g_{LoRA}^{A}, U^T g_{LoRA}^{A} \right\rangle_F \\ &= \frac{1}{s^2}\left\| U^T g_{LoRA}^{A} \right\|_F^2 \geq 0.\end{aligned}$$

Proof of Theorem 2.2.

- Part 4: $\left\langle g_{LoRA}^{B}, \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] g_{LoRA}^{B} (AA^T)^{-1} \right\rangle_F \geq 0$
- Since $(AA^T)^{-1}$ is PD and $[I - B(B^T B)^{-1} B^T]$ is PSD, there exist invertible matrices U, V, which satisfy $(AA^T)^{-1} = U^T U$ and $[I - B(B^T B)^{-1} B^T] = VV^T$, respectively, by the Cholesky Decomposition. Therefore,

$$\begin{aligned}
\left\langle g_{LoRA}^{B}, \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] g_{LoRA}^{B} (AA^T)^{-1} \right\rangle_F &= \frac{1}{s^2}\left\langle g_{LoRA}^{A}, VV^T g_{LoRA}^{A} UU^T \right\rangle_F \\
&= \frac{1}{s^2}\left\langle V^T g_{LoRA}^{A} U, V^T g_{LoRA}^{A} U \right\rangle_F \\
&= \frac{1}{s^2}\left\| V^T g_{LoRA}^{A} U \right\|_F^2 \geq 0.
\end{aligned}$$

Proof of Theorem 2.2.

Theorem 2.2. *When updating matrices A and B using the closed-form solution from Theorem 2.1, we proceed as follows:*

$$A \leftarrow A - \gamma g^A \tag{11}$$

$$B \leftarrow B - \gamma g^B, \tag{12}$$

where $\gamma \geq 0$ denotes the learning rate. Our method ensures a decrease in the loss, akin to the standard gradient descent algorithm, expressed by:

$$\mathrm{d}L = -\gamma\{\langle g_{lora}^A, \frac{1}{s^2}(B^T B)^{-1} g_{lora}^A \rangle_F + \langle g_{lora}^B, \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] g_{lora}^B (AA^T)^{-1} \rangle_F\} \leq 0. \tag{13}$$

Proof. See Appendix B.3. □

- Therefore, we can conclude that $dL \leq 0$ when using Lora-Pro.

Q) Which one should we use for $X \in \mathbb{R}^{r\times r}$?

Theorem 2.1. *Assume matrices $B \in \mathbb{R}^{m\times r}$, $A \in \mathbb{R}^{r\times n}$ are both full rank. For the objective $\min_{g^A, g^B} \|\tilde{g} - g\|_F^2$, the optimal solutions are given by:*

$$g^A = \frac{1}{s}(B^T B)^{-1} B^T g + XA = \frac{1}{s^2}(B^T B)^{-1} g^A_{lora} + XA, \tag{8}$$

$$g^B = \frac{1}{s}[I - B(B^T B)^{-1} B^T] g A^T (AA^T)^{-1} - BX \tag{9}$$

$$= \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] g^B_{lora} (AA^T)^{-1} - BX. \tag{10}$$

Here, $\boxed{X \in \mathbb{R}^{r\times r}}$ *represents an arbitrary matrix.*

Proof. See Appendix B.2. □

A) The optimal X can be found via Sylvester Equation.

Theorem 2.3. *Consider the optimization problem,*

$$\min_X \|g^A - g^A_{lora}\|_F^2 + \|g^B - g^B_{lora}\|_F^2, \tag{14}$$

where g^A and g^B are the optimal solutions as stated in Theorem 2.1. The optimal X can be determined by solving the Sylvester equation:

$$\boxed{B^T BX + XAA^T = -\frac{1}{s^2}(B^T B)^{-1} g^A_{lora} A^T}, \tag{15}$$

which has a unique solution X provided that $B^T B$ and $-AA^T$ do not have any shared eigenvalues.

Proof. See Appendix B.4. □

- We will skip the details of how to solve this type of Sylvester equation.

Proof of Theorem 2.3.

- Let's denote $L = \left\|g^A - g^A_{LoRA}\right\|_F^2 + \left\|g^B - g^B_{LoRA}\right\|_F^2$.

- Then, we want to find X which satisfies $\frac{\partial L}{\partial X} = 0$.

- Note that we've found from Theorem 2.1. that
  - $g^A = \frac{1}{s^2}(B^T B)^{-1} g^A_{LoRA} + XA$
  - $g^B = \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] g^B_{LoRA} (AA^T)^{-1} - BX$.

- Moreover, we have $g^A_{LoRA} = sB^T g$ and $g^B_{LoRA} = sgA^T$.

Proof of Theorem 2.3.

$$L = \left\| \frac{1}{s^2}(B^T B)^{-1} g_{LoRA}^{A} - sB^T g + XA \right\|_F^2$$

$$+ \left\| \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] sgA^T (AA^T)^{-1} - sgA^T - BX \right\|_F^2$$

$$= \| C_A + XA \|_F^2 + \| C_B - BX \|_F^2 .$$

$$\frac{\partial L}{\partial X} = 2(C_A + XA)A^T - 2B^T(C_B - BX) = 2(C_A A^T + XAA^T - B^T C_B + B^T BX)$$

$$\Rightarrow B^T B X + X AA^T = B^T C_B - C_A A^T. \quad \blacksquare$$

Summary of LoRA-Pro

- Compute standard LoRA gradients g_{LoRA}^{B} and g_{LoRA}^{A}.
- Using Theorem 2.1., get the optimal g^{B} and g^{A} analytically.
  - Which minimize the gap between $\tilde{g}$ and g.
- Using Theorem 2.3, find the optimal X via Sylvester Equation.
  - Which minimizes the gap between g^{A}, g^{B} and g_{LoRA}^{A}, g_{LoRA}^{B}, respectively.
- Back-propagate with g^{B} and g^{A} to update B and A, respectively.

Contents

Summary

- Pros
  - High performance gain
    - With theoretical backgrounds & low computational cost
  - Easy to implement
    - Input: g_{LoRA}^{B} and g_{LoRA}^{A}
    - Output: g^{B} and g^{A}

```python
# Step 2:- run optimizer and upscaling simultaneously
for i, group in enumerate(self.bit16_groups):
    self.timers(OPTIMIZER_GRADIENTS_TIMER).start()
    self.global_step += 1
    partition_id = dist.get_rank(group=self.real_dp_process_group[i])

    self.lorapro_full_adjustment(partition_id)

    if self.cpu_offload:
        single_grad_partition = self.single_partition_of_fp32_groups[i].grad
        self.unscale_and_clip_grads([single_grad_partition], scaled_global_grad_norm)

        self.timers(OPTIMIZER_GRADIENTS_TIMER).stop()
        self.timers(OPTIMIZER_STEP_TIMER).start()
        self._optimizer_step(i)
```

Summary

- Cons
  - LoRA-Pro underperforms on some tasks full fine-tuning struggles on.
    - It aims to mimics full fine-tuning.

	MT-Bench	GSM8K	HumanEval
Full FT	5.30±0.11	**59.36±0.85**	**35.31±2.13**
LoRA	5.61±0.10	42.08±0.04	14.76±0.17
PiSSA	5.30±0.02	44.54±0.27	16.02±0.78
rsLoRA	5.25±0.03	45.62±0.10	16.01±0.79
LoRA+	5.71±0.08	52.11±0.62	18.17±0.52
DoRA	**5.97±0.02**	53.07±0.75	19.75±0.41
AdaLoRA	5.57±0.05	50.72±1.39	17.80±0.44
LoRA-GA	<u>5.95±0.16</u>	53.60±0.30	19.81±1.46
LoRA-GA (rank=32)	5.79±0.09	55.12±0.30	20.18±0.19
LoRA-GA (rank=128)	6.13±0.07	55.07±0.18	23.05±0.37
LoRA-Pro	5.86±0.06	<u>54.23±0.79</u>	<u>22.76±0.35</u>
LoRA-Pro (rank=32)	6.01±0.05	55.14±1.73	28.05±0.00
LoRA-Pro (rank=128)	5.68±0.14	56.48±0.23	34.55±2.46

Q & A

Thank you.

Pseudo-code of LoRA-Pro with SGD

Algorithm 1 LoRA-Pro with SGD optimizer

Require: Given initial learning rate γ, scaling factor s.

1: Initialize time step $t \leftarrow 0$, low-rank matrices $A_0 \in \mathbb{R}^{r \times n}$ and $B_0 \in \mathbb{R}^{m \times r}$
2: **repeat**
3: $\quad t \leftarrow t + 1$
4: $\quad g^A_{lora}, g^B_{lora} \leftarrow \text{SelectBatch}(A_{t-1}, B_{t-1})$ ▷ *Select batch and return the corresponding gradients*
5: $\quad A, B \leftarrow A_{t-1}, B_{t-1}$ ▷ *Obtain the low-rank matrices A and B*
6: $\quad X \leftarrow \text{SolveSylvester}(B^T B X + X A A^T = -\frac{1}{s^2}(B^T B)^{-1} g^A_{lora} A^T)$ ▷ *Compute X by solving the sylvester equation*
7: $\quad g^A = \frac{1}{s^2}(B^T B)^{-1} g^A_{lora} + XA$ ▷ *Adjust the gradients of LoRA with Theorem 2.1*
8: $\quad g^B = \frac{1}{s^2}[I - B(B^T B)^{-1} B^T] g^B_{lora} (AA^T)^{-1} - BX$
9: $\quad A_t \leftarrow A_{t-1} - \gamma g^A$
10: $\quad B_t \leftarrow B_{t-1} - \gamma g^B$
11: **until** *stopping criterion is met*
12: **return** optimized parameters A_t and B_t

Experiments on Natural Language Understanding (T5-Base)

- Target Modules: Q, K, V, Out, FC1, FC2
- $r = 8 \,/\, \alpha = 16 \,/\, s = \frac{\alpha}{r}$

Method	MNLI	SST2	CoLA	QNLI	MRPC	Average
Full FT	**86.33±0.00**	**94.75±0.21**	80.70±0.24	93.19±0.22	84.56±0.73	87.91
LoRA	85.30±0.04	94.04±0.11	69.35±0.05	92.96±0.09	68.38±0.01	82.08
PiSSA	85.75±0.07	94.07±0.06	74.27±0.39	93.15±0.14	76.31±0.51	84.71
rsLoRA	85.73±0.10	94.19±0.23	72.32±1.12	93.12±0.09	52.86±2.27	79.64
LoRA+	85.81±0.09	93.85±0.24	77.53±0.20	93.14±0.03	74.43±1.39	84.95
LoRA-GA	85.70±0.09	94.11±0.18	80.57±0.20	93.18±0.06	85.29±0.24	87.77
DoRA	85.67±0.09	94.04±0.53	72.04±0.94	93.04±0.06	68.08±0.51	82.57
AdaLoRA	85.45±0.11	93.69±0.20	69.16±0.24	91.66±0.05	68.14±0.28	81.62
LoRA-Pro	86.03±0.19	94.19±0.13	**81.94±0.24**	**93.42±0.05**	**86.60±0.14**	**88.44**

Experiments on Language Generation (Llama-2-7B)

- $r = 8 \, / \, \alpha = 16 \, / \, s = \frac{\alpha}{\sqrt{r}}$

	MT-Bench	GSM8K	HumanEval
Full FT	5.30±0.11	**59.36±0.85**	**35.31±2.13**
LoRA	5.61±0.10	42.08±0.04	14.76±0.17
PiSSA	5.30±0.02	44.54±0.27	16.02±0.78
rsLoRA	5.25±0.03	45.62±0.10	16.01±0.79
LoRA+	5.71±0.08	52.11±0.62	18.17±0.52
DoRA	**5.97±0.02**	53.07±0.75	19.75±0.41
AdaLoRA	5.57±0.05	50.72±1.39	17.80±0.44
LoRA-GA	<u>5.95±0.16</u>	53.60±0.30	19.81±1.46
LoRA-GA (rank=32)	5.79±0.09	55.12±0.30	20.18±0.19
LoRA-GA (rank=128)	6.13±0.07	55.07±0.18	23.05±0.37
LoRA-Pro	5.86±0.06	<u>54.23±0.79</u>	<u>22.76±0.35</u>
LoRA-Pro (rank=32)	6.01±0.05	55.14±1.73	28.05±0.00
LoRA-Pro (rank=128)	5.68±0.14	56.48±0.23	34.55±2.46

Experiments on Image Classification (CLIP-ViT-B/16)

- $r = 8 \,/\, \alpha = 16$
- LoRA adaptors are attached to the visual backbone only.

Method	Cars	DTD	EuroSAT	GTSRB	RESISC45	SUN397	SVHN	Average
Zero-shot	63.75	44.39	42.22	35.22	56.46	62.56	15.53	45.73
Full FT	84.23±0.06	77.44±0.19	98.09±0.03	94.31±0.28	93.95±0.0	75.35±0.10	93.04±0.18	88.06
LoRA	72.81±0.13	73.92±0.38	96.93±0.07	92.40±0.10	90.03±0.14	70.12±0.18	88.02±0.07	83.46
rsLoRA	82.38±0.20	78.03±0.76	98.06±0.08	95.04±0.11	93.96±0.18	75.38±0.24	92.74±0.18	87.94
LoRA+	72.87±0.18	74.07±0.45	97.01±0.02	92.42±0.18	89.96±0.11	70.17±0.15	88.08±0.05	83.51
DoRA	73.72±0.06	73.72±0.33	96.95±0.01	92.38±0.17	90.03±0.08	70.20±0.19	88.23±0.05	83.48
LoRA-GA	85.18±0.41	77.50±0.12	98.05±0.27	95.28±0.10	94.43±0.19	75.44±0.06	93.68±0.35	88.51
LoRA-Pro	**85.87±0.08**	**78.64±0.25**	**98.46±0.03**	**95.66±0.05**	**94.75±0.21**	**76.42±0.14**	**94.63±0.20**	**89.20**

Ablation Study for the Choice of X

choice of X	MT-Bench	GSM8K	HumanEval
Zero	5.76±0.02	53.83±1.16	22.96±1.96
Sylvester (Thm. 2.3)	5.86±0.06	54.23±0.79	22.76±0.35
Symmetry (Eq. (16))	5.63±0.12	54.46±0.88	22.56±1.06

$$X = -\frac{1}{2s}B(B^T B)^{-1}B^T g A(A^T A)^{-1}A = -\frac{1}{2s^2}B(B^T B)^{-1}B^T g_{lora}^{B}(A^T A)^{-1}A. \tag{16}$$

Justification for Training Costs

Table 5: We compare LoRA, LoRA-Pro, and Full Fine-Tuning in terms of memory cost, training time, and performance on the MT-Bench, GSM8K, and HumanEval datasets. Memory cost is measured using a single A6000 GPU with a batch size of 1. Training time is recorded on the WizardLM dataset using 8 A100 GPUs with DeepSpeed ZeRO-2 stage optimization.

	Memory Cost	Training Time	MT-Bench	GSM8K	HumanEval
Full FT	$>$ 48 GB	2h 33min	5.30±0.11	59.36±0.85	35.31±2.13
LoRA	22.26 GB	1h 22min	5.61±0.10	42.08±0.04	14.76±0.17
LoRA-GA	22.60 GB	1h 25min	5.95±0.16	53.60±0.30	19.81±1.46
LoRA-Pro	23.05 GB	1h 23min	5.86±0.06	54.23±0.79	22.76±0.35