北京大学 PEKING UNIVERSITY

# Understanding and Enforcing Weight Disentanglement in Task Arithmetic

Shangge Liu[1] , Yuehan Yin[1] , Lei Wang[2] , Qi Fan[1] , Yinghuan Shi[1] ,
Wenbin Li[1*] , Yang Gao[1] , Dacheng Tao[3]

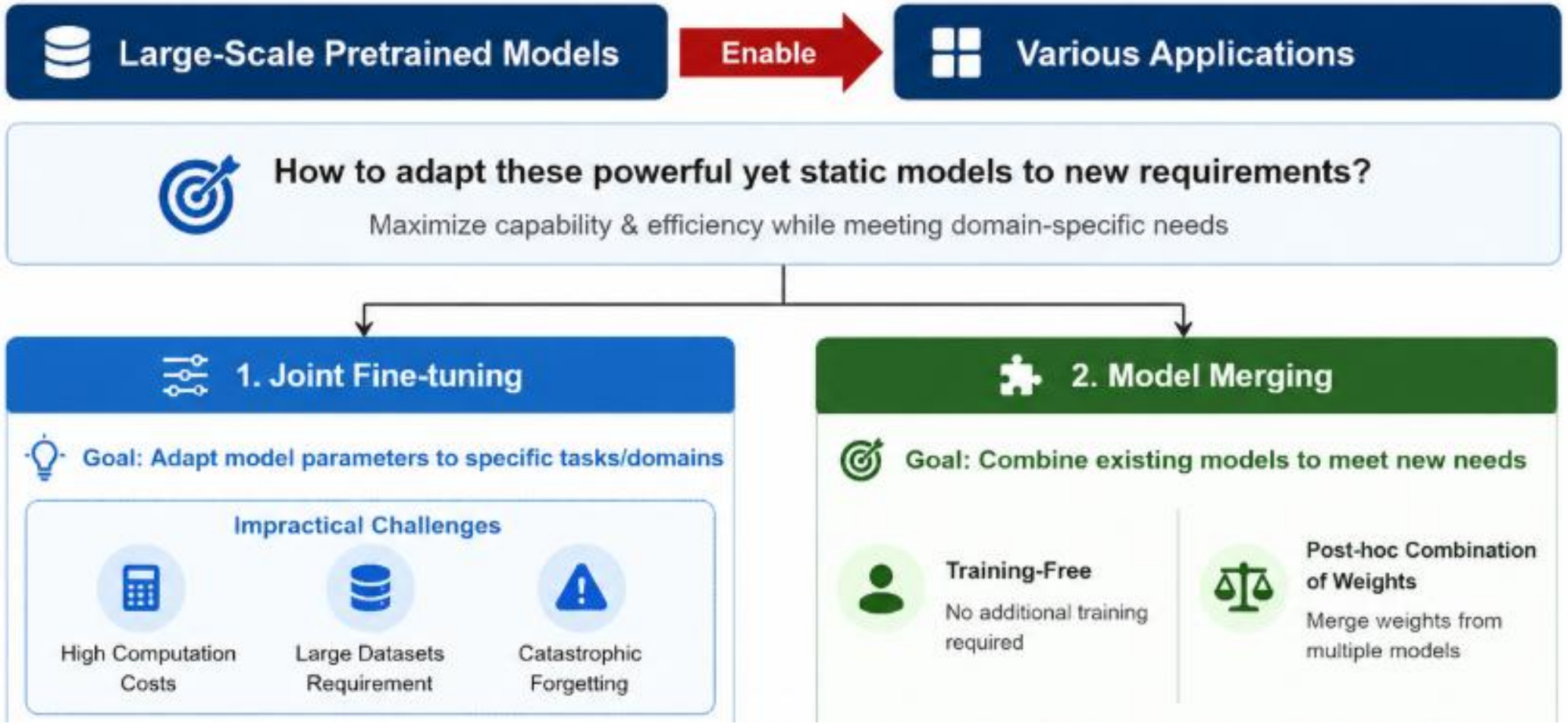
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]University of Wollongong, Australia [3]Nanyang Technological University, Singapore

CVPR 2026 (Oral)

Presenter: Jiaming Yang

2026.5.24

# Background: Model Merging and Task Arithmetic



# Background: Model Merging and Task Arithmetic

Model Merging

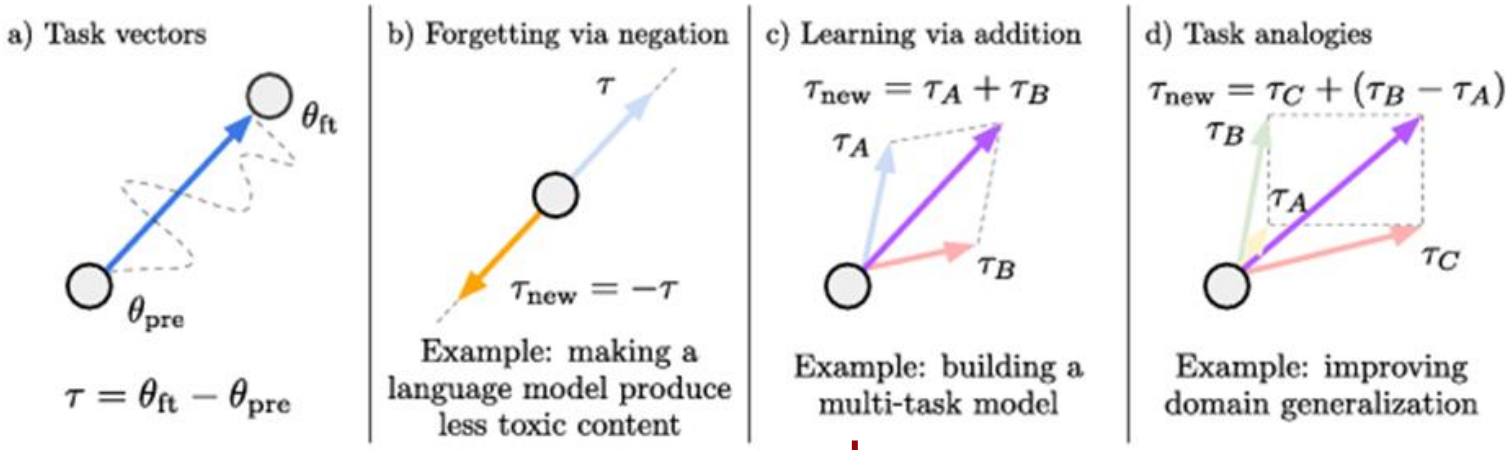
- Task arithmetic: Elegant paradigm
  - Task $t$
  - Parameter shift: $\boldsymbol{\tau_t = \theta_t - \theta_0}$
  - Add/Subtract task vectors:
    - Compose, remove, or even draw analogies between tasks

Gabriel Ilharco et al. **Editing models with task arithmetic**. *In International Conference on Learning Representations (ICLR),* 2023.

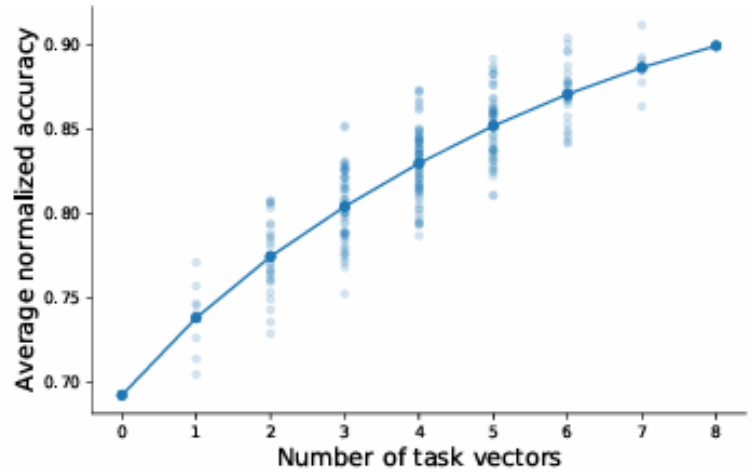
# Background: Model Merging and Task Arithmetic

Model Merging

- Task Arithmetic:




Norm Acc:
*average performance ratio of the merged multi-task model to individually fine-tuned single-task models*




Adding task vectors builds multi-task models

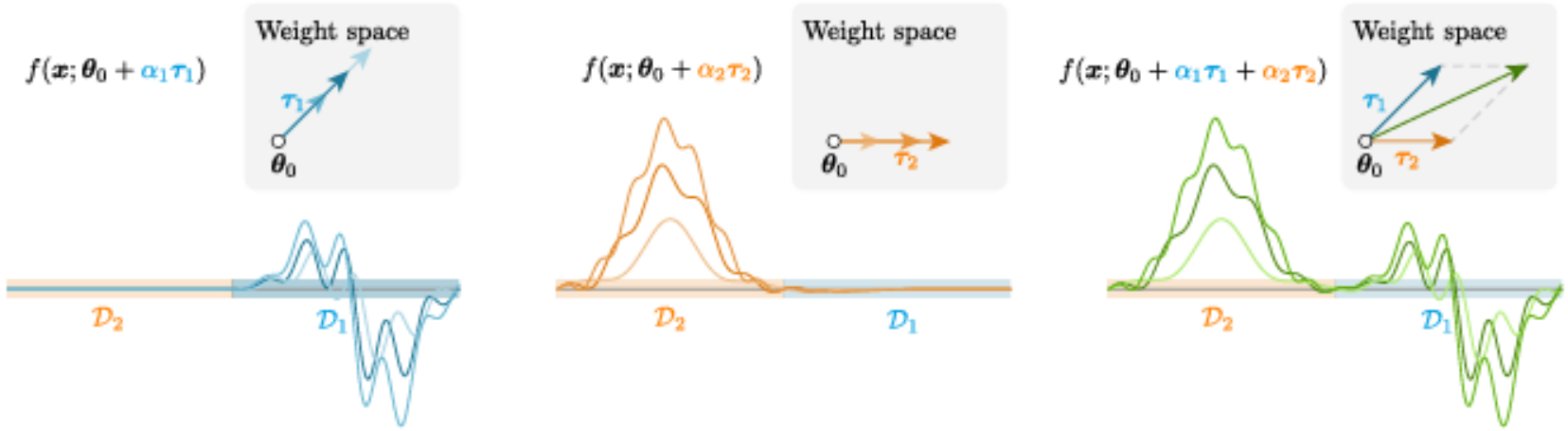
- ***Why does it Work?***

Results

# Background: Model Merging and Task Arithmetic

## Weight Disentanglement

- Task vectors are isolated to their respective data domains

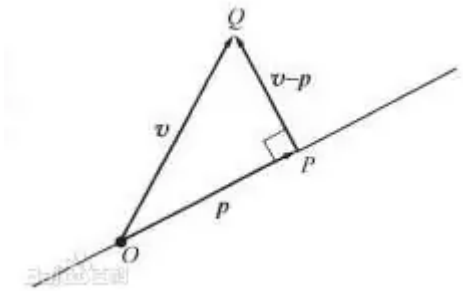


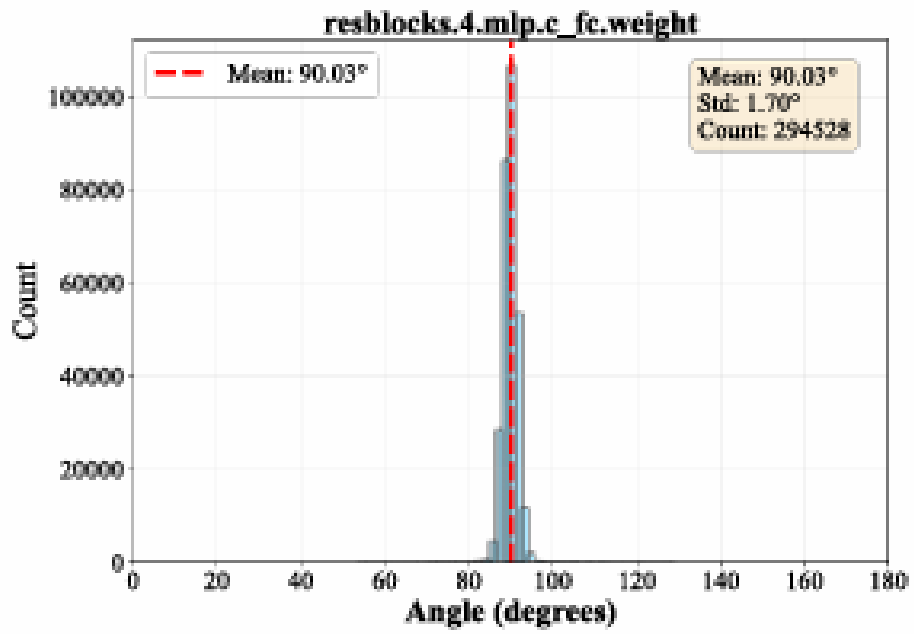

- Phenomenological Description
  - ➢ Not an explanation with properties of model/vectors

Guillermo Ortiz-Jimenez et al. **Task arithmetic in the tangent space: Improved editing of pre-trained models**. *In Advances in Neural Information Processing Systems(NeurIPS),* 2023.

# Background: Orthogonality in Neural Networks

Orthogonality



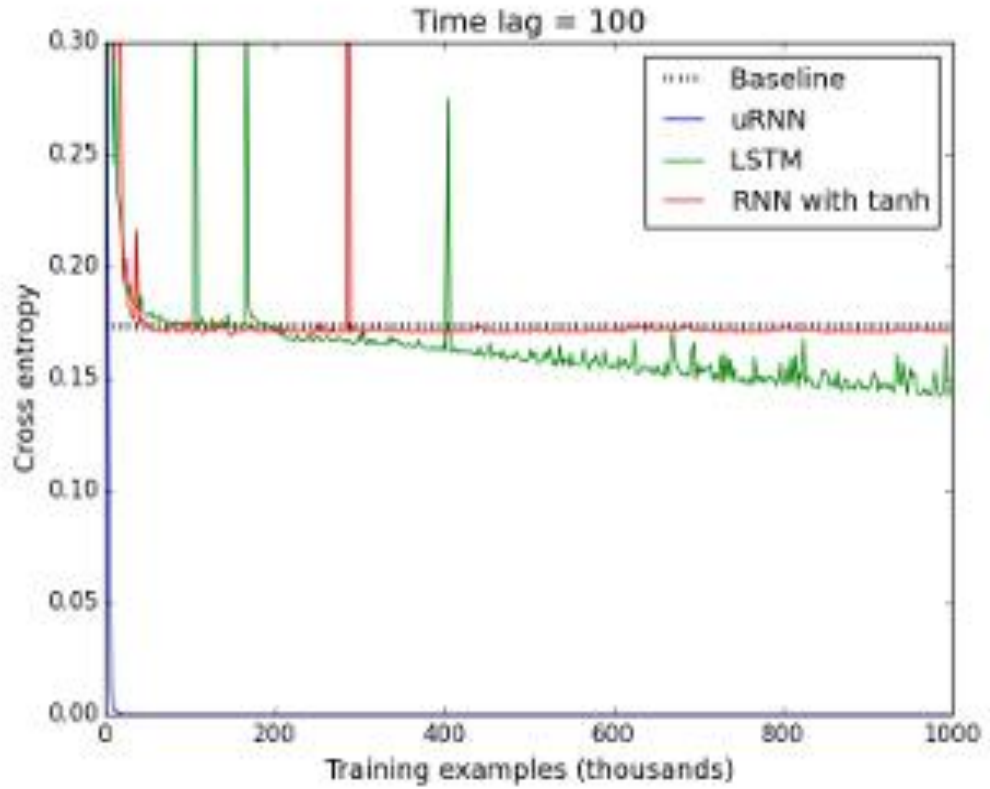




- In Neural Networks
  - ➢ Geometric Property of Weights
  - ➢ Improving training stability, generalization, and efficiency

# Background: Orthogonality in Neural Networks

Applications of Orthogonality

- In RNN
  - Addressing the issue of vanishing and exploding gradients, especially when trying to learn long-term dependencies.

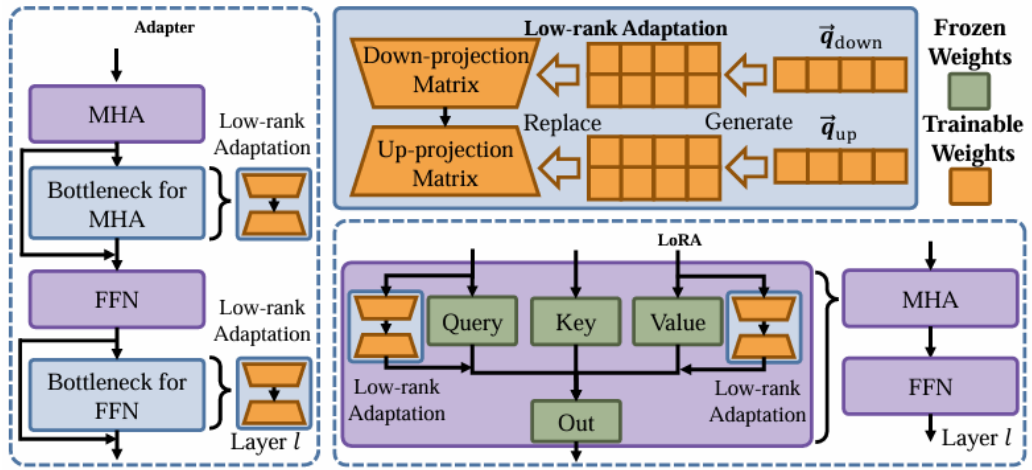


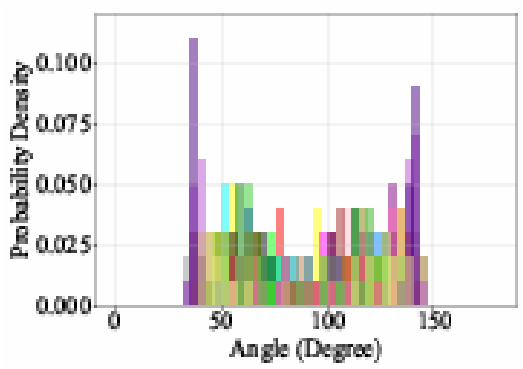

Martin Arjovsky et al, **Unitary evolution recurrent neural networks.** *In Proceedings of the 33nd International Conference on Machine Learning (ICML),* 2016.

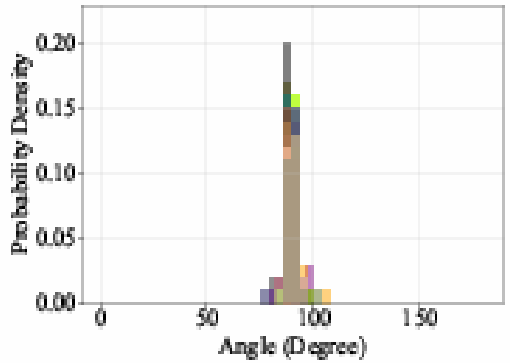
# Background: Orthogonality in Neural Networks

Applications of Orthogonality

- In PEFT
  - Preserve the pretrained semantics and concepts




Yiting Yang et al. **Efficient adaptation of pre-trained vision transformer underpinned by approximately orthogonal fine tuning strategy**. *In Proc. of IEEE/CVF International Conference on Computer Vision (ICCV), 2025*

# Before We Move on:

Remember 2 main questions in Task Arithmetic we try to answer:

1. What makes a model good for task arithmetic?
2. How to construct a good task vector?

$$\boldsymbol{\theta_t = \theta_0 + \tau_t}$$

# Preliminaries and Problem Formulation

Setup and Notation

- Initial pretrained weights: $\theta_0$
- New weights: $\theta_t^*$
- Task vector: $\tau_t = \theta_t^* - \theta_0$
- Model combination: $\theta_{MT} = \theta_0 + \sum_{t=1}^{T} \alpha_t \tau_t$
  - $\alpha_t$ denotes scalar coefficient of each task

# Preliminaries and Problem Formulation

## Weight Disentanglement Property

- *Def*:

**Definition 1** (Weight Disentanglement). *A model $f$ satisfies weight disentanglement at $\theta_0$ with respect to a set of tasks $\mathcal{T}$ with data domains $\{\mathcal{D}_t\}_{t=1}^{T}$ if, for any set of scalar coefficients $\{\alpha_t\}_{t=1}^{T}$, the following relationship holds,*

$$f(x;\theta_0+\sum_{t=1}^{T}\alpha_t\tau_t)=\begin{cases} f(x;\theta_0+\alpha_i\tau_i), & \text{if } x\in\mathcal{D}_i \\ f(x;\theta_0). & \text{if } x\notin\bigcup_{t=1}^{T}\mathcal{D}_t \end{cases}$$

- Clarify:
  - In-domain Data → Task Vector
  - Out-of-domain → Pre-trained Behavior

# Preliminaries and Problem Formulation

## Neural Tangent Kernel Linearization Hypothesis

- *Hypothesis*:

  which approximates the model's output for a small parameter change $\tau$ with a first-order Taylor expansion around $\theta_0$,

  $$f(x;\theta_0+\tau) \approx f(x;\theta_0) + \tau^\top \nabla_\theta f(x;\theta_0). \quad (4)$$

  Here, $J(x) := \nabla_\theta f(x;\theta_0)$ is the Jacobian of the model's output with respect to its parameters.

- Clarify:
  - Small parameter update → tangent kernel

# Preliminaries and Problem Formulation

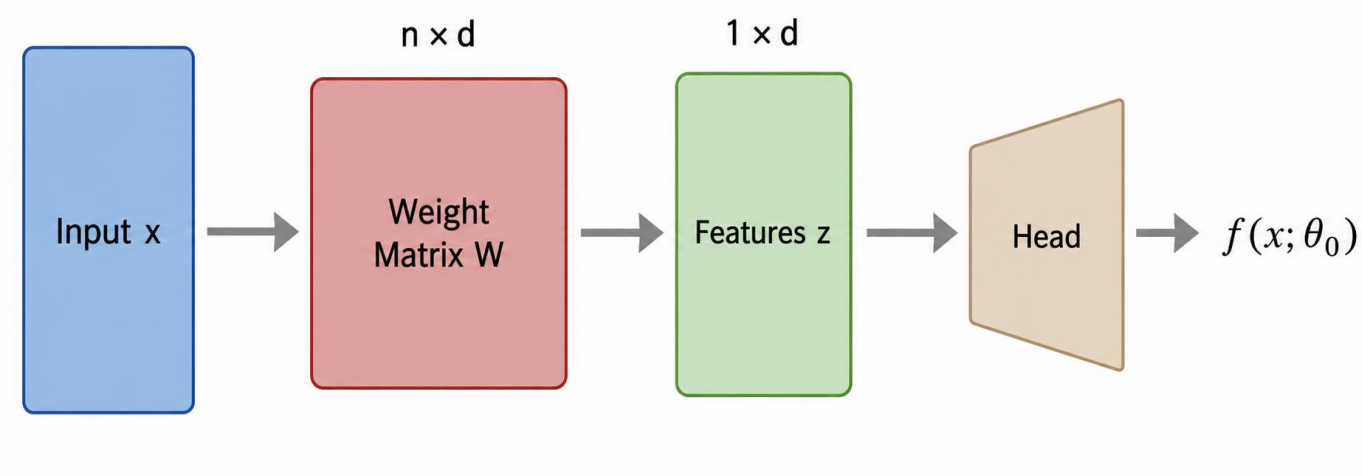
## Task-Feature Specialization

*Assumption:*

- $I_j \cap I_t = \emptyset$

  *j, t* denotes tasks, *I* marks *set*

**Definition 2** (Task-Specialized Feature Set). *For a given linear layer with weight matrix $W$, we consider each column vector $\{w_k\}_{k=1}^{d}$ as extracting a "base feature" whose activation is $z_k$. For a task $t$ with data domain $\mathcal{D}_t$, we define its specialized feature set $I_t \subseteq \{1, \ldots, d\}$ as the set of indices for which the model's final output $f(x; \theta_0)$ is sensitive to the activation $z_k$ for inputs $x \in \mathcal{D}_t$. Formally, for any $k \notin I_t$, we have $\mathbb{E}_{x \sim \mathcal{D}_t}[|\frac{\partial f(x;\theta_0)}{\partial z_k}|] = 0$.*




E.g.: d=5
Classification:{1,2}
Segmentation:{3,4,5}

# Preliminaries and Problem Formulation

Weight Vector Orthogonality

Column Vectors are mutually orthogonal

A) Block Orthogonality:

- Any two vectors from different sets are orthogonal

B) Column-wise Orthogonality:

- All pairs of distinct column vectors are orthogonal

# Preliminaries and Problem Formulation

An Equivalent Condition for Disentanglement

$$f(x;\theta_0+\tau_t+\tau_j) \;=\; f(x;\theta_0+\tau_t), \qquad \forall\, x \in \mathcal{D}_t.$$

- $t, j$ denotes two different tasks

Then introduce the NTK Linearization Hypothesis:

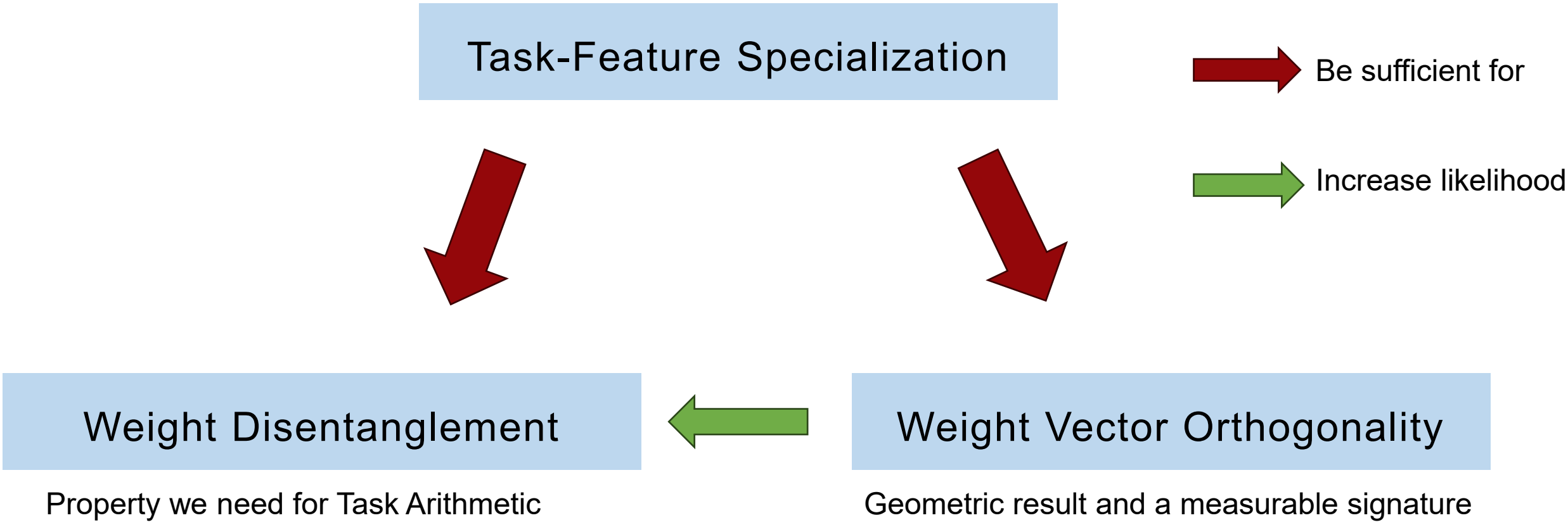
- *Lemma:*

$$\tau_j^{\top} J(x) = 0, \quad \forall x \in \mathcal{D}_t.$$

- An identical representation, $t, j$ denotes two tasks, $J(x)$ denotes Jacobian Matrix of input $x$

# Framework and Theory

Our ideal assumption and theoretical basis



# Framework and Theory

TFS → WD

- *Thm:*

$$\tau_j^\top J(x) = 0, \quad \forall x \in \mathcal{D}_t.$$

**Expanding for a linear layer**

$$\text{Interference}_W(x) = \sum_{k=1}^{d} \langle (\tau_j)_k, \nabla_{w_k} f(x; \theta_0) \rangle,$$

- To proof that every term in this summation is approximately zero;
- $(\tau_j)_k$ denotes weights update on the *k*-th column for task *j*
- $\nabla_{w_k} f(x; \theta_0)$ denotes gradient of the model output

# Framework and Theory

TFS ⟹ WD

- Analysis of the gradient term
  - Chain Rule:

$$\nabla_{w_k} f(x;\theta_0) = \frac{\partial f(x;\theta_0)}{\partial w_k} = \boxed{\frac{\partial f(x;\theta_0)}{\partial z_k}} \cdot \frac{\partial z_k}{\partial w_k}.$$

  - $\forall x \in D_t$:

From TFS set definition

$$k \notin I_t \implies \nabla_{w_k} f(x;\theta_0) \approx 0.$$

# Framework and Theory

TFS  WD

- Analysis of the task vector term
  - Based on TFS *Assumption:*

$$(\tau_j)_k = w_k^{(S)} - w_k^{(0)} = \sum_{s=0}^{S-1} \left( w_k^{(s+1)} - w_k^{(s)} \right)$$

$$= -\eta \sum_{s=0}^{S-1} \mathbb{E}_{x \sim \mathcal{D}_j} \left[ \nabla_{w_k} \mathcal{L}_j(x; \theta^{(s)}) \right].$$

$$\mathbb{E}_{x \sim \mathcal{D}_j} \left[ \nabla_{w_k} \mathcal{L}_j(x; \theta^{(s)}) \right]$$

$$= \mathbb{E}_{x \sim \mathcal{D}_j} \left[ \underbrace{\frac{\partial \mathcal{L}_j}{\partial f(x; \theta^{(s)})}}_{\text{non-zero,bounded}} \cdot \underbrace{\frac{\partial f(x; \theta^{(s)})}{\partial z_k}}_{\text{Expectation} \approx 0} \cdot \underbrace{\frac{\partial z_k}{\partial w_k}}_{\text{non-zero,bounded}} \right]$$

    - $S$ stands for training steps
  - Thus*:*

$$k \notin I_j \implies (\tau_j)_k \approx 0.$$

# Framework and Theory

TFS  WD

- $I_j \cap I_t = \emptyset$

  Case A: $k \notin I_t \implies \nabla_{w_k} f(x;\theta_0) \approx 0.$

  Case B: $k \notin I_j \implies (\tau_j)_k \approx 0.$

- Thus: $\text{Interference}_W(x) = \sum_{k=1}^{d} \langle (\tau_j)_k, \nabla_{w_k} f(x;\theta_0) \rangle,$

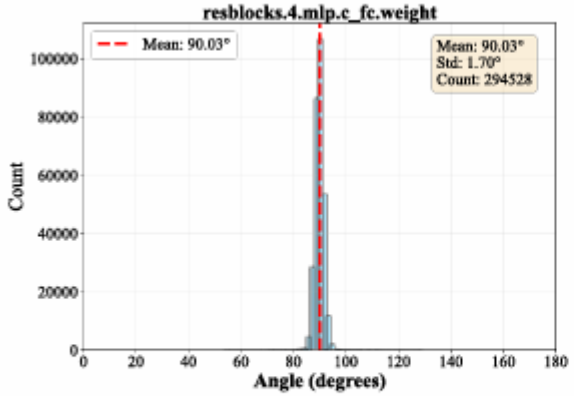
  is approximately zero

# Framework and Theory

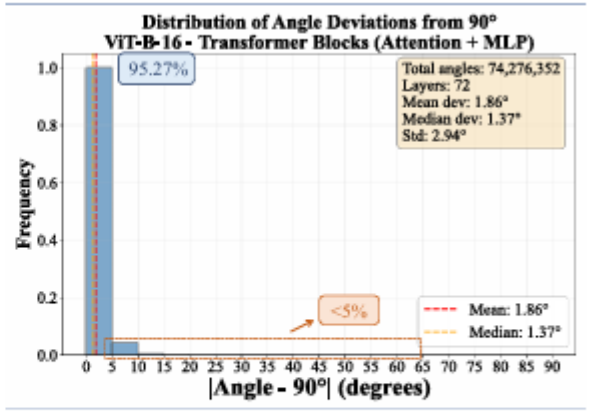
TFS → WVO

*Corollary:*

- TFS leads to Block Orthogonality
- Empirical Evidence:




(a) The distribution of angles between column vector pairs in a weight matrix.




(b) Statistical summary of angular deviations from 90° across all linear layers of the model.

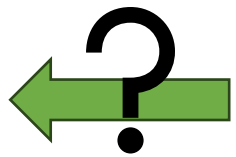
Column-wise Orthogonality is shown:
By decorrelating features within the same task.

# Framework and Theory

**What makes a pretrained model good for Task Arithmetic?**

**TFS property gives a sufficient condition**

- However, TFS is an abstract property, WVO provides a concrete, measurable signature

Weight Disentanglement  Weight Vector Orthogonality

# Framework and Theory

Relationships between TFS, WVO and WD

➢ Bayesian Analysis

- Event

  A:TFS

  B:WD

  C:WVO

- Claim: $P(B|C) > P(B)$

# Framework and Theory

Relationships between TFS, WVO and WD

- *Proof sketch*:

$$P(B|A) = 1 \quad \text{and} \quad P(C|A) = 1.$$

$$P(B|C) = \underbrace{P(B|A,C)}_{1} P(A|C) + \underbrace{P(B|\neg A,C)}_{q} P(\neg A|C).$$

$$P(B) = P(B|A)P(A) + P(B|\neg A)P(\neg A).$$

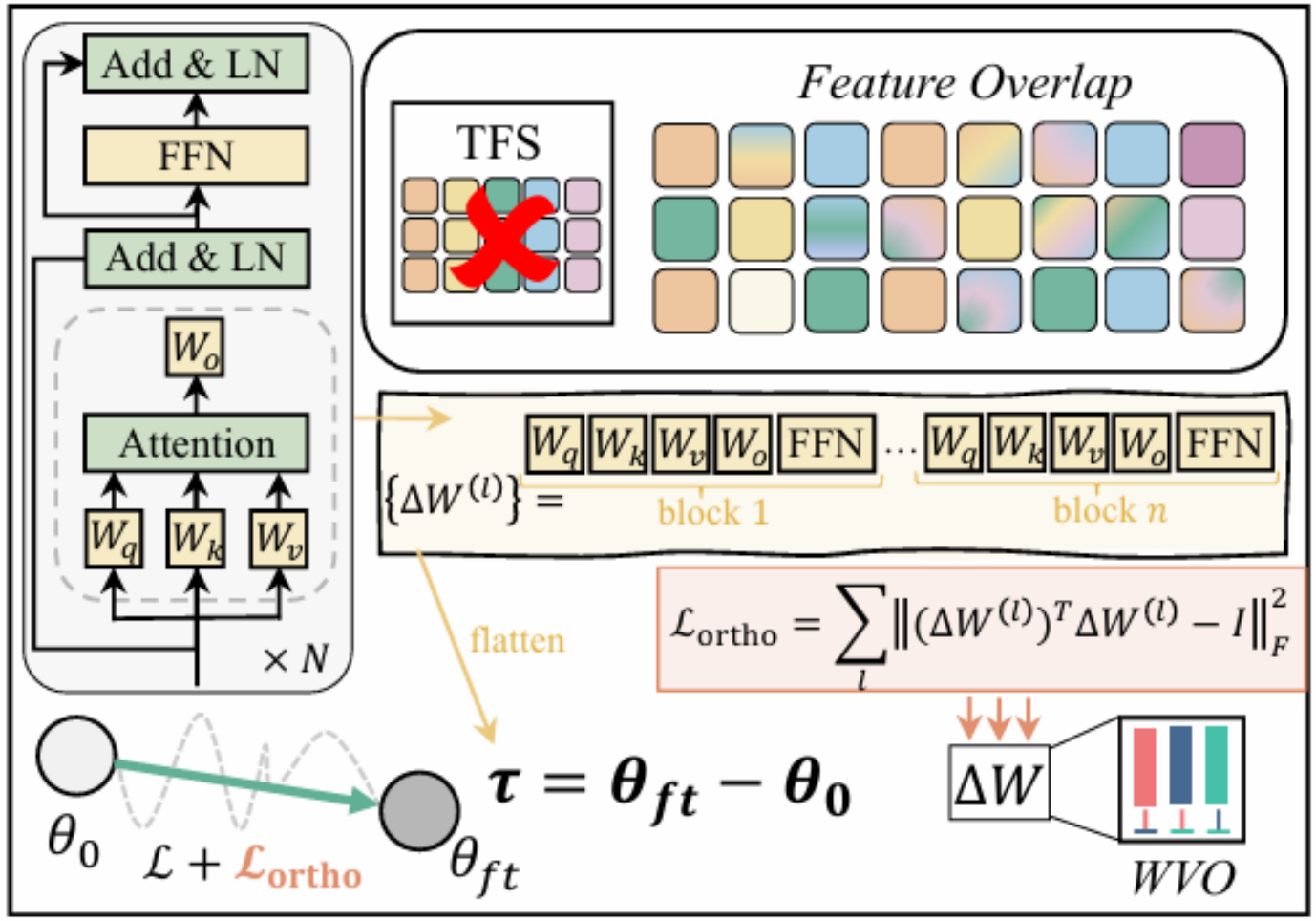
$$P(B|\neg A, C) \approx P(B|\neg A) = q$$

$$P(A|C) > P(A)$$

# Methods

Overview

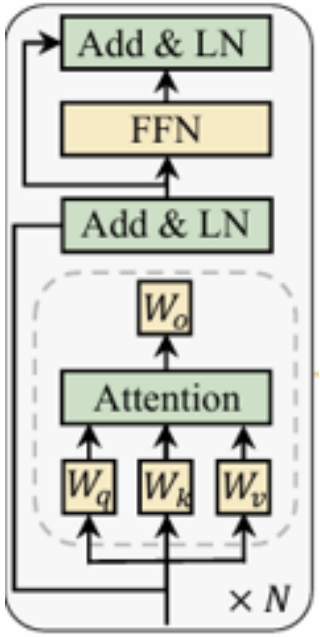
- Enforcing column-wise orthogonality on fine-tuning weight updates (ΔW) through a simple regularization
- Promote weight disentanglement in a plug-and-play manner.



# Methods

Challenges of Feature Overlap

- TFS is an ideal property, there exists shared features between tasks in real models
- **How can we actively construct good task vectors that promote disentanglement?**




# Methods

Orthogonal Regularizations

- *Def:*
  - The sum of penalties over all tuned linear layers

$$\mathcal{L}_{ortho}(\Delta\theta) = \sum_{l} \|(\Delta W^{(l)})^{\top} \Delta W^{(l)} - I\|_F^2, \qquad \mathcal{L} = \mathcal{L}_{\text{task}}(\theta_0 + \Delta\theta) + \lambda \cdot \mathcal{L}_{\text{ortho}}(\Delta\theta),$$

  - Thus making a better task vector:

# Methods

Theoretical Justification

- *Proof*
  - Reframe Interference

Bounded To be zero

$$|\tau_j^\top J(x)| \approx ||\tau_j||_2 \cdot ||J(x)||_2 \cdot |\cos \angle(\tau_j, \tau_t)|.=0?$$

  - Regularizer:
    - Norm Control
    - Angle Control

# Experiments

Settings

- Benchmark
  - 8 diverse image classification datasets
- Model
  - CLIP-pretrained VisionTransformer
- Tasks
  - Addition
  - Negation

# Experiments

Settings

- Metrics
  - Abs.Acc $\text{Abs.Acc.} = \frac{1}{T}\sum_{t=1}^{T}\text{acc}(\theta_{MT}, \mathcal{D}_t)$
  - Norm.Acc $\text{Norm.Acc.} = \left(\frac{1}{T}\sum_{t=1}^{T}\frac{\text{acc}(\theta_{MT}, \mathcal{D}_t)}{\text{acc}(\theta_t^*, \mathcal{D}_t)}\right) \times 100\%$
  - Tar.Acc minimum achievable accuracy on the target (forgetting) task, averaged across all target tasks
  - Con.Acc retained accuracy on control task (e.g., ImageNet) during task negation, serving as a safeguard to measure how much of the pre-trained model's general capability is preserved while forgetting the target task

# Experiments

## Task Addition

- Quantitative Results

| Method | ViT-B-32, 8 tasks | | ViT-B-16, 8 tasks | | ViT-L-14, 8 tasks | |
|---|---|---|---|---|---|---|
| | Abs.Acc.(↑) | Norm.Acc. (↑) | Abs.Acc.(↑) | Norm.Acc. (↑) | Abs.Acc.(↑) | Norm.Acc. (↑) |
| zero-shot | 47.74 | / | 54.22 | / | 64.54 | / |
| Non-linear Finetuning [16] | 70.32 | 77.56 | 75.39 | 75.39 | 84.07 | 89.19 |
| Non-lin. FT+**OrthoReg** (ours) | **73.41** | **93.93** | **77.68** | **93.62** | **88.23** | **100.08** |
| Δ | +3.09 | +16.37 | +2.29 | +18.23 | +4.16 | +10.89 |
| Tangent Task Arithmetic [32] | 74.68 | 85.27 | 78.97 | 87.48 | 86.19 | 93.14 |
| TTA+**OrthoReg** (ours) | **76.35** | **91.81** | **79.85** | **88.02** | **87.52** | **96.44** |
| Δ | +1.67 | +6.54 | +0.88 | +0.54 | +1.33 | +3.30 |
| Attention-Only Fine-tuning [19] | 78.07 | 86.99 | 80.71 | 87.64 | 87.81 | 93.59 |
| ATT-FT+**OrthoReg** (ours) | **80.87*** | **99.76** | **83.37*** | **98.77** | **90.41*** | **100.05** |
| Δ | +2.80 | +12.77 | +2.66 | +11.13 | +2.60 | +6.46 |
| LoRA-ATT | 73.84 | 84.29 | 75.51 | 83.17 | 87.02 | 93.33 |
| LoRA-ATT+**OrthoReg** (ours) | **76.00** | **86.10** | **79.67** | **87.70** | **89.16** | **95.50** |
| Δ | +2.16 | +1.81 | +4.16 | +4.53 | +2.14 | +2.17 |

# Experiments

## Task Negation

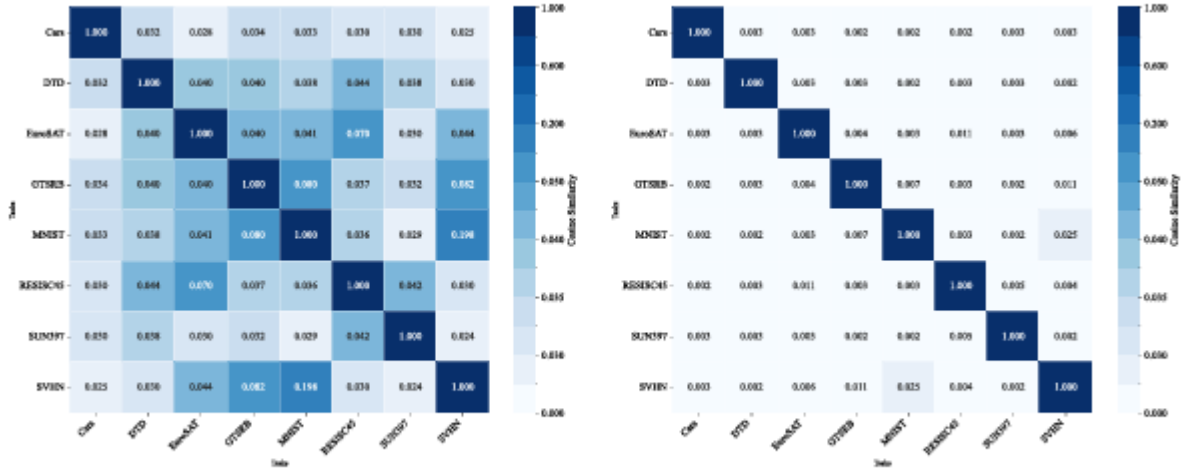
- Quantitative Results

| Method | ViT-B-32, 8 tasks | | ViT-B-16, 8 tasks | | ViT-L-14, 8 tasks | |
|---|---|---|---|---|---|---|
| | Tar.Acc.(↓) | Con.Acc. (↑) | Tar.Acc.(↓) | Con.Acc. (↑) | Tar.Acc.(↓) | Con.Acc. (↑) |
| zero-shot | 47.74 | 66.70 | 54.22 | 68.34 | 64.54 | 77.44 |
| Non-linear Finetuning [16] | 25.05 | 63.91 | 20.29 | 66.38 | 18.09 | 74.39 |
| Non-lin. FT+**OrthoReg** (ours) | **18.55** | 64.07 | **19.51** | 67.42 | **16.33** | 75.39 |
| Δ | -6.50 | +0.16 | -0.78 | +1.04 | -1.76 | +1.00 |
| Tangent Task Arithmetic [32] | 11.47 | 63.99 | 9.33 | 66.82 | 8.36 | 74.39 |
| TTA+**OrthoReg** (ours) | **11.39*** | 64.07 | **7.49*** | 66.73 | **8.36*** | 74.87 |
| Δ | -0.06 | +0.08 | -1.84 | -0.09 | +0.00 | +0.48 |
| Attention-Only Fine-tuning [19] | 19.39 | 64.90 | 19.20 | 67.75 | 24.85 | 76.42 |
| ATT-FT+**OrthoReg** (ours) | **15.67** | 64.16 | **14.78** | 66.81 | **14.67** | 75.40 |
| Δ | -3.72 | -0.74 | -4.42 | -0.94 | -10.18 | -1.02 |
| LoRA-ATT | 20.10 | 64.51 | 19.44 | 67.28 | 22.17 | 75.81 |
| LoRA-ATT+**OrthoReg** (ours) | **19.19** | 64.43 | **17.25** | 67.08 | **13.94** | 74.45 |
| Δ | -0.91 | -0.08 | -2.19 | -0.20 | -8.23 | -1.36 |

# Experiments

## Validation of Inter-Task Orthogonality

- Cosine similarity between different task vectors



(a) Non-lin. FT (b) Non-lin. FT+OrthoReg

# Further Discussion

## Comparison and Connection with Previous Work

| | **OrthoReg (Our Method)** | **TTA** |
|---|---|---|
| **Core Idea** | Explicit Orthogonality Enforcement | Implicitly Constrained to Tangent Space |
| **Operation Target** | Column Vectors of Weight Update $\Delta W$ | Entire Task Vectors |
| **Implementation** | Regularization Term | Tangent Space Projection |
| **Common Essence** | Orthogonal Task Vectors → Minimal Interference | |
| **Key Difference** | Simple & Universal, Compatible with Any Fine-tuning | High Computational Cost |

| **Fine-tuning Method** | **Total Params (M)** | **Trainable Params (M)** | **Training Time (Min)** | **Peak GPU Mem (MB)** | **Abs. Acc. (%)** |
|---|---|---|---|---|---|
| *Full Fine-tuning Methods* | | | | | |
| Non-lin. FT [16] (Baseline) | 342.56 | 342.56 | 158.21 | 42589.22 | 84.07 |
| TTA [32] (Linearized) | 685.12 | 342.56 | 280.86 | 68031.34 | 86.19 |
| Non-lin. FT + OrthoReg (ours) | 342.56 | 342.56 | 177.04 | 44500.27 | 88.23 |

# Further Discussion

**Would it still WORK on different downstream tasks?**

- Current methods using Task Arithmetic still focus on the benchmark with the same image classification task on 8 different datasets
- For different downstream tasks?
  - Would orthogonality in all layers become a too strong constraint?
  - Hierarchical orthogonality may constitutes a potential solution.
  - Frozen backbone and Task-Arithmetic adapter/LoRA may offer another promising avenue.

# Further Discussion

## Gaps between the theory and the method

- In the theoretic framework part:
  - Weight Vector Orthogonality means orthogonality between two **complete** column vectors

- In the method pipeline:
  - We use the penalty to constraint orthogonality between **weight updates**, then to make task vectors orthogonal mutually

# Takeaways

**Task Arithmetic gives another route to model fine-tuning**

- Weight Disentanglement： Desired property
- Task Feature Specialization: The common cause
- Weight Vector Orthogonality: geometric consequence, also a measurable signature
- PTMs with the TFS property are proved to fit task arithmetic

**The Application of Orthogonality**

- Orthogonal task vectors better qualified for task arithmetic paradigm

北京大学
PEKING UNIVERSITY

# Thanks for listening!