PACE: marrying generalization in PArameter-efficient fine-tuning with Consistency rEgularization

Yao Ni[†] Shan Zhang[‡, †] Piotr Koniusz[§, †]

[†]The Australian National University [§]Data61♥CSIRO
[‡]Australian Institute for Machine Learning, The University of Adelaide

NeurIPS 2024 Spotlight

Australian National University

THE UNIVERSITY of ADELAIDE

CSIRO

PACE: marrying generalization in PArameter-efficient fine-tuning with Consistency rEgularization

Yao Ni[†] Shan Zhang[‡,†] Piotr Koniusz[§,†]

[†]The Australian National University [§]Data61♥CSIRO

[‡]Australian Institute for Machine Learning, The University of Adelaide

NeurIPS 2024 Spotlight

Yao Ni Seeking PostDoc Position. Scan his CV.

Background: Parameter-Efficient Fine-Tuning

- Models (GPTs, Vision Transformers) are becoming increasingly large.
- Full parameter fine-tuning is resource-intensive.

Background: Parameter-Efficient Fine-Tuning

- Models (GPTs, Vision Transformers) are becoming increasingly large.
- Full parameter fine-tuning is resource-intensive.

Parameter-Efficient Fine-Tuning (PEFT): Adjusting a small subset of parameters.

- Higher Performance.
- Less parameter storage.

Background: Parameter-Efficient Fine-Tuning

- Models (GPTs, Vision Transformers) are becoming increasingly large.
- Full parameter fine-tuning is resource-intensive.

Parameter-Efficient Fine-Tuning (PEFT): Adjusting a small subset of parameters.

- Higher Performance.
- Less parameter storage.

Issues: lack of generalizability & forgetting pre-trained knowledge.

Background: Parameter-Efficient Fine-Tuning

- Models (GPTs, Vision Transformers) are becoming increasingly large.
- Full parameter fine-tuning is resource-intensive.

Parameter-Efficient Fine-Tuning (PEFT): Adjusting a small subset of parameters.

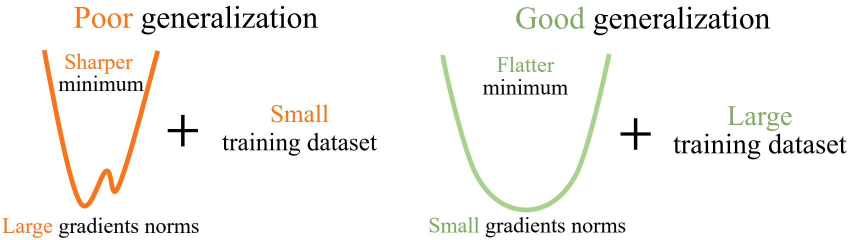
- Higher Performance.
- Less parameter storage.

Issues: lack of generalizability & forgetting pre-trained knowledge.

Goal: improve generalization & retain pre-trained knowledge.

Motivation: Theorem 1

Theorem 1: Smaller gradient norm and larger dataset lead to better generalization on unseen data.



Motivation: Theorem 1

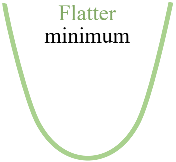
Theorem 1: Smaller gradient norm and larger dataset lead to better generalization on unseen data.

Poor generalization

Large gradients norms

+ Small training dataset

Good generalization



Small gradients norms

↓

Robust to unseen data

+ Large training dataset

Motivation: Theorem 1

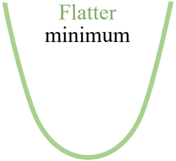
Theorem 1: Smaller gradient norm and larger dataset lead to better generalization on unseen data.

Poor generalization

Large gradients norms

+ Small training dataset

Good generalization



Small gradients norms

↓

Robust to unseen data

+ Large training dataset

↓

Rich knowledge for unseen data

Solution for better generalization and retain knowledge

Smaller Gradients Norms
Larger dataset

Solution for better generalization and retain knowledge

Smaller Gradients Norms ← Regularize gradients
Larger dataset

Solution for better generalization and retain knowledge

Smaller Gradients Norms $\leftarrow$ Regularize gradients

Larger dataset

small dataset in downstream tasks

Solution for better generalization and retain knowledge

Smaller Gradients Norms $\leftarrow$ Regularize gradients

Larger dataset

- small dataset in downstream tasks
- Retain knowledge by fine-tuned pre-trained alignment (FPA)

Solution for better generalization and retain knowledge

Smaller Gradients Norms ← Regularize gradients

Larger dataset

- small dataset in downstream tasks
- Retain knowledge by fine-tuned pre-trained alignment (FPA)

Pre-trained
$\boldsymbol{\theta}_0$

$\boldsymbol{x}$

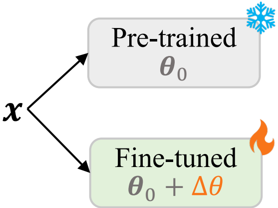
Fine-tuned
$\boldsymbol{\theta}_0 + \Delta\theta$

Solution for better generalization and retain knowledge

Smaller Gradients Norms ← Regularize gradients

Larger dataset

- small dataset in downstream tasks
- Retain knowledge by fine-tuned pre-trained alignment (FPA)

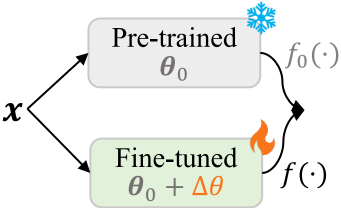


Solution for better generalization and retain knowledge

Smaller Gradients Norms ← Regularize gradients

Larger dataset

- small dataset in downstream tasks
- Retain knowledge by fine-tuned pre-trained alignment (FPA)

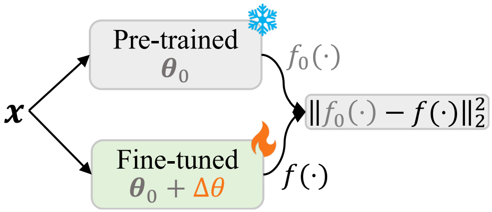


Solution for better generalization and retain knowledge

Smaller Gradients Norms ← Regularize gradients

Larger dataset

- small dataset in downstream tasks
- Retain knowledge by fine-tuned pre-trained alignment (FPA)

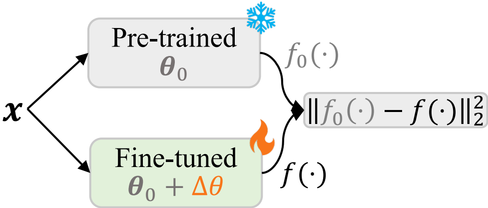


Solution for better generalization and retain knowledge

Smaller Gradients Norms ← Regularize gradients

Larger dataset

- small dataset in downstream tasks
- Retain knowledge by fine-tuned pre-trained alignment (FPA)



Prop 1. Naive alignment does not guarantee smaller gradient norms

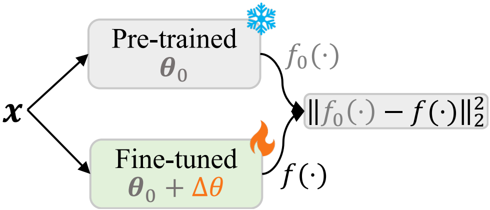
Solution for better generalization and retain knowledge

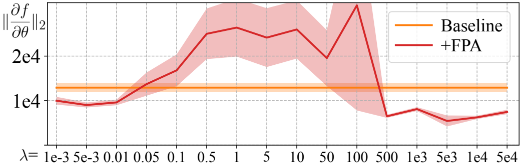
Smaller Gradients Norms ← Regularize gradients

Larger dataset

small dataset in downstream tasks

Retain knowledge by fine-tuned pre-trained alignment (FPA)





Gradient norms & reg. strength λ (CIFAR-100, ViT-B/16)

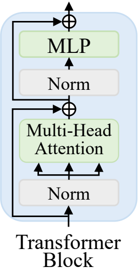
Prop 1. Naive alignment does not guarantee smaller gradient norms

Our method: PACE

To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.

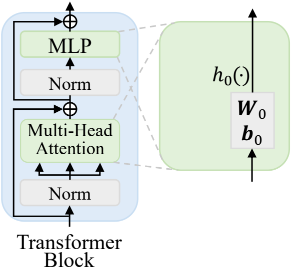
Our method: PACE

To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.



Our method: PACE

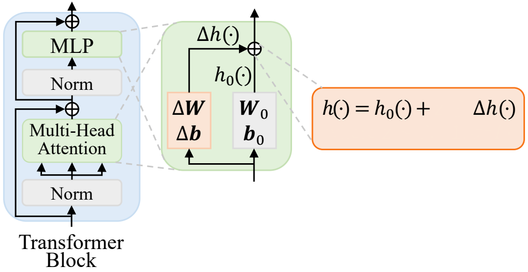
To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.



$\boldsymbol{W}_0 \& \boldsymbol{b}_0$, $\Delta\boldsymbol{W} \& \Delta\boldsymbol{b}$: pre-trained/adapter linear weights;

Our method: PACE

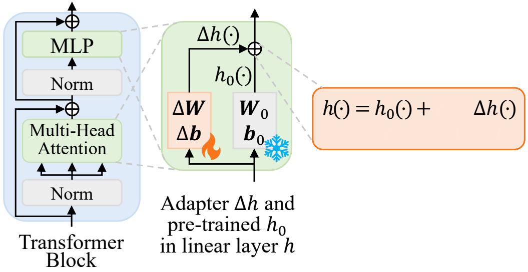
To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.



$\boldsymbol{W}_0 \& \boldsymbol{b}_0, \Delta \boldsymbol{W} \& \Delta \boldsymbol{b}$: pre-trained/adapter linear weights;

Our method: PACE

To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.

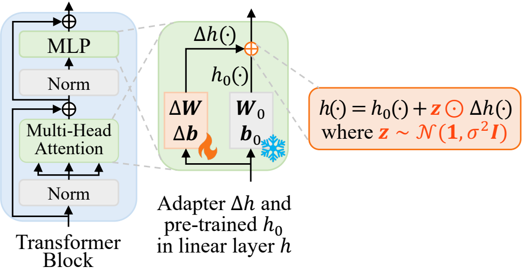


$\boldsymbol{W}_0 \& \boldsymbol{b}_0, \Delta\boldsymbol{W} \& \Delta\boldsymbol{b}$: pre-trained/adapter linear weights;

Our method: PACE

To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.

Transformer block with adapter perturbed by noise

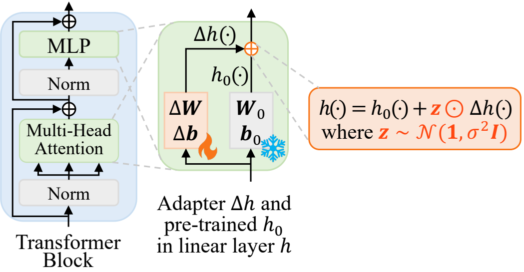


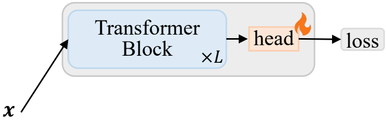
$\boldsymbol{W}_0 \& \boldsymbol{b}_0, \Delta\boldsymbol{W} \& \Delta\boldsymbol{b}$: pre-trained/adapter linear weights;

Our method: PACE

To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.

Transformer block with adapter perturbed by noise

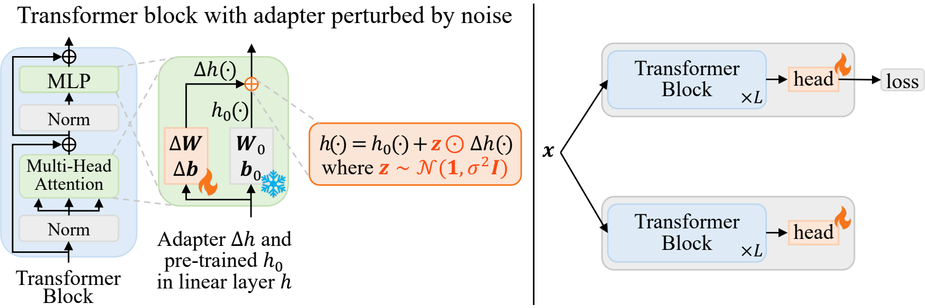




$\boldsymbol{W}_0\&\boldsymbol{b}_0, \Delta\boldsymbol{W}\&\Delta\boldsymbol{b}$: pre-trained/adapter linear weights; $\boldsymbol{x}$: sample; L: number of blocks

Our method: PACE

To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.

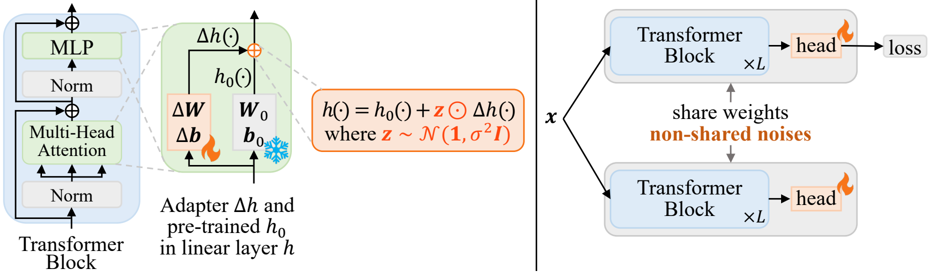


$\boldsymbol{W}_0 \& \boldsymbol{b}_0, \Delta \boldsymbol{W} \& \Delta \boldsymbol{b}$: pre-trained/adapter linear weights; $\boldsymbol{x}$: sample; L: number of blocks

Our method: PACE

To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.

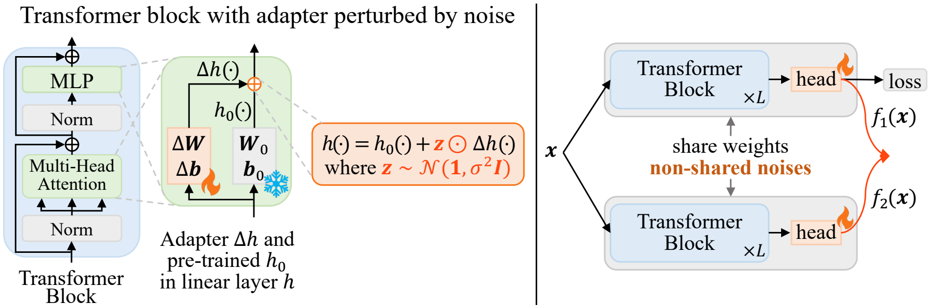
Transformer block with adapter perturbed by noise



W_0&b_0, ΔW&Δb: pre-trained/adapter linear weights; x: sample; L: number of blocks

Our method: PACE

To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.

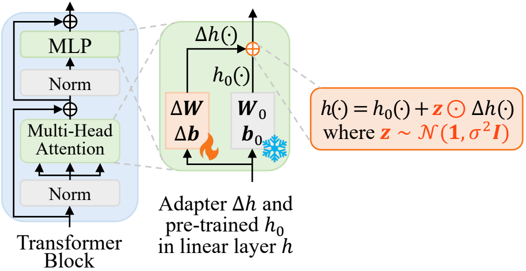


$\boldsymbol{W}_0 \& \boldsymbol{b}_0, \Delta\boldsymbol{W} \& \Delta\boldsymbol{b}$: pre-trained/adapter linear weights; $\boldsymbol{x}$: sample; L: number of blocks

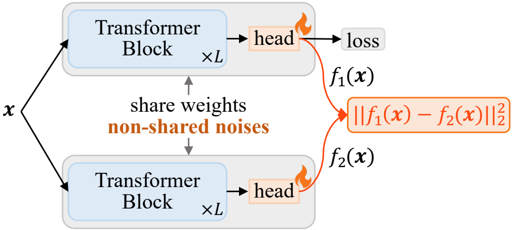
Our method: PACE

To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.

Transformer block with adapter perturbed by noise



Consistency regularization between two outputs of $\boldsymbol{x}$

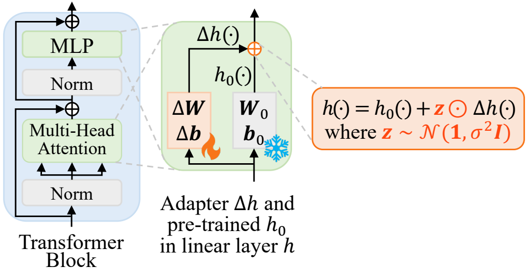


$\boldsymbol{W}_0 \& \boldsymbol{b}_0, \Delta \boldsymbol{W} \& \Delta \boldsymbol{b}$: pre-trained/adapter linear weights; $\boldsymbol{x}$: sample; L: number of blocks

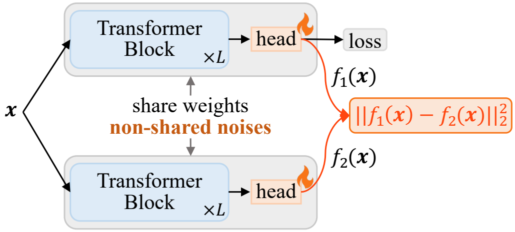
Our method: PACE

To regularize gradients and align fine-tuned pre-trained models,
PACE perturbs adapter features and enforces consistency across perturbations.

Transformer block with adapter perturbed by noise



Consistency regularization between two outputs of $\boldsymbol{x}$



$\boldsymbol{W}_0 \& \boldsymbol{b}_0, \Delta \boldsymbol{W} \& \Delta \boldsymbol{b}$: pre-trained/adapter linear weights; $\boldsymbol{x}$: sample; L: number of blocks

PACE improves generalization and retains pre-trained knowledge

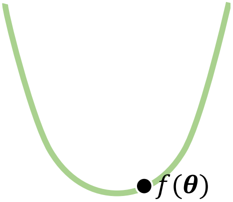
PACE: Theorem 2

Theorem 2: PACE regularizes first- and second-order gradients

PACE: Theorem 2

Theorem 2: PACE regularizes first- and second-order gradients

Large grad norm

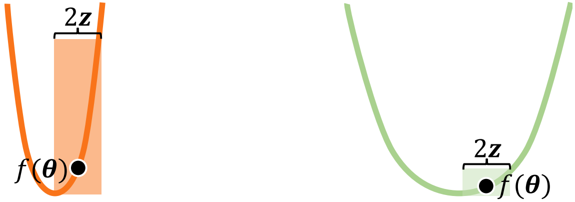


Small grad norm

$\boldsymbol{\theta}$: model weights;

PACE: Theorem 2

Theorem 2: PACE regularizes first- and second-order gradients



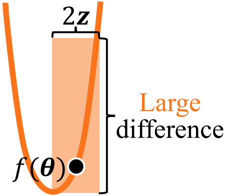
Large grad norm

Small grad norm

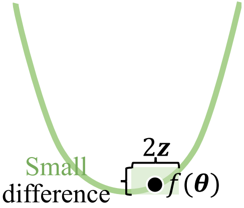
$\boldsymbol{\theta}$: model weights; $\boldsymbol{z}$: noise.

PACE: Theorem 2

Theorem 2: PACE regularizes first- and second-order gradients



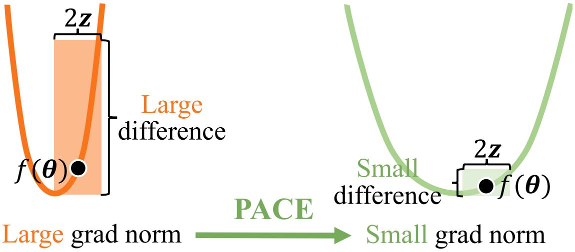
Large grad norm



Small grad norm

$\boldsymbol{\theta}$: model weights; $\boldsymbol{z}$: noise.

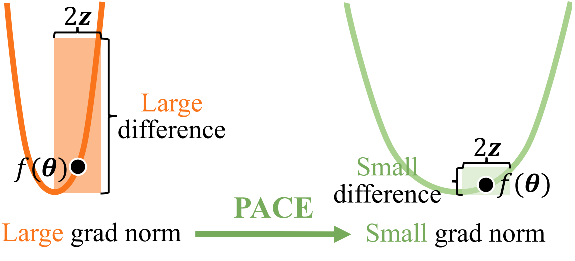
Theorem 2: PACE regularizes first- and second-order gradients



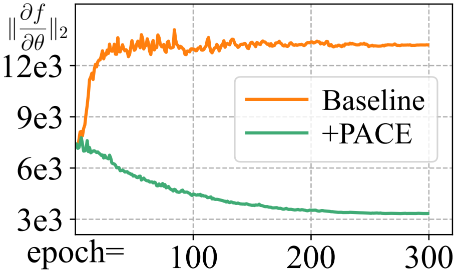
$\boldsymbol{\theta}$: model weights; $\boldsymbol{z}$: noise.

PACE: Theorem 2

Theorem 2: PACE regularizes first- and second-order gradients



$\boldsymbol{\theta}$: model weights; $\boldsymbol{z}$: noise.



Gradient norms on CIFAR-100 w/ ViT-B/16

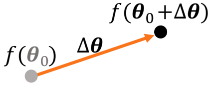
PACE:Theorem 3

Theorem 3: PACE minimize fine-tuned pre-trained distance to retain knowledge.

$f(\boldsymbol{\theta}_0)$

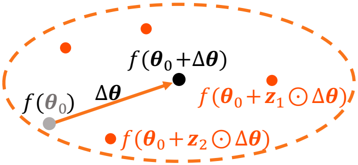
PACE:Theorem 3

Theorem 3: PACE minimize fine-tuned pre-trained distance to retain knowledge.



PACE:Theorem 3

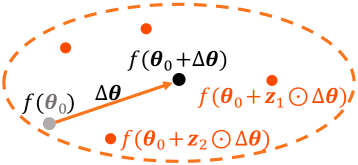
Theorem 3: PACE minimize fine-tuned pre-trained distance to retain knowledge.



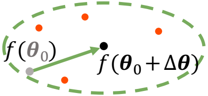
Large FP-distance

PACE:Theorem 3

Theorem 3: PACE minimize fine-tuned pre-trained distance to retain knowledge.

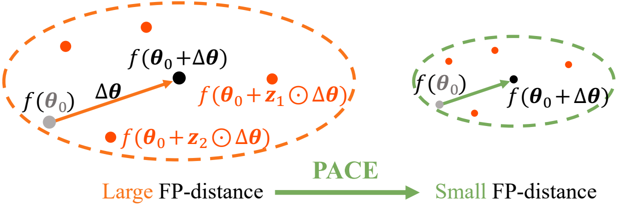


Large FP-distance



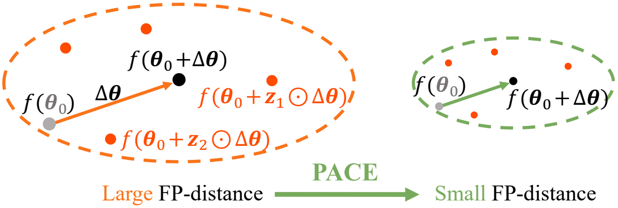
Small FP-distance

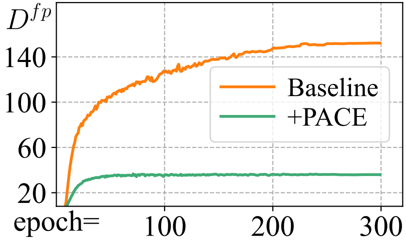
Theorem 3: PACE minimize fine-tuned pre-trained distance to retain knowledge.



PACE:Theorem 3

Theorem 3: PACE minimize fine-tuned pre-trained distance to retain knowledge.





Distance between fine-tuned and pre-trained models (D^{fp}) on CIFAR-100 w/ ViT-B/16.

Experiments: Image Classification

Results on VTAB-1K with ViT-B/16.

Method	Natural							Specialized				Structured								
	Cifar100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	NsORB-Ele	Mean Acc.
Full	68.9	87.7	64.3	97.3	86.9	87.4	38.8	79.7	95.7	84.2	73.9	56.3	58.6	41.7	65.5	57.5	46.7	25.7	29.1	68.9
Linear	64.4	85.0	63.2	97.0	86.3	36.6	51.0	78.5	87.5	68.5	74.0	34.3	30.6	33.2	55.4	12.5	20.0	9.6	19.2	57.6
VPT-Deep	78.8	90.8	65.8	98.0	88.3	78.1	49.6	81.8	96.1	83.4	68.4	68.5	60.0	46.5	72.8	73.6	47.9	32.9	37.8	72.0
Adapter	69.2	90.1	68.0	98.8	89.9	82.8	54.3	84.0	94.9	81.9	75.5	80.9	65.3	48.6	78.3	74.8	48.5	29.9	41.6	73.9
AdaptFormer	70.8	91.2	70.5	99.1	90.9	86.6	54.8	83.0	95.8	84.4	76.3	81.9	64.3	49.3	80.3	76.3	45.7	31.7	41.1	74.7
LoRA	67.1	91.4	69.4	98.8	90.4	85.3	54.0	84.9	95.3	84.4	73.6	82.9	69.2	49.8	78.5	75.7	47.1	31.0	44.0	74.5
NOAH	69.6	92.7	70.2	99.1	90.4	86.1	53.7	84.4	95.4	83.9	75.8	82.8	68.9	49.9	81.7	81.8	48.3	32.8	44.2	74.2
RepAdapter	69.0	92.6	**75.1**	99.4	91.8	90.2	52.9	87.4	95.9	87.4	75.5	75.9	62.3	53.3	80.6	77.3	54.9	29.5	37.9	76.1
RLRR	75.6	92.4	72.9	99.3	91.5	89.8	57.0	86.8	95.2	85.3	75.9	79.7	64.2	53.9	82.1	83.9	53.7	33.4	43.6	76.7
GLoRA	76.4	92.9	74.6	**99.6**	**92.5**	91.5	57.8	87.3	**96.8**	88.0	76.0	83.1	67.3	54.5	**86.2**	83.8	52.9	37.0	41.4	78.0
Baseline	74.9	93.3	72.0	99.4	91.0	91.5	54.8	83.2	95.7	86.9	74.2	83.0	70.5	51.9	81.4	77.9	51.7	33.6	44.4	76.4
+PACE	**79.0**	**94.2**	73.6	99.4	92.4	**93.7**	**58.0**	**87.4**	96.4	**89.3**	**77.1**	**84.9**	**70.9**	**54.9**	84.3	**84.7**	**57.3**	**39.3**	**44.8**	**79.0**

Experiments: Text classification & generation

Results for GLUE w/ RoBERTa$_{\text{base}}$. Matthew's/Pearson correlation for COLA/STSB, and accuracy for others.

Method	COLA	STSB	MRPC	RTE	QNLI	SST2	Avg.
Full	63.6	91.2	90.2	78.7	92.8	94.8	85.2
BitFit	62.0	90.8	**92.7**	81.5	91.8	93.7	85.4
Adapt	62.6	90.3	88.4	75.9	93.0	94.7	84.2
VeRA	65.6	90.7	89.5	78.7	91.8	94.6	85.2
LoRA	63.4	91.5	89.7	86.6	93.3	95.1	86.6
+PACE	**66.2**	**92.0**	91.4	**86.9**	**93.6**	**95.6**	**87.6**

Results for GSM-8K w/ Phi-3-mini-4k-instruct.

Method	Accuracy
Pre-trained	62.01
Full	73.16
LoRA	75.66
+PACE	**78.77**

Conclusions

Conclusions:

- PACE perturbs adapter features and enforces consistency regularization across perturbations.
- PACE regularizes gradients for improved generalization and reduces fine-tuned pre-trained distance to retain knowledge.