Understanding and Enforcing Weight Disentanglement in Task Arithmetic

Supplementary Material

A. Note on the Scope of Analysis: Why Focus on Linear Layers

Throughout our theoretical analysis, we primarily focus on the parameters of linear layers, such as fully-connected (FC) layers and the projection matrices within attention mechanisms. We omit biases and parameters from normalization layers (*e.g.*, LayerNorm).

This simplification is well-justified, as linear layers constitute the vast majority of parameters [12, 25] in modern large-scale models like Transformers [20], and their behavior consequently dictates the model's overall functional transformations and capacity for learning task-specific knowledge. Moreover, this focus aligns with established practices in the model merging literature, where complex strategies are often applied exclusively to linear layers [9, 16, 22], suggesting a secondary role for biases and normalization parameters in the task interference phenomena we aim to mitigate. The centrality of these layers is further underscored by the success of parameter-efficient fine-tuning (PEFT) methods like LoRA [7], which demonstrate that model adaptation for new tasks primarily occurs within these linear components.

Given this convergence of evidence, concentrating our geometric analysis on linear layers allows us to build a tractable yet powerful theoretical framework that captures the core mechanisms of task arithmetic.

B. Justification for Two-Task Simplification

In our main analysis (Section 4.1), we simplify the full definition of weight disentanglement (Definition 1) to a two-task, in-domain scenario as,

$$f(x;\theta_0+\tau_t+\tau_j) \;=\; f(x;\theta_0+\tau_t), \qquad \forall\, x\in\mathcal{D}_t. \quad (1)$$

This appendix provides a detailed justification for why this simplification is sufficient and does not result in a loss of generality. Our simplification is reasonable for two primary reasons.

First, our subsequent proofs focus on demonstrating that the pairwise interference term $\tau_j^\top J(x)$ is approximately zero for any x in the data domain $\mathcal{D}_t$ of a different task t. This is the core of the disentanglement mechanism under the NTK linearization hypothesis. Due to the linearity of this interference term with respect to the task vectors, proving the disappearance of pairwise interference is sufficient for the general multi-task case. Specifically, if $\tau_j^\top J(x)\approx 0$ for all $j\neq t$, then the total interference from all other tasks in the merged model also vanishes,

$$\sum_{j\neq t}\alpha_j(\tau_j^\top J(x)) \approx \sum_{j\neq t}\alpha_j\cdot 0 = 0. \quad (2)$$

Therefore, focusing on two-task interaction $f(x;\theta_0+\tau_t+\tau_j)$ and omitting the scaling coefficients α during the proof does not compromise the generality of our conclusions.

Second, our analysis concentrates on the "in-domain disentanglement" condition because it addresses the central challenge of eliminating crosstalk between actively composed tasks. The"out-of-domain preservation" condition, $f(x;\theta_0+\sum_{t=1}^T\alpha_t\tau_t)=f(x;\theta_0)$ for $x\notin\bigcup_{t=1}^T\mathcal{D}_t$, can be established using the same underlying logic. For an out-of-domain sample x_{ood}, its processing should ideally not rely on the specialized features of any task t. This implies that the interference term $\tau_t^\top J(x_{ood})$ should be approximately zero for all task vectors τ_t. This is a direct extension of the principle we prove for the in-domain case. By establishing the core argument for pairwise in-domain disentanglement, we effectively provide the necessary and sufficient reasoning to prove the full weight disentanglement property.

C. Proof of Lemma 1

In this part, we provide the detailed proof for Lemma 1, which establishes the equivalence between the functional property of weight disentanglement and a geometric orthogonality condition under the NTK linearization hypothesis.

Lemma 1. *Under the NTK linearization hypothesis, weight disentanglement between tasks t and j is equivalent to the interference term from task j being approximately zero on the data domain of task t:*

$$\tau_j^\top J(x)=0, \quad \forall x\in\mathcal{D}_t. \quad (3)$$

Proof. Our starting point is the simplified, two-task definition of weight disentanglement, which states that for any input x from the data domain of task t, the following approximation should hold:

$$f(x;\theta_0+\tau_t+\tau_j)=f(x;\theta_0+\tau_t), \quad \forall x\in\mathcal{D}_t. \quad (4)$$

We apply the first-order Taylor approximation from the NTK hypothesis to both sides of this equation.

For the left-hand side (LHS), the total parameter perturbation from the pre-trained state θ_0 is $(\tau_t+\tau_j)$. The linearization is therefore,

$$\begin{aligned}\text{LHS} &\approx f(x;\theta_0)+(\tau_t+\tau_j)^\top J(x)\\ &= f(x;\theta_0)+\tau_t^\top J(x)+\tau_j^\top J(x). \end{aligned} \quad (5)$$

For the right-hand side (RHS), the perturbation is simply τ_t. The linearization is,

$$\text{RHS} \approx f(x;\theta_0) + \tau_t^\top J(x). \tag{6}$$

By substituting these approximations from Equation (5) and Equation (6) back into the original weight disentanglement condition (Equation (4)), we obtain,

$$f(x;\theta_0) + \tau_t^\top J(x) + \tau_j^\top J(x) \approx f(x;\theta_0) + \tau_t^\top J(x). \tag{7}$$

Canceling the common terms $f(x;\theta_0)$ and $\tau_t^\top J(x)$ from both sides of the approximation leaves us with the final, equivalent condition:

$$\tau_j^\top J(x) = 0, \quad \forall x \in \mathcal{D}_t. \tag{8}$$

This shows that, under NTK linearization, the functional requirement that task j does not interfere with task t is equivalent to the geometric condition that the task vector τ_j is orthogonal to the model's gradient Jacobian $J(x)$ for all data points x in the domain of task t. □

D. Detailed Proof of Theorem 1

D.1. Proof of Theorem 1

In this section, we provide the formal proof for Theorem 1.

Theorem 1. *Under the NTK linearization hypothesis (Section 3.3) and the Task-Feature Specialization property, weight disentanglement between tasks t and j holds.*

Proof. According to Lemma 1, our goal is to prove that the interference term $\tau_j^\top J(x)$ is approximately zero for any $x \in \mathcal{D}_t$. We can decompose this total interference into contributions from each linear layer. For clarity, we analyze the interference arising from a single weight matrix $W \in \mathbb{R}^{m\times d}$ and show it is zero. The conclusion generalizes to the entire model by summation.

The interference contributed by W is $\langle (\tau_j)_W, J_W(x)\rangle$, where $(\tau_j)_W$ and $J_W(x)$ are the components of the task vector and Jacobian corresponding to W. By decomposing this along the column vectors $\{w_1, \ldots, w_d\}$ of W, we get,

$$\text{Interference}_W(x) = \sum_{k=1}^{d} \langle (\tau_j)_k, \nabla_{w_k} f(x;\theta_0)\rangle, \tag{9}$$

where $(\tau_j)_k$ is the update applied to column w_k. We will show every term in this summation is approximately zero.

Analysis of the gradient term ($\nabla_{w_k} f(x;\theta_0)$). For an input $x \in \mathcal{D}_t$, the gradient of the model output with respect to a weight column w_k can be expressed using the chain rule,

$$\nabla_{w_k} f(x;\theta_0) = \frac{\partial f(x;\theta_0)}{\partial w_k} = \frac{\partial f(x;\theta_0)}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_k}. \tag{10}$$

According to Definition 2, if the feature index k is not in the specialized set for task t (*i.e.*, $k \notin I_t$), the model's output is insensitive to it, meaning $\frac{\partial f(x;\theta_0)}{\partial z_k} \approx 0$. For $x \in \mathcal{D}_t$,

$$k \notin I_t \implies \nabla_{w_k} f(x;\theta_0) \approx 0. \tag{11}$$

Analysis of the task Vector term ($(\tau_j)_k$). The task vector component $(\tau_j)_k$ is the accumulated update to weight w_k from fine-tuning on task j. By definition, if feature k is not specialized for task j (*i.e.*, $k \notin I_j$), the loss function for task j is insensitive to it. This means the gradients with respect to w_k computed on the data domain $\mathcal{D}_j$ are consistently negligible. Since $(\tau_j)_k$ is the sum of these negligible gradients, it will be approximately zero. (A detailed proof is provided in Appendix D.2 as Proposition 1).

$$k \notin I_j \implies (\tau_j)_k \approx 0. \tag{12}$$

Now, we examine each term $\langle (\tau_j)_k, \nabla_{w_k} f(x;\theta_0)\rangle$ in the summation for index $k \in \{1, \ldots, d\}$. There are two mutually exclusive possibilities.

Case A: $k \in I_j$. By the Task-Feature Specialization property ($I_t \cap I_j = \emptyset$), it must be that $k \notin I_t$. From gradient analysis (Equation (11)), this implies $\nabla_{w_k} f(x;\theta_0) \approx 0$.

Case B: $k \notin I_j$. From task vector analysis (Equation (12)), this implies $(\tau_j)_k \approx 0$.

In both cases, the term $\langle (\tau_j)_k, \nabla_{w_k} f(x;\theta_0)\rangle$ vanishes. Since this holds for all k, the interference from this layer, $\text{Interference}_W(x)$, is approximately zero. As this applies to all layers, the total interference $\tau_j^\top J(x) \approx 0$. By Lemma 1, this proves that weight disentanglement holds. □

D.2. Supporting Proposition for Theorem 1

In this part, we provide a detailed proof for the proposition referenced in the proof of Theorem 1. This proposition formalizes the intuition that if a task does not depend on a specific feature, the fine-tuning process for that task will not significantly alter the weights associated with that feature.

Proposition 1. *Under the NTK Linearization hypothesis (Section 3.3) and the Task-Feature Specialization property, consider the fine-tuning process for task j on its data domain $\mathcal{D}_j$. If a feature index k does not belong to the specialized feature set for task j (i.e., $k \notin I_j$), then the corresponding component of the resulting task vector, $(\tau_j)_k$, is approximately zero.*

$$k \notin I_j \implies (\tau_j)_k \approx 0. \tag{13}$$

Proof. The task vector τ_j is defined as the total change in parameters after fine-tuning on task j, starting from the pre-trained weights θ_0,

$$\tau_j = \theta_j^* - \theta_0, \tag{14}$$

where θ_j^* are the final fine-tuned parameters. The component $(\tau_j)_k$ specifically represents the change in the weight column w_k of a given linear layer.

Let's model the fine-tuning process as a sequence of updates using a gradient-based optimizer, such as Stochastic Gradient Descent (SGD). For a total of S update steps, the weight column w_k is updated iteratively. The update rule for w_k at step s is,

$$w_k^{(s+1)} = w_k^{(s)} - \eta \cdot \mathbb{E}_{x\sim\mathcal{D}_j}\left[\nabla_{w_k}\mathcal{L}_j(x;\theta^{(s)})\right], \quad (15)$$

where η is the learning rate, and $\theta^{(s)}$ represents the model parameters at step s, with the initial state being $\theta^{(0)} = \theta_0$.

Consistent with the perspective from work on Adaptive Weight Disentanglement (AWD) [21] that views the task vector as the sum of accumulated gradients, the total change in the weight column w_k, which is the task vector component $(\tau_j)_k$, is the sum of all single-step updates over the course of training,

$$\begin{aligned}(\tau_j)_k = w_k^{(S)} - w_k^{(0)} &= \sum_{s=0}^{S-1}\left(w_k^{(s+1)} - w_k^{(s)}\right) \\ &= -\eta\sum_{s=0}^{S-1}\mathbb{E}_{x\sim\mathcal{D}_j}\left[\nabla_{w_k}\mathcal{L}_j(x;\theta^{(s)})\right].\end{aligned} \quad (16)$$

To prove that $(\tau_j)_k \approx 0$, we need to show that the expected gradient $\mathbb{E}_{x\sim\mathcal{D}_j}\left[\nabla_{w_k}\mathcal{L}_j(x;\theta^{(s)})\right]$ is approximately zero at every step s of the fine-tuning process.

Let's analyze the gradient for a single data point $x \in \mathcal{D}_j$ using the chain rule,

$$\nabla_{w_k}\mathcal{L}_j(x;\theta^{(s)}) = \frac{\partial\mathcal{L}_j}{\partial f(x;\theta^{(s)})}\cdot\frac{\partial f(x;\theta^{(s)})}{\partial z_k}\cdot\frac{\partial z_k}{\partial w_k}, \quad (17)$$

where z_k is the activation of the base feature corresponding to w_k. We analyze each term in this product.

- $\frac{\partial\mathcal{L}_j}{\partial f(x;\theta^{(s)})}$. This is the derivative of the loss with respect to the model's final output. Before the model has fully converged, this term is generally non-zero and bounded.
- $\frac{\partial z_k}{\partial w_k}$. For a standard linear layer where $z_k = (w_k)^\top \mathrm{In}(x)$, this derivative is simply the input to the layer, $\mathrm{In}(x)$. This term is also non-zero and bounded.
- $\frac{\partial f(x;\theta^{(s)})}{\partial z_k}$. It measures the sensitivity of the final model output to the intermediate feature activation z_k. Our core assumption is that $k \notin I_j$. By Definiton 2 (Task-Specialized Feature Set), this means that at the pre-trained state θ_0, the model's output is insensitive to z_k in expectation over the data domain $\mathcal{D}_j$,

$$\mathbb{E}_{x\sim\mathcal{D}_j}\left\|\frac{\partial f(x;\theta_0)}{\partial z_k}\right\| \approx 0. \quad (18)$$

The fine-tuning process occurs in the neighborhood of θ_0. Under the NTK linearization hypothesis, the parameter changes are small, and the model's Jacobian is assumed to be stable. Therefore, for all steps s in the fine-tuning process, $\theta^{(s)}$ remains close to θ_0, and the sensitivity of the model's output to feature z_k also remains negligible,

$$\mathbb{E}_{x\sim\mathcal{D}_j}\left\|\frac{\partial f(x;\theta^{(s)})}{\partial z_k}\right\| \approx 0 \quad \text{for } s = 0, 1, \ldots, S-1. \quad (19)$$

Now, let's take the expectation of the full gradient expression (Equation (17)) over the data domain $\mathcal{D}_j$,

$$\begin{aligned}&\mathbb{E}_{x\sim\mathcal{D}_j}\left[\nabla_{w_k}\mathcal{L}_j(x;\theta^{(s)})\right] \\ &= \mathbb{E}_{x\sim\mathcal{D}_j}\left[\underbrace{\frac{\partial\mathcal{L}_j}{\partial f(x;\theta^{(s)})}}_{\text{non-zero,bounded}}\cdot\underbrace{\frac{\partial f(x;\theta^{(s)})}{\partial z_k}}_{\text{Expectation}\approx 0}\cdot\underbrace{\frac{\partial z_k}{\partial w_k}}_{\text{non-zero,bounded}}\right].\end{aligned} \quad (20)$$

Since the expectation of the sensitivity term $\frac{\partial f}{\partial z_k}$ is approximately zero, and the other terms are bounded, the expectation of their product will also be approximately zero.

This holds for every step s of the fine-tuning process. Substituting this result back into Equation (16), we find that the total update $(\tau_j)_k$ is a finite sum of near-zero vectors,

$$(\tau_j)_k = -\eta\sum_{s=0}^{S-1}\underbrace{\mathbb{E}_{x\sim\mathcal{D}_j}\left[\nabla_{w_k}\mathcal{L}_j(x;\theta^{(s)})\right]}_{\approx 0} \approx 0. \quad (21)$$

This demonstrates that if a feature is not part of a task's specialized set, the corresponding weights will remain virtually unchanged during fine-tuning for that task.

This completes the proof.

□

E. Proof of Corollary 1

This part provides the detailed proof for Corollary 1, which establishes that the Task-Feature Specialization (TFS) property, a functional characteristic of an ideal pre-trained model, gives rise to a specific geometric structure in its parameters, namely, weight vector block-orthogonality. This formalizes the connection, that Weight Vector Orthogonality (WVO) is presented as a geometric consequence of TFS.

Corollary 1. *Given a model that adheres to the Task-Feature Specialization (TFS) property, its weight matrices will exhibit Block Orthogonality.*

E.1. TFS Implies Cross-Task Feature Decorrelation

We begin by proving a key statistical consequence of TFS, which will be instrumental in our main proof. The functional separation defined by TFS has a direct consequence on the statistical properties of the feature activations. We formalize this as the following proposition.

Proposition 2. *Under the Task-Feature Specialization (TFS) property, for any two distinct tasks $t \neq j$, and for any pair of features with indices $k \in I_t$ and $l \in I_j$, their activations z_k and z_l are approximately decorrelated over a mixed data distribution μ. That is,*

$$Cov_\mu(z_k, z_l) \approx 0. \tag{22}$$

Proof. Let us assume the contrary. Suppose TFS holds, but two features z_k (specialized for task t, *i.e.*, $k \in I_t$) and z_l (specialized for task j, *i.e.*, $l \in I_j$) are statistically correlated. For simplicity, we can model this correlation with an approximate linear relationship,

$$z_k \approx a \cdot z_l + b + \xi, \tag{23}$$

where $a \neq 0$ is a correlation coefficient, b is a bias, and ξ is uncorrelated noise. This model implies that a change in z_l systematically induces a change in z_k.

Now, consider the total derivative of the model's final output $f(x; \theta_0)$ with respect to the activation z_l for an input x from task t's data domain, $\mathcal{D}_t$. Using the chain rule, the change in f with respect to a change in z_l has two paths: a direct path ($z_l \to f$) and an indirect path through the correlated feature z_k ($z_l \to z_k \to f$).

$$\frac{df(x;\theta_0)}{dz_l} = \frac{\partial f}{\partial z_l} + \frac{\partial f}{\partial z_k}\frac{\partial z_k}{\partial z_l} \tag{24}$$

We analyze each term in the context of TFS for an input $x \in \mathcal{D}_t$.

- $\frac{\partial f}{\partial z_l}$: Since $x \in \mathcal{D}_t$ and the feature l is specialized for task j ($l \in I_j$), the TFS assumption ($I_t \cap I_j = \emptyset$) implies $l \notin I_t$. By Definition 2, the model's output is insensitive to z_l on this data domain. Thus, $\mathbb{E}_{x\sim\mathcal{D}_t}[|\frac{\partial f}{\partial z_l}|] \approx 0$.
- $\frac{\partial f}{\partial z_k}$: Since $x \in \mathcal{D}_t$ and the feature k is specialized for task t ($k \in I_t$), the model's output is sensitive to z_k. Thus, $\mathbb{E}_{x\sim\mathcal{D}_t}[|\frac{\partial f}{\partial z_k}|]$ is significantly non-zero.
- $\frac{\partial z_k}{\partial z_l}$: From our linear correlation model, this derivative is the correlation coefficient a, which we assumed to be non-zero.

Substituting these into the chain rule expression and taking the expectation over $\mathcal{D}_t$,

$$\begin{aligned}\mathbb{E}_{x\sim\mathcal{D}_t}\left|\frac{df}{dz_l}\right| &\approx \mathbb{E}_{x\sim\mathcal{D}_t}\left|\underbrace{\frac{\partial f}{\partial z_l}}_{\approx 0} + \underbrace{\frac{\partial f}{\partial z_k}}_{\text{non-zero}} \cdot \underbrace{\frac{\partial z_k}{\partial z_l}}_{\text{non-zero, } a}\right| \\ &\approx |a| \cdot \mathbb{E}_{x\sim\mathcal{D}_t}\left|\frac{\partial f}{\partial z_k}\right|.\end{aligned} \tag{25}$$

Since $|a| \neq 0$ and $\mathbb{E}[|\frac{\partial f}{\partial z_k}|]$ is significantly non-zero, the result is a significantly non-zero value. This means that the model's output f shows a non-negligible total sensitivity to the activation z_l on data from task t.

This result, however, directly contradicts the premise of TFS. If a model has truly specialized feature k for task t and feature l for task j, its function for task t should not be affected by perturbations in z_l. The total effect of z_l on the output, not just the partial derivative, should be negligible.

The contradiction arose from our initial assumption of correlation ($a \neq 0$). Therefore, that assumption must be false. We conclude that for TFS to hold, features specialized for different tasks must be statistically decorrelated. □

E.2. Detailed proof of Corollary 1

Proof. The proof proceeds by first relating the geometric property of the weight matrix ($W^\top W$) to a statistical property of the feature activations (the covariance matrix Σ_z), and then showing that TFS imposes a block-diagonal structure on this covariance matrix.

Step 1: Connecting Weight Geometry to Feature Covariance.

Consider a single linear layer with weight matrix $W = [w_1, \ldots, w_d] \in \mathbb{R}^{m\times d}$, input $\mathrm{In}(x) \in \mathbb{R}^m$, and feature activations $z = W^\top \mathrm{In}(x) \in \mathbb{R}^d$. We compute the covariance matrix Σ_z of the feature activations under a mixed data distribution μ,

$$\Sigma_z = \mathbb{E}_{x\sim\mu}[(z-\mu_z)(z-\mu_z)^\top], \quad \text{where } \mu_z = \mathbb{E}_{x\sim\mu}[z]. \tag{26}$$

In modern deep neural networks, the presence of normalization layers like Layer Normalization (LN) [3] or Batch Normalization (BN) [10] is standard practice. A primary function of these layers is to standardize the activations, dynamically regulating their mean and variance [3, 8, 10]. This forces the mean of the layer's input, $\mu_{\mathrm{In}} = \mathbb{E}_{x\sim\mu}[\mathrm{In}(x)]$, to be approximately zero.

Consequently, the mean of the output feature activations is also approximately zero,

$$\mu_z = \mathbb{E}_{x\sim\mu}[W^\top \mathrm{In}(x)] = W^\top \mathbb{E}_{x\sim\mu}[\mathrm{In}(x)] = W^\top \mu_{\mathrm{In}} \approx 0. \tag{27}$$

With this zero-mean property, the covariance matrix Σ_z simplifies to the second-moment matrix,

$$\Sigma_z = \mathbb{E}_{x\sim\mu}[zz^\top] = \mathbb{E}_{x\sim\mu}[W^\top \mathrm{In}(x)\mathrm{In}(x)^\top W]. \tag{28}$$

Since the weight matrix W is constant with respect to the input x, we can move it outside the expectation:

$$\Sigma_z = W^\top \left(\mathbb{E}_{x\sim\mu}[\mathrm{In}(x)\mathrm{In}(x)^\top]\right) W. \tag{29}$$

At this point, we analyze the term $\mathbb{E}_{x\sim\mu}[\mathrm{In}(x)\mathrm{In}(x)^\top]$, which represents the second moment matrix of the layer's input. As argued before, normalization layers standardize activations. Beyond just enforcing a zero mean, this process also regulates variance, driving the covariance matrix of the

layer's input, Σ_{In}, towards a whitened state [3, 8, 10, 19]. The covariance matrix of the input is defined as,

$$\begin{aligned}\Sigma_{\text{In}} &= \mathbb{E}_{x\sim\mu}[(\text{In}(x)-\mu_{\text{In}})(\text{In}(x)-\mu_{\text{In}})^\top] \\ &= \mathbb{E}_{x\sim\mu}[\text{In}(x)\text{In}(x)^\top] - \mu_{\text{In}}\mu_{\text{In}}^\top\end{aligned} \tag{30}$$

Given that the input is whitened, we have $\Sigma_{\text{In}} \approx I_m$ and $\mu_{\text{In}} \approx 0$. Substituting these into the definition gives us the second-moment matrix of the input,

$$\mathbb{E}_{x\sim\mu}[\text{In}(x)\text{In}(x)^\top] = \Sigma_{\text{In}} + \mu_{\text{In}}\mu_{\text{In}}^\top \approx I_m + 0\cdot 0^\top = I_m. \tag{31}$$

Substituting this result back into the expression for Σ_z, we arrive at the crucial link between the weights' geometry and the features' statistics,

$$\Sigma_z = W^\top I_m W = W^\top W. \tag{32}$$

This equation shows that in this case the Gram matrix of the weights, $W^\top W$, is identical to the covariance matrix of the feature activations, Σ_z. Proving that W has block-orthogonal columns is now equivalent to proving that its Gram matrix $W^\top W$ is block-diagonal, which in turn is equivalent to proving that Σ_z is block-diagonal.

Step 2:Proving the Block-Diagonal Structure of Σ_z.

An element $(\Sigma_z)_{kl}$ of the covariance matrix is, by definition, the covariance between z_k and z_l, *i.e.*, $(\Sigma_z)_{kl} = \text{Cov}_\mu(z_k, z_l)$.

Let's consider two distinct feature indices, $k \neq l$.

Case 1: Features are specialized for different tasks. Suppose $k \in I_t$ and $l \in I_j$ for two tasks $t \neq j$. According to Proposition 2, which we derived from the TFS property, the activations of these features are decorrelated over the mixed distribution μ. Therefore, we directly have,

$$(\Sigma_z)_{kl} = \text{Cov}_\mu(z_k, z_l) \approx 0. \tag{33}$$

Case 2: Features are specialized for the same task. Suppose $k, l \in I_t$ for some task t, with $k \neq l$. Our theory does not make any assumption about intra-task feature decorrelation. Therefore, the term $(\Sigma_z)_{kl} = \text{Cov}_\mu(z_k, z_l)$ is not guaranteed to be zero and may be non-zero in general.

Step 3: Conclusion of Block-Orthogonality

From Step 2, we have shown that the off-diagonal elements of the covariance matrix Σ_z are approximately zero whenever the indices correspond to different tasks. The elements corresponding to pairs of features within the same task may be non-zero. This means Σ_z has a block-diagonal structure,

$$\Sigma_z = W^\top W \approx \begin{pmatrix} \mathbf{B}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}_T \end{pmatrix}. \tag{34}$$

where $\mathbf{B}_t$ is the (generally non-diagonal) covariance submatrix for features whose indices are in the set I_t, and the $\mathbf{0}$ blocks represent matrices with near-zero entries.

The (k, l)-th element of the Gram matrix $W^\top W$ is the inner product of the column vectors $\langle w_k, w_l\rangle$. The block-diagonal structure of $W^\top W$ directly implies that if indices k and l belong to different blocks (*i.e.*, $k \in I_t$ and $l \in I_j$ with $t \neq j$), their corresponding entry in the Gram matrix is approximately zero,

$$\langle w_k, w_l\rangle = (W^\top W)_{kl} \approx 0. \quad \text{for } k \in I_t, l \in I_j, t \neq j \tag{35}$$

This is precisely the definition of block-orthogonality for the columns of the weight matrix W. The set of column vectors $\{w_k\}_{k\in I_t}$ forms a subspace that is orthogonal to the subspace spanned by $\{w_l\}_{l\in I_j}$ for any $j \neq t$.

This completes the proof. □

F. Bayesian Analysis of the Relationship between TFS, WVO, and WD

This part provides a formal Bayesian analysis to justify the claim made in Section 4.2.4, that observing Weight Vector Orthogonality (WVO) in a pre-trained model strongly increases our belief that it will exhibit Weight Disentanglement (WD). This analysis formalizes the intuition that WVO acts as a powerful diagnostic clue for the desirable, yet abstract, property of Task-Feature Specialization (TFS).

Let us define three distinct events.

- Event A: The model has achieved ideal Task-Feature Specialization (TFS). This represents the underlying, unobservable abstract property where the model allocates disjoint sets of internal features to different tasks.
- Event B: The model exhibits Weight Disentanglement (WD). This is the desired functional outcome where task vectors can be composed without destructive interference.
- Event C: The model's parameters possess Weight Vector Orthogonality (WVO). This is a concrete, measurable geometric property of the model's weight matrices.

Our core theory, as established in Section 4.2, posits that TFS is a sufficient condition for both WD (Theorem 1) and WVO (Corollary 1). We can formalize this as a logical implication,

$$A \implies (B \wedge C). \tag{36}$$

This means that if Event A is true, then both Event B and Event C must also be true. Consequently, we have the conditional probabilities,

$$P(B|A) = 1 \quad \text{and} \quad P(C|A) = 1. \tag{37}$$

Our goal is to demonstrate that observing WVO (Event C) provides evidence for WD (Event B). In probabilistic terms, we aim to show that the posterior probability of WD

given WVO is greater than the prior probability of WD,

$$P(B|C) > P(B). \tag{38}$$

First, we can expand the conditional probability $P(B|C)$ by conditioning on whether TFS (Event A) has occurred,

$$P(B|C) = P(B|A,C)P(A|C) + P(B|\neg A,C)P(\neg A|C). \tag{39}$$

Let's analyze the terms in this expression.

1. $P(B|A,C)$: Since Event A (TFS) is a sufficient condition for Event B (WD), if A is true, B must be true, regardless of C. Therefore, $P(B|A,C) = 1$.

2. $P(B|\neg A,C)$: This is the probability of observing WD when TFS is not present, even though WVO is. Without the foundational structure of TFS, WD is not guaranteed. It might occur due to other unknown reasons or by chance, but we can reasonably assume this probability is significantly less than 1. Let's denote this probability as q, where $0 \leq q < 1$.

Substituting these into the equation, we get,

$$P(B|C) = 1 \cdot P(A|C) + q \cdot P(\neg A|C). \tag{40}$$

Rearranging this gives,

$$P(B|C) = q + (1-q)P(A|C). \tag{41}$$

Now, we examine the crucial term $P(A|C)$, which represents our updated belief in TFS after having observed WVO. Using Bayes' theorem,

$$P(A|C) = \frac{P(C|A)P(A)}{P(C)}. \tag{42}$$

As established earlier, $P(C|A) = 1$. This simplifies the expression to,

$$P(A|C) = \frac{P(A)}{P(C)}. \tag{43}$$

Here, $P(A)$ is our prior belief that a model has achieved TFS, and $P(C)$ is the prior probability of observing WVO. WVO is a specific geometric structure that is not guaranteed to occur in any arbitrary neural network; its emergence is non-trivial. Therefore, it is safe to assume that $P(C) < 1$.

This leads to a key inequality,

$$P(A|C) = \frac{P(A)}{P(C)} > P(A). \tag{44}$$

This inequality formally captures our intuition: observing the geometric signature of WVO (Event C) strengthens our belief that the model has developed the underlying functional structure of TFS (Event A).

To complete the proof, we compare the expression for $P(B|C)$ with the unconditional prior probability of WD, $P(B)$. Using the law of total probability again,

$$P(B) = P(B|A)P(A) + P(B|\neg A)P(\neg A). \tag{45}$$

We know $P(B|A) = 1$. For the term $P(B|\neg A)$, we introduce a reasonable assumption: in the absence of the common cause (TFS), its consequences (WD and WVO) are approximately conditionally independent.

$$P(B|\neg A, C) \approx P(B|\neg A). \tag{46}$$

This assumption is justified because if the fundamental mechanism (TFS) that links WD and WVO is absent, the correlation between them should vanish or be significantly diminished. Any residual correlation would be a minor influence. Under this assumption, $P(B|\neg A) \approx P(B|\neg A, C) = q$.

Substituting this into the expression for $P(B)$:

$$P(B) \approx 1 \cdot P(A) + q \cdot (1 - P(A)) = q + (1-q)P(A). \tag{47}$$

We now have two expressions to compare:

1. $P(B|C) = q + (1-q)P(A|C)$;
2. $P(B) \approx q + (1-q)P(A)$.

We have proved that $P(A|C) > P(A)$. Since $q < 1$, the term $(1-q)$ is positive. It therefore follows directly that,

$$P(B|C) > P(B) \tag{48}$$

This result provides a rigorous probabilistic foundation for our central thesis. It demonstrates that observing the measurable geometric property of Weight Vector Orthogonality is a strong piece of evidence that increases the likelihood that the model also possesses the desired functional property of Weight Disentanglement. This justifies using WVO as a diagnostic tool to assess a model's suitability for task arithmetic.

G. Detailed Proof of Theorem 2

G.1. Proof of Theorem 2

Theorem 2. *Under the NTK linearization hypothesis (Section 3.3), even if the Task-Feature Specialization property does not hold (i.e., $I_t \cap I_j \neq \emptyset$), constraining the task update matrices $\{\Delta W_t^{(l)}\}$ to be approximately internally orthogonal (as encouraged by the regularization in Definition 4) actively promotes weight disentanglement between tasks t and j.*

Proof. According to Lemma 1, our goal is to demonstrate that the interference from task j on the data domain of task t is approximately zero, *i.e.*, $\tau_j^\top J(x) \approx 0$. The interference term's magnitude can be expressed as,

$$|\tau_j^\top J(x)| = ||\tau_j||_2 \cdot ||J(x)||_2 \cdot |\cos \angle(\tau_j, J(x))|. \tag{49}$$

The proof proceeds in four steps. We first reframe the angle term, then demonstrate how our regularizer controls both the norm and angle terms, and finally synthesize the results.

Step 1: Directional Alignment.

First, we establish that for a typical input $x \in \mathcal{D}_t$, its Jacobian $J(x)$ is directionally aligned with the task vector τ_t. The direction of τ_t is determined by the average Jacobian over the task's data domain, $\mu_J := \mathbb{E}_{x\in\mathcal{D}_t}[J(x)]$. Under a reasonable data consistency assumption, the gradients of different samples are statistically consistent rather than random, the direction of a typical $J(x)$ aligns with that of μ_J and, by extension, with τ_t. This alignment, rigorously proven in Appendix G.2, allows us to reframe the term's angle using the angle between the two task vectors,

$$|\tau_j^\top J(x)| \approx ||\tau_j||_2 \cdot ||J(x)||_2 \cdot |\cos\angle(\tau_j, \tau_t)|. \tag{50}$$

Step 2: Norm Control.

Our second step is to show that the orthogonal regularization term $\mathcal{L}_{\text{ortho}}$ effectively bounds the norm of the task vectors. The regularizer penalizes the deviation of each update matrix ΔW from the identity. By solving a constrained optimization problem, we can prove that the Frobenius norm of an update matrix ΔW is strictly bounded by its deviation from orthogonality. As formalized in Proposition Proposition 3 (see Appendix G.3), if $\|\Delta W^\top \Delta W - I\|_F^2 \leq \xi$, then the norm is bounded by,

$$\|\Delta W\|_F^2 \leq d + \sqrt{d\xi}, \tag{51}$$

where d is the number of columns. As the task vector's total norm is determined by the norms of its constituent update matrices, $\|\tau_j\|_2^2 = \sum_l \|\Delta W_j^{(l)}\|_F^2$, our regularizer effectively constrains the overall magnitude of τ_j.

Step 3: Angle Control.

Our third and most critical step is to demonstrate that the regularization statistically promotes orthogonality between different task vectors, *i.e.*, $\mathbb{E}[|\cos\angle(\tau_j, \tau_t)|] \approx 0$.

The core mechanism is that the internal orthogonal structure imposed on each update matrix ΔW induces inter-task statistical orthogonality between the resulting task vectors τ_t and τ_j. This can be understood through the lens of Polar Decomposition [6], which allows us to express any approximately orthogonal update matrix ΔW as $\Delta W = QP$, where Q is a strictly orthonormal matrix (an element of the Stiefel manifold $V_d(\mathbb{R}^m)$) and P is a symmetric positive semi-definite matrix that is very close to the identity (as formalized in Proposition 4 in Appendix G.4.1).

Consequently, the inner product of two task vectors, $\langle \tau_t, \tau_j \rangle$, which is a sum of layer-wise inner products $\sum_l \langle \text{vec}(\Delta W_t^{(l)}), \text{vec}(\Delta W_j^{(l)}) \rangle$, is dominated by the sum of inner products of their orthonormal components, $\sum_l \langle \text{vec}(Q_t^{(l)}), \text{vec}(Q_j^{(l)}) \rangle$ (see Appendix G.4 for a detailed derivation). As established in Lemma 2 (Appendix G.4.3), two matrices independently and uniformly drawn from the Stiefel manifold are, when vectorized, statistically orthogonal. Their inner product has an expected value of zero and its probability distribution is sharply peaked at zero. This strong statistical tendency towards orthogonality at each layer propagates to the entire task vectors, ensuring that τ_t and τ_j are highly likely to be nearly orthogonal. The detailed proof can be seen in Appendix G.4.2

Step 4: Completing the Proof. We now synthesize the results. The magnitude of the interference term is given by,

$$|\tau_j^\top J(x)| \approx \underbrace{||\tau_j||_2}_{\text{Bounded}} \cdot \underbrace{||J(x)||_2}_{\text{Inherently Bounded}} \cdot \underbrace{|\cos\angle(\tau_j, \tau_t)|}_{\text{Statistically near zero}} \tag{52}$$

The dual control mechanism of our regularization ensures that this product is approximately zero in expectation. The norm $\|\tau_j\|$ is bounded (Step 2), $\|J(x)\|$ is bounded for any given input and model, and the cosine of the angle between task vectors is statistically driven towards zero (Step 3). Consequently, the expected interference is negligible,

$$\mathbb{E}[|\tau_j^\top J(x)|] \approx 0. \tag{53}$$

By Lemma 1, this establishes that weight disentanglement is approximately achieved. This completes the proof. □

G.2. Proof of Directional Alignment (Step 1)

In this section, we provide a rigorous proof for the claim that for a typical input $x \in \mathcal{D}_t$, its Jacobian vector $J(x)$ is directionally aligned with the task vector τ_t.

Proof. The proof proceeds in two parts: first, relating the direction of τ_t to the average Jacobian μ_J, and second, relating the direction of an individual $J(x)$ to μ_J.

Part 1: Direction of the Task Vector τ_t.

As clarified in Equation (16), the task vector τ_t is the result of accumulated gradients during fine-tuning. In the initial phase of fine-tuning, where the parameters θ are close to θ_0, the direction of τ_t is dominated by the average gradient of the task loss $\mathcal{L}_t$ over the data domain $\mathcal{D}_t$, evaluated at θ_0.

$$\tau_t \propto -\mathbb{E}_{(x,y)\sim\mathcal{D}_t}\left[\nabla_\theta \mathcal{L}_t(f(x;\theta_0), y)\right]. \tag{54}$$

Using the chain rule, $\nabla_\theta \mathcal{L}_t = \frac{\partial \mathcal{L}_t}{\partial f} \cdot \nabla_\theta f = \frac{\partial \mathcal{L}_t}{\partial f} \cdot J(x)$. The expression becomes,

$$\tau_t \propto -\mathbb{E}_{x\sim\mathcal{D}_t}\left[\frac{\partial \mathcal{L}_t}{\partial f} \cdot J(x)\right]. \tag{55}$$

For a well-posed learning task, the loss derivative $\frac{\partial \mathcal{L}_t}{\partial f}$ (which indicates how the loss changes with respect to the model's output) can be assumed to be an approximately constant scalar k_t across the dataset. This yields,

$$\tau_t \propto -k_t \cdot \mathbb{E}_{x\in\mathcal{D}_t}[J(x)]. \tag{56}$$

Let $\mu_J := \mathbb{E}_{x\in\mathcal{D}_t}[J(x)]$ be the average Jacobian vector over the data domain of task t. We thus establish the first directional link,

$$\text{Direction}(\tau_t) = \text{Direction}(\mu_J). \tag{57}$$

Part 2: Direction of an Individual Jacobian $J(x)$.

Next, we formalize an intuitive hypothesis. For a well-defined, non-random machine learning task, the loss function's gradient directions for different samples within its data domain should exhibit statistical consistency, rather than pointing randomly in all directions throughout the parameter space. This consistency is fundamental to the model's ability to learn generalizable patterns from data. Applied to our scenario, this implies that the distribution of the Jacobian vectors $J(x)$ should not be overly dispersed.

We formalize this as the Data Consistency Assumption.

Assumption 1 (Data Consistency Assumption)**.** *For a well-defined task, the Jacobian vectors of individual samples are statistically concentrated around their mean. This means the variance of the Jacobians, $\sigma_J^2 := \mathbb{E}_{x\in\mathcal{D}_t}\left[\|J(x)-\mu_J\|_2^2\right]$, is significantly smaller than the squared norm of their mean,*

$$\sigma_J^2 \ll \|\mu_J\|_2^2. \tag{58}$$

By Chebyshev's inequality [17], for any constant $C > 1$, we have,

$$\begin{aligned}\mathbb{P}\left(\|J(x)-\mu_J\|_2^2 \geq C^2\sigma_J^2\right) &\leq \frac{\mathbb{E}\left[\|J(x)-\mu_J\|_2^2\right]}{C^2\sigma_J^2} \\ &= \frac{\sigma_J^2}{C^2\sigma_J^2} = \frac{1}{C^2}.\end{aligned} \tag{59}$$

This implies that the squared Euclidean distance between the random vector $J(x)$ and its mean μ_J is bounded by $C^2\sigma_J^2$ with a probability of at least $1 - 1/C^2$. In other words, for a "typical" (*i.e.*, high-probability) sample x', its Jacobian vector $J(x)$ satisfies,

$$\|J(x')-\mu_J\|_2 < C\sigma_J. \tag{60}$$

Now, we bound the angle $\theta_{x'} = \angle(J(x'), \mu_J)$ for such a typical sample. Consider the triangle formed by the origin and the endpoints of the vectors $J(x')$ and μ_J. By the properties of vector geometry (related to the Law of Sines [4]), the sine of the angle $\theta_{x'}$ is bounded by the ratio of the length of the opposing side to the length of the adjacent side,

$$\sin(\theta_{x'}) \leq \frac{\|J(x')-\mu_J\|_2}{\|J(x')\|_2}. \tag{61}$$

We have an upper bound for the numerator, $\|J(x)-\mu_J\|_2 < C\sigma_J$. For the denominator, we use the reverse triangle inequality [18] to find a lower bound,

$$\begin{aligned}\|J(x')\|_2 &= \|\mu_J + (J(x')-\mu_J)\|_2 \\ &\geq \|\mu_J\|_2 - \|J(x')-\mu_J\|_2 \\ &> \|\mu_J\|_2 - C\sigma_J.\end{aligned} \tag{62}$$

Substituting these bounds, we get,

$$\sin(\theta_{x'}) < \frac{C\sigma_J}{\|\mu_J\|_2 - C\sigma_J} = \frac{C(\sigma_J/\|\mu_J\|_2)}{1 - C(\sigma_J/\|\mu_J\|_2)}. \tag{63}$$

Given Assumption 1, the ratio $\sigma_J/\|\mu_J\|_2$ is a value much smaller than 1. Therefore, the right-hand side of the inequality is a very small positive number. Since $\sin(\theta_{x'})$ is very small, the angle $\theta_{x'}$ must also be very close to zero. This establishes our second directional link,

$$\text{Direction}(J(x)) \approx \text{Direction}(\mu_J), \quad \text{for a typical } x \in \mathcal{D}_t. \tag{64}$$

Combining the two parts, we have shown that for a typical sample $x \in \mathcal{D}_t$,

$$\text{Direction}(J(x)) \approx \text{Direction}(\mu_J) \approx \text{Direction}(\tau_t). \tag{65}$$

This directional alignment justifies the approximation used in the main proof, allowing the angle between τ_j and $J(x)$ to be replaced by the angle between τ_j and τ_t. This completes the proof. □

G.3. Proposition 3 and Proof (Norm Control)

Proposition 3. *The Frobenius norm of a matrix is bounded by its deviation from orthonormality. Specifically, for a matrix $W \in \mathbb{R}^{m\times d}$, if its deviation from being identity is bounded by $\|W^\top W - I_d\|_F^2 \leq \xi$ for some constant $\xi \geq 0$, then its squared Frobenius norm is bounded by,*

$$\|W\|_F^2 \leq d + \sqrt{d\xi}. \tag{66}$$

Several prior works have implicitly or explicitly leveraged the norm-controlling property of orthogonality [2, 15, 23]. Here, we provide a formal and rigorous proof to establish this principle.

Proof. We aim to find the maximum possible value of $\|W\|_F^2$ under the given constraint. This can be formulated as a constrained optimization problem,

$$\begin{aligned}\max_W \quad & \|W\|_F^2. \\ \text{s.t.} \quad & \|W^\top W - I_d\|_F^2 \leq \xi\end{aligned} \tag{67}$$

To solve this, we use the Singular Value Decomposition (SVD) [5] of W. Let $W = U\Sigma V^\top$, where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{d\times d}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{m\times d}$ is a rectangular diagonal matrix with non-negative singular values $\{\sigma_1, \sigma_2, \ldots, \sigma_d\}$ on its diagonal.

First, we rewrite the objective function in terms of the singular values. Because Frobenius norm is invariant under orthogonal transformations, we can get,

$$\|W\|_F^2 = \|U\Sigma V^\top\|_F^2 = \|\Sigma\|_F^2 = \sum_{i=1}^{d}\sigma_i^2. \tag{68}$$

Next, we rewrite the constraint. We have $W^\top W = (U\Sigma V^\top)^\top(U\Sigma V^\top) = V\Sigma^\top U^\top U\Sigma V^\top = V(\Sigma^\top\Sigma)V^\top$. Let $D = \Sigma^\top\Sigma$, which is a $d \times d$ diagonal matrix with diagonal elements $D_{ii} = \sigma_i^2$. Again, using the orthogonal invariance of the Frobenius norm,

$$\begin{aligned}\|W^\top W - I_d\|_F^2 &= \|VDV^\top - VI_dV^\top\|_F^2 \\ &= \|V(D - I_d)V^\top\|_F^2 = \|D - I_d\|_F^2.\end{aligned} \tag{69}$$

Since $D - I_d$ is a diagonal matrix, its squared Frobenius norm is the sum of the squares of its diagonal elements,

$$\|D - I_d\|_F^2 = \sum_{i=1}^{d}(\sigma_i^2 - 1)^2. \tag{70}$$

The original problem is now equivalent to a simpler optimization problem over the squared singular values. Let $x_i = \sigma_i^2 \geq 0$,

$$\begin{aligned}\max \quad & \sum_{i=1}^{d} x_i. \\ \text{s.t.} \quad & \sum_{i=1}^{d}(x_i - 1)^2 \leq \xi\end{aligned} \tag{71}$$

To find the maximum, the constraint must be active, *i.e.*, $\sum_{i=1}^{d}(x_i - 1)^2 = \xi$. We use the method of Lagrange multipliers [14]. The Lagrangian is,

$$\mathcal{L}(\mathbf{x}, \lambda) = \sum_{i=1}^{d} x_i - \lambda\left(\sum_{i=1}^{d}(x_i - 1)^2 - \xi\right). \tag{72}$$

Taking the partial derivative with respect to x_j and setting it to zero,

$$\frac{\partial\mathcal{L}}{\partial x_j} = 1 - \lambda \cdot 2(x_j - 1) = 0. \tag{73}$$

$$x_j - 1 = \frac{1}{2\lambda} \implies x_j = 1 + \frac{1}{2\lambda}. \tag{74}$$

This shows that at the optimal point, all x_j must be equal. Let $x_1 = x_2 = \cdots = x_d = x^*$.

Substituting $x_i = x^*$ into the active constraint,

$$\sum_{i=1}^{d}(x^* - 1)^2 = d(x^* - 1)^2 = \xi. \tag{75}$$

Solving for x^*, we get,

$$(x^* - 1)^2 = \frac{\xi}{d} \implies x^* - 1 = \pm\sqrt{\frac{\xi}{d}}, \tag{76}$$

$$x^* = 1 \pm \sqrt{\frac{\xi}{d}}. \tag{77}$$

To maximize the objective function $\sum x_i = d \cdot x^*$, we must choose the positive root,

$$x^*_{\text{max}} = 1 + \sqrt{\frac{\xi}{d}}. \tag{78}$$

Finally, the maximum value of the objective function is,

$$\begin{aligned}\max \|W\|_F^2 &= \sum_{i=1}^{d} x^*_{\text{max}} = d \cdot x^*_{\text{max}} \\ &= d\left(1 + \sqrt{\frac{\xi}{d}}\right) = d + \sqrt{d\xi}.\end{aligned} \tag{79}$$

This establishes the upper bound and completes the proof. □

G.4. Detailed Proof of Angle Control Mechanism

This section provides the full proof for Step 3 of Theorem 2, showing that our orthogonal regularization statistically promotes orthogonality between different task vectors.

G.4.1. Proposition 4 and Detailed Proof

Proposition 4. *Let $P \in \mathbb{R}^{d\times d}$ be a symmetric positive semi-definite matrix. If $\|P^2 - I_d\|_F \leq \sqrt{\xi}$, then $\|P - I_d\|_F$ is also bounded, and specifically satisfies,*

$$\|P - I_d\|_F \leq \|P^2 - I_d\|_F. \tag{80}$$

Proof. Since P is symmetric, it has an eigenvalue decomposition $P = U\Lambda U^\top$, where U is an orthogonal matrix and Λ is a diagonal matrix of non-negative eigenvalues $\lambda_1, \ldots, \lambda_d \geq 0$. The Frobenius norm is invariant under orthogonal transformations. Thus, we can express the norms in terms of the eigenvalues,

$$\begin{aligned}\|P - I_d\|_F^2 &= \|U\Lambda U^\top - UI_dU^\top\|_F^2 \\ &= \|U(\Lambda - I_d)U^\top\|_F^2 \\ &= \|\Lambda - I_d\|_F^2 \\ &= \sum_{i=1}^{d}(\lambda_i - 1)^2.\end{aligned} \tag{81}$$

Similarly, since $P^2 = (U\Lambda U^\top)(U\Lambda U^\top) = U\Lambda^2U^\top$,

$$\begin{aligned}\|P^2 - I_d\|_F^2 &= \|U(\Lambda^2 - I_d)U^\top\|_F^2 \\ &= \|\Lambda^2 - I_d\|_F^2 \\ &= \sum_{i=1}^{d}(\lambda_i^2 - 1)^2\end{aligned} \tag{82}$$

Now we compare the terms for each eigenvalue,

$$(\lambda_i^2 - 1)^2 = ((\lambda_i - 1)(\lambda_i + 1))^2 = (\lambda_i - 1)^2 \cdot (\lambda_i + 1)^2 \tag{83}$$

Since P is positive semi-definite, $\lambda_i \geq 0$. This implies $\lambda_i + 1 \geq 1$, and therefore $(\lambda_i + 1)^2 \geq 1$. Multiplying both sides by the non-negative quantity $(\lambda_i - 1)^2$, we get

$$(\lambda_i - 1)^2 \cdot (\lambda_i + 1)^2 \geq (\lambda_i - 1)^2 \cdot 1. \tag{84}$$

This means $(\lambda_i^2 - 1)^2 \geq (\lambda_i - 1)^2$ for all i. Summing over all i,

$$\sum_{i=1}^{d} (\lambda_i^2 - 1)^2 \geq \sum_{i=1}^{d} (\lambda_i - 1)^2. \tag{85}$$

Substituting the norm expressions back, we have,

$$\|P^2 - I_d\|_F^2 \geq \|P - I_d\|_F^2. \tag{86}$$

Taking the square root of both sides yields the desired result,

$$\|P - I_d\|_F \leq \|P^2 - I_d\|_F. \tag{87}$$

□

G.4.2. Proof of Angle Control

Proof. Our goal is to show that enforcing an internal orthogonal structure on the update matrices ΔW_t and ΔW_j statistically drives their corresponding task vectors τ_t and τ_j towards orthogonality. That is, $\mathbb{E}[|\cos \angle(\tau_t, \tau_j)|] \approx 0$. This is equivalent to showing that the inner product $\langle \tau_t, \tau_j \rangle$ is statistically concentrated around zero.

The total inner product is the sum of layer-wise inner products,

$$\langle \tau_t, \tau_j \rangle = \sum_{l \in \text{Layers}} \langle \text{vec}(\Delta W_t^{(l)}), \text{vec}(\Delta W_j^{(l)}) \rangle. \tag{88}$$

We analyze the inner product for a single layer, dropping the superscript (l) for clarity: $\langle \text{vec}(\Delta W_t), \text{vec}(\Delta W_j) \rangle$.

Our $\mathcal{L}_{\text{ortho}} = \|\Delta W^\top \Delta W - I\|_F^2$ encourages the resulting update matrix ΔW^* to be approximately orthogonal, satisfying $\|(\Delta W^*)^\top \Delta W^* - I\|_F^2 \leq \xi$ for a small ξ.

Using Polar Decomposition [6], any such matrix ΔW^* can be uniquely decomposed into $\Delta W^* = QP$, where $Q \in V_d(\mathbb{R}^m)$ is a matrix with orthonormal columns (an element of the Stiefel manifold) and $P = \sqrt{(\Delta W^*)^\top \Delta W^*}$ is a symmetric positive semi-definite matrix.

Substituting this relation into our regularization constraint, $\|(\Delta W^*)^\top \Delta W^* - I\|_F^2 \leq \xi$, we have $\|P^2 - I\|_F^2 \leq \xi$. By Proposition 4, this implies that P is close to the identity matrix, *i.e.*, $\|P - I\|_F$ is also small. We can thus write $P = I + E$, where $E = P - I$ is an "error" matrix with a small Frobenius norm $\|E\|_F$.

Therefore, the update matrices for tasks t and j can be written as,

$$\Delta W_t = Q_t + Q_t E_t, \tag{89}$$

$$\Delta W_j = Q_j + Q_j E_j, \tag{90}$$

where Q_t, Q_j are matrices on Stiefel manifold, and E_t, E_j are error matrices with small norms controlled by ξ.

Now, we analyze the inner product of their vectorized forms,

$$\begin{aligned} &\langle \text{vec}(\Delta W_t), \text{vec}(\Delta W_j) \rangle \\ &= \langle \text{vec}(Q_t + Q_t E_t), \text{vec}(Q_j + Q_j E_j) \rangle. \end{aligned} \tag{91}$$

Expanding this expression yields four terms,

$$\begin{aligned} &= \underbrace{\langle \text{vec}(Q_t), \text{vec}(Q_j) \rangle}_{\text{Main Term}} + \underbrace{\langle \text{vec}(Q_t), \text{vec}(Q_j E_j) \rangle}_{\text{Error Term 1}} \\ &+ \underbrace{\langle \text{vec}(Q_t E_t), \text{vec}(Q_j) \rangle}_{\text{Error Term 2}} + \underbrace{\langle \text{vec}(Q_t E_t), \text{vec}(Q_j E_j) \rangle}_{\text{Error Term 3}}. \end{aligned} \tag{92}$$

We analyze the expectation of each term, assuming that the fine-tuning processes for distinct tasks t and j result in independently sampled matrices from the space of approximately orthogonal matrices.

Main Term. Q_t and Q_j are independent, random matrices from the Stiefel manifold $V_d(\mathbb{R}^m)$. According to Lemma 2 (proven in Appendix G.4.3), the expected value of their inner product is zero,

$$\mathbb{E}[\langle \text{vec}(Q_t), \text{vec}(Q_j) \rangle] = 0. \tag{93}$$

And, Lemma 2 states that the probability distribution of this inner product is sharply concentrated around zero.

Error Terms. We bound the magnitude of the error terms using the Cauchy-Schwarz inequality.

For Error Term 1,

$$|\langle \text{vec}(Q_t), \text{vec}(Q_j E_j) \rangle| \leq \|\text{vec}(Q_t)\|_2 \cdot \|\text{vec}(Q_j E_j)\|_2. \tag{94}$$

Since Q_t has d orthonormal columns, $\|\text{vec}(Q_t)\|_2^2 = \|Q_t\|_F^2 = d$. Since Q_j is an orthogonal transformation, $\|\text{vec}(Q_j E_j)\|_2 = \|Q_j E_j\|_F = \|E_j\|_F$. Thus, the term is bounded by $\sqrt{d} \cdot \|E_j\|_F$. As $\|E_j\|_F$ is a small value controlled by the regularizer, this error term is negligible.

Error Term 2 is similarly bounded by $\sqrt{d} \cdot \|E_t\|_F$ and is also negligible.

Error Term 3 is bounded by $\|\text{vec}(Q_t E_t)\|_2 \cdot \|\text{vec}(Q_j E_j)\|_2 = \|E_t\|_F \cdot \|E_j\|_F$, which is a second-order small term and even more negligible.

Since the main term has an expected value of zero and the error terms are negligible, the expected inner product for a single layer is approximately zero.

$$\mathbb{E}[\langle \text{vec}(\Delta W_t), \text{vec}(\Delta W_j) \rangle] \approx 0. \tag{95}$$

By linearity of expectation, the expected inner product of the full task vectors is also approximately zero,

$$\mathbb{E}[\langle \tau_t, \tau_j \rangle] = \sum_l \mathbb{E}[\langle \text{vec}(\Delta W_t^{(l)}), \text{vec}(\Delta W_j^{(l)}) \rangle] \approx 0. \tag{96}$$

Because the distribution of the main term at each layer is sharply peaked at zero, the distribution of the sum (the total inner product) will also be sharply peaked at zero. This implies that τ_t and τ_j are statistically very likely to be orthogonal, and thus $\mathbb{E}[|\cos\angle(\tau_t, \tau_j)|] \approx 0$. This completes the proof of the angle control mechanism. □

G.4.3. Lemma 2 and Detailed Proof: Stiefel Manifold Inner Product

Lemma 2. *Let A and B be two matrices independently and uniformly sampled from the Stiefel manifold $V_d(\mathbb{R}^m)$ [1] (the set of $m \times d$ matrices with orthonormal columns). Let $Z = \langle vec(A), vec(B)\rangle$. Then,*

(1) The expected value of the inner product is zero: $\mathbb{E}[Z] = 0$.

(2)The probability distribution of Z is sharply concentrated around 0.

Proof. **Part 1: Proof of Zero Expectation.**

The inner product can be written as the trace of the matrix product: $Z = \text{Tr}(A^\top B)$. Due to the independence of A and B, the expectation of the product is the product of expectations,

$$\mathbb{E}[Z] = \mathbb{E}_A[\mathbb{E}_B[\text{Tr}(A^\top B)|A]] = \mathbb{E}_A[\text{Tr}(A^\top \mathbb{E}_B[B])]. \tag{97}$$

Let's compute $\mathbb{E}[B]$. The distribution of B is the uniform (Haar) measure on $V_d(\mathbb{R}^m)$. This distribution is invariant under left-multiplication by any orthogonal matrix $Q \in O(m)$, where $O(m)$ is the group of $m \times m$ orthogonal matrices. This means that for any $Q \in O(m)$, the random matrix QB has the same distribution as B. Therefore,

$$\mathbb{E}[B] = \mathbb{E}[QB] = Q\mathbb{E}[B]. \tag{98}$$

This equality must hold for all $Q \in O(m)$. Let's consider a specific reflection matrix Q that negates the first coordinate, *e.g.*, $Q = \text{diag}(-1, 1, \ldots, 1)$. If the first row of $\mathbb{E}[B]$ were a non-zero vector $\mathbf{r}$, then the first row of $Q\mathbb{E}[B]$ would be $-\mathbf{r}$. The equality $\mathbb{E}[B] = Q\mathbb{E}[B]$ would imply $\mathbf{r} = -\mathbf{r}$, which is only possible if $\mathbf{r} = \mathbf{0}$. This logic applies to every row by choosing appropriate reflection matrices. Therefore, the only matrix that satisfies this condition for all $Q \in O(m)$ is the zero matrix.

$$\mathbb{E}[B] = \mathbf{0}. \tag{99}$$

Substituting this back into the expectation for Z, we get,

$$\mathbb{E}[Z] = \text{Tr}(\mathbb{E}[A^\top] \cdot \mathbf{0}) = 0. \tag{100}$$

This proves the first part of the lemma.

Part 2: Proof of Concentration around Zero.

This is a geometric argument. The vectors $\text{vec}(A)$ and $\text{vec}(B)$ are not arbitrary vectors in $\mathbb{R}^{m\times d}$. They are constrained to lie on the submanifold $\text{vec}(V_d(\mathbb{R}^m))$. The condition $A^\top A = I_d$ imposes $\frac{d(d+1)}{2}$ independent constraints on the elements of A. This means the dimension of the Stiefel manifold $V_d(\mathbb{R}^m)$ is $\dim(V) = md - \frac{d(d+1)}{2}$.

The co-dimension of this submanifold within the ambient space $\mathbb{R}^{m\times d}$ is $\frac{d(d+1)}{2}$, which is positive for $d \geq 1$. The condition for orthogonality, $\langle \text{vec}(A), \text{vec}(B)\rangle = 0$, defines a hyperplane in the product space. The probability density of Z at a value z_0 is proportional to the "volume" of the surface defined by $\langle \text{vec}(A), \text{vec}(B)\rangle = z_0$ on the product manifold $V_d(\mathbb{R}^m) \times V_d(\mathbb{R}^m)$.

Intuitively, because the vectors are already living in a lower-dimensional space due to the internal orthogonality constraints, the additional constraint of being orthogonal to another such vector is "easier" to satisfy. The intersection of the hyperplane $\langle \mathbf{a}, \mathbf{b}\rangle = 0$ with the product manifold is larger than its intersection with the product of two spheres of the same dimension. This geometric fact leads to a higher probability density at $Z = 0$, creating a sharp peak in the distribution. This indicates that two random matrices from Stiefel manifold are much more likely to be nearly orthogonal than two completely random unit vectors in $\mathbb{R}^{m\times d}$. □

H. Comparative Analysis with TTA

H.1. Theoretical Connection

In this section, we establish a theoretical connection between our proposed method (OrthoReg) and Tangent Task Arithmetic (TTA) [16]. We demonstrate that both methods, despite their different implementations, derive their effectiveness from a shared underlying mechanism: promoting orthogonality between different task vectors (*i.e.*, $\langle \tau_t, \tau_j\rangle \approx 0$ for $t \neq j$). This inter-task vector orthogonality is a key driver for achieving weight disentanglement.

As proven in Theorem 2 (specifically, the Angle Control mechanism in Appendix G.4.2), our OrthoReg achieves this goal explicitly. By enforcing an internal orthogonal structure on each update matrix ΔW, it statistically drives the resulting full task vectors towards orthogonality.

In contrast, TTA achieves this goal implicitly by leveraging the geometric properties of the pre-trained model's Neural Tangent Kernel (NTK). We now provide a detailed derivation to formalize this connection.

TTA operates by performing fine-tuning in the tangent space of the pre-trained model θ_0. The model's output is approximated by its first-order Taylor expansion,

$$f(x; \theta_0 + \tau) \approx f(x; \theta_0) + \tau^\top J(x), \tag{101}$$

where $J(x) = \nabla_\theta f(x; \theta_0)$ is the Jacobian. The optimization is performed over the task vector τ directly. For a task t with data $\{(x_i, y_i)\}_{i=1}^{N_t}$ from domain $\mathcal{D}_t$, the TTA objective can be formulated as a regularized empirical risk minimization problem, for instance, using a mean-squared error loss:

$$\min_{\tau_t} \frac{1}{N_t} \sum_{i=1}^{N_t} \left\| (f(x_i; \theta_0) + \tau_t^\top J(x_i)) - y_i \right\|_2^2 + \lambda \|\tau_t\|_2^2.$$

Table 1. Computational cost comparison on the Cars dataset using a ViT-L-14 model. The table highlights the efficiency of OrthoReg. The final column shows the Absolute Accuracy from the task addition benchmark (as seen in Table 1 of the main paper). While applying OrthoReg to Non-linear Fine-tuning (Non-lin. FT) achieves performance that is superior to Tangent Task Arithmetic (TTA) and significantly better than the baseline Non-lin. FT, this table further demonstrates its superior computational efficiency. As seen, TTA incurs substantial overhead in both training time and memory, whereas OrthoReg adds only a modest cost to the baseline. The colored cells visually emphasize the significant difference in computational cost between TTA and our proposed method.

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 [9] (Baseline)	342.56	342.56	158.21	42589.22	84.07
TTA [16] (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
Parameter-Efficient Fine-tuning (Attention-Only)					
ATT-FT [11]	342.56	100.66	126.28	36591.06	87.81
ATT-FT + OrthoReg (ours)	342.56	100.66	132.96	36976.50	90.41

This is a linear ridge regression problem in the variable τ_t. According to the Representer Theorem, the optimal solution τ_t^* must lie in the subspace spanned by the Jacobians of the training data points. Therefore, τ_t^* can be expressed as a linear combination of these Jacobians,

$$\tau_t^* = \sum_{i=1}^{N_t} \alpha_i J(x_i), \tag{102}$$

where $\{\alpha_i\}$ are scalar coefficients determined by the optimization.

Now, consider the inner product between two task vectors, τ_t^* and τ_j^*, obtained by applying TTA to two different tasks, t and j,

$$\langle \tau_t^*, \tau_j^* \rangle = \left\langle \sum_{i=1}^{N_t} \alpha_i J(x_i), \sum_{k=1}^{N_j} \beta_k J(x_k) \right\rangle, \tag{103}$$

where $\{x_i\} \subset \mathcal{D}_t$ and $\{x_k\} \subset \mathcal{D}_j$. By linearity of the inner product, this becomes,

$$\langle \tau_t^*, \tau_j^* \rangle = \sum_{i=1}^{N_t} \sum_{k=1}^{N_j} \alpha_i \beta_k \langle J(x_i), J(x_k) \rangle. \tag{104}$$

The term $\langle J(x_i), J(x_k) \rangle$ is precisely the definition of the Neural Tangent Kernel (NTK) evaluated at the pair of inputs (x_i, x_k),

$$k_{\text{NTK}}(x_i, x_k) = J(x_i)^\top J(x_k). \tag{105}$$

Therefore, the inner product of the task vectors is a weighted sum of NTK values between the data points of the two tasks,

$$\langle \tau_t^*, \tau_j^* \rangle = \sum_{i=1}^{N_t} \sum_{k=1}^{N_j} \alpha_i \beta_k \, k_{\text{NTK}}(x_i, x_k). \tag{106}$$

A central empirical finding of the TTA paper [16] is that the NTK of large pre-trained models, such as CLIP, exhibits a strong localization property. This property means that the kernel function value is significant only when both inputs are from the same task domain and decays rapidly to near-zero when the inputs are from different, unrelated task domains. Formally, for distinct tasks $t \neq j$,

$$k_{\text{NTK}}(x_i, x_k) \approx 0 \quad \text{for all } x_i \in \mathcal{D}_t \text{ and } x_k \in \mathcal{D}_j. \tag{107}$$

Substituting this result into our expression for the inner product, we find that every term in the double summation is approximately zero. Consequently, the entire sum is approximately zero,

$$\langle \tau_t^*, \tau_j^* \rangle \approx \sum_{i=1}^{N_t} \sum_{k=1}^{N_j} \alpha_i \beta_k \cdot 0 \approx 0. \tag{108}$$

This derivation shows that TTA's effectiveness in promoting weight disentanglement stems from its ability to implicitly construct task vectors that are nearly orthogonal to each other. This orthogonality is not an explicit constraint but rather an emergent property arising from the localized structure of the pre-trained model's NTK.

Our analysis thus unifies our method and TTA under a common principle: inter-task vector orthogonality is a core mechanism for achieving weight disentanglement. Our OrthoReg method provides a more direct, explicit to enforce this geometric property, which explains its ability to further enhance the performance of TTA and other task arithmetic methods, as demonstrated in our experiments.

H.2. Experimental Performance Comparison and Analysis

As established in Section 4.4 and Appendix H.1, both our OrthoReg method and Tangent Task Arithmetic (TTA) [16]

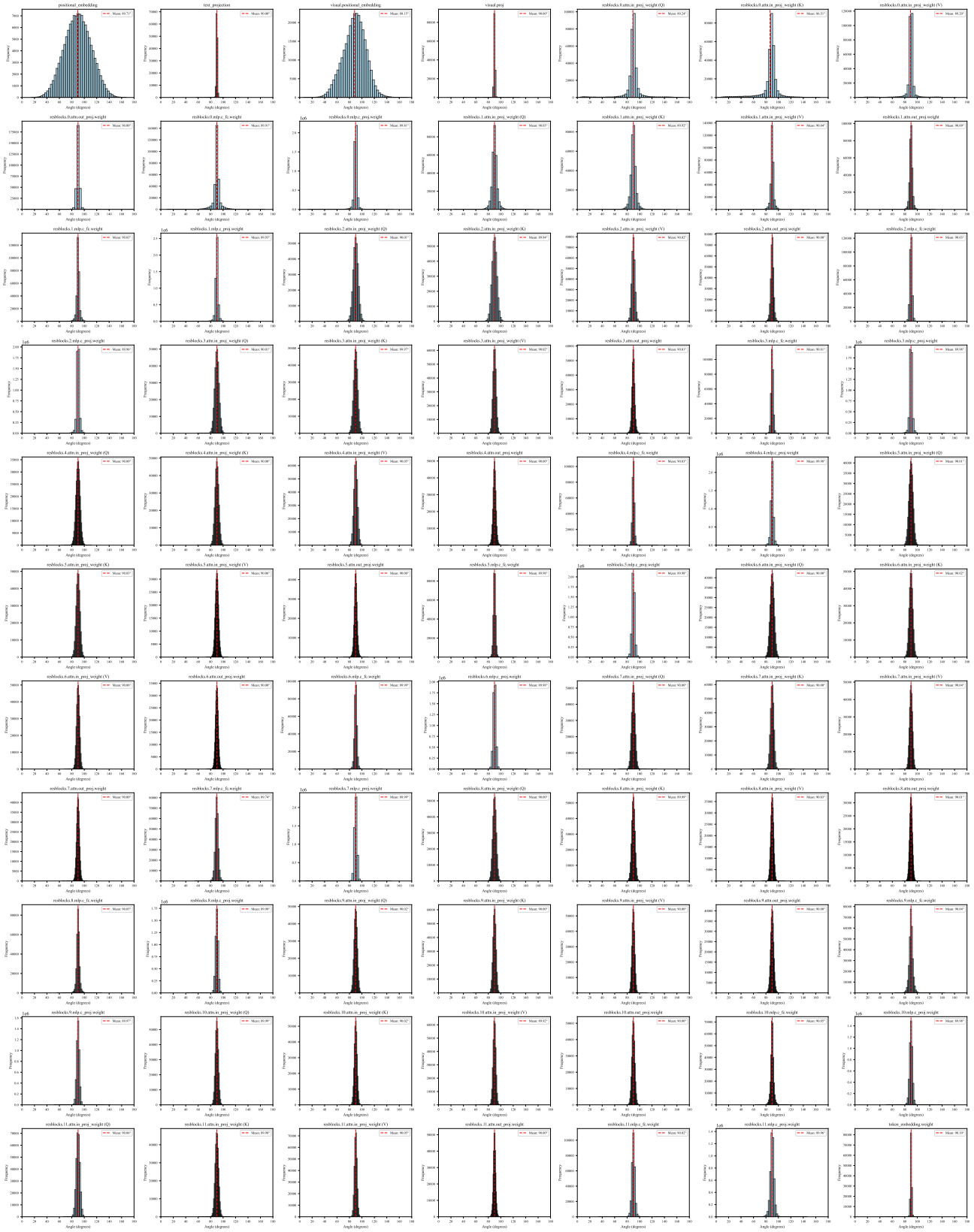
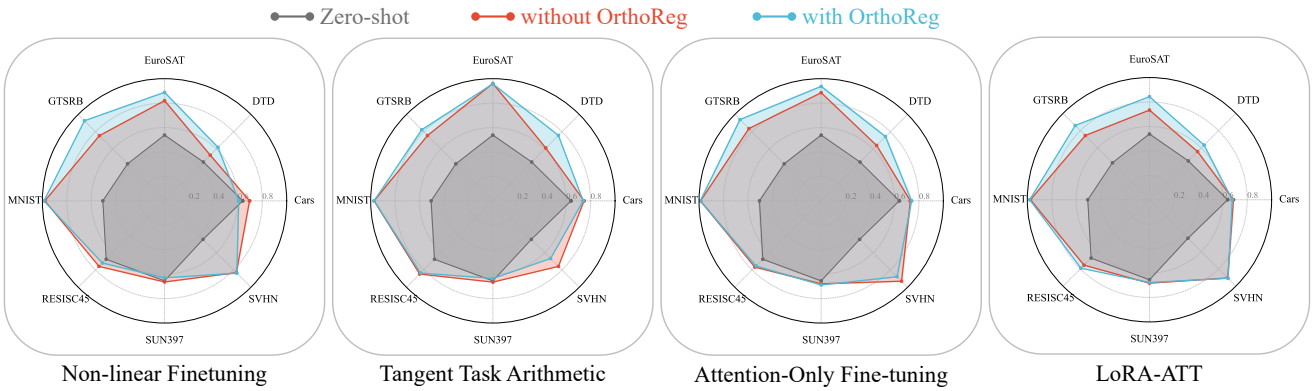
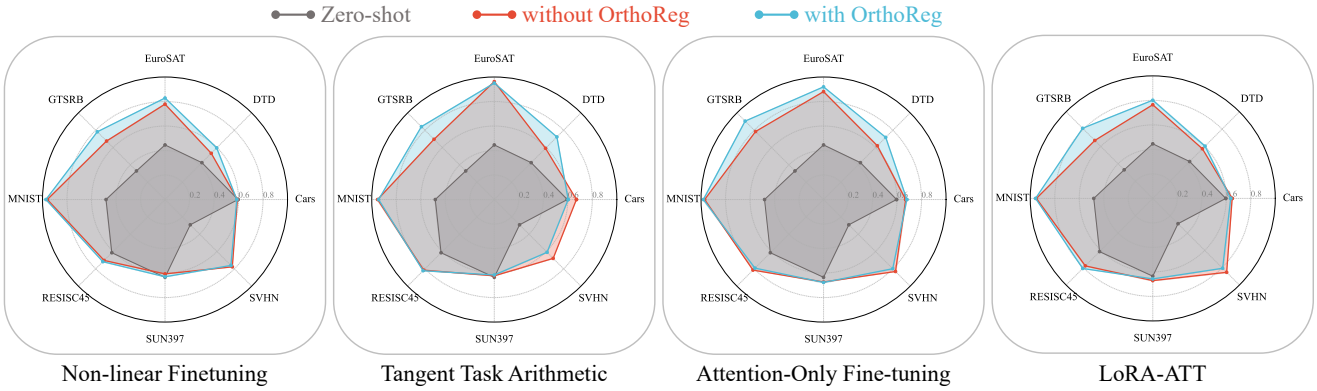


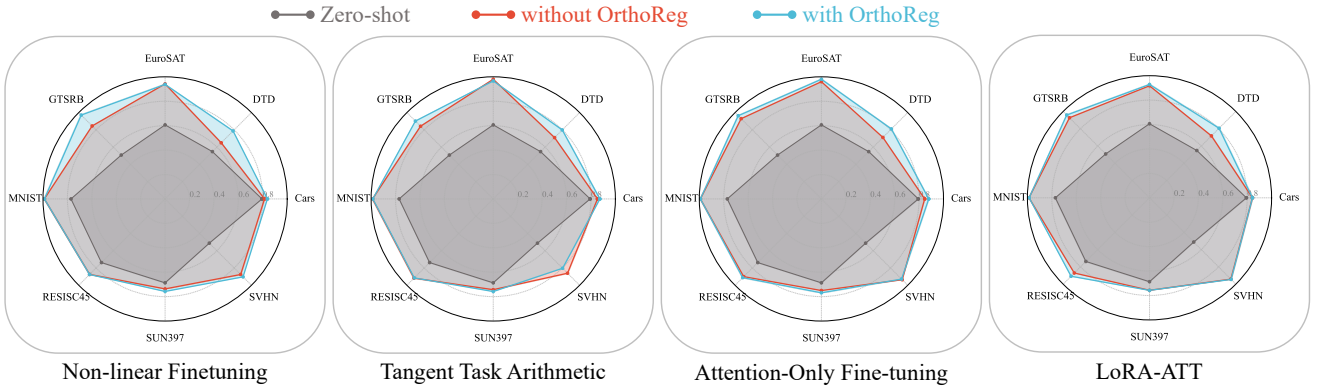
Figure 1. Angle distributions of weight matrix columns for all layers in ViT-B/16. Each subplot displays a histogram of the angles (in degrees) between all pairs of column vectors for a specific weight matrix. The red dashed line indicates the 90° point of perfect orthogonality. The plots are ordered sequentially, starting with the embedding layers, followed by the 12 transformer blocks.



(a) ViT-B-16



(b) ViT-B-32



(c) ViT-L-14

Figure 2. The accuracy of merged models across the eight benchmark tasks for different ViT architectures. Each subplot shows the performance for a specific baseline method: zero-shot (gray), the baseline's merged model (red), and the baseline enhanced with our orthogonal regularization (blue). The rows correspond to models: (a) ViT-B-16, (b) ViT-B-32, and (c) ViT-L-14.

succeed by promoting inter-task vector orthogonality. However, we posited that OrthoReg offers a more direct, efficient, and scalable approach by avoiding the costly Jacobian computations inherent to TTA. This section provides an empirical analysis to validate this claim by comparing the computational costs, specifically training time and peak GPU memory usage of TTA against standard fine-tuning methods enhanced with our OrthoReg regularizer.

Experimental Setup. We conduct a controlled experiment on the Cars dataset [13] using the ViT-L-14 model architecture. We measure the wall-clock training time and peak GPU memory consumption for a single fine-tuning run.

Table 2. The minimum average Target Accuracy (Tar.Acc.) achievable while maintaining at least 90% of the zero-shot accuracy on the ImageNet control task (Con.Acc.). Our proposed orthogonal regularization (+OrthoReg) shows a consistent and significant improvement in forgetting the target task. An asterisk (*) denotes the best (lowest) target accuracy for each model architecture.

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 [9]	17.34	60.80	14.92	63.63	13.51	72.51
Non-lin. FT**+OrthoReg** (ours)	**14.14**	60.84	**13.78**	65.69	**12.69**	74.17
Δ	-3.20	+0.04	-1.14	+2.06	-0.82	+1.66
Tangent Task Arithmetic [16]	7.36	62.08	6.68	65.49	5.07	72.51
TTA**+OrthoReg** (ours)	**6.66**	62.19	**4.77**	65.13	**3.83**	72.87
Δ	-0.70	+0.11	-1.91	-0.36	-1.24	+0.36
Attention-Only Fine-tuning [11]	19.11	64.82	19.01	67.67	24.85	76.42
ATT-FT**+OrthoReg** (ours)	**10.75**	62.18	**10.63**	64.10	**11.47**	73.17
Δ	-8.36	-2.64	-8.38	-3.57	-13.38	-3.25
LoRA-ATT	16.85	63.23	19.44	67.28	21.23	75.41
LoRA-ATT**+OrthoReg** (ours)	**14.59**	61.68	**17.25**	67.08	**10.10**	72.19
Δ	-2.26	-1.55	-2.19	-0.20	-11.13	-3.22

Table 3. The minimum average Target Accuracy (Tar.Acc.) achievable while maintaining at least 80% of the zero-shot accuracy on the ImageNet control task (Con.Acc.). Our proposed orthogonal regularization (+OrthoReg) shows a consistent and significant improvement in forgetting the target task. An asterisk (*) denotes the best (lowest) target accuracy for each model architecture.

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 [9]	11.97	54.43	11.65	59.20	12.67	70.59
Non-lin. FT**+OrthoReg** (ours)	**10.24**	57.06	**10.40**	61.39	**9.30**	72.33
Δ	-1.73	+2.63	-1.25	+2.19	-3.37	+1.74
Tangent Task Arithmetic [16]	5.70	60.76	5.61	64.53	2.84	70.81
TTA**+OrthoReg** (ours)	**3.26**	59.26	**2.10**	62.61	**1.86**	70.23
Δ	-2.44	-1.50	-3.51	-1.92	-0.98	-0.58
Attention-Only Fine-tuning [11]	19.11	64.82	19.01	67.67	24.85	76.42
ATT-FT**+OrthoReg** (ours)	**7.23**	58.38	**8.08**	61.21	**8.12**	68.46
Δ	-11.88	-6.44	-10.93	-6.46	-16.73	-7.96
LoRA-ATT	15.58	62.4	15.83	62.40	21.23	75.41
LoRA-ATT**+OrthoReg** (ours)	**11.00**	58.47	**9.19**	60.41	**7.68**	69.83
Δ	-4.58	-3.93	-6.64	-1.99	-13.55	-5.58

Results and Analysis. The results, summarized in Table 1 are organized to highlight the efficiency trade-offs between different full-parameter fine-tuning strategies and their parameter-efficient counterparts.

The primary comparison focuses on the full fine-tuning methods. Standard Non-linear Fine-tuning (Non-lin. FT) serves as our baseline, completing training in 158.21 minutes and consuming 42589.22 MB of peak GPU memory. In stark contrast, TTA [16], which operates on a linearized model, is substantially more resource-intensive. It requires 280.86 minutes (a 77.5% increase in time) and 68031.34 MB of memory (a 59.7% increase), confirming that its reliance on Jacobian computations imposes a significant computational burden.

Our proposed OrthoReg, when applied to Non-lin. FT, introduces only a moderate overhead for its regularization calculations, resulting in a total cost of 177.04 minutes and 44500.27 MB of memory during the training phase. Crucially, this is significantly more efficient than TTA in both time and memory, while achieving superior or comparable

task-addition performance as shown in the main text and the last column of Table 1 (*e.g.*, for ViT-L-14, Non-lin. FT + OrthoReg achieves 88.23% Abs.Acc. vs. TTA's 86.19%). This demonstrates that OrthoReg provides a more efficient path to enforcing the properties that benefit task arithmetic.

This efficiency advantage is also evident in the parameter-efficient setting. As shown in the lower section of Table 1, applying OrthoReg to ATT-FT baseline results in only a minimal increase in computational cost. The training time rises modestly from 126.28 to 132.96 minutes, and peak memory usage increases marginally from 36591.06 MB to 36976.50 MB. However, the performance increases considerably from 87.81% to 90.41%. This demonstrates that the substantial performance improvements gained from OrthoReg come at a very low computational price, further highlighting its practicality.

In conclusion, these experiments provide strong empirical evidence that OrthoReg achieves the goal of promoting task vector orthogonality more efficiently than TTA. This efficiency, combined with the superior performance demonstrated in our main results, establishes OrthoReg as a more effective and accessible tool for reliable task arithmetic.

I. Experiments Details

The Normalized Accuracy (Norm.Acc.) metric evaluates the performance of the merged multi-task model (θ_{MT}) relative to individually fine-tuned single-task models (θ_t^*). It is defined as the average of the performance ratios across all T tasks. A score of 100% indicates that the merged model performs, on average, on par with the individual specialist models, suggesting a successful composition with minimal negative interference.

The formula is given by,

$$\text{Norm.Acc.} = \left(\frac{1}{T} \sum_{t=1}^{T} \frac{\text{acc}(\theta_{MT}, \mathcal{D}_t)}{\text{acc}(\theta_t^*, \mathcal{D}_t)} \right) \times 100\%, \quad (109)$$

where T is the total number of tasks being merged, $\text{acc}(\theta_{MT}, \mathcal{D}_t)$ is the accuracy of the merged model on test set for task t and $\text{acc}(\theta_t^*, \mathcal{D}_t)$ is the accuracy of the model fine-tuned only on task t, evaluated on its own test set.

This definition is consistent with the evaluation protocol established in prior work [9, 11, 16].

J. More Experimental Results

J.1. Detailed Visualization of Orthogonality

To provide comprehensive empirical support for the claim made in Section 4.2.3, this part presents a detailed visualization of the weight vector angle distributions for all linear layers within the pre-trained CLIP ViT-B/16 model. Figure 1 displays the histograms for each weight matrix.

As illustrated in Figure 1, a clear and consistent pattern emerges across the model's layers. We observe two distinct behaviors. (1) Embedding Layers. The first two subplots correspond to the patch_embedding and pos_embedding layers. These layers show broader, more Gaussian-like distributions, which is understandable given their unique function of mapping raw inputs into the initial embedding space. As our analysis primarily concerns the transformation dynamics within the main model body, these layers are not the central focus of our study. (2) Transformer Blocks. In stark contrast, nearly all subsequent weight matrices, which constitute the core computational machinery of the model, including the query, key, value (QKV) projections, attention output projections (proj), and MLP layers within all 12 transformer blocks, exhibit angle distributions that are sharply and narrowly peaked at 90 degrees.

This detailed, per-layer visualization provides robust evidence that near-orthogonality is not an isolated occurrence but a pervasive geometric property of the pre-trained model's core processing blocks.

J.2. Detailed Per-Task Performance Visualization

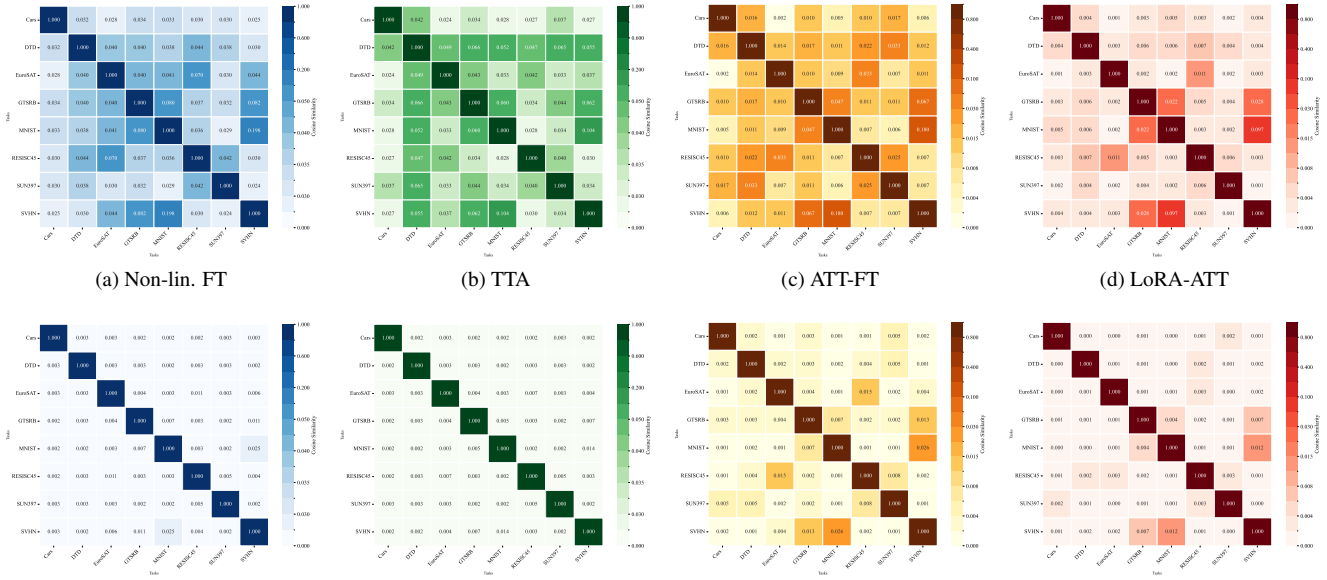
This section supplements the analysis in Section 5.2 by providing the comprehensive per-task performance radar charts for all evaluated architectures: ViT-L-14, ViT-B-16, and ViT-B-32. The results shown in Figure 2 reinforce and expand upon the findings presented in the main body. We consistently observe that applying OrthoReg (the blue area) leads to a larger performance footprint compared to the baselines (the red area) across the vast majority of tasks, methods, and architectures. This further corroborates our claim that OrthoReg is a model-agnostic regularizer that effectively mitigates task interference, leading to broad performance gains in multi-task scenarios.

J.3. Details About Task Negation

In this section, we provide additional details for the task negation experiments discussed in Section 5.3. When the accuracy requirement on the control task is further relaxed, such as to 90% (see Table 2) or 80% (see Table 3), the effect of task negation becomes progressively stronger, resulting in lower accuracy on the target task. Moreover, our OrthoReg regularizer can further enhance the negation effect while still meeting the control-task accuracy threshold. In some cases, it even improves control-task accuracy while reducing target-task accuracy. These results demonstrate that our method effectively disentangles task-specific feature information, substantially reducing undesired interference with non-target tasks during the task negation process.

J.4. Visualization of Task Vector Similarity

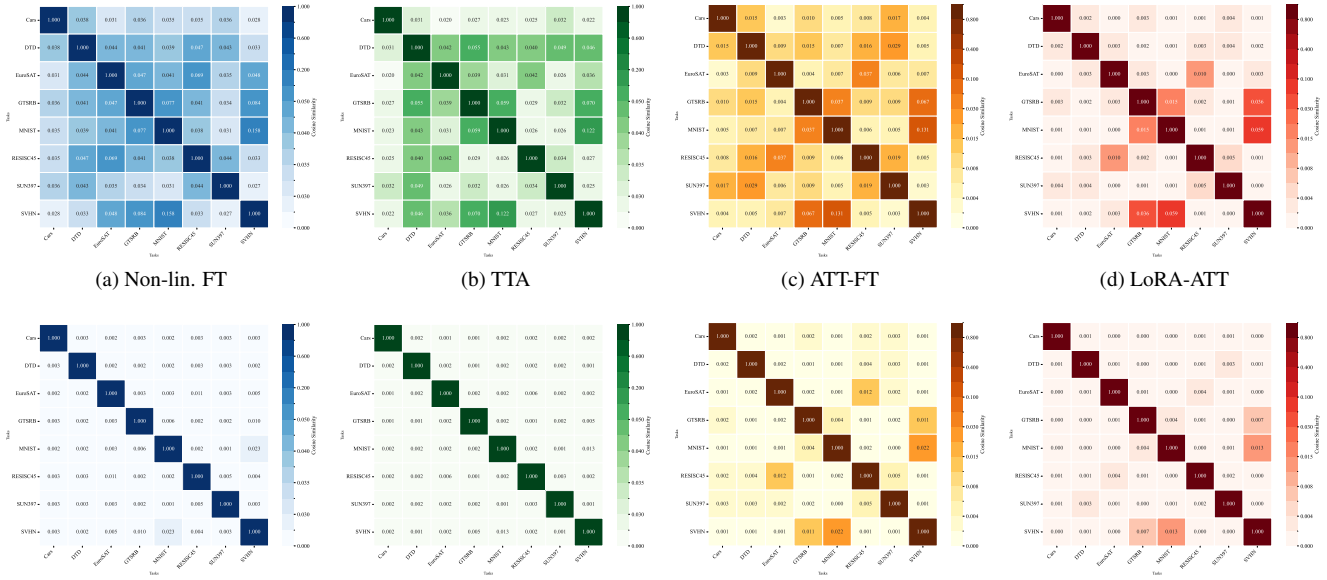
To supplement the analysis in Section 5.4, this section provides additional task vector similarity heatmaps. These fig-



(a) Non-lin. FT (b) TTA (c) ATT-FT (d) LoRA-ATT

(e) Non-lin. FT+OrthoReg (f) TTA+OrthoReg (g) ATT-FT+OrthoReg (h) LoRA-ATT+OrthoReg

Figure 3. Pairwise cosine similarity heatmaps of task vectors for ViT-B-16 across different methods. The top row shows the baseline methods, where significant off-diagonal correlation (brighter colors) is visible. The bottom row shows the same methods with our OrthoReg regularizer. The consistently darker off-diagonal values in the bottom row provide strong empirical validation that OrthoReg successfully produces more orthogonal task vectors, mitigating a key source of task interference.

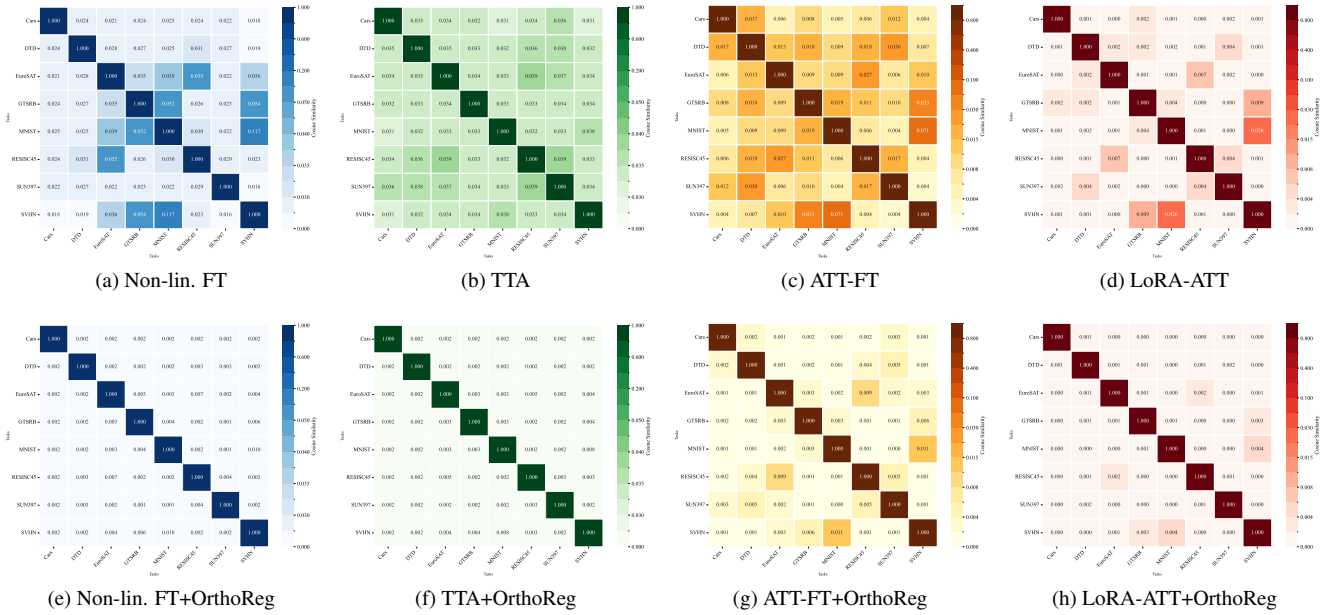


(a) Non-lin. FT (b) TTA (c) ATT-FT (d) LoRA-ATT

(e) Non-lin. FT+OrthoReg (f) TTA+OrthoReg (g) ATT-FT+OrthoReg (h) LoRA-ATT+OrthoReg

Figure 4. Pairwise cosine similarity heatmaps of task vectors for VIT-B-32 across different methods. The top row shows the baseline methods, where significant off-diagonal correlation (brighter colors) is visible. The bottom row shows the same methods with our OrthoReg regularizer. The consistently darker off-diagonal values in the bottom row provide strong empirical validation that OrthoReg successfully produces more orthogonal task vectors, mitigating a key source of task interference.

ures (Figure 3, Figure 4, Figure 5) illustrate the effect of OrthoReg across different baseline methods and model architectures, consistently demonstrating that our method produces more orthogonal task vectors.



(a) Non-lin. FT (b) TTA (c) ATT-FT (d) LoRA-ATT

(e) Non-lin. FT+OrthoReg (f) TTA+OrthoReg (g) ATT-FT+OrthoReg (h) LoRA-ATT+OrthoReg

Figure 5. Pairwise cosine similarity heatmaps of task vectors for ViT-L-14 across different methods. The top row shows the baseline methods, where significant off-diagonal correlation (brighter colors) is visible. The bottom row shows the same methods with our OrthoReg regularizer. The consistently darker off-diagonal values in the bottom row provide strong empirical validation that OrthoReg successfully produces more orthogonal task vectors, mitigating a key source of task interference.

Table 4. Performance comparison of different LoRA module configurations with and without orthogonality regularization. The last row under each module shows the improvement (Δ) from OrthoReg.

LoRA Modules	**Finetuning Mode**	**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.(↑)
qkvofp (All)	LoRA	73.03	81.89	75.18	81.83	85.44	90.98
	+OrthoReg	74.71	84.31	78.07	85.23	87.69	93.67
	Δ	**+1.68**	**+2.42**	**+2.89**	**+3.40**	**+2.25**	**+2.69**
qkvo– (Attn All)	LoRA	73.95	84.19	76.31	84.04	87.13	93.49
	+OrthoReg	76.20	86.55	80.48	91.97	89.14	95.49
	Δ	**+2.25**	**+2.36**	**+4.17**	**+7.93**	**+2.01**	**+2.00**
qkv— (Q,K,V)	LoRA	70.14	80.98	74.69	82.82	85.03	91.67
	+OrthoReg	73.68	84.40	78.10	86.27	87.56	93.97
	Δ	**+3.54**	**+3.42**	**+3.41**	**+3.45**	**+2.53**	**+2.30**
q-v— (Q,V only)	LoRA	69.25	80.30	75.15	83.35	84.39	91.11
	+OrthoReg	72.71	83.77	77.03	85.37	86.58	93.29
	Δ	**+3.46**	**+3.47**	**+1.88**	**+2.02**	**+2.19**	**+2.18**
—-fp (MLP only)	LoRA	69.19	78.01	71.24	78.02	81.98	87.78
	+OrthoReg	68.92	77.77	72.05	78.72	82.80	88.13
	Δ	**-0.27**	**-0.24**	**+0.81**	**+0.70**	**+0.82**	**+0.35**

J.5. Detailed Ablation Study on LoRA Components

This part provides additional details and results to supplement the LoRA ablation study presented in Section 5.1.

J.5.1. Rationale for Module Selection

The selection of different module subsets for our LoRA-based ablation study was designed to systematically probe the effect of OrthoReg on distinct functional components of the Vision Transformer.

- All Tunable Layers. qkvofp: This represents the most comprehensive PEFT approach, applying LoRA to all available linear layers (attention and MLP). It serves as a baseline to evaluate the effect of tuning the entire model in a parameter-efficient manner.
- MLP Layers Only. —fp: This configuration isolates the FFN or MLP blocks. By tuning only these layers, we can assess their specific contribution to task adaptation and how OrthoReg influences them in isolation.
- Attention Subsets. qkvo–, qkv—, and q-v—: These configurations focus on the multi-head self-attention mechanism, which is widely considered crucial for capturing task-specific patterns.
  - qkvo– tunes all four projection matrices (query, key, value, and output), representing a full intervention within the attention block.
  - qkv— omits the output projection, allowing us to gauge its importance.
  - q-v— is a particularly important configuration. Prior work [24] has identified that fine-tuning only the query and value matrices can be a highly effective and parameter-efficient strategy.

By comparing these configurations, we can draw nuanced conclusions about where task-specific knowledge is stored and how promoting orthogonality in different components contributes to the final performance of task arithmetic.

J.5.2. results

Table 4 summarizes the effect of applying OrthoReg across different LoRA module configurations. Overall, OrthoReg consistently improves performance in all settings except the MLP-only configuration. The largest gains appear in attention-related modules , such as qkvo– , with improvements up to +4.17 points on ViT-B-16. This aligns with the common understanding that attention layers carry most of the task-specific information, and orthogonalizing their updates most effectively reduces feature entanglement.

Full-layer tuning (qkvofp) also benefits substantially from OrthoReg, indicating that larger tunable subspaces allow orthogonality constraints to better isolate task-relevant directions. The Q,V-only configuration (q-v—), previously identified as an efficient tuning strategy, also shows stable improvements when combined with OrthoReg.

The only exception is the MLP-only setup, where OrthoReg slightly reduces accuracy on smaller models. This suggests that MLP layers contribute less task-specific variation, and enforcing orthogonality may occasionally restrict useful shared representations.

Overall, the results confirm that OrthoReg most strongly enhances the components responsible for task-discriminative behavior, leading to more accurate task vectors and more reliable task arithmetic.

References

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008. 11

[2] Martín Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1120–1128. JMLR.org, 2016. 8

[3] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. 4, 5

[4] H.S.M. Coxeter and S.L. Greitzer. *Geometry Revisited*. Mathematical Association of America, 1967. 8

[5] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. 8

[6] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012. 7, 10

[7] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 1

[8] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 791–800. Computer Vision Foundation / IEEE Computer Society, 2018. 4, 5

[9] Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 1, 12, 15, 16

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456. JMLR.org, 2015. 4, 5

[11] Ruochen Jin, Bojian Hou, Jiancong Xiao, Weijie J. Su, and Li Shen. Fine-tuning attention modules only: Enhancing weight disentanglement in task arithmetic. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 12, 15, 16

[12] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. 1

[13] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, December 2013*, 2013. 14

[14] David G Luenberger, Yinyu Ye, et al. *Linear and nonlinear programming*. Springer, 1984. 9

[15] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 8

[16] Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, pages 66727–66754. Curran Associates, Inc., 2023. 1, 11, 12, 15, 16

[17] T.J. Rivlin. *The Chebyshev Polynomials*. Wiley, 1974. 8

[18] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 1976. 8

[19] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2488–2498, 2018. 5

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. 1

[21] Feng Xiong, Runxi Cheng, Wang Chen, Zhanqiu Zhang, Yiwen Guo, Chun Yuan, and Ruifeng Xu. Multi-task model merging via adaptive weight disentanglement. *CoRR*, abs/2411.18729, 2024. 3

[22] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 1

[23] Yiting Yang, Hao Luo, Yuan Sun, Qingsen Yan, Haokui Zhang, Wei Dong, Guoqing Wang, Peng Wang, Yang Yang, and Hengtao Shen. Efficient adaptation of pre-trained vision transformer underpinned by approximately orthogonal fine-tuning strategy. *CoRR*, abs/2507.13260, 2025. 8

[24] Maxime Zanella and Ismail Ben Ayed. Low-rank few-shot adaptation of vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024*, pages 1593–1603. IEEE, 2024. 19

[25] Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Moefication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 877–890. Association for Computational Linguistics, 2022. 1