SD-LoRA: Scalable Decoupled Low-Rank Adaptation for Class Incremental Learning

Yichen Wu[1,2,*], **Hongming Piao**[1,*], **Long-Kai Huang**[4,†], **Renzhen Wang**[3], **Wanhua Li**[2],
Hanspeter Pfister[2], **Deyu Meng**[3,6], **Kede Ma**[1,†], **Ying Wei**[5,†]
[1]City University of Hong Kong, [2]Harvard University, [3]Xi'an Jiaotong University,
[4]Tencent AI Lab, [5]Zhejiang University, [6]Pengcheng Laboratory

Presenter: Hongming Piao

Motivation

Method	Rehearsal-free	Inference Efficiency	End-to-end Optimization
L2P (Wang et al., 2022b)	✓	✗	✗
DualPrompt (Wang et al., 2022a)	✓	✗	✗
CODA-Prompt (Smith et al., 2023)	✓	✗	✓
HiDe-Prompt (Wang et al., 2024a)	✗	✗	✓
InfLoRA (Liang & Li, 2024)	✗	✓	✓
SD-LoRA(Ours)	✓	✓	✓

Motivation

Method	Rehearsal-free	Inference Efficiency	End-to-end Optimization
L2P (Wang et al., 2022b)	✓	✗	✗
DualPrompt (Wang et al., 2022a)	✓	✗	✗
CODA-Prompt (Smith et al., 2023)	✓	✗	✓
HiDe-Prompt (Wang et al., 2024a)	✗	✗	✓
InfLoRA (Liang & Li, 2024)	✗	✓	✓
SD-LoRA(Ours)	✓	✓	✓

Sample-dependent inference with foundation models

- Complex designing of the matching mechanism.
- Extra-computation during inference.

Motivation

Method	Rehearsal-free	Inference Efficiency	End-to-end Optimization
L2P (Wang et al., 2022b)	✓	✗	✗
DualPrompt (Wang et al., 2022a)	✓	✗	✗
CODA-Prompt (Smith et al., 2023)	✓	✗	✓
HiDe-Prompt (Wang et al., 2024a)	✗	✗	✓
InfLoRA (Liang & Li, 2024)	✗	✓	✓
SD-LoRA(Ours)	✓	✓	✓

Sample-dependent inference with foundation methods

- Require complex designing of the matching mechanism.
- Extra-computation during inference.

Need to store huge learned tasks' features to avoid catastrophic forgetting.

Motivation

Method	Rehearsal-free	Inference Efficiency	End-to-end Optimization
L2P (Wang et al., 2022b)	✓	✗	✗
DualPrompt (Wang et al., 2022a)	✓	✗	✗
CODA-Prompt (Smith et al., 2023)	✓	✗	✓
HiDe-Prompt (Wang et al., 2024a)	✗	✗	✓
InfLoRA (Liang & Li, 2024)	✗	✓	✓
SD-LoRA(Ours)	✓	✓	✓

Sample-dependent inference with foundation methods

- Require complex designing of the matching mechanism.
- Extra-computation during inference.

Need to store huge learned tasks' features to avoid catastrophic forgetting.

Motivation

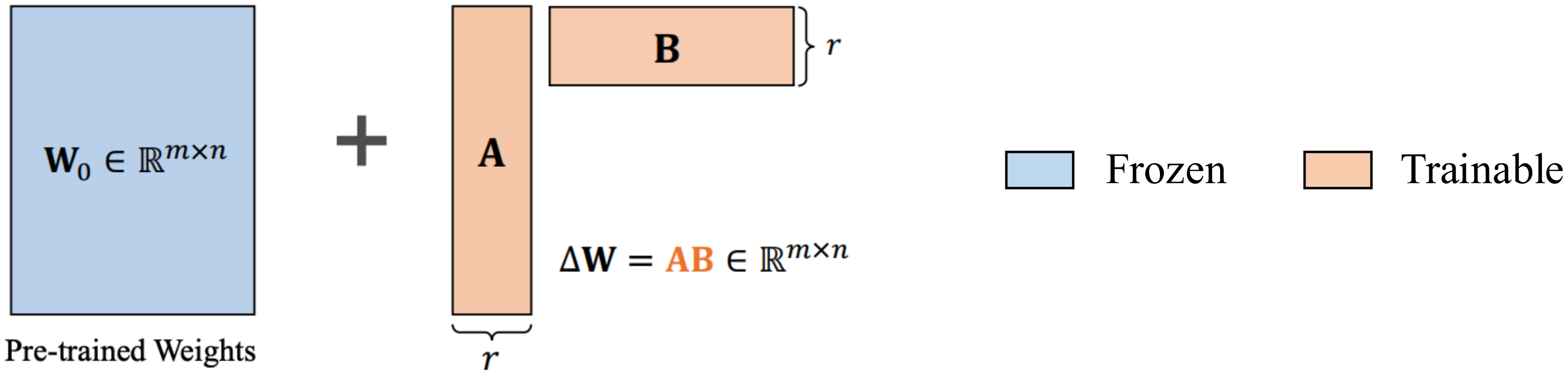
Improve the inference efficiency.

Avoid catastrophic forgetting without the huge storage of the learned tasks' features.

The Proposed SD-LoRA

1) Improve the inference efficiency

(Training)



(Testing)

$$h' = \mathbf{W}_0 x + \Delta\mathbf{W} x = \underbrace{(\mathbf{W}_0 + \mathbf{AB})}_{\mathbf{W}'} x$$

Avoid additional inference overhead
by incorporating the LoRAs into pre-trained weights

The Proposed SD-LoRA

2)Avoid catastrophic forgetting without the huge storage of the learned tasks' features

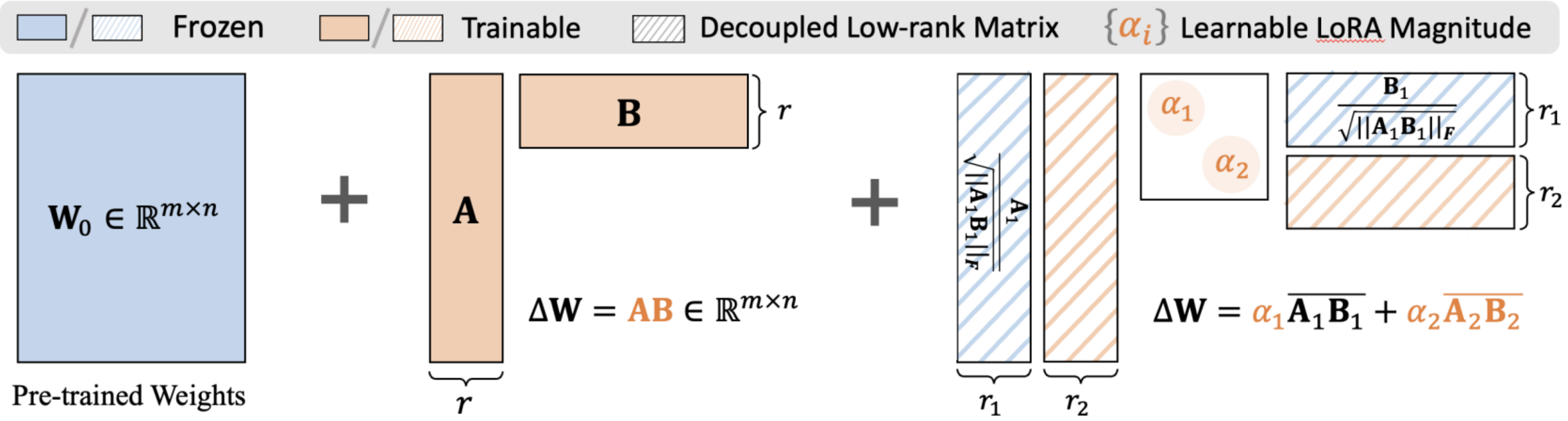
Decouple the magnitude and direction of the learned **AB**

$$\Delta\mathbf{W} = ||\mathbf{AB}||_F \cdot \overline{\mathbf{AB}} = ||\mathbf{AB}||_F \cdot \frac{\mathbf{AB}}{||\mathbf{AB}||_F}$$

The Proposed SD-LoRA

2)Avoid catastrophic forgetting without the huge storage of the learned tasks' features

(Training)



(a) Vanilla LoRA

(b) SD-LoRA

(Testing)

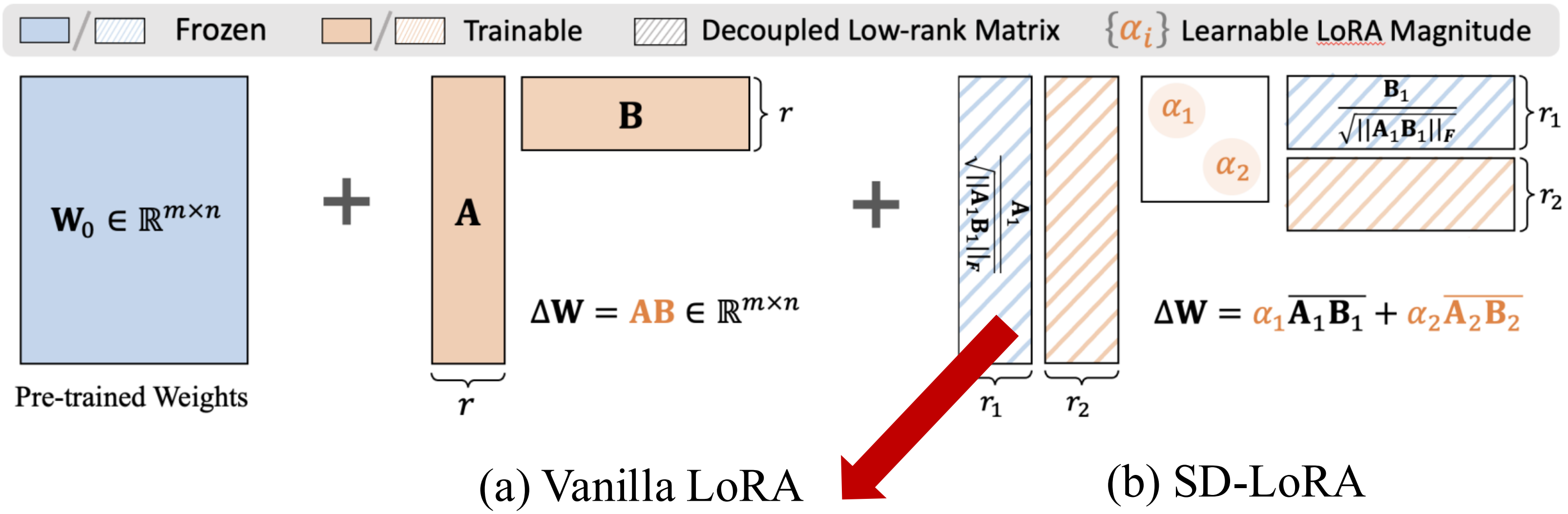
$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$

1) Improve the inference efficiency

The Proposed SD-LoRA

2)Avoid catastrophic forgetting without the huge storage of the learned tasks' features

(Training)



(a) Vanilla LoRA (b) SD-LoRA

Task specific knowledge (direction)

(Testing)

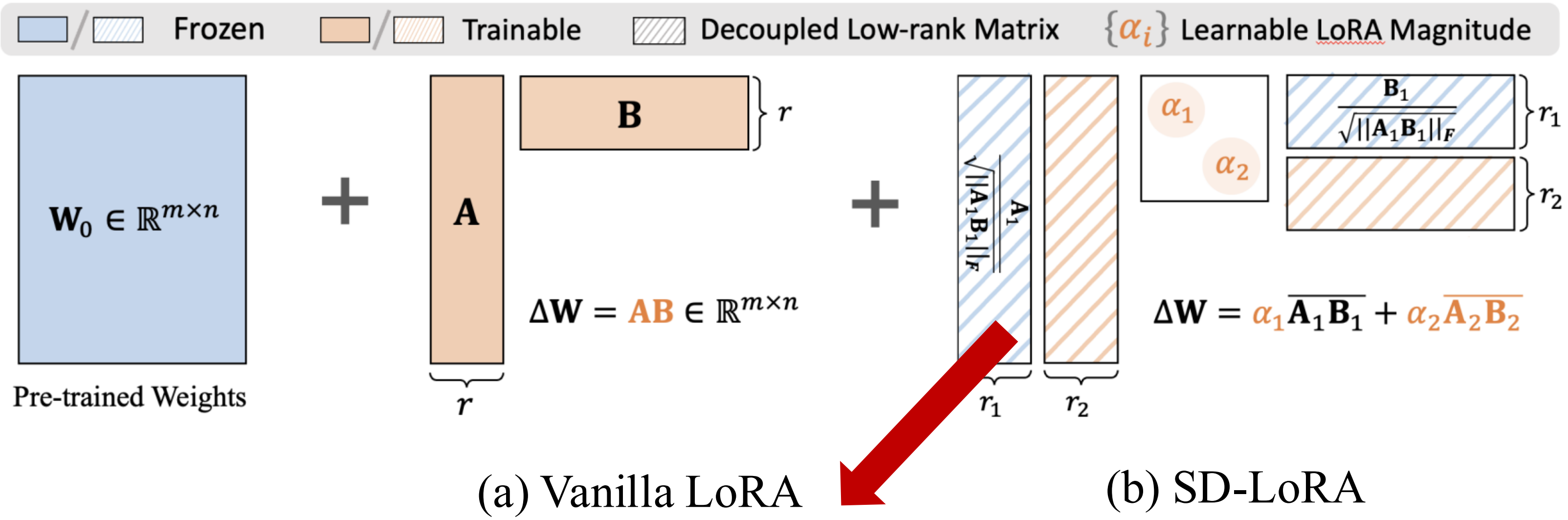
$$h' = (\mathbf{W}_0 + \alpha_1\overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2\overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t\overline{\mathbf{A}_t\mathbf{B}_t})x$$

1) Improve the inference efficiency

The Proposed SD-LoRA

2)Avoid catastrophic forgetting without the huge storage of the learned tasks' features

(Training)



(a) Vanilla LoRA (b) SD-LoRA

Task specific knowledge (direction)

(Testing)

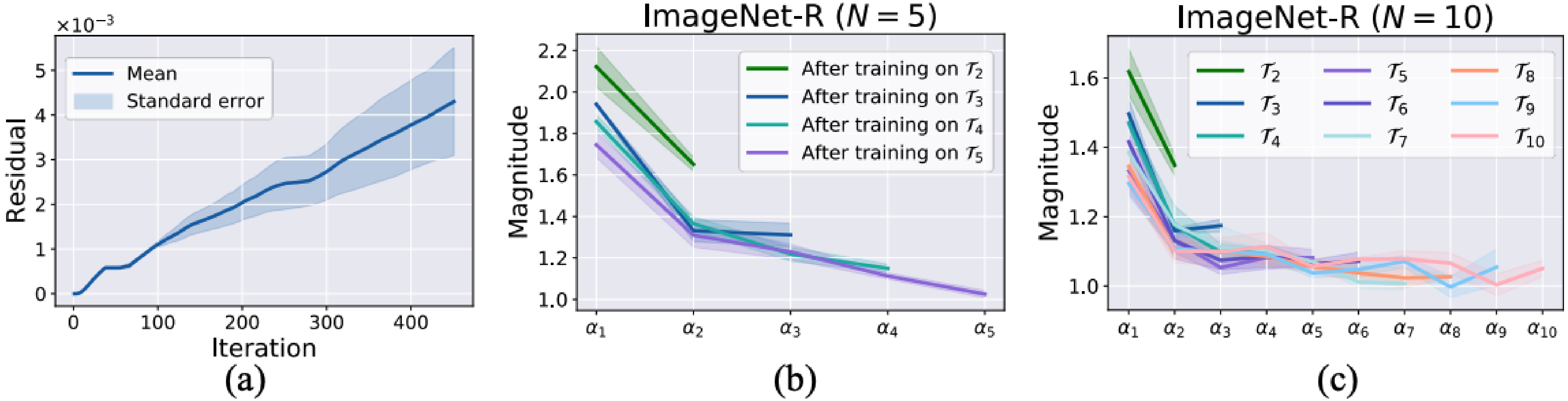
$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$

1) Improve the inference efficiency

Why SD-LoRA avoid catastrophic forgetting?

The directions of previous tasks are important

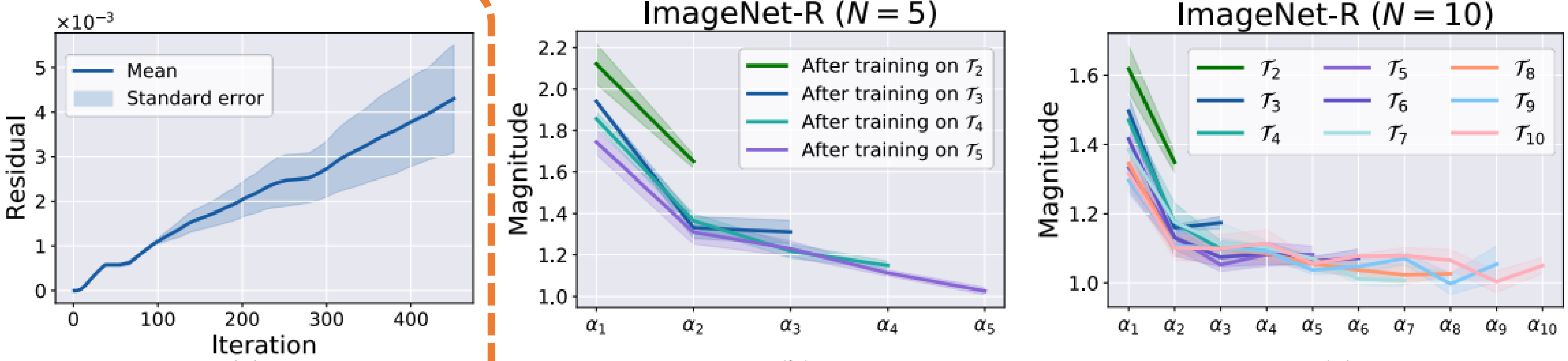
$$h' = (\mathbf{W}_0 + \alpha_1\overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2\overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t\overline{\mathbf{A}_t\mathbf{B}_t})x$$



The directions of previous tasks are important

$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$

$\overline{\mathbf{A}_t\mathbf{B}_t}$ ↕ $\{\overline{\mathbf{A}_k\mathbf{B}_k}\}_{k=1}^{t-1}$

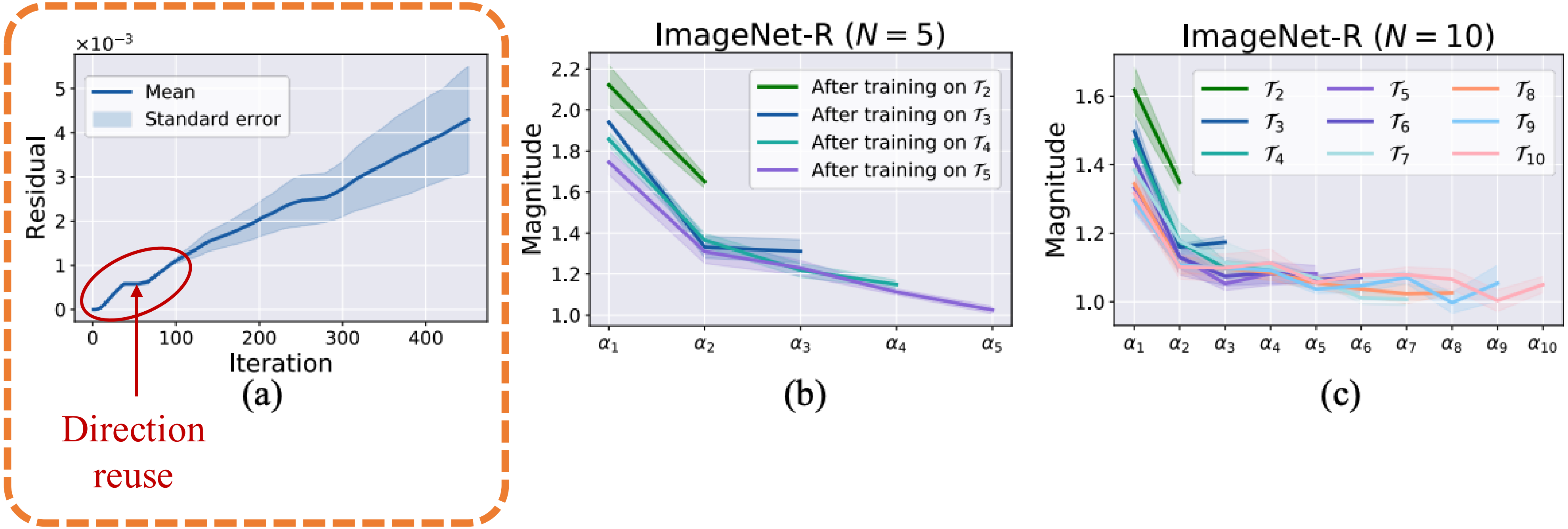


(a) (b) (c)

The directions of previous tasks are important

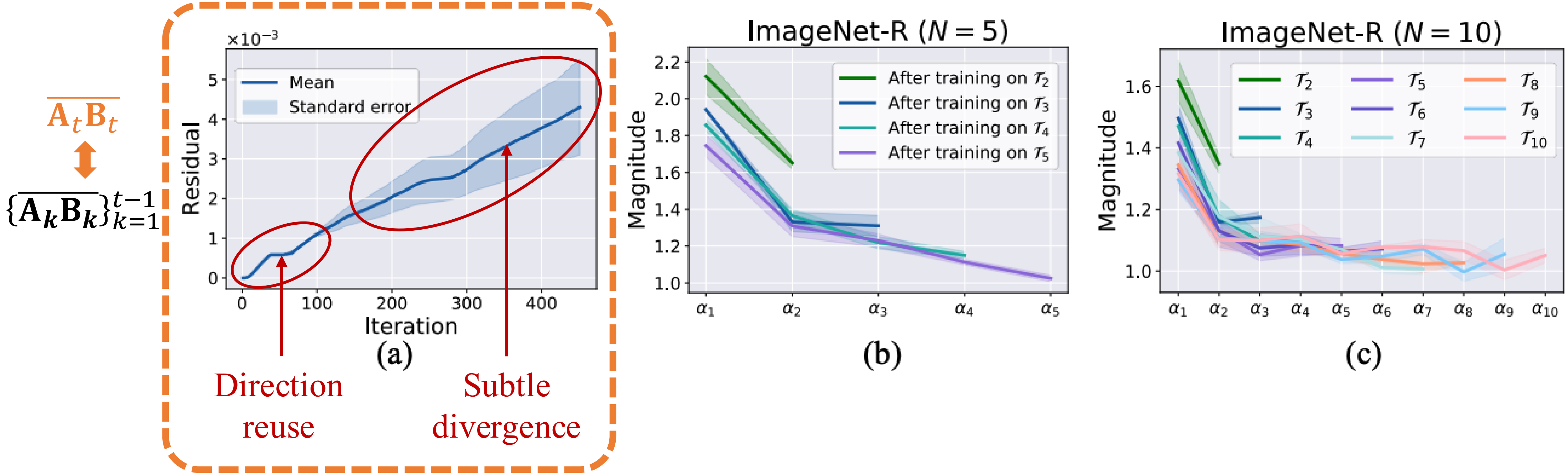
$$h' = (\mathbf{W}_0 + \alpha_1\overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2\overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t\overline{\mathbf{A}_t\mathbf{B}_t})x$$

$\overline{\mathbf{A}_t\mathbf{B}_t}$ ⇕ $\{\overline{\mathbf{A}_k\mathbf{B}_k}\}_{k=1}^{t-1}$



The directions of previous tasks are important

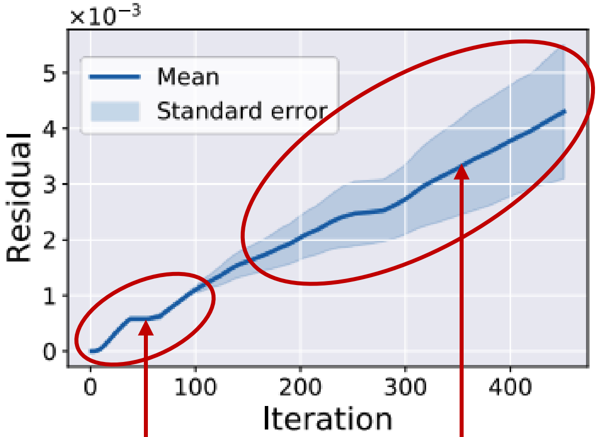
$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$



The directions of previous tasks are important

$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$

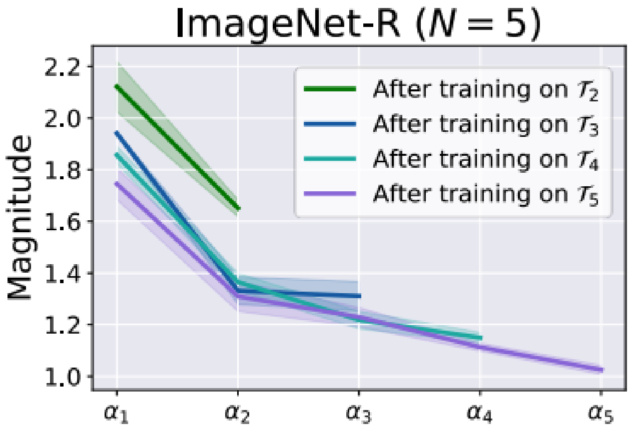
$\overline{\mathbf{A}_t\mathbf{B}_t}$ ⬍ $\{\overline{\mathbf{A}_k\mathbf{B}_k}\}_{k=1}^{t-1}$



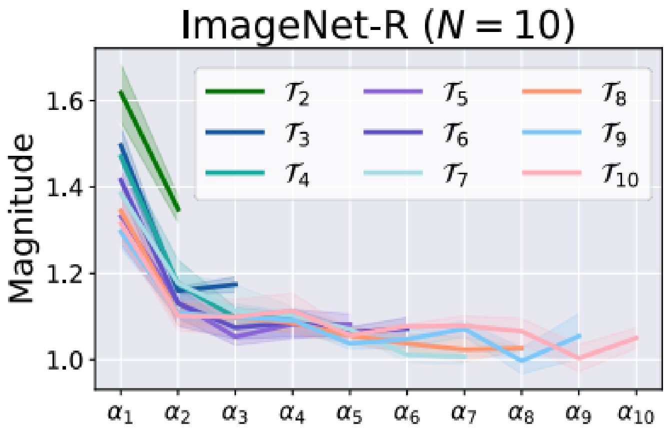
(a)

Direction reuse

Subtle divergence



(b)



(c)

Newly learned direction $\overline{\mathbf{A}_t\mathbf{B}_t}$ highly related to previously learned ones

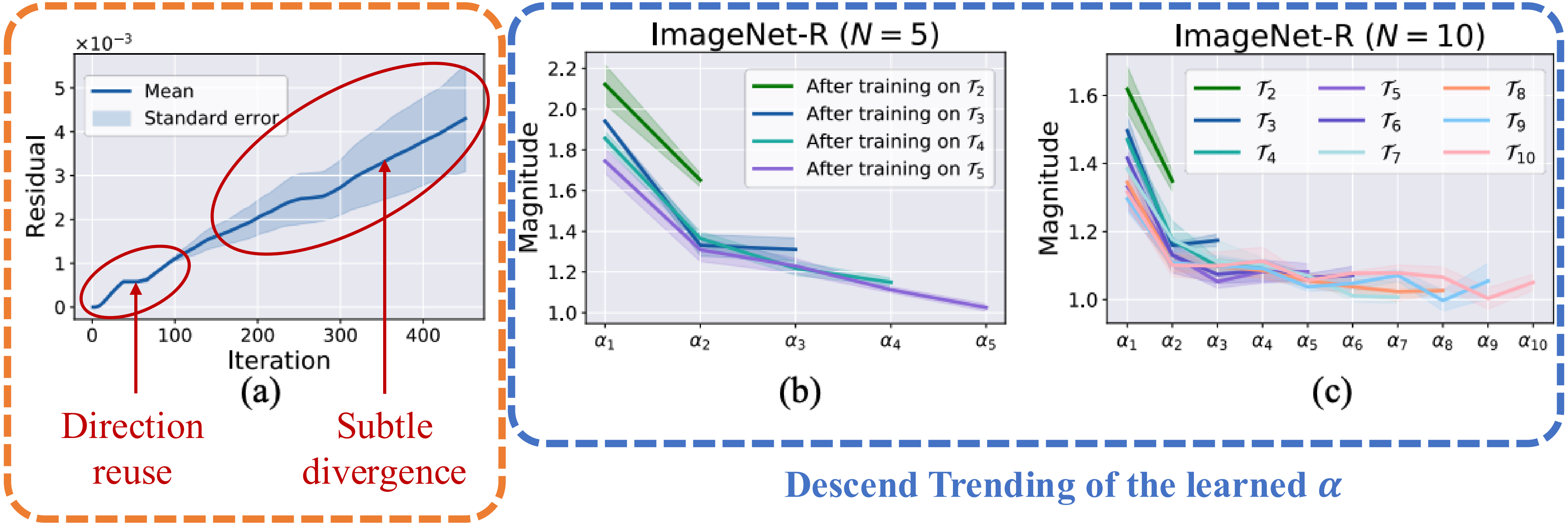
The directions of previous tasks are important

$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$

$\overline{\mathbf{A}_t\mathbf{B}_t}$

$\updownarrow$

$\{\overline{\mathbf{A}_k\mathbf{B}_k}\}_{k=1}^{t-1}$

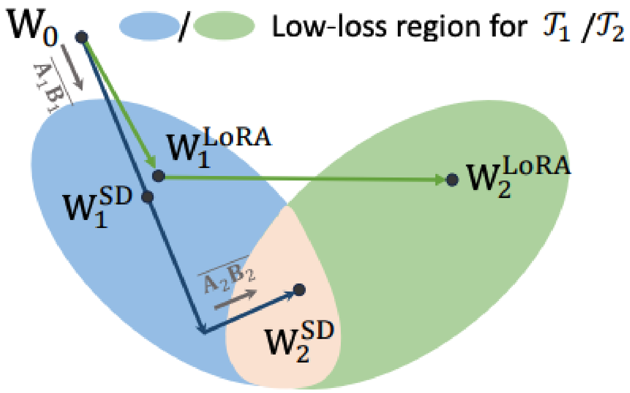


Descend Trending of the learned α

Newly learned direction $\overline{\mathbf{A}_t\mathbf{B}_t}$ highly related to previously learned ones

SD-LoRA effectively uncovers a low-loss path

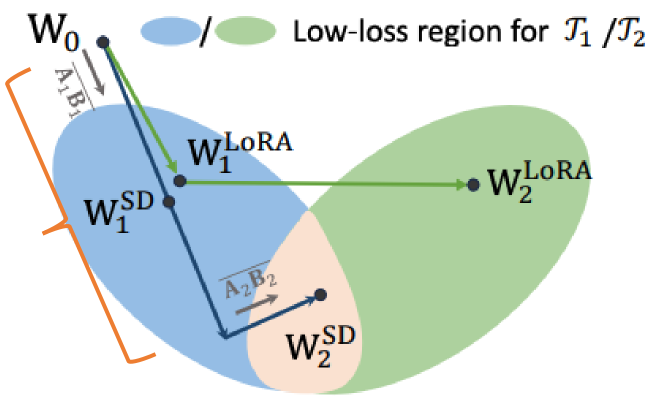
$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$



(a)

SD-LoRA effectively uncovers a low-loss path

$$h' = (\mathbf{W}_0 + \alpha_1\overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2\overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t\overline{\mathbf{A}_t\mathbf{B}_t})x$$

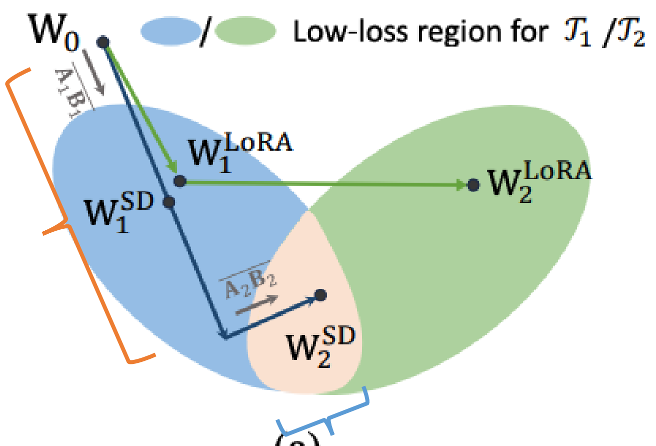


(a)

- α encourages updates along the key directions learned from earlier tasks, rapidly approaching the shared low-loss region for multiple tasks.

SD-LoRA effectively uncovers a low-loss path

$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$



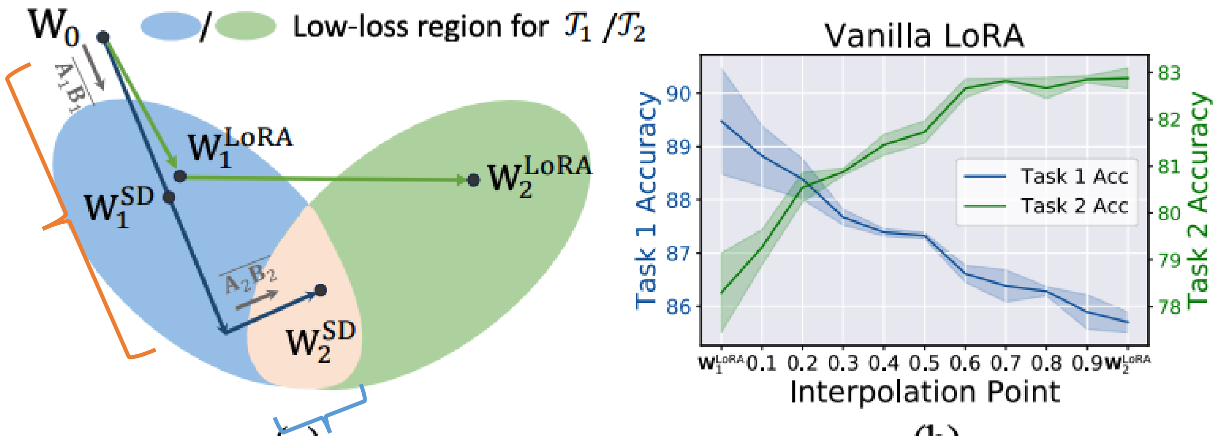
(a)

- α encourages updates along the key directions learned from earlier tasks, rapidly approaching the shared low-loss region for multiple tasks.

- By incrementally introducing LoRA, it fine-tunes these directions, allowing the model to accurately converge on the shared low-loss region for different tasks.

SD-LoRA effectively uncovers a low-loss path

$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$

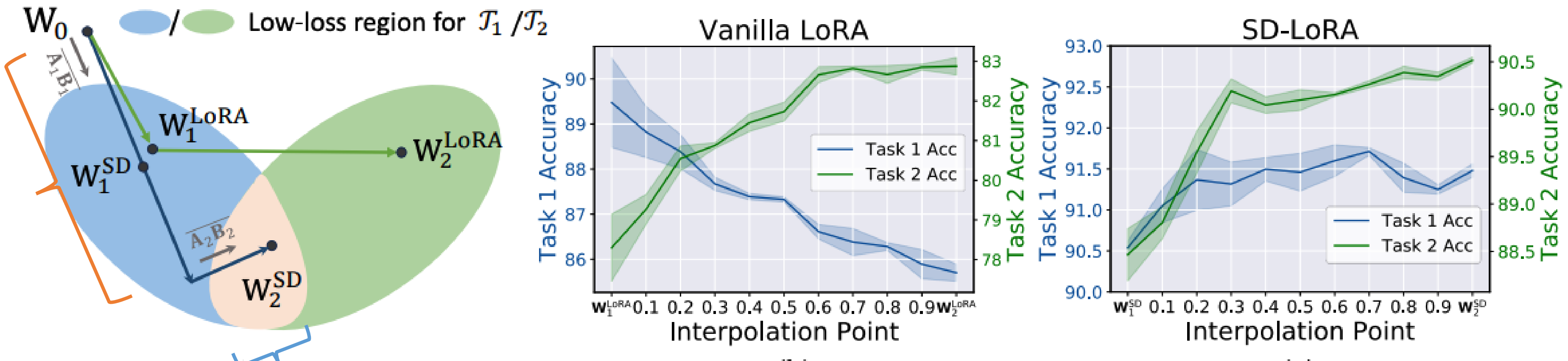


(a) (b)

- α encourages updates along the key directions learned from earlier tasks, rapidly approaching the shared low-loss region for multiple tasks.
- By incrementally introducing LoRA, it fine-tunes these directions, allowing the model to accurately converge on the shared low-loss region for different tasks.

SD-LoRA effectively uncovers a low-loss path

$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$

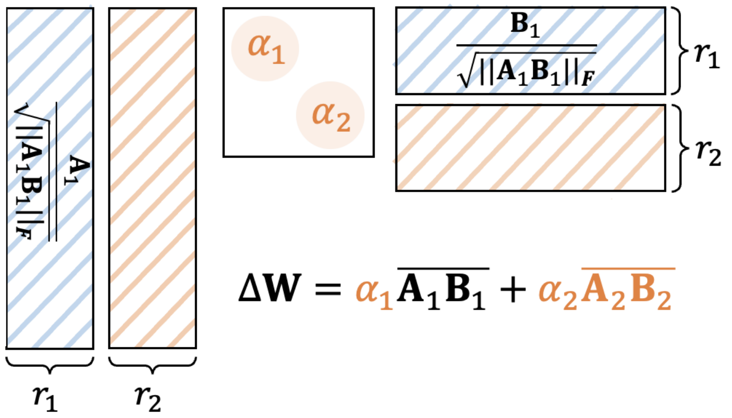


(a) (b) (c)

- α encourages updates along the key directions learned from earlier tasks, rapidly approaching the shared low-loss region for multiple tasks.
- By incrementally introducing LoRA, it fine-tunes these directions, allowing the model to accurately converge on the shared low-loss region for different tasks.

Theoretical Analysis

Theoretically explain why the previously learned LoRA directions are so critical.



Theorem 1. *Suppose the assumptions stated in Appendix A.1 hold, where ϵ_1 is a small constant. Let $\delta \in (0,1)$ be such that $\delta \leq \min_{k\in\{1,\dots,j\}} \frac{\sigma_k - \sigma_{k+1}}{\sigma_k}$. Fix any tolerance level ϵ_2 satisfying $\epsilon_2 \leq \frac{1}{m+n+r}$. Let η denote the learning rate for updating the matrices $\mathbf{A}$ and $\mathbf{B}$, and define $\Delta\mathbf{W}^{[:i]}$ as the rank-i approximation of $\Delta\mathbf{W}^\star$, obtained by retaining the top-i principal components.*

Then, there exist some numerical constants c and c′, and a sequence of iteration indices:

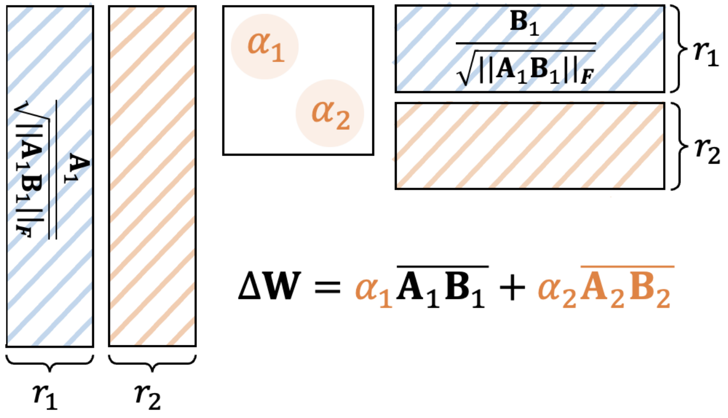
$$i_1 \leq i_2 \leq \ldots \leq i_j \leq \frac{c'}{\delta\eta\sigma_j} \log\left(\frac{\kappa_j}{\delta\epsilon_2}\right)$$

such that, with high probability, gradient descent with step size $\eta \leq c\min\{\delta, 1-\delta\}\frac{\sigma_j^2}{\sigma_1^3}$ and initialization scaling factor $\rho \leq \left(\frac{c\delta\epsilon_2}{\kappa_j}\right)^{\frac{1}{c\delta}}$ ensures that the approximation error satisfies

$$\left\|\mathbf{A}_{i_k}\mathbf{B}_{i_k} - \Delta\mathbf{W}^{[:k]}\right\|_{\text{op}} \leq \epsilon_2\sigma_1 + \epsilon_1, \quad \forall k = 1, 2, \ldots, j.$$

Theoretical Analysis

Theoretically explain the previously learned LoRA directions are so critical.



Theorem 1. *Suppose the assumptions stated in Appendix A.1 hold, where ϵ_1 is a small constant. Let $\delta \in (0,1)$ be such that $\delta \leq \min_{k\in\{1,\ldots,j\}} \frac{\sigma_k - \sigma_{k+1}}{\sigma_k}$. Fix any tolerance level ϵ_2 satisfying $\epsilon_2 \leq \frac{1}{m+n+r}$. Let η denote the learning rate for updating the matrices $\mathbf{A}$ and $\mathbf{B}$, and define $\Delta\mathbf{W}^{[:i]}$ as the rank-i approximation of $\Delta\mathbf{W}^\star$, obtained by retaining the top-i principal components.*

Then, there exist some numerical constants c and c′, and a sequence of iteration indices:

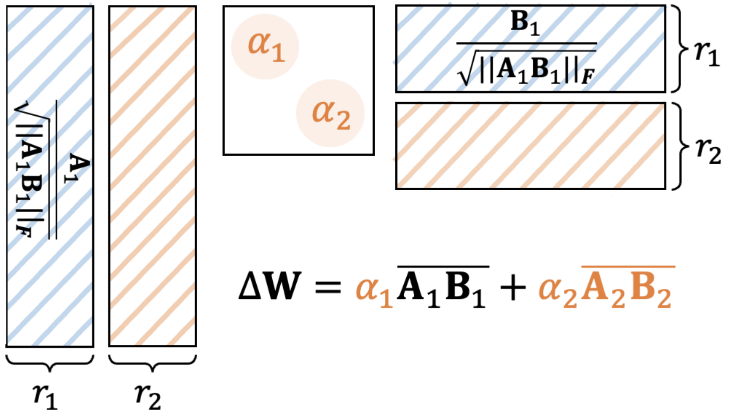
$$i_1 \leq i_2 \leq \ldots \leq i_j \leq \frac{c'}{\delta\eta\sigma_j} \log\left(\frac{\kappa_j}{\delta\epsilon_2}\right)$$

such that, with high probability, gradient descent with step size $\eta \leq c\min\{\delta, 1-\delta\}\frac{\sigma_j^2}{\sigma_1^3}$ and initialization scaling factor $\rho \leq \left(\frac{c\delta\epsilon_2}{\kappa_j}\right)^{\frac{1}{c\delta}}$ ensures that the approximation error satisfies

$$\left\|\mathbf{A}_{i_k}\mathbf{B}_{i_k} - \Delta\mathbf{W}^{[:k]}\right\|_{\text{op}} \leq \epsilon_2\sigma_1 + \epsilon_1, \quad \forall k = 1, 2, \ldots, j.$$

Theoretical Analysis

Theoretically explain the previously learned LoRA directions are so critical.



Theorem 1. *Suppose the assumptions stated in Appendix A.1 hold, where ϵ_1 is a small constant. Let $\delta \in (0,1)$ be such that $\delta \leq \min_{k\in\{1,\ldots,j\}} \frac{\sigma_k-\sigma_{k+1}}{\sigma_k}$. Fix any tolerance level ϵ_2 satisfying $\epsilon_2 \leq \frac{1}{m+n+r}$. Let η denote the learning rate for updating the matrices $\mathbf{A}$ and $\mathbf{B}$, and define $\Delta\mathbf{W}^{[:i]}$ as the rank-i approximation of $\Delta\mathbf{W}^\star$, obtained by retaining the top-i principal components.*

Then, there exist some numerical constants c and c', and a sequence of iteration indices:

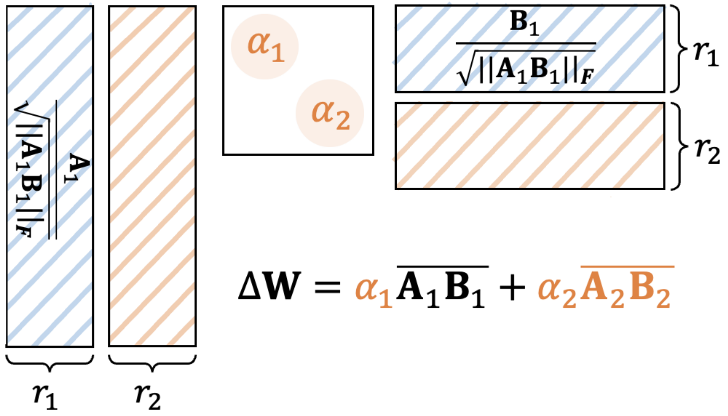
$$i_1 \leq i_2 \leq \ldots \leq i_j \leq \frac{c'}{\delta\eta\sigma_j}\log\left(\frac{\kappa_j}{\delta\epsilon_2}\right)$$

such that, with high probability, gradient descent with step size $\eta \leq c\min\{\delta, 1-\delta\}\frac{\sigma_j^2}{\sigma_1^3}$ and initialization scaling factor $\rho \leq \left(\frac{c\delta\epsilon_2}{\kappa_j}\right)^{\frac{1}{c\delta}}$ ensures that the approximation error satisfies

$$\left\|\mathbf{A}_{i_k}\mathbf{B}_{i_k} - \Delta\mathbf{W}^{[:k]}\right\|_{\text{op}} \leq \epsilon_2\sigma_1 + \epsilon_1, \quad \forall k = 1,2,\ldots,j.$$

Theoretical Analysis

Theoretically explain the previously learned LoRA directions are so critical.



Theorem 1. *Suppose the assumptions stated in Appendix A.1 hold, where ϵ_1 is a small constant. Let $\delta \in (0,1)$ be such that $\delta \leq \min_{k\in\{1,\ldots,j\}} \frac{\sigma_k - \sigma_{k+1}}{\sigma_k}$. Fix any tolerance level ϵ_2 satisfying $\epsilon_2 \leq \frac{1}{m+n+r}$. Let η denote the learning rate for updating the matrices $\mathbf{A}$ and $\mathbf{B}$, and define $\Delta\mathbf{W}^{[:i]}$ as the rank-i approximation of $\Delta\mathbf{W}^\star$, obtained by retaining the top-i principal components.*

Then, there exist some numerical constants c and c', and a sequence of iteration indices:

$$i_1 \leq i_2 \leq \ldots \leq i_j \leq \frac{c'}{\delta\eta\sigma_j}\log\left(\frac{\kappa_j}{\delta\epsilon_2}\right)$$

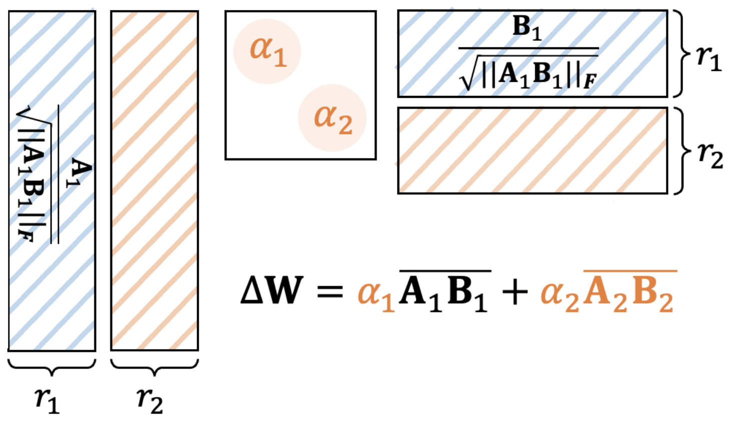
such that, with high probability, gradient descent with step size $\eta \leq c\min\{\delta, 1-\delta\}\frac{\sigma_j^2}{\sigma_1^3}$ and initialization scaling factor $\rho \leq \left(\frac{c\delta\epsilon_2}{\kappa_j}\right)^{\frac{1}{c\delta}}$ ensures that the approximation error satisfies

$$\left\|\mathbf{A}_{i_k}\mathbf{B}_{i_k} - \Delta\mathbf{W}^{[:k]}\right\|_{\text{op}} \leq \epsilon_2\sigma_1 + \epsilon_1, \quad \forall k = 1, 2, \ldots, j.$$

As the continual training progress, the learned matrix AB gradually approximate the principal components of $\Delta\mathbf{W}^*$

Efficient Variants of SD-LoRA

$$h' = (\mathbf{W}_0 + \alpha_1 \overline{\mathbf{A}_1\mathbf{B}_1} + \alpha_2 \overline{\mathbf{A}_2\mathbf{B}_2} + \cdots + \alpha_t \overline{\mathbf{A}_t\mathbf{B}_t})x$$



Reduce the rank of the newly introduced LoRA

SD-LoRA-RR

$$r_1 = r_2 = \ldots > r_\mu = r_{\mu+1} = \ldots > r_\nu = r_{\nu+1} = \ldots = r_N$$

Don't need to introduce the extra LoRA part

SD-LoRA-KD (Knowledge Distillation)

$$\{\Delta\alpha_k\}_{k=1}^{t-1} = \arg\min_{\{\alpha'_k\}_{k=1}^{t-1}} \left\| \overline{\mathbf{A}_t\mathbf{B}_t} - \sum_{k=1}^{t-1} \alpha'_k \overline{\mathbf{A}_k\mathbf{B}_k} \right\|_F^2$$

$$\boldsymbol{h}' = \left(\mathbf{W}_0 + (\alpha_1 + \Delta\alpha_1)\overline{\mathbf{A}_1\mathbf{B}_1} + (\alpha_2 + \Delta\alpha_2)\overline{\mathbf{A}_2\mathbf{B}_2} + \ldots + (\alpha_{t-1} + \Delta\alpha_{t-1})\overline{\mathbf{A}_{t-1}\mathbf{B}_{t-1}}\right)\boldsymbol{x}$$

Represent new directions by the subspace spanned by previously learned directions

Experimental Results

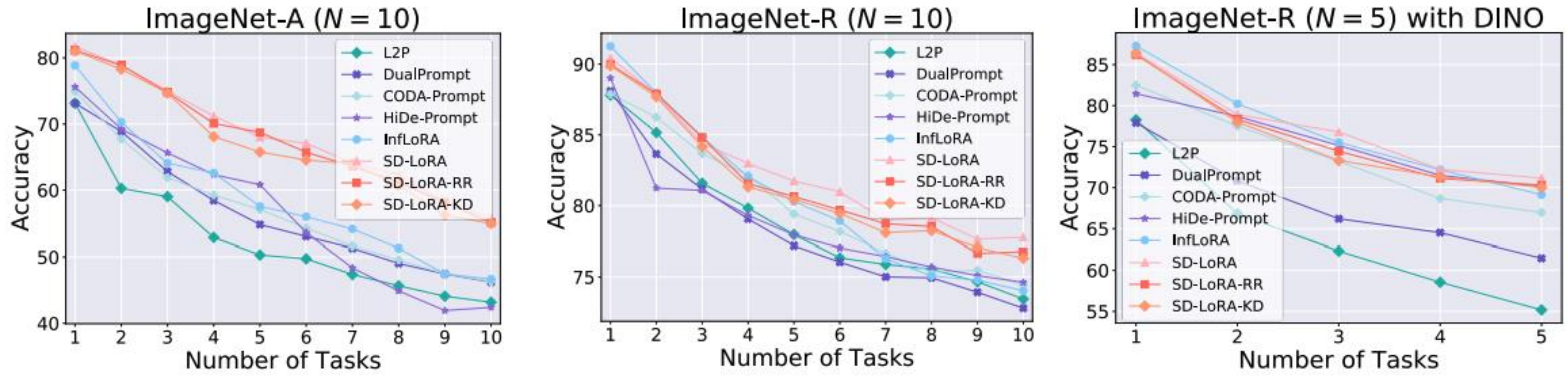
The performance on different task lengths.

Method	ImageNet-R ($N = 5$)		ImageNet-R ($N = 10$)		ImageNet-R ($N = 20$)	
	Acc ↑	AAA ↑	Acc ↑	AAA ↑	Acc ↑	AAA ↑
Full Fine-Tuning	$64.92_{(0.87)}$	$75.57_{(0.50)}$	$60.57_{(1.06)}$	$72.31_{(1.09)}$	$49.95_{(1.31)}$	$65.32_{(0.84)}$
L2P	$73.04_{(0.71)}$	$76.94_{(0.41)}$	$71.26_{(0.44)}$	$76.13_{(0.46)}$	$68.97_{(0.51)}$	$74.16_{(0.32)}$
DualPrompt	$69.99_{(0.57)}$	$72.24_{(0.41)}$	$68.22_{(0.20)}$	$73.81_{(0.39)}$	$65.23_{(0.45)}$	$71.30_{(0.16)}$
CODA-Prompt	$76.63_{(0.27)}$	$80.30_{(0.28)}$	$74.05_{(0.41)}$	$78.14_{(0.39)}$	$69.38_{(0.33)}$	$73.95_{(0.63)}$
HiDe-Prompt	$74.77_{(0.25)}$	$78.15_{(0.24)}$	$74.65_{(0.14)}$	$78.46_{(0.18)}$	$73.59_{(0.19)}$	$77.93_{(0.19)}$
InfLoRA	$76.95_{(0.23)}$	$81.81_{(0.14)}$	$74.75_{(0.64)}$	$80.67_{(0.55)}$	$69.89_{(0.56)}$	$76.68_{(0.57)}$
SD-LoRA	**79.15**$_{(0.20)}$	**83.01**$_{(0.42)}$	**77.34**$_{(0.35)}$	**82.04**$_{(0.24)}$	**75.26**$_{(0.37)}$	$80.22_{(0.72)}$
SD-LoRA-RR	$79.01_{(0.26)}$	$82.50_{(0.38)}$	$77.18_{(0.39)}$	$81.74_{(0.24)}$	$74.05_{(0.51)}$	**80.65**$_{(0.35)}$
SD-LoRA-KD	$78.85_{(0.29)}$	$82.47_{(0.58)}$	$77.03_{(0.67)}$	$81.52_{(0.26)}$	$74.12_{(0.66)}$	$80.11_{(0.75)}$

The performance on different continual learning benchmarks.

Method	ImageNet-A ($N = 10$)		DomainNet ($N = 5$)	
	Acc ↑	AAA ↑	Acc ↑	AAA ↑
Full Fine-Tuning	$16.31_{(7.89)}$	$30.04_{(13.18)}$	$51.46_{(0.47)}$	$67.08_{(1.13)}$
L2P (Wang et al., 2022b)	$42.94_{(1.27)}$	$51.40_{(1.95)}$	$70.26_{(0.25)}$	$75.83_{(0.98)}$
DualPrompt (Wang et al., 2022a)	$45.49_{(0.96)}$	$54.68_{(1.24)}$	$68.26_{(0.90)}$	$73.84_{(0.45)}$
CODA-Prompt (Smith et al., 2023)	$45.36_{(0.78)}$	$57.03_{(0.94)}$	$70.58_{(0.53)}$	$76.68_{(0.44)}$
HiDe-Prompt (Wang et al., 2024a)	$42.70_{(0.60)}$	$56.32_{(0.40)}$	$72.20_{(0.08)}$	$77.01_{(0.04)}$
InfLoRA (Liang & Li, 2024)	$49.20_{(1.12)}$	$60.92_{(0.61)}$	$71.59_{(0.23)}$	$78.29_{(0.50)}$
SDLoRA	**55.96**$_{(0.73)}$	**64.95**$_{(1.63)}$	**72.82**$_{(0.37)}$	**78.89**$_{(0.50)}$
SD-LoRA-RR	$55.59_{(1.08)}$	$64.59_{(1.91)}$	$72.58_{(0.40)}$	$78.79_{(0.78)}$
SD-LoRA-KD	$54.24_{(1.12)}$	$63.89_{(0.58)}$	$72.15_{(0.50)}$	$78.44_{(0.66)}$

The detailed performance on the streaming tasks and the results on different backbones



(a) (b) (c)

Thanks!