On the Emergence of Cross-Task Linearity in Pretraining-Finetuning Paradigm

Zhanpeng Zhou [*1] **Zijun Chen** [*1 2] **Yilan Chen** [3] **Bo Zhang** [2] **Junchi Yan** [1]

Abstract

The pretraining-finetuning paradigm has become the prevailing trend in modern deep learning. In this work, we discover an intriguing linear phenomenon in models that are initialized from a common pretrained checkpoint and finetuned on different tasks[1], termed as *Cross-Task Linearity (CTL)*. Specifically, we show that if we linearly interpolate the weights of two finetuned models, the features in the weight-interpolated model are often approximately equal to the linear interpolation of features in two finetuned models at each layer. We provide comprehensive empirical evidence supporting that CTL consistently occurs for finetuned models that start from the same pretrained checkpoint. We conjecture that in the pretraining-finetuning paradigm, neural networks approximately function as linear maps, mapping from the parameter space to the feature space. Based on this viewpoint, our study unveils novel insights into explaining model merging/editing, particularly by translating operations from the parameter space to the feature space. Furthermore, we delve deeper into the root cause for the emergence of CTL, highlighting the role of pretraining. We released our source code at https://github.com/zzp1012/Cross-Task-Linearity.

1. Introduction

Pretrained models have become the fundamental infrastructure of modern machine learning systems, and finetuning has evolved as a predominant way for adapting the pretrained model to various downstream tasks. Despite the prominent success, our understanding of the pretraining-finetuning paradigm still lags behind. There is a growing interest in unraveling the hidden mechanisms of pretraining and finetuning, particularly in human preference alignment (Ouyang et al., 2022), interpretability (Elhage et al., 2021; Olsson et al., 2022), and AI ethics (Weidinger et al., 2021) etc.

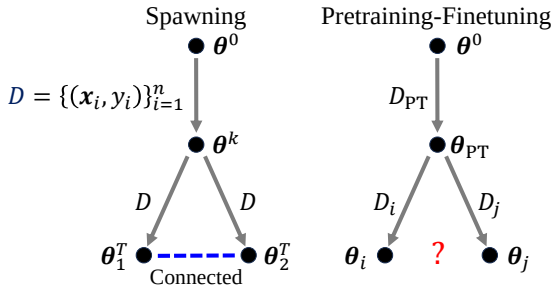


Figure 1. The spawning method and the pretraining-finetuning paradigm. $\boldsymbol{\theta}^0$ denotes random initialization of the network weights. For spawning, the network is first trained for k epochs to get $\boldsymbol{\theta}^k$, then spawned into two copies and updated until convergence to get $\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T$. Note $\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T$ are trained on same task but with different SGD noise. With a proper chosen k , $\boldsymbol{\theta}_1^T$ and $\boldsymbol{\theta}_2^T$ can satisfy LMC and LLFC. For pretraining-finetuning, the network is first trained on pretraining task $\mathcal{D}_{\mathrm{PT}}$ to get $\boldsymbol{\theta}_{\mathrm{PT}}$. Then $\boldsymbol{\theta}_{\mathrm{PT}}$ is finetuned on $\mathcal{D}_i$ and $\mathcal{D}_j$ to get $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$. $\mathcal{D}_i$ and $\mathcal{D}_j$ can be different.

Recent works on *Linear Mode Connectivity (LMC)* (Nagarajan & Kolter, 2019; Frankle et al., 2020) and *Layerwise Linear Feature Connectivity (LLFC)* (Zhou et al., 2023) shed light on understanding the training dynamics and hidden mechanisms of neural networks. LMC depicts a linear path in the parameter space of a network where the loss remains approximately constant (see Definition 3.1). In other words, linearly interpolating the weights of two different models, which are of the same architecture and trained on the same task, could lead to a new model that achieves similar performance as the two original models. LLFC indicates that the features in the weight-interpolated model are proportional to the linear interpolation of the features in the two original models (see Definition 3.2). Frankle et al. (2020) observed LMC for networks that are jointly trained for a short time before undergoing independent training on the same task, termed as *spawning method* (see Figure 1). Zhou et al. (2023) discovered the models that linearly connected

[*]Equal contribution [1]School of Artificial Intelligence & Department of Computer Science and Engineering & MoE Lab of AI, Shanghai Jiao Tong University, Shanghai, China [2]Shanghai Artificial Intelligence Laboratory [3]Computer Science and Engineering, University of California San Diego. Correspondence to: Bo Zhang <bo.zhangzx@gmail.com>, Junchi Yan <yanjunchi.sjtu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]In this work, task refers to dataset used for finetuning unless otherwise stated.

	Performance	Equality	Task
LMC (Frankle et al., 2020)	Approx. equal	Approx. equal	Single
LLFC (Zhou et al., 2023)	Flexible	Proportional to	Single
CTL (Ours)	Flexible	Approx. equal	Multiple

Table 1. Comparison: LMC requires models with approximately equal performance; LLFC depicts a proportional relation only. Both LMC and LLFC focus on models trained on the same task. CTL extends LLFC to models finetuned on different tasks.

in the loss landscape are also linearly connected in feature space, i.e., satisfy LMC and LLFC simultaneously.

As shown in Figure 1, a connection is identified between the pretraining-finetuning paradigm and the spawning method, as both entail training models from a same pretrained checkpoint. Therefore, a natural question arises: are models, initialized from a common pretrained checkpoint[2] but finetuned on *different tasks*, linearly connected in the loss landscape or feature space, akin to the models obtained by the spawning method satisfying LMC and LLFC?

In this work, we discover that the finetuned models are linearly connected in internal features even though there is no such connectivity in the loss landscape, i.e., LLFC holds but LMC not. Indeed, we identify a stronger notion of linearity than LLFC: if we linearly interpolate the weights of two models that finetuned on different tasks, the features in the weight-interpolated model are *approximately equal* to the linear interpolation of features in the two finetuned models at each layer, namely *Cross-Task Linearity (CTL)* as termed in this paper (see comparison among LMC, LLFC, and CTL in Table 1). To be precise, let $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ be the weights of two finetuned models, and $f^{(\ell)}(\boldsymbol{\theta})$ be the features in the model of weights $\boldsymbol{\theta}$ at ℓ-th layer. We say that $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ satisfy CTL if $\forall \ell, \forall \alpha \in [0, 1]$,

$$f^{(\ell)}(\alpha \boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) \approx \alpha f^{(\ell)}(\boldsymbol{\theta}_i) + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_j).$$

CTL may not be universal for arbitrary networks trained on tasks, yet we provide comprehensive empirical evidence supporting that CTL consistently occurs for the finetuned models across a wide range of settings. We conjecture that in the pretraining-finetuning paradigm, neural networks can roughly function as linear maps, mapping from the parameter space to the feature space.

Based on the observed CTL in the pretraining-finetuning paradigm, we obtain novel insights into two widely-used model merging/editing techniques: *model averaging* (Izmailov et al., 2018; Matena & Raffel, 2022; Rame et al., 2023; 2022; Wortsman et al., 2022a;b) and *task arithmetic* (Ilharco et al., 2022; 2023; Ortiz-Jimenez et al., 2023).

i) Model averaging takes the average of weights of multiple models, which are finetuned on the same task but with different hyperparameter configurations, so as to improve accuracy and robustness. We explain the averaging of weights as the averaging of features at each layer, building a stronger connection between model averaging and logits ensemble than before.

ii) Task arithmetic merges the weights of models, that are finetuned on different tasks, via simple arithmetic operations, shaping the behaviour of the resulting model accordingly. We translate the arithmetic operation in the parameter space into the operations in the feature space, yielding a feature-learning explanation for task arithmetic.

Furthermore, we delve deeper into the root cause of CTL. We empirically investigate various factors contributing to the holding of CTL, highlighting the role of pretraining. We also take a primary attempt to prove CTL and find that the emergence of CTL is associated with the flatness of the network landscape and the distance between the weights of two finetuned models.

In summary, our work reveals a linear connection between finetuned models, offering significant insights into model merging/editing techniques. This, in turn, advances our understanding of underlying mechanisms of pretraining and finetuning from a feature-centric perspective.

2. Related Work

(Linear) Mode Connectivity. Freeman & Bruna (2017); Draxler et al. (2018); Garipov et al. (2018) noted Mode Connectivity (MC), where different minima in the loss landscape can be connected by a non-linear path of nearly constant loss. Nagarajan & Kolter (2019); Frankle et al. (2020) discovered that the path of nearly constant loss can be linear, for models that are jointly trained for a short time before undergoing independent training, termed Linear Mode Connectivity (LMC). Fort et al. (2020) analyzed LMC from a perspective of the Neural Tangent Kernel dynamics. Entezari et al. (2022); Ainsworth et al. (2023) showed that even independently trained networks can satisfy LMC after accounting for permutation invariance. Studies (Liang et al., 2018; Venturi et al., 2019; Nguyen et al., 2019; Nguyen, 2019; Kuditipudi et al., 2019; Ferbach et al., 2023; Zhao et al., 2023; Zhou et al., 2023) have attempted to prove (linear) mode connectivity from various perspectives. Adilova et al. (2023) studied the layerwise behaviour of LMC under federated learning settings. Qin et al. (2022) studied MC in the context of pretrained language models. Mirzadeh et al. (2021); Juneja et al. (2023) investigated finetuning from the lens of LMC. Zhou et al. (2023) identified a stronger connectivity than LMC, namely Layerwise Linear Feature Connectivity (LLFC), and observed LLFC always co-occurs with LMC. (Chen et al., 2024) expand the concept of feature similarity with LLFC.

[2]In this work, we consider finetuned models that start from a common pretrained checkpoint.

Model Merging/Editing. Recent studies find averaging the parameters of finetuned models over the same task leads to improved performance and generalization abilities (Izmailov et al., 2018; Matena & Raffel, 2022; Rame et al., 2023; 2022; Wortsman et al., 2022a;b). Moreover, the averaging of weights from models finetuned over tasks enables multi-task abilities (Ilharco et al., 2022; Li et al., 2022; Yadav et al., 2023; Jin et al., 2023; Stoica et al., 2023; Yu et al., 2023). Singh & Jaggi (2020); Liu et al. (2022) show that the weights of independently trained neural networks can be merged after aligning the neurons. Moreover, Ilharco et al. (2023); Ortiz-Jimenez et al. (2023) extend the simple averaging to arithmetic operations in the parameter space, enabling a finer-grained control of the model behaviours.

3. Backgrounds and Preliminaries

Notation Setup. Unless explicitly stated otherwise, we consider a classification dataset/task, denoted as $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^{d_0}$ is the input and $y_i \in [c]$ is the label of the i-th datapoint. Here, d_0 is the input dimension, $[c] = \{1, 2, \ldots, c\}$ and c is the number of classes. We use $\boldsymbol{X} \in \mathbb{R}^{d_0 \times n}$ to stack all the input data into a matrix.

We consider an L-layer neural network defined as $f(\boldsymbol{\theta}; \boldsymbol{x})$, where $\boldsymbol{\theta}$ denotes the model parameters, $\boldsymbol{x}$ is the input, and $f(\boldsymbol{\theta}; \boldsymbol{x}) \in \mathbb{R}^c$. $f^{(\ell)}(\boldsymbol{\theta}; \boldsymbol{x}) \in \mathbb{R}^{d_\ell}$ represents the internal feature (post-activation) in the network at the ℓ-th layer. Here, d_ℓ denotes the dimension of the ℓ-th layer ($0 \leq \ell \leq L$) and $f^{(L)}(\boldsymbol{\theta}; \boldsymbol{x}) = f(\boldsymbol{\theta}; \boldsymbol{x})$. For an input matrix $\boldsymbol{X}$, we use $f^{(\ell)}(\boldsymbol{\theta}; \boldsymbol{X}) \in \mathbb{R}^{d_\ell \times n}$ to denote the collection of features on all the datapoints. When $\boldsymbol{X}$ is clear from the context, we simply write $f^{(\ell)}(\boldsymbol{\theta}) = f^{(\ell)}(\boldsymbol{\theta}; \boldsymbol{X})$. The expected loss on dataset $\mathcal{D}$ is denoted by $\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}}\left[L\left(f\left(\boldsymbol{\theta}; \boldsymbol{x}\right), y\right)\right]$, where L represents the loss function. Our analysis focuses on models trained on a training set, with all investigations evaluated on a separate test set.

Linear Mode Connectivity (LMC).

Definition 3.1 (**Linear Mode Connectivity** (Nagarajan & Kolter, 2019; Frankle et al., 2020))**.** Given dataset $\mathcal{D}$ and two modes[3] $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ such that $\mathcal{L}(\boldsymbol{\theta}_i) \approx \mathcal{L}(\boldsymbol{\theta}_j)$ on $\mathcal{D}$, we say $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ are linearly connected in the loss landscape if they satisfy $\forall \alpha \in [0, 1]$,

$$\mathcal{L}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) \approx \mathcal{L}(\boldsymbol{\theta}_i) \approx \mathcal{L}(\boldsymbol{\theta}_j).$$

As Definition 3.1 shows, LMC indicates different optima can be connected via a simple linear path of nearly constant loss. Previous studies (Frankle et al., 2020) observed LMC for networks that start from a common pretrained checkpoint and undergo independent training on the same task until convergence, commonly referred as *spawning method* (Fort et al., 2020; Zhou et al., 2023).

[3]A *mode* refers to the obtained solution after training.

Layerwise Linear Feature Connectivity (LLFC).

Definition 3.2 (**Layerwise Linear Feature Connectivity** (Zhou et al., 2023))**.** Given dataset $\mathcal{D}$ and two modes $\boldsymbol{\theta}_i, \boldsymbol{\theta}_j$ of an L-layer neural network f, the modes $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ are said to be linearly connected in feature space on $\mathcal{D}$ if $\forall \ell \in [L], \forall \alpha \in [0, 1]$ such that,

$$f^{(\ell)}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) \propto \alpha f^{(\ell)}(\boldsymbol{\theta}_i) + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_j).$$

In Definition 3.2, LLFC states that the features (post-activation) in the interpolated model $\boldsymbol{\theta}_\alpha = \alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j$ are proportional to the linear interpolation of the features in $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ at each layer.

Zhou et al. (2023) introduced LLFC, which defines a stronger notion of linear connectivity than LMC, and noted its consistent co-occurrence with LMC. Specifically, if two modes $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ satisfy LMC, then they also approximately satisfy LLFC. Moreover, it can be proven that LLFC directly induces LMC for models with equal loss (see Theorem 3.3). Therefore, they believed that LLFC is a more fundamental property than LMC.

Theorem 3.3 (**LLFC Induces LMC** (Proof in Appendix B.2))**.** *Given dataset $\mathcal{D}$, convex loss function L, and two modes $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ with equal loss on $\mathcal{D}$, i.e., $\mathcal{L}(\boldsymbol{\theta}_i) = \mathcal{L}(\boldsymbol{\theta}_j)$, suppose the two modes $\boldsymbol{\theta}_i, \boldsymbol{\theta}_j$ satisfy LLFC on $\mathcal{D}$ with exact equality, then for all $\alpha \in [0, 1]$,*

$$\mathcal{L}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) \leq \mathcal{L}(\boldsymbol{\theta}_i) = \mathcal{L}(\boldsymbol{\theta}_j).$$

Main Experimental Setup. In Section 4.1, we conduct experiments on standard continue learning benchmark datasets, including Rotated MNIST (LeCun et al., 1998) and Split CIFAR-100 (Krizhevsky et al., 2009), with MLP and ResNet-18 (He et al., 2016). We follow the same training procedures and hyper-parameters as in Mirzadeh et al. (2021). In Sections 4.2 and 4.3, we directly adopt the finetuned ViTs (Dosovitskiy et al., 2020)/T5s (Raffel et al., 2020) checkpoints open-sourced by Wortsman et al. (2022a); Ilharco et al. (2023) and perform experiments on various image and text datasets. Due to space limit, we defer more experimental settings to Appendix C.1.

4. Cross-Task Linearity

In this section, we provide empirical evidence indicating that the finetuned models are linearly connected in the feature space, even though there is no such connectivity in the loss landscape, i.e. LLFC holds but LMC not. Taking a step further, we identify a stronger notion of linearity, namely Cross-Task Linearity (CTL), which approximately characterizes neural networks as linear maps in the pretraining-finetuning paradigm. From these observations, we offer novel insights into model averaging and task arithmetic.

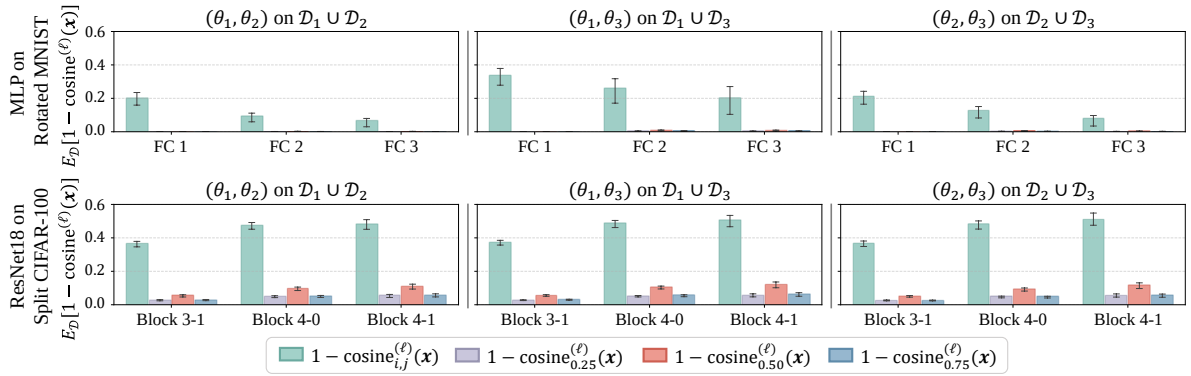


Figure 2. Verification of CTL. Compare $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}^{(\ell)}_{\alpha}(\boldsymbol{x})]$ with $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}^{(\ell)}_{i,j}(\boldsymbol{x})]$. Here, $\{\boldsymbol{\theta}_i\}_{i=1}^3$ and $\{\mathcal{D}_i\}_{i=1}^3$ denotes finetuned models and their corresponding downstream tasks. For Rotated MNIST, models are pretrained on MNIST and finetuned on variants of MNIST where digits are at different angles. For Split CIFAR-100, models are pretrained and finetuned on disjoint sets of 5 classes from CIFAR-100. The bottom and top of the error bar represent the lower and upper quartile of the values across the dataset, respectively. The results are reported for last three layers/blocks, with $\alpha \in \{0.25, 0.5, 0.75\}$. More results in Appendix C.3.

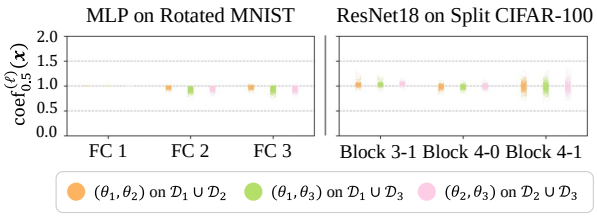


Figure 3. Verification of CTL. Distribution of $\text{coef}^{(\ell)}_{\alpha}(\boldsymbol{x})$ across the dataset. Here, $\alpha = 0.5$. $\{\boldsymbol{\theta}_i\}_{i=1}^3$ and $\{\mathcal{D}_i\}_{i=1}^3$ denotes finetuned models and their corresponding downstream tasks. We follow the same training settings as in Figure 2. The results are reported for last three layers/blocks. More results in Appendix C.3.

4.1. Extend LMC and LLFC to CTL

LMC Fails in the Pretraining-Finetuning Paradigm. LMC might not hold for the finetuned models, as the preconditions of Definition 3.1 are not met, i.e., the models finetuned on different tasks might not have approximately equal optimal loss on the same task. Indeed, the studies on LMC are motivated by the interest in studying optima of the same loss landscape. It means LMC depicts the property of models trained on the same task. Therefore, it is clear that finetuned models might not satisfy LMC, even on the pretraining task, where catastrophic forgetting may occur (McCloskey & Cohen, 1989). Similar phenomena were observed in (Mirzadeh et al., 2021; Juneja et al., 2023).

LLFC Holds in the Pretraining-Finetuning Paradigms. Despite no connectivity in the loss landscape, we surprisingly find that the finetuned models are linearly connected in the feature space. Here, we extend the original LLFC to the cases where models are finetuned on different tasks.

To verify LLFC for finetuned models, we conduct extensive experiments across a range of settings. Specifically, we consider a set of finetuned models[4], $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i\}_{i=1}^k$, which are initialized from a common pretrained checkpoint $\boldsymbol{\theta}_{\text{PT}}$ but finetuned on different tasks. Here, the downstream tasks for finetuning are denoted as $\{\mathcal{D}_i\}_{i=1}^k$, respectively. Then, for each pair of finetuned models $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \in \boldsymbol{\Theta}^2$, on each datapoint $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}_i \cup \mathcal{D}_j$, we measure the cosine similarity between the features in the weight-interpolated model $\boldsymbol{\theta}_\alpha = \alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j$ and the linear interpolation of the features in $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ at each layer ℓ, denoted as $\text{cosine}^{(\ell)}_{\alpha}(\boldsymbol{x}) = \cos[f^{(\ell)}(\boldsymbol{\theta}_\alpha;\boldsymbol{x}), \alpha f^{(\ell)}(\boldsymbol{\theta}_i;\boldsymbol{x}) + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_j;\boldsymbol{x})]$. We compare $\text{cosine}^{(\ell)}_{\alpha}$ to the baseline cosine similarity between the features in $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ in the same layer, i.e, $\text{cosine}^{(\ell)}_{i,j}(\boldsymbol{x}) = \cos[f^{(\ell)}(\boldsymbol{\theta}_i;\boldsymbol{x}), f^{(\ell)}(\boldsymbol{\theta}_j;\boldsymbol{x})]$. In Figure 2, the values of $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}^{(\ell)}_{\alpha}(\boldsymbol{x})]$ consistently approach 0 across a range of layers, α, and various pairs of $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$, under different task settings. The small error bars indicate a consistent behaviour across each datapoint in $\mathcal{D}_i \cup \mathcal{D}_j$. Additionally, the values of baseline $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}^{(\ell)}_{i,j}(\boldsymbol{x})]$ deviate from 0, excluding the trivial scenario where $f^{(\ell)}(\boldsymbol{\theta}_i)$ and $f^{(\ell)}(\boldsymbol{\theta}_j)$ are already close enough. The results confirm that LLFC holds in the pretraining-finetuning paradigm.

CTL Occurs in the Pretraining-Finetuning Paradigm. Building upon the observations of LLFC for finetuned models, we identify a stronger notion of linearity than LLFC, termed as Cross-Task Linearity (CTL). Precisely, given a pair of finetuned models $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \in \boldsymbol{\Theta}^2$ and downstream tasks $\mathcal{D}_i$ and $\mathcal{D}_j$ respectively, we say them satisfy CTL on $\mathcal{D}_i \cup \mathcal{D}_j$ if $\forall \ell \in [L], \forall \alpha \in [0,1]$,

$$f^{(\ell)}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) \approx \alpha f^{(\ell)}(\boldsymbol{\theta}_i) + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_j).$$

Beyond LLFC, CTL enforces the approximate equality. We

[4]For simplicity, we often denote models of the same architecture as $\boldsymbol{\theta}$ instead of $f(\boldsymbol{\theta})$.

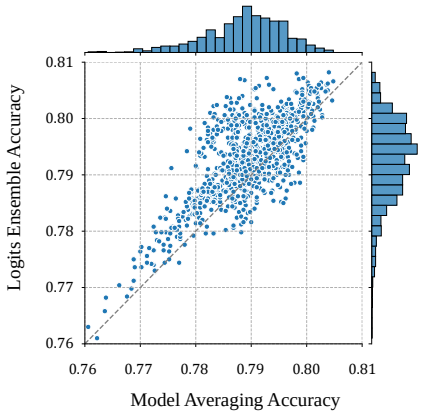


Figure 4. Linear correlation between the model averaging accuracy and the logits ensemble accuracy. Each datapoint represents three models fine-tuned on ImageNet with varying hyperparameters, denoted as $\{\boldsymbol{\theta}\}_{i=1}^3$. The x-axis represents accuracy of $f(\frac{1}{3}\sum_{i=1}^3 \boldsymbol{\theta}_i)$, while the y-axis represents accuracy of $\frac{1}{3}\sum_{i=1}^3 f(\boldsymbol{\theta}_i)$. The grey dashed line represents $y = x$.

have validated the features in weight-interpolated model $\boldsymbol{\theta}_\alpha$ and the linear interpolation of features in $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ have similar directions. To further validate CTL, we compare the length of their features at each layer ℓ. Specifically, for each pair of finetuned models $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \in \boldsymbol{\Theta}^2$, on each datapoint $\boldsymbol{x} \in \mathcal{D}_i \cup \mathcal{D}_j$, we measure $\mathrm{coef}_\alpha^{(\ell)}(\boldsymbol{x}) = \frac{\|f^{(\ell)}(\boldsymbol{\theta}_\alpha;\boldsymbol{x})\|\,\mathrm{cosine}_\alpha^{(\ell)}(\boldsymbol{x})}{\|\alpha f^{(\ell)}(\boldsymbol{\theta}_i;\boldsymbol{x})+(1-\alpha)f^{(\ell)}(\boldsymbol{\theta}_j;\boldsymbol{x})\|}$ [5]. In Figure 3, the values of $\mathrm{coef}_\alpha^{(\ell)}(\boldsymbol{x})$ are close to 1 across various layers and different pairs of $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$, under different task settings. Together with the results in Figure 2, we confirm that CTL often occurs for the finetuned models.

In summary, we find that neural networks approximately function as linear maps in the pretraining-finetuning paradigm, mapping from the parameter space to the feature space. This viewpoint enables us to study model merging/editing from a feature-learning perspective.

Conjecture 4.1 (Transitivity of CTL.). *Given models $\boldsymbol{\theta}_i, \boldsymbol{\theta}_j, \boldsymbol{\theta}_k$. We have $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_k)$ satisfy CTL if $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ and $(\boldsymbol{\theta}_j, \boldsymbol{\theta}_k)$ satisfy CTL.*

In addition, we conjecture the transitivity of CTL (see Conjecture 4.1). This is inferred from our results in Figures 2, 3 and 13 to 16 that $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_k)$ is observed to satisfy CTL when $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ and $(\boldsymbol{\theta}_j, \boldsymbol{\theta}_k)$ satisfy CTL. We will later leverage the transitivity of CTL to prove the theorems.

4.2. Insights into Model Averaging

Recent studies (Wortsman et al., 2022a;b) discovered that averaging the weights of multiple models fine-tuned on the same task but with different hyperparameter configurations often leads to improved accuracy and robustness. This approach, termed as *model averaging*, can be formulated as $f(\frac{1}{k}\sum_{i=1}^k \boldsymbol{\theta}_i)$. Here, $\{\boldsymbol{\theta}_i\}_{i=1}^k$ represents the set of finetuned models, and the downstream task for finetuning is denoted as $\mathcal{D}_{\mathrm{FT}}$. Alternatively, as another way to combine multiple models, the logits ensemble simply averages the outputs of different models, i.e., $\frac{1}{k}\sum_{i=1}^k f(\boldsymbol{\theta}_i)$. Both methods are effective in improving overall model performance in practice and indeed a linear correlation has been observed between the accuracy of model averaging and logits ensemble (see Figure 4). Here, we build a stronger connection between model averaging and logits ensemble in the feature space.

Specifically, we discover that the features in model averaging can be approximated by the averaging of features in each individual finetuned model, i.e., $\forall \ell \in [L]$,

$$f^{(\ell)}\left(\frac{1}{k}\sum_{i=1}^k \boldsymbol{\theta}_i\right) \approx \frac{1}{k}\sum_{i=1}^k f^{(\ell)}(\boldsymbol{\theta}_i). \tag{1}$$

We conduct extensive experiments to validate our discovery. Similar to Section 4.1, on each datapoint $\boldsymbol{x} \in \mathcal{D}_{\mathrm{FT}}$, we measure the cosine similarity between the features in model averaging $f^{(\ell)}(\frac{1}{k}\sum_{i=1}^k \boldsymbol{\theta}_i)$ and the averaging of features in each model $\frac{1}{k}\sum_{i=1}^k f^{(\ell)}(\boldsymbol{\theta}_i)$ at each layer ℓ, denoted as $\mathrm{cosine}_{avg}^{(\ell)}(\boldsymbol{x}) = \cos[f^{(\ell)}(\frac{1}{k}\sum_{i=1}^k \boldsymbol{\theta}_i;\boldsymbol{x}), \frac{1}{k}\sum_{i=1}^k f^{(\ell)}(\boldsymbol{\theta}_i;\boldsymbol{x})]$ Additionally, we compare $\mathrm{cosine}_{avg}^{(\ell)}(\boldsymbol{x})$ with the baseline $\mathrm{cosine}_{base}^{(\ell)}(\boldsymbol{x}) = \frac{1}{k}\sum_{i=1}^k \cos[f^{(\ell)}(\frac{1}{k}\sum_{j=1}^k \boldsymbol{\theta}_j;\boldsymbol{x}), f^{(\ell)}(\boldsymbol{\theta}_i;\boldsymbol{x})]$. We compute $\mathrm{coef}_{avg}^{(\ell)}(\boldsymbol{x}) = \frac{\|f^{(\ell)}(\frac{1}{k}\sum_{i=1}^k \boldsymbol{\theta}_i;\boldsymbol{x})\|\mathrm{cosine}_{avg}^{(\ell)}(\boldsymbol{x})}{\|\frac{1}{k}\sum_{i=1}^k f^{(\ell)}(\boldsymbol{\theta}_i;\boldsymbol{x})\|}$ to validate the features have similar length. In Figure 5, the values of $\mathbb{E}_{\mathcal{D}}[1 - \mathrm{cosine}_{avg}^{(\ell)}(\boldsymbol{x})]$ closely approach 0 compared with the baseline $\mathbb{E}_{\mathcal{D}}[1 - \mathrm{cosine}_{base}^{(\ell)}(\boldsymbol{x})]$, and in Figure 6, the values of $\mathrm{coef}_{avg}^{(\ell)}(\boldsymbol{x})$ closely approximate 1. In conclusion, model averaging roughly aggregates the features in each individual finetuned model at each layer.

It is not difficult to see that our discovery could directly imply the observed linear correlation between the model averaging accuracy and the logits ensemble accuracy, particularly when Equation (1) is applied to the output layer. Apparently, our discovery unveils a finer-grained characterization of the linear correlation between model averaging and logits ensemble. Indeed, our discovery can be viewed as a generalization of CTL to the case of multiple models in the pretraining-finetuning paradigm (see Theorem 4.2). Hence, we conclude that CTL establishes a stronger connection between model averaging and logits ensemble in the feature space, thus further explaining the effectiveness of model averaging from a feature-learning perspective.

Theorem 4.2 (CTL Generalizes to Multiple Models (Proof in Appendix B.3)**).** *Given dataset $\mathcal{D}$ and a set of modes $\boldsymbol{\Theta}$ where each pair of modes $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \in \boldsymbol{\Theta}^2$ satisfy CTL on $\mathcal{D}$, assume transitivity of CTL (see Conjecture 4.1),*

[5] $\|f^{(\ell)}(\boldsymbol{\theta}_\alpha;\boldsymbol{x})\|\,\mathrm{cosine}_\alpha^{(\ell)}(\boldsymbol{x})$ denotes the length of the projection of $f^{(\ell)}(\boldsymbol{\theta}_\alpha;\boldsymbol{x})$ onto the vector $\alpha f^{(\ell)}(\boldsymbol{\theta}_i;\boldsymbol{x}) + (1-\alpha)f^{(\ell)}(\boldsymbol{\theta}_j;\boldsymbol{x})$.

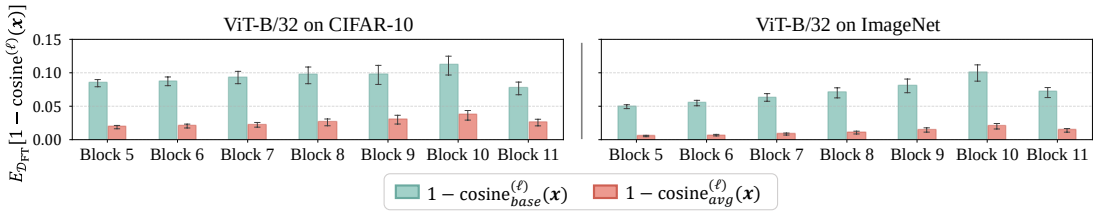


Figure 5. Verification of CTL in model averaging. Compare $\mathbb{E}_{\mathcal{D}}[1-\mathrm{cosine}^{(\ell)}_{avg}(\boldsymbol{x})]$ with $\mathbb{E}_{\mathcal{D}}[1-\mathrm{cosine}^{(\ell)}_{base}(\boldsymbol{x})]$. The bottom and top of the error bar represent the lower and upper quartile of the values across the dataset, respectively. The results are reported for last 7 blocks of ViT-B/32 models, that are finetuned on CIFAR-10 and ImageNet, respectively. More results in Appendix C.4.

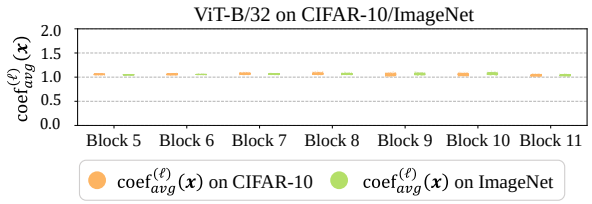


Figure 6. Verification of CTL in model averaging. Distribution of $\mathrm{coef}^{(\ell)}_{avg}(\boldsymbol{x})$ across the dataset. The results are reported for last 7 blocks of ViT-B/32 models finetuned on CIFAR-10 and ImageNet. More results in Appendix C.4.

then for any $\{\boldsymbol{\theta}_i\}_{i=1}^k \in \boldsymbol{\Theta}$ *and* $\{\alpha_i\}_{i=1}^k \in [0,1]$, *subject to the constraint that* $\sum_{i=1}^k \alpha_i = 1$, *we have*

$$f^{(\ell)}\left(\sum_{i=1}^k \alpha_i \boldsymbol{\theta}_i\right) \approx \sum_{i=1}^k \alpha_i f^{(\ell)}(\boldsymbol{\theta}_i), \quad \forall \ell \in [L].$$

4.3. Insights into Task Arithmetic

Ilharco et al. (2023) introduced task arithmetic for editing pretrained models using task vectors, which are obtained by subtracting the pretrained weights from the finetuned weights. Specifically, considering a pretrained model $\boldsymbol{\theta}_{\mathrm{PT}}$ and a set of finetuned models $\{\boldsymbol{\theta}_i\}_{i=1}^k$ with corresponding downstream tasks $\{\mathcal{D}_i\}_{i=1}^k$, the task vectors $\{\tau_i\}_{i=1}^k$ are defined as $\tau_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_{PT}$. Arithmetic operations, including addition and negation, can be applied to the task vectors to obtain a new task vector τ_{new}, and the new task vector is then applied to the pretrained weights with a scaling term λ, i.e., $\boldsymbol{\theta}_{\mathrm{new}} = \boldsymbol{\theta}_{\mathrm{PT}} + \lambda\tau_{\mathrm{new}}$. It allows to control the behavior of the edited model via simple arithmetic operations on task vectors. In this subsection, we aim to explain the effectiveness of task arithmetic from a feature-learning perspective.

CTL Explains Learning via Addition. An intriguing discovery in task arithmetic is that the addition of task vectors builds multi-task models. For instance, with a proper chosen λ, $f\left(\boldsymbol{\theta}_{\mathrm{PT}} + \lambda\left(\tau_i + \tau_j\right)\right)$ demonstrate comparable performance on both tasks $\mathcal{D}_i$ and $\mathcal{D}_j$. Despite this surprising observation, it is not well understood why addition in the parameter space leads to the multi-task abilities.

We aim to interpret the addition operation from a feature-learning perspective. Assuming CTL holds for the edited models, we can easily derive that $\forall \ell \in [L]$,

$$\begin{aligned} & f^{(\ell)}\left(\boldsymbol{\theta}_{\mathrm{PT}} + \lambda(\tau_i + \tau_j)\right) \\ \approx & \frac{1}{2} f^{(\ell)}(\boldsymbol{\theta}_{\mathrm{PT}} + 2\lambda\tau_i) + \frac{1}{2} f^{(\ell)}(\boldsymbol{\theta}_{\mathrm{PT}} + 2\lambda\tau_j). \end{aligned} \tag{2}$$

We conduct experiments to verify Equation (2). Specifically, given a pair of task vectors (τ_i, τ_j), on each datapoint $\boldsymbol{x} \in \mathcal{D}_i \cup \mathcal{D}_j$, we measure the cosine similarity between LHS and RHS of Equation (2), i.e., $\mathrm{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; 2\lambda\tau_i, 2\lambda\tau_j) = \cos[f^{(\ell)}\left(\boldsymbol{\theta}_{\mathrm{PT}} + \lambda(\tau_i + \tau_j)\right), \frac{1}{2} f^{(\ell)}(\boldsymbol{\theta}_{\mathrm{PT}} + 2\lambda\tau_i; \boldsymbol{x}) + \frac{1}{2} f^{(\ell)}(\boldsymbol{\theta}_{\mathrm{PT}} + 2\lambda\tau_j; \boldsymbol{x})]$ at each layer ℓ. Similarly as before, we compare it with the baseline cosine similarity, i.e., $\mathrm{cosine}^{(\ell)}_{base}(\boldsymbol{x}; 2\lambda\tau_i, 2\lambda\tau_j) = \cos[f^{(\ell)}(\boldsymbol{\theta}_{\mathrm{PT}} + 2\lambda\tau_i; \boldsymbol{x}), f^{(\ell)}(\boldsymbol{\theta}_{\mathrm{PT}} + 2\lambda\tau_j; \boldsymbol{x})]$. Additionally, we examine the approximate equality via $\mathrm{ceof}^{\ell}_{arith}(\boldsymbol{x}; 2\lambda\tau_i, 2\lambda\tau_j) = \frac{\|f^{(\ell)}\left(\boldsymbol{\theta}_{\mathrm{PT}} + \lambda(\tau_i + \tau_j)\right)\| \, \mathrm{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; 2\lambda\tau_i, 2\lambda\tau_j)}{\|\frac{1}{2} f^{(\ell)}(\boldsymbol{\theta}_{\mathrm{PT}} + 2\lambda\tau_i; \boldsymbol{x}) + \frac{1}{2} f^{(\ell)}(\boldsymbol{\theta}_{\mathrm{PT}} + 2\lambda\tau_j; \boldsymbol{x})\|}$. In Figure 7, the values of $\mathbb{E}_{\mathcal{D}}[1 - \mathrm{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; 2\lambda\tau_i, 2\lambda\tau_j)]$ are close to 0 compared with $\mathbb{E}_{\mathcal{D}}[1 - \mathrm{cosine}^{(\ell)}_{base}(\boldsymbol{x}; 2\lambda\tau_i, 2\lambda\tau_j)]$, and in Figure 8, the values of $\mathrm{ceof}^{\ell}_{arith}(\boldsymbol{x}; 2\lambda\tau_i, 2\lambda\tau_j)$ are distributed around 1. Hence we conclude that the features in the model applied with the addition of two task vectors can be approximated by the addition of the features in two models, each applied with a single task vector.

Though Equation (2) has transformed the addition from the parameter space to the feature space, the reason why addition in the feature space constructs multi-task models remains unclear. In fact, we discover that if we replace the features in the pretrained model by the average of the features in two finetuned model, i.e., $f^{L \leftarrow (\ell+1)}(\boldsymbol{\theta}_{\mathrm{PT}}; \frac{1}{2} f^{(\ell)}(\boldsymbol{\theta}_i) + \frac{1}{2} f^{(\ell)}(\boldsymbol{\theta}_j))$, the model with replaced features could demonstrate abilities on both $\mathcal{D}_i$ and $\mathcal{D}_j$. Here, $f^{L \leftarrow (\ell+1)}(\boldsymbol{\theta}; \cdot)$ denotes the mapping from the internal features of the network $f(\boldsymbol{\theta})$ at ℓ-th layer to the final output. This feature replacement shares a similar methodology with the model stitching[6], and thus, we term the model with replaced features as the stitched model. In Figure 9, across var-

[6]Model stitching (Lenc & Vedaldi, 2015; Bansal et al., 2021) is a widely used technique for analyzing the internal representations of networks. It stitches the front part of one model with the back

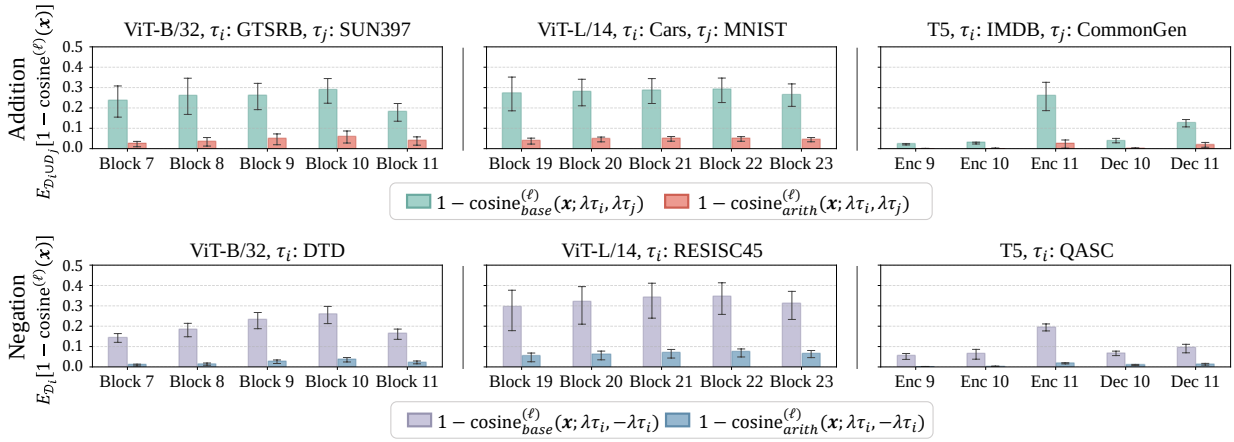


Figure 7. Verification of CTL in task arithmetic. **Addition**: Compare $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)]$ with $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{base}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)]$. **Negation**: Compare $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)]$ with $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{base}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)]$. The bottom and top of the error bar represent the lower and upper quartile of the values across the dataset, respectively. The results are reported for last 5 blocks of finetuned models under different settings, with $\lambda = 0.4$ (Ilharco et al., 2023). More results in Appendices C.5 and C.6.

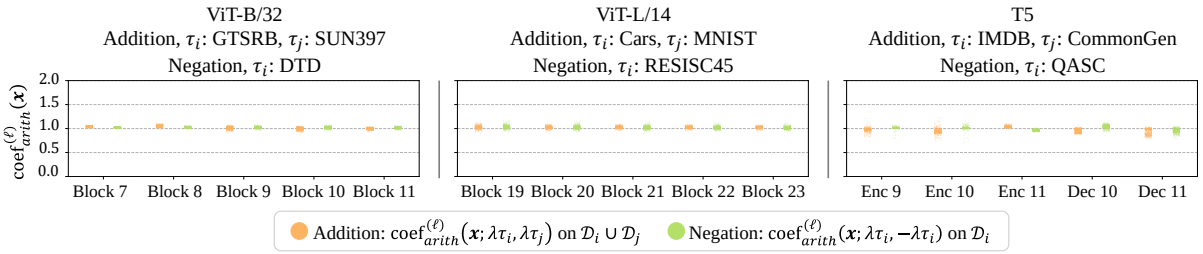


Figure 8. Verification of CTL in task arithmetic. **Addition**: Distribution of $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)$. **Negation**: Distribution of $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)$. The results are reported for last 5 blocks of finetuned models under different settings, with $\lambda = 0.4$ (Ilharco et al., 2023). More results in Appendices C.5 and C.6.

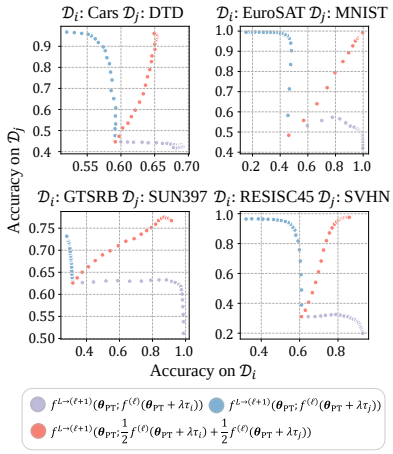


Figure 9. Accuracy of $f^{L\leftarrow(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; \frac{1}{2}f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} + \lambda\tau_i) + \frac{1}{2}f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} + \lambda\tau_j))$, $f^{L\leftarrow(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} + \lambda\tau_i))$ and $f^{L\leftarrow(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} + \lambda\tau_j))$ on both $\mathcal{D}_i$ (x-axis) and $\mathcal{D}_j$ (y-axis). Results are reported for ViT-B/32 with various combinations of task vectors and different values of $\lambda \in [0.05, 1]$. The stitching layers ℓ is chosen to be Block-9.

ious combinations of τ_i and τ_j and different values of λ, the stitched model $f^{L\leftarrow(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; \frac{1}{2}f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} + \lambda\tau_i) + \frac{1}{2}f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} + \lambda\tau_j))$ achieves comparable performance on both $\mathcal{D}_i$ and $\mathcal{D}_j$, while $f^{L\leftarrow(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} + \lambda\tau_i))$ and $f^{L\leftarrow(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} + \lambda\tau_j))$ are only capable of single tasks. Therefore, we conclude that the addition in the feature space actually aggregates the task-specific information from both tasks, thereby bridging the multi-task abilities and CTL.

CTL Explains Forgetting via Negation. Another surprising finding in task arithmetic is that negating a task vector removes the ability of the pretrained model on the corresponding task. Specifically, the edited model $f(\boldsymbol{\theta}_{\text{PT}} - \lambda\tau_i)$ forgets its proficiency on $\mathcal{D}_i$ while maintaining its performance elsewhere. Further exploration of the underlying reasons for this forgetting effect is encouraged.

We still explain the negation operation from a feature-learning perspective. Assume CTL satisfied for the edited

part of another model by a learnable linear layer. If stitched model retains a good performance on target task, we say that the two model share a similar representation at the stitching layer. In our case, no learnable linear layer is employed.

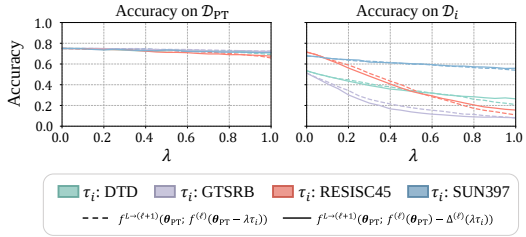


Figure 10. Accuracy of $f^{L\to(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}}) - \Delta^{(\ell)}(\lambda\tau_i))$ and $f^{L\to(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} - \lambda\tau_i))$ on $\mathcal{D}_{\text{PT}}$ and $\mathcal{D}_i$ v.s. λ. Results are reported for ViT-L/14 with different task vectors. The stitching layer ℓ is chosen to be Block-19.

models, we can simply obtain that, $\forall \ell \in [L]$,

$$f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}}) \approx \frac{1}{2} f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} + \lambda\tau_i) + \frac{1}{2} f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} - \lambda\tau_i). \quad (3)$$

To verify Equation (3), given a task vector τ_i, on each datapoint $\boldsymbol{x} \in \mathcal{D}_i$, we measure $\text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)$ and compare it with $\text{cosine}^{(\ell)}_{base}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)$. We also compute $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)$. The results in Figures 7 and 8 validate our hypothesis in Equation (3).

Equation (3) interprets the negation in the parameter space as the negation in the feature space, as can be rewritten as:

$$f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} - \lambda\tau_i) \approx f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}}) - \Delta^{(\ell)}(\lambda\tau_i), \quad (4)$$

where $\Delta^{(\ell)}(\lambda\tau_i) = f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} + \lambda\tau_i) - f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}})$. Intuitively, $\Delta^{(\ell)}(\lambda\tau_i)$ encodes the extra information specific to the task $\mathcal{D}_i$. Therefore, $f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} - \lambda\tau_i)$ loses the task-specific information of $\Delta^{(\ell)}(\lambda\tau_i)$, while retaining most information of $f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}})$.

We now examine the ability of the negation in the feature space through model stitching. Specifically, we measure the accuracy of the stitched model $f^{L\leftarrow(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}}) - \Delta^{(\ell)}(\lambda\tau_i))$ on the downstream task $\mathcal{D}_i$ and the pretraining task $\mathcal{D}_{\text{PT}}$. In Figure 10, with the increase of λ, the accuracy of $f^{L\leftarrow(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}}) - \Delta^{(\ell)}(\lambda\tau_i))$ drops significantly on $\mathcal{D}_i$ while keeping nearly constant on $\mathcal{D}_{\text{PT}}$. We also evaluate $f^{L\leftarrow(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}} - \lambda\tau_i))$, which shows a similar performance to $f^{L\leftarrow(\ell+1)}(\boldsymbol{\theta}_{\text{PT}}; f^{(\ell)}(\boldsymbol{\theta}_{\text{PT}}) - \Delta^{(\ell)}(\lambda\tau_i))$ on both tasks, thus further validating Equation (4). In conclusion, CTL translates the negation in the parameter space as the negation in the feature space, which further induces the aforementioned forgetting effect.

CTL Implies Task Arithmetic. Ortiz-Jimenez et al. (2023) proposed the weight disentanglement as the necessary condition to perform task arithmetic. We show that the weight disentanglement is roughly a consequence of CTL (see Theorem 4.3 and discussion in Appendix A).

Theorem 4.3 (CTL Connects to Weight Disentanglement. (Proof in Appendix B.4)). *Given pretrained model $\boldsymbol{\theta}_{\text{PT}}$ and a set of task vectors $\Upsilon = \{\tau_i\}_{i=1}^k$, suppose each pair of edited models $(\boldsymbol{\theta}_{\text{PT}} + \lambda_i\tau_i, \boldsymbol{\theta}_{\text{PT}} + \lambda_j\tau_j)$ satisfy CTL when $\lambda_i, \lambda_j \in [-\beta, \beta]$, assume the transitivity of CTL (see Conjecture 4.1), then $\forall \{\alpha_i\}_{i=1}^k \in [-\frac{\beta}{k+1}, \frac{\beta}{k+1}]$,*

$$f\left(\boldsymbol{\theta}_{\text{PT}} + \sum_{i=1}^{k} \alpha_i\tau_i; \boldsymbol{x}\right) \approx \sum_{i=1}^{k} g_i(\alpha_i\tau_i; \boldsymbol{x}) + g_0(\boldsymbol{x})$$

where $g_i(\alpha_i\tau_i; \boldsymbol{x}) = \frac{1}{k+1} f(\boldsymbol{\theta}_{\text{PT}} + (k+1)\alpha_i\tau_i; \boldsymbol{x})$ and $g_0(\boldsymbol{x}) = \frac{1}{k+1} f(\boldsymbol{\theta}_{\text{PT}}; \boldsymbol{x})$.

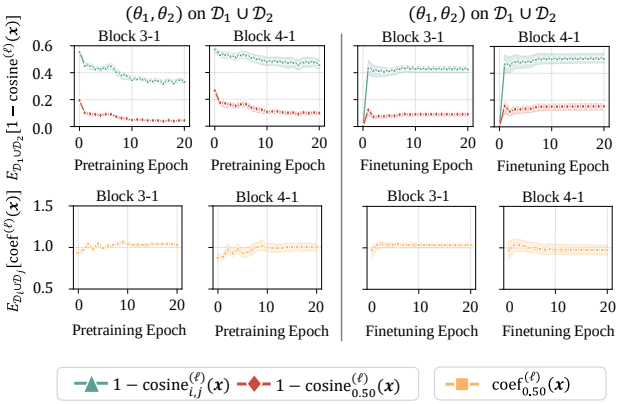


Figure 11. The impact of the number of pretraining/finetuning epochs on the emergence of CTL. We show the change of $1 - \text{cosine}^{(\ell)}_{0.5}(\boldsymbol{x})$ and $\text{coef}^{(\ell)}_{0.5}(\boldsymbol{x})$ w.r.t. the number of pretraining/finetuning epoch. **Left**: We vary the number of pretraining epochs from 0 to 20 and fix the number of finetuning epochs to 10. **Right**: We fix the number of pretraining epochs to 10 and vary the number of finetuning epochs from 0 to 20. Results are reported for ResNet-18s pretrained and finetuned on Split CIFAR-100.

5. Unveiling the Root Cause of CTL

We have seen CTL consistently occurs in the pretraining-finetuning paradigm, roughly characterizing networks as linear maps from the parameter space to the feature space. In this section, we aim to unveil the root cause of CTL. We explore various factors contributing to the emergence of CTL, emphasizing the role of pretraining. We also take a theoretical attempt to prove CTL.

Factors Contributing to CTL. We investigate the impact of two factors, the number of pretraining/finetuning epoch and the task similarity, on the emergence of CTL.

***i)* The number of pretraining/finetuning epochs.** First, we study the impact of the pretraining epochs on CTL. Specifically, we vary the number of pretraining epochs and fix the the number of finetuning epochs. Consistently with Section 4.1, we measure $\text{cosine}^{(\ell)}_{0.5}(\boldsymbol{x})$, $\text{cosine}^{(\ell)}_{i,j}(\boldsymbol{x})$ and $\text{coef}^{(\ell)}_{0.5}(\boldsymbol{x})$ for each pair of finetuned models. In Figure 11 (left), $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{0.5}(\boldsymbol{x})]$ decreases as the number of pretraining epochs increases and the values of $\text{coef}^{(\ell)}_{0.5}(\boldsymbol{x})$ grad-

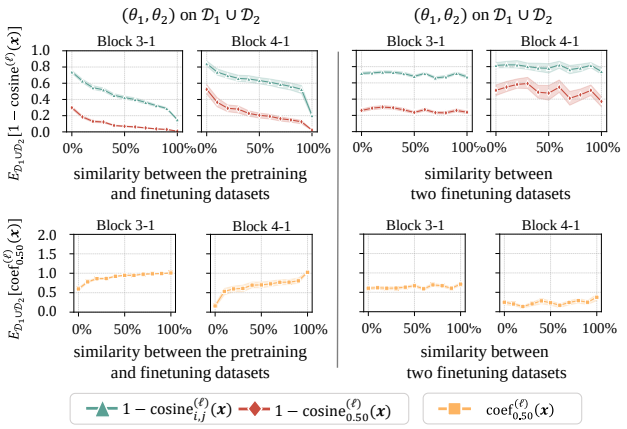


Figure 12. The impact of task similarity on the emergence of CTL. We show the change of $1 - \text{cosine}_{0.5}^{(\ell)}(\boldsymbol{x})$ and $\text{coef}_{0.5}^{(\ell)}(\boldsymbol{x})$ w.r.t. the ratio of replaced samples (task similarity). **Left**: We replace the samples from the two finetuning dataset with the samples from the pretraining dataset. **Right**: We replace the samples from one finetuning dataset with the samples from the other finetuning dataset. Results are reported for ResNet-18s pretrained and finetuned on Split ImageNet.

ually concentrate around 1. Results indicate that increasing pretraining epochs promotes the emergence of CTL.

Second, we study the impact of the finetuning epochs on CTL. Similarly, we fix the number of pretraining epochs and vary the number of pretraining epochs. In Figure 11 (right), the values of $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}_{0.5}^{(\ell)}(\boldsymbol{x})]$ deviate slightly from 0 once finetuning for at least one epoch, while there is no clear trend for $\text{coef}_{0.5}^{(\ell)}(\boldsymbol{x})$. Results imply that the number of finetuning epochs has lesser impact on CTL.

***ii)* The task similarity.** First, we study the impact of the similarity between the pretraining and finetuning tasks on CTL. However, directly quantifying task similarity is challenging. Therefore, we adopt an alternative method: replacing the samples from the finetuning tasks $\mathcal{D}_i$ and $\mathcal{D}_j$ with the samples from the pretraining task $\mathcal{D}_{\text{PT}}$ and finetuning the models on $\mathcal{D}_i$ and $\mathcal{D}_j$ with these replaced samples. The ratio of replaced samples, i.e., $\frac{\text{\# of replaced samples}}{\text{\# of total samples in } \mathcal{D}_i \setminus \mathcal{D}_j}$, serves as a measure of task similarity[7]. We then calculate $\text{cosine}_{0.5}^{(\ell)}(\boldsymbol{x})$ and $\text{coef}_{0.5}^{(\ell)}(\boldsymbol{x})$ for each pair of finetuned models at different ratio of replaced samples. In Figure 12 (left), as the ratio of replaced samples increases, $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}_{0.5}^{(\ell)}(\boldsymbol{x})]$ decreases, and the values of $\text{coef}_{0.5}^{(\ell)}(\boldsymbol{x})$ gradually approaches 1. Results indicate that the similarity between pretraining task and finetuning task promotes the emergence of CTL.

Second, we study the impact of the similarity between the two finetuning tasks on CTL. Similarly, we replace the samples from one finetuning task with the samples from the other finetuning task and use the ratio of replaced samples as a measure of task similarity[8]. In Figure 12 (right), the values of $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}_{0.5}^{(\ell)}(\boldsymbol{x})]$ remain nearly constant regardless of how similar the two finetuning tasks are, and the distributions of $\text{coef}_{0.5}^{(\ell)}(\boldsymbol{x})$ change little as well. Results demonstrate that the similarity between finetuning tasks has lesser effects on CTL.

In summary, our investigation indicate that the number of pretraining epochs and the similarity between pretraining and finetuning tasks promote the emergence of CTL, emphasizing the role of pretraining.

Theoretical Attempt to Prove CTL. In addition, we take a first step to prove CTL. Specifically, we prove that the emergence of CTL is related to the flatness of the landscape of $f(\cdot)$ and the distance between two finetuned models (see Theorem 5.1). We affirm our theorem by demonstrating a strong correlation between $\delta_{i,j}$ and $\frac{\alpha(1-\alpha)\lambda_{\max}}{2} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2$ (Results in Appendix C.2).

Theorem 5.1 (**The Emergence of CTL** (Proof in Appendix B.2))**.** *Suppose $f(\boldsymbol{\theta}) : \mathbb{R}^p \mapsto \mathbb{R}$ is third differentiable in an open convex set $\boldsymbol{\Theta}$ and the its Hessian norm at $\boldsymbol{\theta}_0$ is bounded by $\lambda_{min} \leq \left\|\nabla^2 f(\boldsymbol{\theta}_0)\right\| \leq \lambda_{max}$, then*

$$\begin{aligned}\delta_{i,j} &= |f(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) - \alpha f(\boldsymbol{\theta}_i) - (1-\alpha) f(\boldsymbol{\theta}_j)| \\ &\leq \frac{\alpha(1-\alpha)\lambda_{max}}{2} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2 + \mathcal{E},\end{aligned}$$

where $\mathcal{E} = O\left(\max\left(\|\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0\|^3, \alpha\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_0\|^3, (1-\alpha)\|\boldsymbol{\theta}_j - \boldsymbol{\theta}_0\|^3\right)\right)$ is the high order term.

Remark 5.2. Previous studies found linearizing models is insufficient to explain LMC and task arithmetic (Fort et al., 2020; Ortiz-Jimenez et al., 2023). In Theorem 5.1, instead of linearizing models, we provide a more realistic setting.

6. Conclusion and Limitations

In this work, we identified Cross-Task Linearity (CTL) as a prevalent phenomenon that consistently occurs for finetuned models, approximately characterizing neural networks as linear maps in the pretraining-finetuning paradigm. Based on the observed CTL, we obtained novel insights into two widely-used model merging/editing techniques: model averaging and task arithmetic. Furthermore, we studied the root cause of CTL, highlighting the role of pretraining.

Our current work primarily focuses on empirical findings, despite a theoretical attempt to prove CTL in Section 5. We defer a thorough theoretical analysis to future work. Additionally, on the practical side, we leave comprehensive exploration of CTL on Large Language Models to future.

[7]Notably, when the ratio is 1, $\mathcal{D}_i$, $\mathcal{D}_j$ and $\mathcal{D}_{\text{PT}}$ are identical.

[8]When the ratio is 1, $\mathcal{D}_i$ and $\mathcal{D}_j$ are identical yet different from $\mathcal{D}_{\text{PT}}$

Acknowledgments

This work was in part supported by the National Natural Science Foundation of China (62222607), the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), the National Key R&D Program of China (2022ZD0160104) and Shanghai Rising Star Program (23QD1401000).

Impact Statement

This paper refers to the understanding of deep neural networks especially for pretrained models which themselves have significant impact to the society. In particular, we hope our technique can help better removing the unwanted behavior of trained neural networks.

References

Adilova, L., Fischer, A., and Jaggi, M. Layerwise linear mode connectivity. *arXiv preprint arXiv:2307.06966*, 2023.

Ainsworth, S., Hayase, J., and Srinivasa, S. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CQsmMYmlP5T.

Bansal, Y., Nakkiran, P., and Barak, B. Revisiting model stitching to compare neural representations. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=ak06J5jNR4.

Chen, Y., Zhou, Z., and Yan, J. Going beyond neural network feature similarity: The network feature complexity and its interpretation using category theory. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4bSQ3lsfEV.

Cheng, G., Han, J., and Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

contributors, T. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Entezari, R., Sedghi, H., Saukh, O., and Neyshabur, B. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=dNigytemkL.

Fabbri, A. R., Li, I., She, T., Li, S., and Radev, D. R. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*, 2019.

Ferbach, D., Goujaud, B., Gidel, G., and Dieuleveut, A. Proving linear mode connectivity of neural networks via optimal transport. *arXiv preprint arXiv:2310.19103*, 2023.

Fort, S., Dziugaite, G. K., Paul, M., Kharaghani, S., Roy, D. M., and Ganguli, S. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33: 5850–5861, 2020.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Freeman, C. D. and Bruna, J. Topology and geometry of half-rectified network optimization. In *International Conference on Learning Representations*,

2017. URL https://openreview.net/forum?id=Bk0FWVcgx.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Helber, P., Bischke, B., Dengel, A., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Ilharco, G., Wortsman, M., Gadre, S. Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., and Schmidt, L. Patching open-vocabulary models by interpolating weights. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=CZZFRxbOLC.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. P., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. In Globerson, A. and Silva, R. (eds.), *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pp. 876–885. AUAI Press, 2018. URL http://auai.org/uai2018/proceedings/papers/313.pdf.

Jin, X., Ren, X., Preotiuc-Pietro, D., and Cheng, P. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Juneja, J., Bansal, R., Cho, K., Sedoc, J., and Saphra, N. Linear connectivity reveals generalization strategies. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hY6M0JHl3uL.

Khot, T., Clark, P., Guerquin, M., Jansen, P., and Sabharwal, A. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8082–8090, 2020.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013. URL https://api.semanticscholar.org/CorpusID:14342571.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kuditipudi, R., Wang, X., Lee, H., Zhang, Y., Li, Z., Hu, W., Ge, R., and Arora, S. Explaining landscape connectivity of low-cost solutions for multilayer nets. *Advances in neural information processing systems*, 32, 2019.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

LeCun, Y. and Cortes, C. The mnist database of handwritten digits. 2005. URL https://api.semanticscholar.org/CorpusID:60282629.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lenc, K. and Vedaldi, A. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 991–999, 2015.

Li, M., Gururangan, S., Dettmers, T., Lewis, M., Althoff, T., Smith, N. A., and Zettlemoyer, L. Branch-train-merge: Embarrassingly parallel training of expert language models. In *First Workshop on Interpolation Regularizers and Beyond at NeurIPS 2022*, 2022. URL https://openreview.net/forum?id=SQgVgE2Sq4.

Liang, S., Sun, R., Li, Y., and Srikant, R. Understanding the loss surface of neural networks for binary classification. In *International Conference on Machine Learning*, pp. 2835–2843. PMLR, 2018.

Lin, B. Y., Zhou, W., Shen, M., Zhou, P., Bhagavatula, C., Choi, Y., and Ren, X. Commongen: A constrained text generation challenge for generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*, 2019.

Liu, C., Lou, C., Wang, R., Xi, A. Y., Shen, L., and Yan, J. Deep neural network fusion via graph matching with applications to model ensemble and federated learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 13857–13869. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/liu22k.html.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Matena, M. S. and Raffel, C. Merging models with fisher-weighted averaging. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=LSKlp_aceOC.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Mirzadeh, S. I., Farajtabar, M., Gorur, D., Pascanu, R., and Ghasemzadeh, H. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Fmg_fQYUejf.

Nagarajan, V. and Kolter, J. Z. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Nguyen, Q. On connected sublevel sets in deep learning. In *International conference on machine learning*, pp. 4790–4799. PMLR, 2019.

Nguyen, Q., Mukkamala, M. C., and Hein, M. On the loss landscape of a class of deep neural networks with no bad local valleys. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJgXsjA5tQ.

Ni, J., Ábrego, G. H., Constant, N., Ma, J., Hall, K. B., Cer, D., and Yang, Y. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*, 2021.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Ortiz-Jimenez, G., Favero, A., and Frossard, P. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=0A9f2jZDGW.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Gray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Qin, Y., Qian, C., Yi, J., Chen, W., Lin, Y., Han, X., Liu, Z., Sun, M., and Zhou, J. Exploring mode connectivity for pre-trained language models. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6726–6746, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.451. URL https://aclanthology.org/2022.emnlp-main.451.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Rame, A., Kirchmeyer, M., Rahier, T., Rakotomamonjy, A., patrick gallinari, and Cord, M. Diverse weight averaging for out-of-distribution generalization. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=tq_J_MqB3UB.

Rame, A., Ahuja, K., Zhang, J., Cord, M., Bottou, L., and Lopez-Paz, D. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28656–28679. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/rame23a.html.

Singh, S. P. and Jaggi, M. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pp. 1453–1460. IEEE, 2011.

Stoica, G., Bolya, D., Bjorner, J., Hearn, T., and Hoffman, J. Zipit! merging models from different tasks without training, 2023.

Venturi, L., Bandeira, A. S., and Bruna, J. Spurious valleys in one-hidden-layer neural network optimization landscapes. *J. Mach. Learn. Res.*, 20:133:1–133:34, 2019. URL http://jmlr.org/papers/v20/18-674.html.

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., Kenton, Z., Brown, S., Hawkins, W., Stepleton, T., Biles, C., Birhane, A., Haas, J., Rimell, L., Hendricks, L. A., Isaac, W., Legassick, S., Irving, G., and Gabriel, I. Ethical and social risks of harm from language models, 2021.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022a.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., and Schmidt, L. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7959–7971, June 2022b.

Xiao, J., Ehinger, K. A., Hays, J., Torralba, A., and Oliva, A. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119:3–22, 2016.

Yadav, P., Tam, D., Choshen, L., Raffel, C., and Bansal, M. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

Yu, L., Yu, B., Yu, H., Huang, F., and Li, Y. Language models are super mario: Absorbing abilities from homologous models as a free lunch, 2023.

Zhao, B., Dehmamy, N., Walters, R., and Yu, R. Understanding mode connectivity via parameter space symmetry. In *UniReps: the First Workshop on Unifying Representations in Neural Models*, 2023. URL https://openreview.net/forum?id=aP2a5i1iUf.

Zhou, Z., Yang, Y., Yang, X., Yan, J., and Hu, W. Going beyond linear mode connectivity: The layerwise linear feature connectivity. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=vORUHrVEnH.

A. Connection with Prior Works

A recent work (Ortiz-Jimenez et al., 2023) argue against the linearization hypothesis, which suggesting that finetuning occurs in the linear regime, by demonstrating the significant impact of non-linear terms on model behavior during training and merging. There is seemingly a contradiction between our main discovery, CTL, and the findings of Ortiz-Jimenez et al. (2023), as CTL roughly characterizing the neural network as linear maps in the pretraining-finetuning paradigm. In this section, we clarify that our main discovery, CTL, indeed does not contradict with the findings of Ortiz-Jimenez et al. (2023).

Let us first revisit the linearization hypothesis. As mentioned in Ortiz-Jimenez et al. (2023), the linearization hypothesis was originally proposed to explain the phenomenon that the outputs of weight-averaged model linearly correlate with the averaging of the outputs of each individual model (Ilharco et al., 2022; Wortsman et al., 2022a;b). In fact, this linear correlation phenomenon can be viewed as a special case of our main discovery, Cross-Task Linearity (CTL), when applied to the output layer. Our work indeed generalizes this phenomenon to the intermediate states at each layer and a broader setting of pretraining-finetuning paradigm (see full discussion in Section 4.2). Therefore, our work is closely aligned with Ilharco et al. (2022); Wortsman et al. (2022a;b).

Now let us turn to Ortiz-Jimenez et al. (2023). They demonstrated that the finetuned models cannot be accurately approximated by their post-hoc linearization and thus rejected the linearization hypothesis. The post-hoc linearization refers to the first-order Taylor expansion of the finetuned models at the pretrained checkpoint and the linearization hypothesis implies the finetuned model can be perfectly approximated by this linearization. However, they found that the performance of these post-hoc linearized models failed to match that of the original finetuned models, neither in single tasks nor in task arithmetic. The post-hoc linearization ensures the strict linearity of the finetuned models, which could directly explain both the linear correlation phenomenon and our main discovery, CTL. However, such linearization exhibits a significant discrepancy with the original finetuned models in terms of performance and thereby cannot fully explain the effectiveness of task arithmetic/model averaging.

In this work, we depart from the linearization hypothesis and instead focus on an approximate version of linearity. i.e, CTL. For the experiment side, we observe the approximate linearity, CTL, rather than strict linearity on the original finetuned model under standard experimental settings. For the theory side, we do not assume any linearization of finetuned models. Though such approximate linearity as well as the previous linear correlation phenomenon are surprising for neural networks, which are often viewed as highly non-linear functions, we take a preliminary attempt to prove that the approximate linearity can emerge when the product of the sharpness of the loss landscape and the squared Euclidean distance between two finetuned models' weights is sufficiently small (see Theorem 5.1). Therefore, our discovery not only establishes a strong connection with previous findings on the linear correlation phenomenon but also departs from the linearization hypothesis, turning to the approximate linearity.

Moreover, our discovery, CTL, also connects to the Weight Disentanglement, proposed in section 4 of Ortiz-Jimenez et al. (2023), regarded as a pre-condition for finetuned models to perform task arithmetic. In Theorem 4.3, we demonstrate that assuming the transitivity of CTL (see Conjecture 4.1), CTL can disentangle the edited models applied with multiple task vectors into a sum of localized components, each controlled by a single task vector. Therefore, CTL closely relates to the Weight Disentanglement. In fact, the condition of the Weight Disentanglement is too idealistic to be satisfied in practice. However, CTL serves as a relaxed version of the Weight Disentanglement, which still effectively explaining the task arithmetic.

In conclusion, our work indeed closely aligns with prior studies, not only Ilharco et al. (2022); Wortsman et al. (2022a;b) but also Ortiz-Jimenez et al. (2023). We depart from the linearization hypothesis and identify an approximate version of linearity, CTL, which generalizes the previous linear correlation phenomenon and still efficiently explains the effectiveness of the model merging/task arithmetic techniques.

B. Missing Proofs

B.1. Preliminary Lemmas

Definition B.1 (**Transitivity of CTL.**). Given models $\boldsymbol{\theta}_i, \boldsymbol{\theta}_j, \boldsymbol{\theta}_k$. We have $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_k)$ satisfy CTL if $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ and $(\boldsymbol{\theta}_j, \boldsymbol{\theta}_k)$ satisfy CTL.

Lemma B.2 (**CTL holds for two-model weight interpolations.**). *Given two models $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ satisfy CTL, then $\forall \gamma \in [0, 1]$, we have $\boldsymbol{\theta}'_i = \gamma \boldsymbol{\theta}_i + (1-\gamma)\boldsymbol{\theta}_j$ and $\boldsymbol{\theta}_j$ satisfy CTL.*

Proof. $\forall \alpha, \gamma \in [0,1], \forall \ell \in [L]$,

$$\begin{aligned} f^{(\ell)}(\alpha \boldsymbol{\theta}_i' + (1-\alpha)\boldsymbol{\theta}_j) &= f^{(\ell)}(\alpha(\gamma\boldsymbol{\theta}_i + (1-\gamma)\boldsymbol{\theta}_j) + (1-\alpha)\boldsymbol{\theta}_j) \\ &= f^{(\ell)}(\alpha\gamma\boldsymbol{\theta}_i + (1-\alpha\gamma)\boldsymbol{\theta}_j) \\ &\approx \alpha\gamma f^{(\ell)}(\boldsymbol{\theta}_i) + (1-\alpha\gamma) f^{(\ell)}(\boldsymbol{\theta}_j) \\ &\approx \alpha(\gamma f^{(\ell)}(\boldsymbol{\theta}_i) + (1-\gamma) f^{(\ell)}(\boldsymbol{\theta}_j)) + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_j) \\ &\approx \alpha f^{(\ell)}(\boldsymbol{\theta}_i') + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_j) \end{aligned}$$

Therefore, $\boldsymbol{\theta}_i'$ and $\boldsymbol{\theta}_j$ satisfy CTL, and this finishes our proof. □

Lemma B.3 (CTL holds for multi-model weight interpolations.). *Given a set of models $\Theta = \{\boldsymbol{\theta}_i\}_{i=1}^k$, suppose each pair of models $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \in \Theta^2$ satisfy CTL, assume transitivity of CTL (see Definition B.1), then for any $\{\alpha_i\}_{i=1}^k \in [0,1]$ subject to $\sum_{i=1}^k \alpha_i = 1$, we have the weight-interpolated model $\boldsymbol{\theta}' = \sum_{i=1}^k \alpha_i \boldsymbol{\theta}_i$ satisfy CTL with each model from Θ.*

Proof. We proceed by induction on k. When $k = 2$, Lemma B.2 directly implies Lemma B.3.

Assume Lemma B.3 holds for some $k' \geq 2$. Assume that for any $\{\alpha_i\}_{i=1}^{k'} \in [0,1]$ subject to $\sum_{i=1}^{k'} \alpha_i = 1$ and any $\boldsymbol{\theta}_j \in \Theta$, we have $\boldsymbol{\theta}' = \sum_{i=1}^{k'} \alpha_i \boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ satisfy CTL.

Now we need to show Lemma B.3 holds when $k = k' + 1$. That is, we need to show for any $\{\alpha_i\}_{i=1}^{k'+1} \in [0,1]$ subject to $\sum_{i=1}^{k'+1} \alpha_i = 1$ and any $\boldsymbol{\theta}_j \in \Theta$, we have $\boldsymbol{\theta}' = \sum_{i=1}^{k'+1} \alpha_i \boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ satisfy CTL. As Lemma B.3 holds when $k = k'$, we have the weight-interpolated model $\boldsymbol{\theta}'' = \sum_{i \neq j} \frac{\alpha_i}{1-\alpha_j} \boldsymbol{\theta}_i$ satisfy CTL with each model $\boldsymbol{\theta}_p \in \Theta \setminus \{\boldsymbol{\theta}_j\}$. As both $(\boldsymbol{\theta}'', \boldsymbol{\theta}_p)$ and $(\boldsymbol{\theta}_p, \boldsymbol{\theta}_j)$ satisfy CTL, we have $\boldsymbol{\theta}''$ and $\boldsymbol{\theta}_j$ satisfy CTL as well. According to Lemma B.2, the weight-interpolated model $\alpha_i \boldsymbol{\theta}_j + (1-\alpha_j)\boldsymbol{\theta}'' = \sum_{i=1}^{k'+1} \alpha_i \boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ satisfy CTL, and this finishes our proof. □

B.2. Proof of Theorem 3.3

Theorem B.4 (LLFC Induces LMC). *Given dataset $\mathcal{D}$, convex loss function L, and two modes $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ with equal loss on $\mathcal{D}$, i.e., $\mathcal{L}(\boldsymbol{\theta}_i) = \mathcal{L}(\boldsymbol{\theta}_j)$, suppose the two modes $\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_j$ satisfy LLFC on $\mathcal{D}$ with exact equality, then for all $\alpha \in [0,1]$,*

$$\mathcal{L}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) \leq \mathcal{L}(\boldsymbol{\theta}_i) = \mathcal{L}(\boldsymbol{\theta}_j).$$

Proof. Since $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ satisfy LLFC on $\mathcal{D}$ with exact equality, we have

$$f\left(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j; \boldsymbol{x}\right) = \alpha f\left(\boldsymbol{\theta}_i; \boldsymbol{x}\right) + (1-\alpha) f\left(\boldsymbol{\theta}_j; \boldsymbol{x}\right), \quad \forall \boldsymbol{x} \in \mathcal{D}.$$

Since the loss function L is convex to model outputs, we have

$$\begin{aligned} L\left(f\left(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j; \boldsymbol{x}\right), y\right) &= L\left(\alpha f\left(\boldsymbol{\theta}_i; \boldsymbol{x}\right) + (1-\alpha) f\left(\boldsymbol{\theta}_j; \boldsymbol{x}\right), y\right) \\ &\leq \alpha L\left(f\left(\boldsymbol{\theta}_i; \boldsymbol{x}\right), y\right) + (1-\alpha) L\left(f\left(\boldsymbol{\theta}_j; \boldsymbol{x}\right), y\right), \quad \forall (\boldsymbol{x}, y) \in \mathcal{D}. \end{aligned}$$

Therefore,

$$\begin{aligned} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}}\left[L\left(f\left(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j; \boldsymbol{x}\right), y\right)\right] &\leq \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}}\left[\alpha L\left(f\left(\boldsymbol{\theta}_i; \boldsymbol{x}\right), y\right)\right] + \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}}\left[(1-\alpha) L\left(f\left(\boldsymbol{\theta}_j; \boldsymbol{x}\right), y\right)\right] \\ \mathcal{L}\left(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j\right) &\leq \alpha\mathcal{L}(\boldsymbol{\theta}_i) + (1-\alpha)\mathcal{L}(\boldsymbol{\theta}_j) \end{aligned}$$

According to the condition that $\mathcal{L}(\boldsymbol{\theta}_i) = \mathcal{L}(\boldsymbol{\theta}_j)$, we finally obtain that

$$\mathcal{L}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) \leq \mathcal{L}(\boldsymbol{\theta}_i) = \mathcal{L}(\boldsymbol{\theta}_j).$$

□

B.3. Proof of Theorem 4.2

Theorem B.5 (CTL Generalizes to Multiple Models). *Given dataset $\mathcal{D}$ and a set of modes $\boldsymbol{\Theta}$ where each pair of modes $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \in \boldsymbol{\Theta}^2$ satisfy CTL on $\mathcal{D}$, assume transitivity of CTL (see Definition B.1), then for any $\{\boldsymbol{\theta}_i\}_{i=1}^k \in \boldsymbol{\Theta}$ and $\{\alpha_i\}_{i=1}^k \in [0,1]$, subject to the constraint that $\sum_{i=1}^k \alpha_i = 1$, we have*

$$f^{(\ell)}\left(\sum_{i=1}^k \alpha_i \boldsymbol{\theta}_i\right) \approx \sum_{i=1}^k \alpha_i f^{(\ell)}(\boldsymbol{\theta}_i), \quad \forall \ell \in [L].$$

Proof. We proceed by induction on k. When $k = 2$, Theorem B.5 clearly holds. For any $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \in \boldsymbol{\Theta}^2$, CTL holds on $\mathcal{D}$. Then, $\forall \alpha_1 \in [0,1]$ and $\alpha_2 = 1 - \alpha_1 \in [0,1]$, we have

$$\begin{aligned} f^{(\ell)}(\alpha_1 \boldsymbol{\theta}_1 + \alpha_2 \boldsymbol{\theta}_2) &= f^{(\ell)}\left(\sum_{i=1}^2 \alpha_i \boldsymbol{\theta}_i\right) \\ &\approx \alpha_1 f^{(\ell)}(\boldsymbol{\theta}_1) + \alpha_2 f^{(\ell)}(\boldsymbol{\theta}_2) \\ &\approx \sum_{i=1}^2 \alpha_i f^{(\ell)}(\boldsymbol{\theta}_i), \quad \forall \ell \in [L]. \end{aligned}$$

Assume Theorem B.5 holds for some $k' \geq 2$. That is, assume that for any $\{\boldsymbol{\theta}_i\}_{i=1}^{k'} \in \boldsymbol{\Theta}$ and $\{\alpha_i\}_{i=1}^{k'} \in [0,1]$, subject to the constraint that $\sum_{i=1}^{k'} \alpha_i = 1$, we have

$$f^{(\ell)}\left(\sum_{i=1}^{k'} \alpha_i \boldsymbol{\theta}_i\right) \approx \sum_{i=1}^{k'} \alpha_i f^{(\ell)}(\boldsymbol{\theta}_i), \quad \forall \ell \in [L].$$

Now we need to show Theorem B.5 holds when $k = k' + 1$. For any set of modes $\{\boldsymbol{\theta}_i\}_{i=1}^{k'+1} \in \boldsymbol{\Theta}$ and any set of coefficients $\{\alpha_i\}_{i=1}^{k'+1} \in [0,1]$, we define $\boldsymbol{\theta}_{avg,k'} = \sum_{i=1}^{k'} \frac{\alpha_i}{1-\alpha_{k'+1}} \boldsymbol{\theta}_i$. According to Lemma B.3, CTL holds for $\boldsymbol{\theta}_{avg,k'}$ and $\boldsymbol{\theta}_{k'+1}$, then we have

$$f^{(\ell)}(\alpha \boldsymbol{\theta}_{k'+1} + (1-\alpha)\boldsymbol{\theta}_{avg,k'}) \approx \alpha f^{(\ell)}(\boldsymbol{\theta}_{k'+1}) + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_{avg,k'}), \quad \forall \alpha \in [0,1], \forall \ell \in [L].$$

Substituting α with $\alpha_{k'+1}$, we can obtain

$$\begin{aligned} f^{(\ell)}(\alpha_{k'+1} \boldsymbol{\theta}_{k'+1} + (1-\alpha_{k'+1})\boldsymbol{\theta}_{avg,k'}) &= f^{(\ell)}\left(\sum_{i=1}^{k'+1} \alpha_i \boldsymbol{\theta}_i\right) \\ &\approx \alpha_{k'+1} f^{(\ell)}(\boldsymbol{\theta}_{k'+1}) + (1-\alpha_{k'+1}) f^{(\ell)}(\boldsymbol{\theta}_{avg,k'}), \quad \forall \alpha \in [0,1], \forall \ell \in [L]. \end{aligned}$$

Knowing that Theorem B.5 holds true when $k = k'$, then we have

$$\begin{aligned} f^{(\ell)}\left(\sum_{i=1}^{k'+1} \alpha_i \boldsymbol{\theta}_i\right) &\approx \alpha_{k'+1} f^{(\ell)}(\boldsymbol{\theta}_{k'+1}) + (1-\alpha_{k'+1}) f^{(\ell)}\left(\sum_{i=1}^{k'} \frac{\alpha_i}{1-\alpha_{k'+1}} \boldsymbol{\theta}_i\right) \\ &\approx \alpha_{k'+1} f^{(\ell)}(\boldsymbol{\theta}_{k'+1}) + (1-\alpha_{k'+1}) \sum_{i=1}^{k'} \frac{\alpha_i}{1-\alpha_{k'+1}} f^{(\ell)}(\boldsymbol{\theta}_i) \\ &\approx \sum_{i=1}^{k'+1} \alpha_i f^{(\ell)}(\boldsymbol{\theta}_i), \quad \forall \ell \in [L]. \end{aligned}$$

Therefore, Theorem B.5 holds true when $k = k' + 1$, and this finishes our proof. □

B.4. Proof of Theorem 4.3

Theorem B.6 (**CTL Connects to Weight Disentanglement.**). *Given pretrained model $\boldsymbol{\theta}_{\mathrm{PT}}$ and a set of task vectors $\Upsilon = \{\tau_i\}_{i=1}^k$, suppose each pair of edited models $(\boldsymbol{\theta}_{\mathrm{PT}} + \lambda_i\tau_i, \boldsymbol{\theta}_{\mathrm{PT}} + \lambda_j\tau_j)$ satisfy CTL when $\lambda_i, \lambda_j \in [-\beta, \beta]$, assume the transitivity of CTL (see Definition B.1), then $\forall\{\alpha_i\}_{i=1}^k \in [-\frac{\beta}{k+1}, \frac{\beta}{k+1}]$,*

$$f\left(\boldsymbol{\theta}_{\mathrm{PT}} + \sum_{i=1}^{k}\alpha_i\tau_i; \boldsymbol{x}\right) \approx \sum_{i=1}^{k} g_i(\alpha_i\tau_i; \boldsymbol{x}) + g_0(\boldsymbol{x})$$

where $g_i(\alpha_i\tau_i; \boldsymbol{x}) = \frac{1}{k+1}f(\boldsymbol{\theta}_{\mathrm{PT}} + (k+1)\alpha_i\tau_i; \boldsymbol{x})$ and $g_0(\boldsymbol{x}) = \frac{1}{k+1}f(\boldsymbol{\theta}_{\mathrm{PT}}; \boldsymbol{x})$.

Proof. We proceed by induction on k. When $k = 1$, above Theorem B.6 clearly holds. For any $\tau_1 \in \Upsilon$ and $\lambda_1 \in [-\beta, \beta]$, CTL holds for $(\boldsymbol{\theta}_{\mathrm{PT}}, \boldsymbol{\theta}_{\mathrm{PT}} + \lambda_1\tau_1)$. Then, $\forall\alpha_1 \in [-\frac{\beta}{2}, \frac{\beta}{2}]$, we have

$$f(\boldsymbol{\theta}_{\mathrm{PT}} + \alpha_1\tau_1) \approx \frac{1}{2}f(\boldsymbol{\theta}_{\mathrm{PT}} + 2\alpha_1\tau_1; \boldsymbol{x}) + \frac{1}{2}f(\boldsymbol{\theta}_{\mathrm{PT}}; \boldsymbol{x})$$

Assume Theorem B.6 holds for some $k' \geq 1$. That is, assume that for any $\{\tau_i\}_{i=1}^{k'} \in \Upsilon$ and $\{\alpha_i\}_{i=1}^{k'} \in [-\frac{\beta}{k'+1}, \frac{\beta}{k'+1}]$, we have

$$f\left(\boldsymbol{\theta}_{\mathrm{PT}} + \sum_{i=1}^{k'}\alpha_i\tau_i; \boldsymbol{x}\right) \approx \sum_{i=1}^{k'}\frac{1}{k'+1}f(\boldsymbol{\theta}_{\mathrm{PT}} + (k'+1)\alpha_i\tau_i; \boldsymbol{x}) + \frac{1}{k'+1}f(\boldsymbol{\theta}_{\mathrm{PT}}; \boldsymbol{x})$$

Now we need to show Theorem B.6 holds when $k = k' + 1$. According to Lemma B.3, it is clear to see that the edited model $\boldsymbol{\theta}_{\mathrm{PT}} + (k'+2)\alpha_{k'+1}\tau_{k'+1}$ and the weight-averaged model $\frac{1}{k'+1}\boldsymbol{\theta}_{\mathrm{PT}} + \frac{1}{k'+1}\sum_{i=1}^{k'}\boldsymbol{\theta}_{\mathrm{PT}} + (k'+2)\alpha_i\tau_i$ satisfy CTL. Then, we have

$$\begin{aligned}
&f\left(\boldsymbol{\theta}_{\mathrm{PT}} + \sum_{i=1}^{k'+1}\alpha_i\tau_i; \boldsymbol{x}\right)\\
=&f\left(\frac{1}{k'+2}\left(\boldsymbol{\theta}_{\mathrm{PT}} + (k'+2)\alpha_{k'+1}\tau_{k'+1}\right) + \frac{k'+1}{k'+2}\left(\frac{1}{k'+1}\boldsymbol{\theta}_{\mathrm{PT}} + \frac{1}{k'+1}\sum_{i=1}^{k'}\boldsymbol{\theta}_{\mathrm{PT}} + (k'+2)\alpha_i\tau_i\right); \boldsymbol{x}\right)\\
\approx&\frac{1}{k'+2}f(\boldsymbol{\theta}_{\mathrm{PT}} + (k'+2)\alpha_{k'+1}\tau_{k'+1}; \boldsymbol{x}) + \frac{k'+1}{k'+2}f\left(\boldsymbol{\theta}_{\mathrm{PT}} + \sum_{i=1}^{k'}\frac{k'+2}{k'+1}\alpha_i\tau_i; \boldsymbol{x}\right)\\
\approx&\frac{1}{k'+2}f(\boldsymbol{\theta}_{\mathrm{PT}} + (k'+2)\alpha_{k'+1}\tau_{k'+1}; \boldsymbol{x}) + \frac{k'+1}{k'+2}\left(\sum_{i=1}^{k'}\frac{1}{k'+1}f(\boldsymbol{\theta}_{\mathrm{PT}} + (k'+2)\alpha_i\tau_i; \boldsymbol{x}) + \frac{1}{k'+1}f(\boldsymbol{\theta}_{\mathrm{PT}}; \boldsymbol{x})\right)\\
\approx&\sum_{i=1}^{k'+1}\frac{1}{k'+2}f(\boldsymbol{\theta}_{\mathrm{PT}} + (k'+2)\alpha_i\tau_i; \boldsymbol{x}) + \frac{1}{k'+2}f(\boldsymbol{\theta}_{\mathrm{PT}}; \boldsymbol{x})
\end{aligned}$$

Therefore, Theorem B.6 holds when $k = k' + 1$, and this finishes our proof. □

B.5. Proof of Theorem 5.1

Theorem B.7 (**The Emergence of CTL**). *Suppose $f(\boldsymbol{\theta}) : \mathbb{R}^p \mapsto \mathbb{R}$ is third differentiable in an open convex set $\boldsymbol{\Theta}$ and the its Hessian norm at $\boldsymbol{\theta}_0$ is bounded by $\lambda_{min} \leq \left\|\nabla^2 f(\boldsymbol{\theta}_0)\right\| \leq \lambda_{max}$, then*

$$\delta_{i,j} = |f(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) - \alpha f(\boldsymbol{\theta}_i) - (1-\alpha)f(\boldsymbol{\theta}_j)| \leq \frac{\alpha(1-\alpha)\lambda_{max}}{2}\left\|\boldsymbol{\theta}_j - \boldsymbol{\theta}_i\right\|^2 + \mathcal{E},$$

where $\mathcal{E} = O\left(\max\left(\left\|\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0\right\|^3, \alpha\left\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_0\right\|^3, (1-\alpha)\left\|\boldsymbol{\theta}_j - \boldsymbol{\theta}_0\right\|^3\right)\right)$ is the high order term.

Proof. Since f is third differentiable in an open convex set $\boldsymbol{\Theta}$, then by Taylor's Theorem, for any $\boldsymbol{\theta}_0, \boldsymbol{\theta} \in \boldsymbol{\Theta}$,

$$f(\boldsymbol{\theta}) = f(\boldsymbol{\theta}_0) + \nabla f(\boldsymbol{\theta}_0)^\top (\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\boldsymbol{\theta} - \boldsymbol{\theta}_0) + R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta} - \boldsymbol{\theta}_0).$$

where the remainder term $R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) = O\left(\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^3\right)$. Suppose $\lambda_{\min} \leq \left\|\nabla^2 f(\boldsymbol{\theta}_0)\right\| \leq \lambda_{\max}$, we have

$$\frac{\lambda_{\min}}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2 \leq \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\boldsymbol{\theta} - \boldsymbol{\theta}_0) \leq \frac{\lambda_{\max}}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2 .$$

Then for $f(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j)$, by Taylor's Theorem,

$$\begin{aligned}
& f(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) \\
&= f(\boldsymbol{\theta}_0) + \nabla f(\boldsymbol{\theta}_0)^\top (\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) \\
&\quad + \frac{1}{2} (\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) + R_{\boldsymbol{\theta}_0,3}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) \\
&= f(\boldsymbol{\theta}_0) + \nabla f(\boldsymbol{\theta}_0)^\top (\alpha (\boldsymbol{\theta}_i - \boldsymbol{\theta}_0) + (1-\alpha) (\boldsymbol{\theta}_j - \boldsymbol{\theta}_0)) \\
&\quad + \frac{1}{2} (\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) + R_{\boldsymbol{\theta}_0,3}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) \\
&= \alpha \left(f(\boldsymbol{\theta}_0) + \nabla f(\boldsymbol{\theta}_0)^\top (\boldsymbol{\theta}_i - \boldsymbol{\theta}_0)\right) + (1-\alpha) \left(f(\boldsymbol{\theta}_0) + \nabla f(\boldsymbol{\theta}_0)^\top (\boldsymbol{\theta}_i - \boldsymbol{\theta}_0)\right) \\
&\quad + \frac{1}{2} (\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) + R_{\boldsymbol{\theta}_0,3}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) \\
&= \alpha \left(f(\boldsymbol{\theta}_i) - \frac{1}{2} (\boldsymbol{\theta}_i - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\boldsymbol{\theta}_i - \boldsymbol{\theta}_0) - R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_0)\right) \\
&\quad + (1-\alpha) \left(f(\boldsymbol{\theta}_j) - \frac{1}{2} (\boldsymbol{\theta}_j - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) - R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta}_j - \boldsymbol{\theta}_0)\right) \\
&\quad + \frac{1}{2} (\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) + R_{\boldsymbol{\theta}_0,3}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) \\
&= \alpha f(\boldsymbol{\theta}_i) + (1-\alpha) f(\boldsymbol{\theta}_j) \\
&\quad + \frac{1}{2} (\alpha (\boldsymbol{\theta}_i - \boldsymbol{\theta}_0) + (1-\alpha) (\boldsymbol{\theta}_j - \boldsymbol{\theta}_0))^\top \nabla^2 f(\boldsymbol{\theta}_0) (\alpha (\boldsymbol{\theta}_i - \boldsymbol{\theta}_0) + (1-\alpha) (\boldsymbol{\theta}_j - \boldsymbol{\theta}_0)) \\
&\quad - \frac{\alpha}{2} (\boldsymbol{\theta}_i - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\boldsymbol{\theta}_i - \boldsymbol{\theta}_0) - \frac{(1-\alpha)}{2} (\boldsymbol{\theta}_j - \boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) \\
&\quad + R_{\boldsymbol{\theta}_0,3}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) - \alpha R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_0) - (1-\alpha) R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) \\
&= \alpha f(\boldsymbol{\theta}_i) + (1-\alpha) f(\boldsymbol{\theta}_j) \\
&\quad - \frac{\alpha(1-\alpha)}{2} (\boldsymbol{\theta}_j - \boldsymbol{\theta}_i)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\boldsymbol{\theta}_j - \boldsymbol{\theta}_i) \\
&\quad + R_{\boldsymbol{\theta}_0,3}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) - \alpha R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_0) - (1-\alpha) R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta}_j - \boldsymbol{\theta}_0).
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
& |f(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) - \alpha f(\boldsymbol{\theta}_i) - (1-\alpha) f(\boldsymbol{\theta}_j)| \\
&= \left| -\frac{\alpha(1-\alpha)}{2} (\boldsymbol{\theta}_j - \boldsymbol{\theta}_i)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\boldsymbol{\theta}_j - \boldsymbol{\theta}_i) + R_{\boldsymbol{\theta}_0,3}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) - \alpha R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_0) - (1-\alpha) R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) \right| \\
&\leq \left| \frac{\alpha(1-\alpha)}{2} (\boldsymbol{\theta}_j - \boldsymbol{\theta}_i)^\top \nabla^2 f(\boldsymbol{\theta}_0) (\boldsymbol{\theta}_j - \boldsymbol{\theta}_i) \right| + |R_{\boldsymbol{\theta}_0,3}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0) - \alpha R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_0) - (1-\alpha) R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta}_j - \boldsymbol{\theta}_0)| \\
&\leq \frac{\alpha(1-\alpha)\lambda_{\max}}{2} \|\boldsymbol{\theta}_j - \boldsymbol{\theta}_i\|^2 + O\left(\max\left(\|\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j - \boldsymbol{\theta}_0\|^3, \alpha \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_0\|^3, (1-\alpha) \|\boldsymbol{\theta}_j - \boldsymbol{\theta}_0\|^3\right)\right).
\end{aligned}$$

where the last inequality is because $\lambda_{\min} \leq \left\|\nabla^2 f(\boldsymbol{\theta}_0)\right\| \leq \lambda_{\max}$ and $R_{\boldsymbol{\theta}_0,3}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) = O\left(\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^3\right)$.

□

C. More Experimental Results

C.1. Detailed Experimental Settings

C.1.1. Experimental Settings in Section 4.1

Multi-Layer Perceptron on the Rotated MNIST Dataset.
Following the settings outlined by Mirzadeh et al. (2021), we adopt the multi-layer perceptron with two hidden layers with 100 units for Rotated MNIST dataset. ReLU activation functions are adopted between linear layers. Therefore, the multi-layer perceptron has 4 linear layers (1 for input, 2 for hidden and 1 for output) and 3 ReLU layers. We pretrain the MLP on normal MNIST and finetune it on Rotated MNIST, where each digit are rotated by a specific angle. We use rotation angle degrees of $\{0°, 22.5°, 45°, 67.5°, 90°\}$. Optimization is done with the default SGD algorithm and the learning rate of 1×10^{-1}, the batch size is set to 64 and the training epoch is set to 1 for both pretraining and finetuning.

We have 4 finetuned MLPs, yielding 6 non-repeated combinations of two finetuned models $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ in total. CTL are evaluated for each combinations on the union of their finetuning tasks $(\mathcal{D}_i \cup \mathcal{D}_j)$ which have $20,000$ test samples.

ResNet-18 on the Split CIFAR-100 Dataset.
Still following the settings outlined by Mirzadeh et al. (2021), we adopt the ResNet-18 architecture (He et al., 2016) on the Split CIFAR-100 dataset. The Split CIFAR-100 dataset is divided by classes, and 5 consecutive categories of CIFAR-100 are grouped into one split, having 20 splits in total. We use the first split as pretraining task and the second to fifth splits as finetuning tasks. We pretrain the ResNet-18 on first split and finetune it on the rest 4 splits respectively, acquiring 4 finetuned ResNet-18 checkpoints. No data augmentation techniques are adopted and optimization is done using the default SGD algorithm with learning rate of 5×10^{-2}. The batch size is set to 64. The training epoch is set to 10 for both pretraining and finetuning.

Similar to the setup of the Rotated MNIST experiment, we have 4 finetuned ResNet-18 models, yielding 6 non-repeated combinations of two finetuned models $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ in total. CTL are evaluated for each combinations and on the union of their finetuning tasks $(\mathcal{D}_i \cup \mathcal{D}_j)$ which have $20,000$ test samples.

C.1.2. Experimental Settings in Section 4.2

Model Averaging Accuracy v.s. Logits Ensemble Accuracy.
We choose 20 out of the 72 ViT-B/32 (Dosovitskiy et al., 2020) checkpoints that are finetuned on ImageNet (Deng et al., 2009) and open-sourced by Wortsman et al. (2022a), yielding $\binom{20}{3} = 1140$ non-repeated combinations of three finetuned ViT-B/32 models. For each combination of the finetuned models, we evaluated model averaging accuracy and logits ensemble accuracy on $10,000$ test samples from ImageNet.

Verification of Equation (1)
For ViT-B/32 on CIFAR-10, We train our ViT-B/32 initialized from same CLIP pretrained checkpoint but finetuned on CIFAR-10 dataset with different hyper-parameters to obtain 5 checkpoints to validate Equation (1). For ViT-B/32 on ImageNet, we choose 10 out of the 72 ViT-B/32 checkpoints that are finetuned on ImageNet and open-sourced by Wortsman et al. (2022a) to validate Equation (1). For both cases, we perform experiments on randomly-selected $10,000$ samples from the test set.

It's worth mentioning that in the forward pass of ViT models, the input in the shape of `(batch_size, patches_num, hidden_dim)` will be permuted to `(patches_num, batch_size, hidden_dim)`. We permute the internal feature back and reshape it into `(batch_size, patches_num × hidden_dim)`. Now, the dimension of the features is simply `patches_num × hidden_dim`.

C.1.3. Experimental Settings in Section 4.3

CTL Explains Learning via Addition.
We present the experimental settings in (*i*) Cross-Task Linearity (CTL) and (*ii*) Model Stitching experiment, respectively.

***i*) Cross-Task Linearity (CTL) experiment.**
Vision Transformer (Dosovitskiy et al., 2020): we evaluate ViT-B/32 and ViT-L/14 on 8 image classification datasets: Cars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Stallkamp et al., 2011), MNIST (LeCun & Cortes, 2005), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2016), SVHN (Netzer et al., 2011)).

8 finetuned ViT-B/32 (ViT-L/14) models generate $\binom{8}{2} = 28$ non-repeated combinations of two task vectors in total. For each combination of the two task vectors, we validate Equation (2) on on the union of their finetuning datasets ($\mathcal{D}_i \cup \mathcal{D}_j$) which has $10,000$ test samples in total.

T5 (Raffel et al., 2020): we evaluate T5 on 6 NLP datasets: IMDB (Maas et al., 2011), RACE (Lai et al., 2017), QASC (Khot et al., 2020), MultiNews (Fabbri et al., 2019), SQuAD (Rajpurkar et al., 2016), CommonGen (Lin et al., 2019), as same setup in Ilharco et al. (2023). 6 finetuned T5-base models generate $\binom{6}{2} = 15$ non-repeated combinations of two task vectors in total. For each combination of the two task vectors, we validate Equation (2) on on the union of their finetuning datasets ($\mathcal{D}_i \cup \mathcal{D}_j$) which has $1,000$ test samples in total. As T5 is a encoder-decoder architecture and sentences are varied in their lengths, we adopt the convention in sentence-T5 (Ni et al., 2021), which uses (i) the average pooling of tokens in the encoder to represent the internal feature of a sentence and (ii) the decoder's hidden states when generating first token (which is equivalent to attention pooling) to represent the feature of a sentence in decoder.

***ii*) Model Stitching experiment.**
We only validate ViT architectures (ViT-B/32, ViT-L/14) on the aforementioned 8 image classification datasets. We follow the **Cross-Task Linearity (CTL) experiment** settings except for the evaluation data size, which is of $2,000$ in this case. Notably, it is impossible for us to include the results for all 28 combinations, and thus, part of our experimental results will be presented, which is the same for the other experiments.

CTL Explains Learning via Negation.
Similar to the **Learning via Addition** setup, the datasets and architectures are kept the same.

***i*) Cross-Task Linearity (CTL) experiment.**
We evaluate both ViT and T5 architectures. For ViT architectures, we evaluate the 8 finetuned models on their corresponding finetuned datasets, each having $10,000$ test samples. For T5 architectures, we evaluate the 6 finetuned models on their downstream finetuned datasets, each having $10,000$ test samples.

***ii*) Model Stitching experiment.**
In Model Stitching experiment, we only validate ViT (ViT-B/32, ViT-L/14) architectures. We follow the same settings as above except for evaluation data size, which is of $2,000$ in this case.

C.1.4. Experimental Settings in Section 5

***i)* The number of pretraining/finetuning epochs.** We adopt the **ResNet-18 on the Split CIFAR-100 Dataset** setting in Appendix C.1.1. For validating the impact of pretraining epochs, we vary the number of pretraining epochs from 0 to 20 and fix the number of finetuning epochs to 10. For validating the impact of finetuning epochs, we fix the number of pretraining epochs to 10 and vary the number of finetuning epochs from 0 to 20. We use the combination of $\mathcal{D}_1$ and $\mathcal{D}_2$.

***ii)* The task similarity.** We use the Split ImageNet-1k (Deng et al., 2009) instead of Split CIFAR-100 as pretraining and finetuning datasets in practice to confidentially make sure the differences between datasets significant. Similar to the setting of **ResNet-18 on the Split CIFAR-100 Dataset** setting in Appendix C.1.1, the Split ImageNet-1k dataset is divided by classes, and 10 consecutive categories are grouped into one split. We use the first split as pretraining datasets and the second/third split as $\mathcal{D}_1$/$\mathcal{D}_2$. We use the default training hyper-parameters in torchvision (contributors, 2016) to pretrain/finetune the ResNet-18.

C.2. Verification of Theorem 5.1

In this section, we conduct experiments to validate the theoretical analysis presented in Theorem 5.1. Specifically, we demonstrate that $\delta_{i,j}$ exhibits a stronger correlation with $\frac{\alpha(1-\alpha)\lambda_{\text{max}}}{2} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2$ compared to λ_{max} or $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2$ alone.

For each pair of $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$, we calculate the distance between finetuned models, i.e., $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2$, and $\delta_{i,j}$, i.e., $|f(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) - \alpha f(\boldsymbol{\theta}_i) - (1-\alpha) f(\boldsymbol{\theta}_j)|$ where $\alpha = 0.5$. We also compute the largest eigenvalue of the Hessian matrix of $f(\cdot)$ at $\boldsymbol{\theta}_0$, i.e., $\lambda_{\max}$. Here, $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ denote models that are initialized from a common checkpoint and finetuned on the same dataset with different hyperparameters. The function $f(\cdot)$ represents the loss function $\mathcal{L}(\cdot)$, and $\boldsymbol{\theta}_0$ is simply chosen as $\boldsymbol{\theta}_{\text{PT}}$.

For ResNet-20 models finetuned on the CIFAR-10 dataset, we find that if we use $\frac{\alpha(1-\alpha)\lambda_{\text{max}}}{2} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2$ to fit a regression model to predict $\delta_{i,j}$, the R-squared value of the model is approximately 0.903. However, if we use only λ_{max} or $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2$ to fit the regression model, the R-squared value of the model is approximately 0.782 or 0.839, respectively. Therefore,

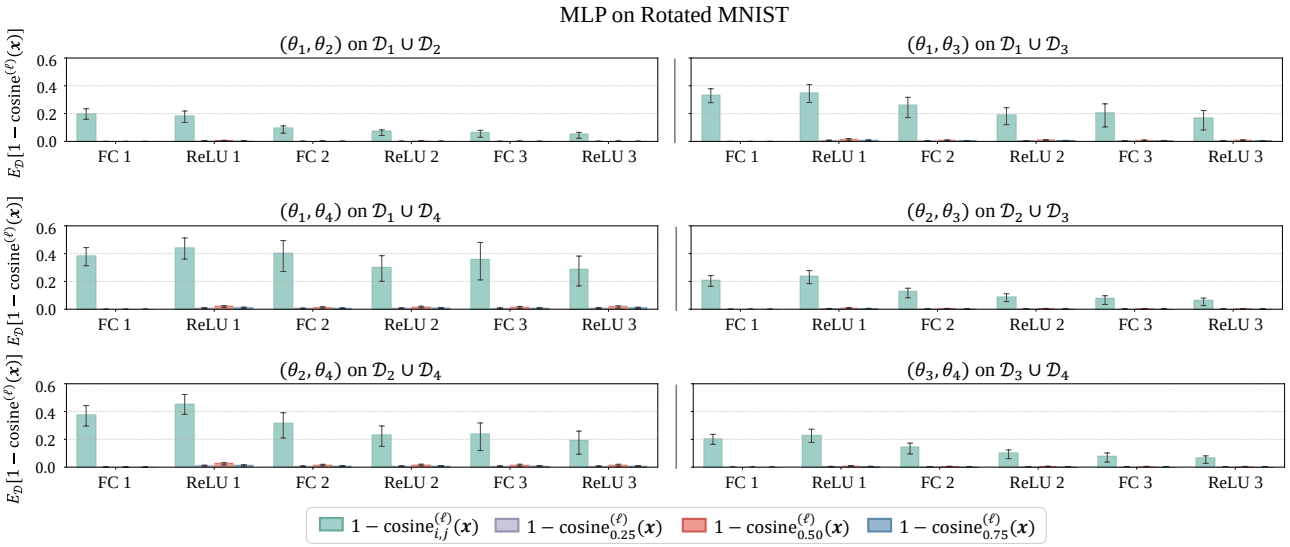


Figure 13. Verification of CTL. Compare $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{\alpha}(\boldsymbol{x})]$ with $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{i,j}(\boldsymbol{x})]$. Here, $\{\boldsymbol{\theta}_i\}_{i=1}^4$ and $\{\mathcal{D}_i\}_{i=1}^4$ denotes four finetuned MLPs on corresponding Rotated MNIST with rotation $\in \{22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ\}$ respectively. The results are reported for all layers of finetuned MLP, with $\alpha \in \{0.25, 0.5, 0.75\}$.

we conclude that $\delta_{i,j}$ indeed demonstrates a strong correlation with $\frac{\alpha(1-\alpha)\lambda_{\max}}{2}\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2$. It is worth noting that such correlation is a joint effect of both $\lambda_{\max}$ and $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2$, implying that either reducing $\lambda_{\max}$ or $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2$ leads to the fulfillment of CTL.

C.3. More Verification of CTL

In this section, we provide more experimental results about the CTL on MLP and ResNet-18 in different task combinations and different layers, which shows the CTL holds in the pretraining-finetuning paradigm. In Figure 13 and Figure 14, we include experimental results of $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{\alpha}(\boldsymbol{x})]$ and $\text{coef}^{(\ell)}_{\alpha}(\boldsymbol{x})$ for MLPs on Rotated MNIST dataset. In Figure 15 and Figure 16, we include experimental results of $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{\alpha}(\boldsymbol{x})]$ and $\text{coef}^{(\ell)}_{\alpha}(\boldsymbol{x})$ for ResNet-18 on Split CIFAR-100 dataset.

C.4. More Verification of Equation (1)

In this section, we provide more experimental results about the CTL in model averaging on ViT-B/32 to validate Equation (1). In Figure 17, we include the $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{avg}(\boldsymbol{x})]$ and $\text{coef}^{(\ell)}_{avg}(\boldsymbol{x})$ for ViT-B/32 on CIFAR-10. In Figure 18, we include the $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{avg}(\boldsymbol{x})]$ and $\text{coef}^{(\ell)}_{avg}(\boldsymbol{x})$ for ViT-B/32 on ImageNet. Results are reported across all blocks of ViT-B/32.

C.5. More Verification of Equation (2)

In this section, we provide more results about the CTL in task arithmetic on ViT-B/32, ViT-L/14 , T5 and Llama-2-13B architectures to validate Equation (2). We report both $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)]$ and $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)$. In Figure 19, we provide more results for ViT-B/32 architecture and all the blocks of ViT-B/32 are reported. In Figure 20, we provide more results for ViT-L/14 architecture and the last 12 blocks of ViT-L/14 are reported. In Figure 21, we provide more results on T5 architecture and the last 6 encoder blocks and last 6 decoder blocks of T5 are reported.

C.6. More Verification of Equation (3)

In this section, we provide results about the CTL in task arithmetic for more task vectors of ViT-B/32, ViT-L/14 and T5 architectures to validate Equation (3). We report both $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)]$ and $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)$. In Figure 22, we provide more results for ViT-B/32 architecture and all the blocks of ViT-B/32 are reported. In Figure 23, we provide more results for ViT-L/14 architecture and last 12 blocks of ViT-L/14 are reported. In Figure 24, we provide more results for T5 architecture and the last 6 encoder blocks and last 6 decoder blocks of T5 are reported.

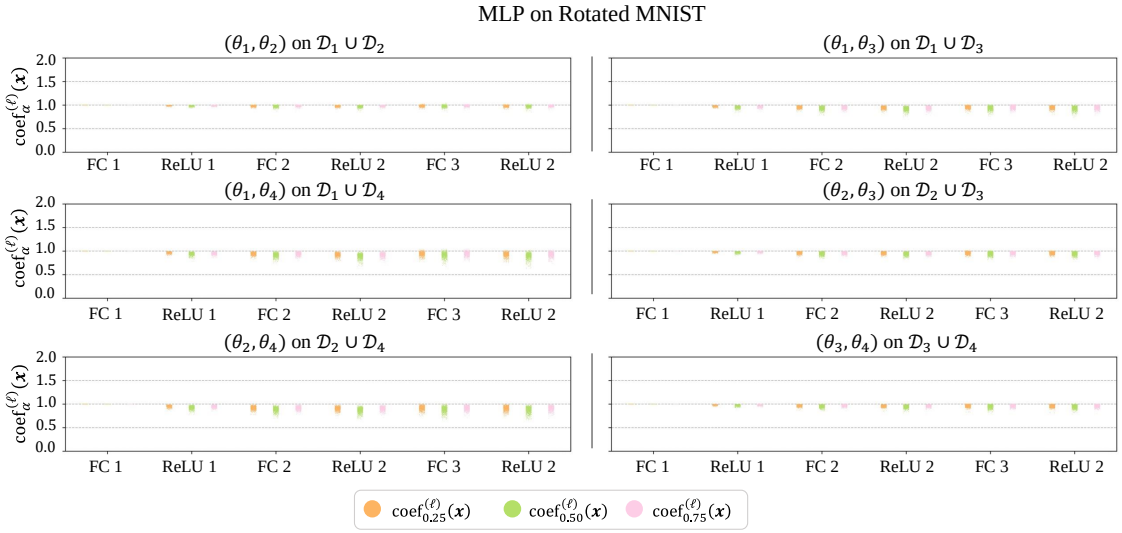


Figure 14. Verification of CTL. Distribution of $\mathrm{coef}_\alpha^{(\ell)}(\boldsymbol{x})$ across the datasets. Here, $\{\boldsymbol{\theta}_i\}_{i=1}^4$ and $\{\mathcal{D}_i\}_{i=1}^4$ denotes four finetuned MLPs on the corresponding Rotated MNIST with rotation $\in \{22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ\}$ respectively. The results are reported for all layers except classification head of finetuned MLPs, with $\alpha \in \{0.25, 0.5, 0.75\}$.

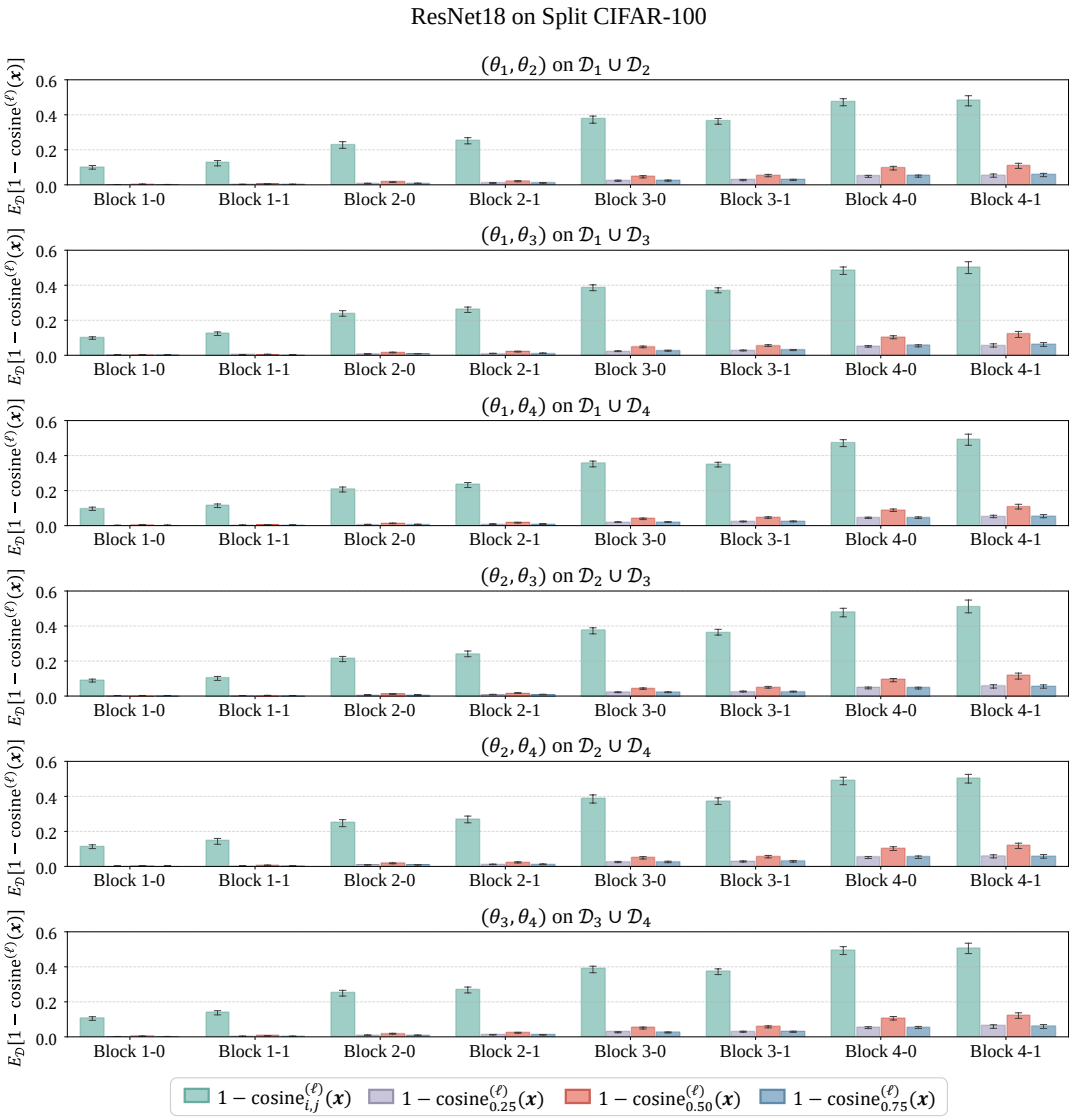


Figure 15. Verification of CTL. Compare $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{\alpha}(\boldsymbol{x})]$ with $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{i,j}(\boldsymbol{x})]$. Here, $\{\boldsymbol{\theta}_i\}_{i=1}^4$ and $\{\mathcal{D}_i\}_{i=1}^4$ denotes four finetuned ResNet-18s and the second to fifth splits in Split CIFAR-100 respectively. The results are reported for all blocks of finetuned ResNet-18 models, with $\alpha \in \{0.25, 0.5, 0.75\}$.

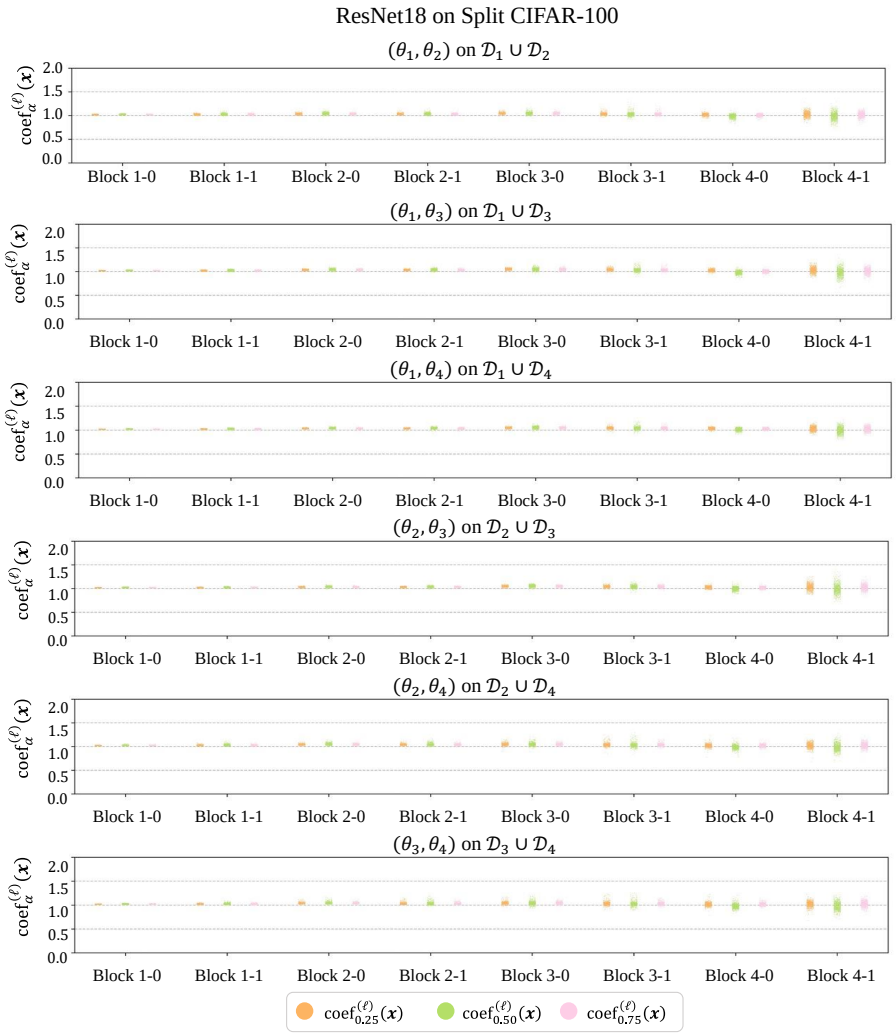


Figure 16. Verification of CTL. Distribution of $\mathrm{coef}_{\alpha}^{(\ell)}(\boldsymbol{x})$ across the datasets. Here, $\{\boldsymbol{\theta}_i\}_{i=1}^4$ and $\{\mathcal{D}_i\}_{i=1}^4$ denotes four finetuned ResNet-18s and the second to fifth splits in Split CIFAR-100 respectively. The results are reported for all blocks of finetuned ResNet-18 models, with $\alpha \in \{0.25, 0.5, 0.75\}$.

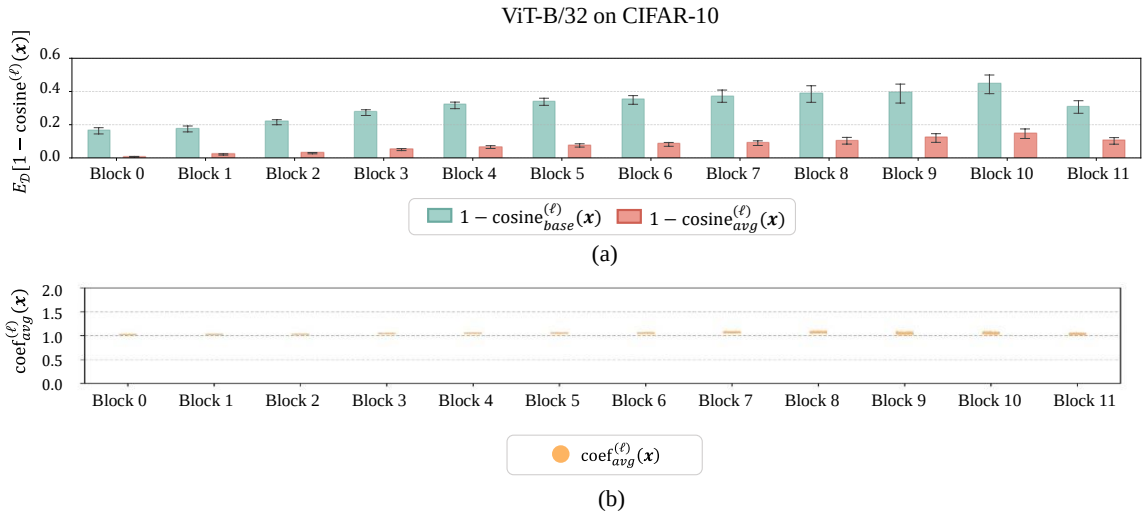


Figure 17. Verification of CTL in model averaging. (a) Compare $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}^{(\ell)}_{avg}(\boldsymbol{x})]$ with $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}^{(\ell)}_{base}(\boldsymbol{x})]$. (b) Distribution of $\text{coef}^{(\ell)}_{avg}(\boldsymbol{x})$ on CIFAR-10. The results are reported for all blocks of ViT-B/32 models finetuned on CIFAR-10 with different hyper-parameters.

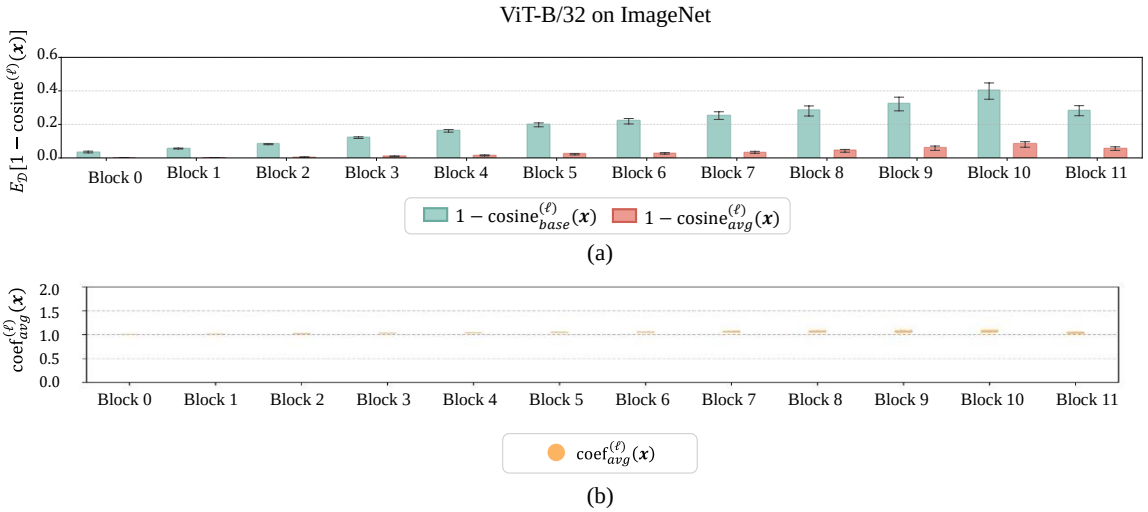


Figure 18. Verification of CTL in model averaging. (a) Compare $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}^{(\ell)}_{avg}(\boldsymbol{x})]$ with $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}^{(\ell)}_{base}(\boldsymbol{x})]$. (b) Distribution of $\text{coef}^{(\ell)}_{avg}(\boldsymbol{x})$ on ImageNet. The results are reported for all blocks of ViT-B/32 models finetuned on ImageNet with different hyper-parameters.

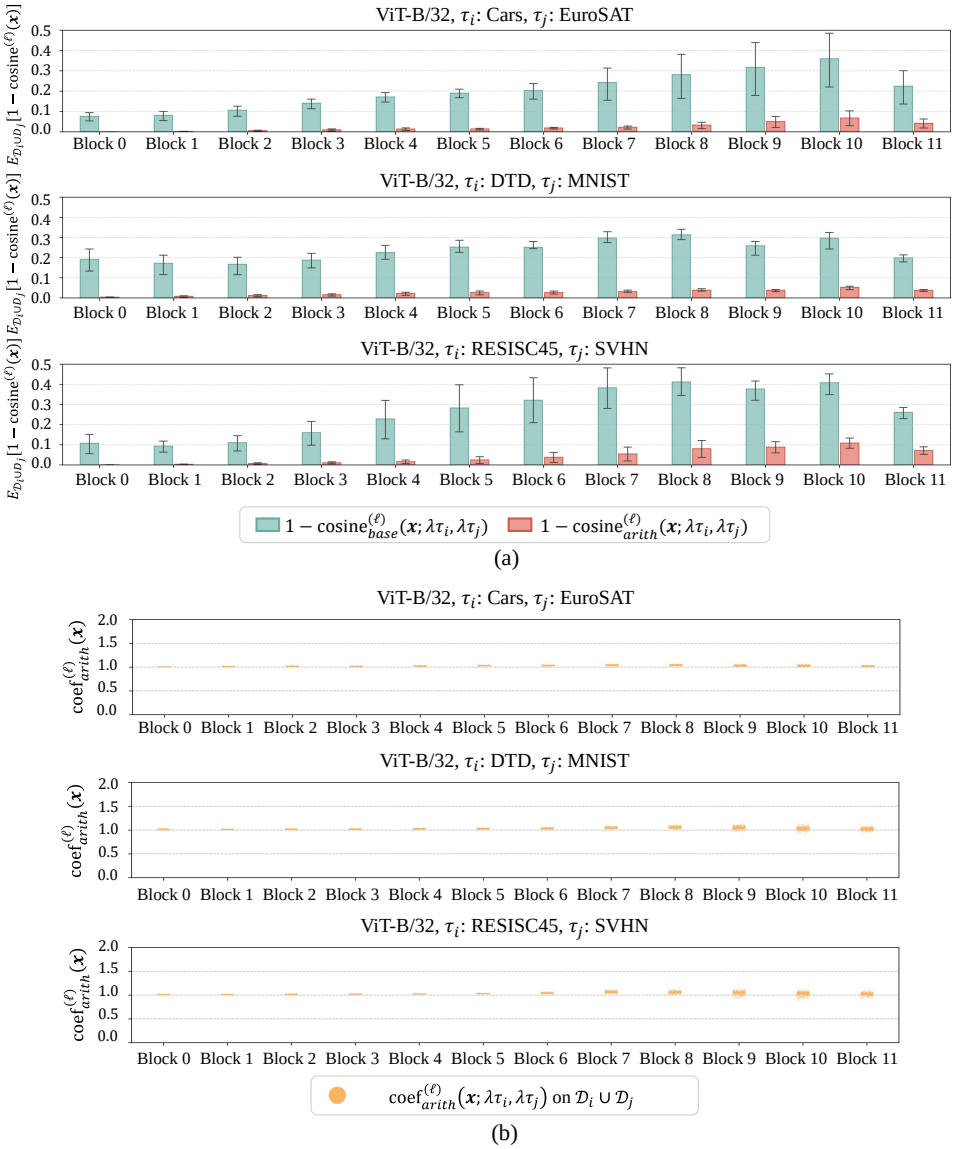
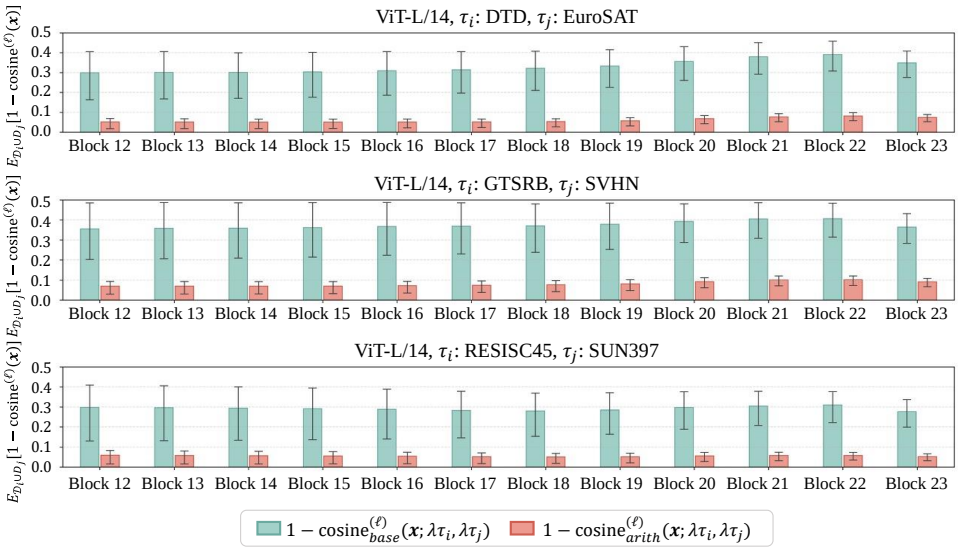
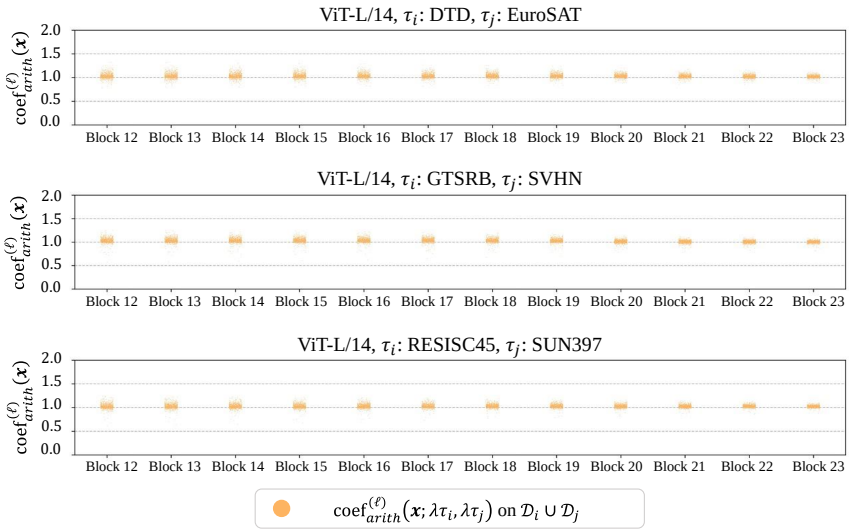


Figure 19. Verification of **Learning via Addition** in task arithmetic. (a) Compare $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)]$ with $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{base}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)]$. The bottom and top of the error bar represent the lower and upper quartile of the values across the dataset, respectively. (b) Distribution of $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)$. The results are reported for all blocks of finetuned ViT-B/32 under different settings, with $\lambda = 0.4$ and $\alpha = 0.5$.



(a)



(b)

Figure 20. Verification of **Learning via Addition** in task arithmetic. (a) Compare $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)]$ with $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{base}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)]$. The bottom and top of the error bar represent the lower and upper quartile of the values across the dataset, respectively. (b) Distribution of $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)$. The results are reported for the last 12 blocks of finetuned ViT-L/14 under different settings, with $\lambda = 0.4$ and $\alpha = 0.5$.

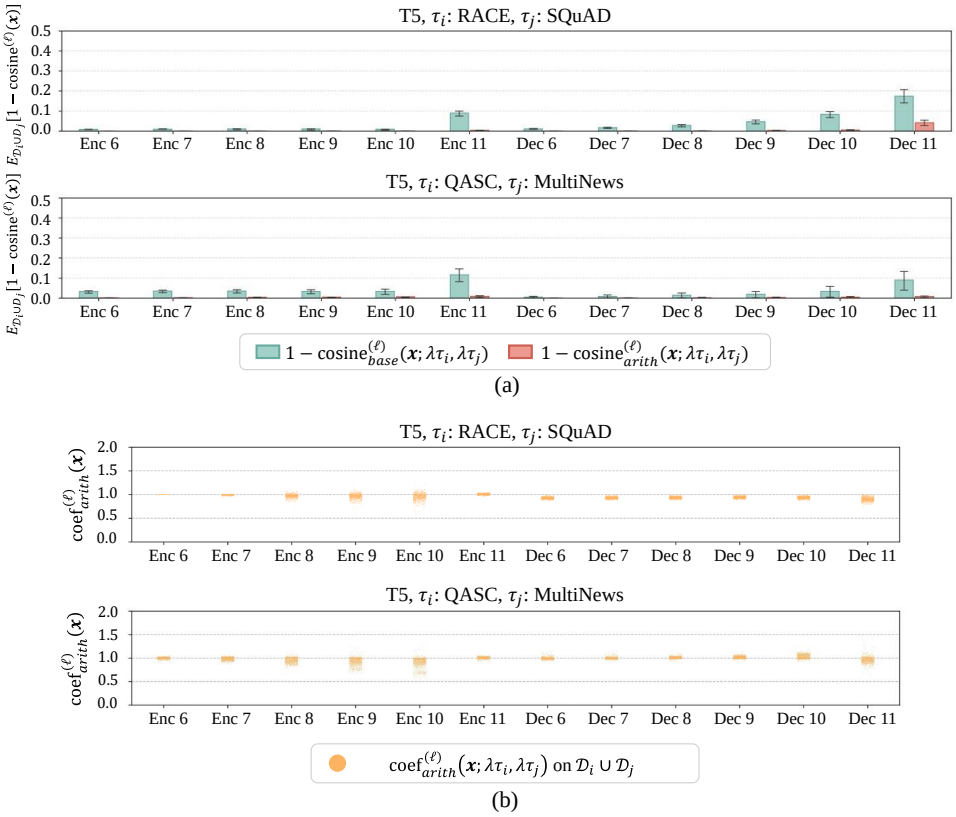
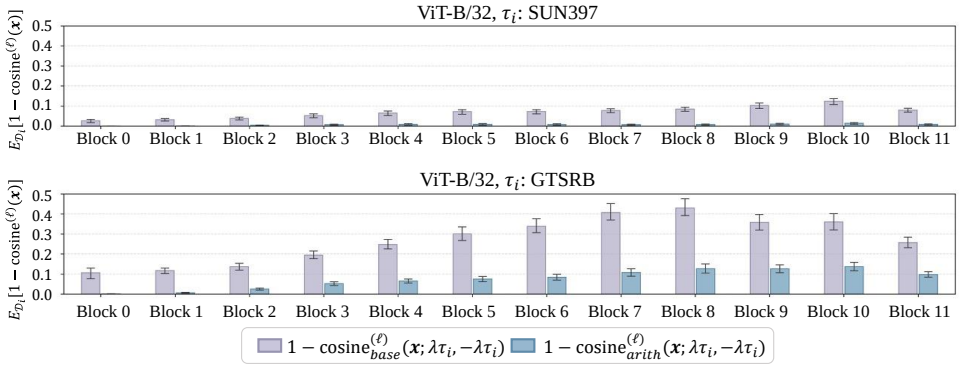
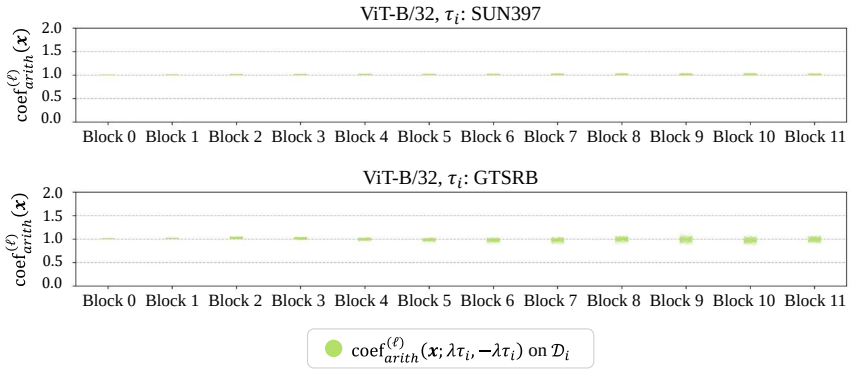


Figure 21. Verification of **Learning via Addition** in task arithmetic. (a) Compare $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)]$ with $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{base}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)]$. The bottom and top of the error bar represent the lower and upper quartile of the values across the dataset, respectively. (b) Distribution of $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, \lambda\tau_j)$. The results are reported for the last 6 encoder blocks and the last 6 decoder blocks of finetuned T5 under different settings, with $\lambda = 0.4$ and $\alpha = 0.5$.



(a)



(b)

Figure 22. (a) Compare $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)]$ with $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{base}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)]$. The bottom and top of the error bar represent the lower and upper quartile of the values across the dataset, respectively. (b) Distribution of $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)$. The results are reported for all blocks of finetuned ViT-B/32 under different settings, with $\lambda = 0.4$ and $\alpha = 0.5$.

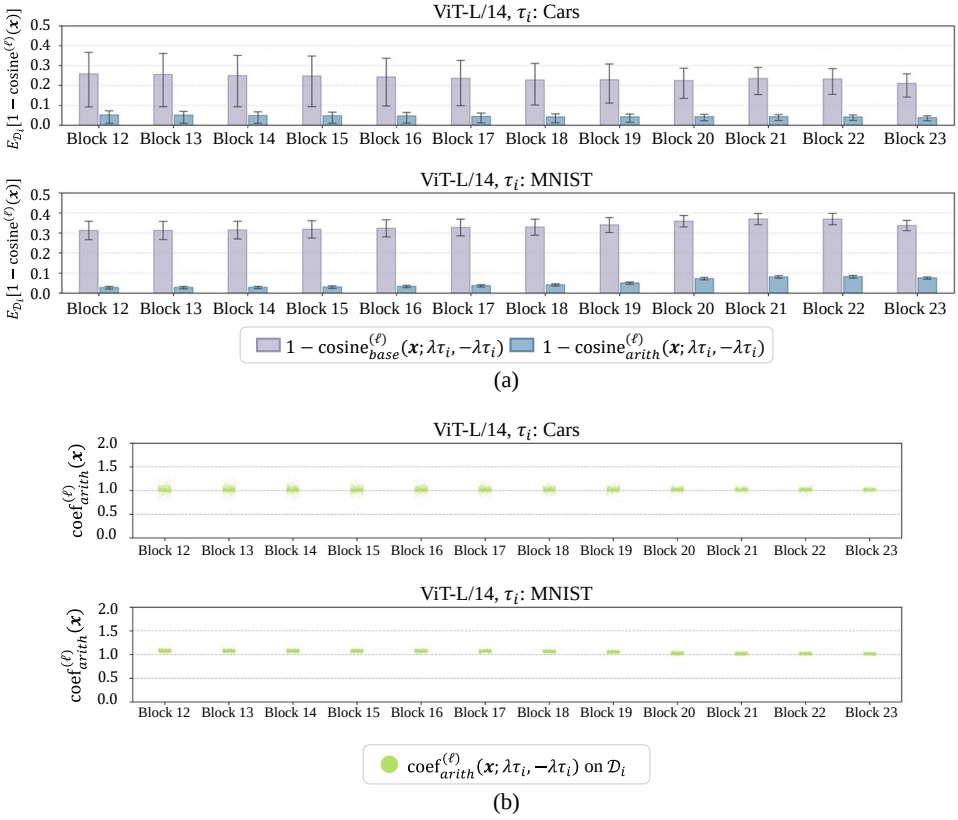
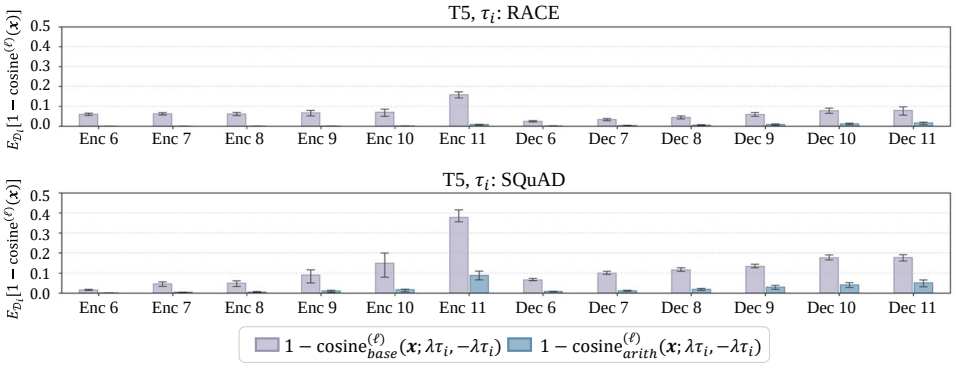
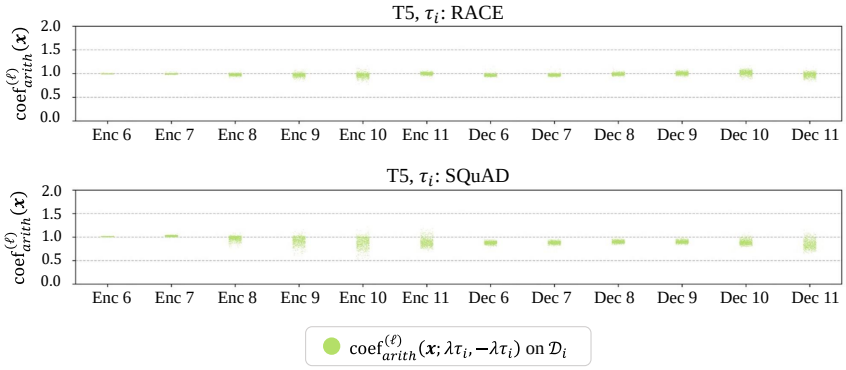


Figure 23. Verification of **Learning via Negation** in task arithmetic. (a) Compare $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)]$ with $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{base}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)]$. The bottom and top of the error bar represent the lower and upper quartile of the values across the dataset, respectively. (b) Distribution of $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)$. The results are reported for the last 12 blocks of finetuned ViT-L/14 under different settings, with $\lambda = 0.4$ and $\alpha = 0.5$.



(a)



(b)

Figure 24. Verification of **Learning via Negation** in task arithmetic. (a) Compare $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)]$ with $\mathbb{E}_{\mathcal{D}}[1 - \text{cosine}^{(\ell)}_{base}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)]$. The bottom and top of the error bar represent the lower and upper quartile of the values across the dataset, respectively. (b) Distribution of $\text{coef}^{(\ell)}_{arith}(\boldsymbol{x}; \lambda\tau_i, -\lambda\tau_i)$. The results are reported for the last 6 encoder blocks and the last 6 decoder blocks of finetuned T5 under different settings, with $\lambda = 0.4$ and $\alpha = 0.5$.