# A Proofs of Theorems and Derivations

## A.1 Proof of Lemma 1

In this section, we prove Lemma 1 in the main paper. This lemma indicates that we can directly imply Linear Mode Connectivity (LMC, see Definition 1) from Layerwise Linear Feature Connectivity (LLFC, see Definition 2) applied to last layer.

**Definition 1 (Linear Mode Connectivity).** *Given a test dataset $\mathcal{D}$ and two modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ such that $\mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_A) \approx \mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_B)$, we say $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ are linearly connected if they satisfy*

$$\mathrm{Err}_{\mathcal{D}}(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B) \approx \mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_A), \qquad \forall \alpha \in [0,1].$$

**Definition 2 (Layerwise Linear Feature Connectivity).** *Given dataset $\mathcal{D}$ and two modes $\boldsymbol{\theta}_A$, $\boldsymbol{\theta}_B$ of an $L$-layer neural network $f$, the modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ are said to be layerwise linearly feature connected if they satisfy*

$$\forall \ell \in [L], \forall \alpha \in [0,1], \exists c > 0, \textit{s.t. } cf^{(\ell)}(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B) = \alpha f^{(\ell)}(\boldsymbol{\theta}_A) + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_B).$$

**Lemma 1.** *Suppose two modes $\boldsymbol{\theta}_A$, $\boldsymbol{\theta}_B$ satisfy LLFC on a dataset $\mathcal{D}$ and*

$$\max\{\mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_A), \mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_B)\} \leq \epsilon,$$

*then we have*

$$\forall \alpha \in [0,1], \mathrm{Err}_{\mathcal{D}}(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B) \leq 2\epsilon.$$

*Proof.* Note that the classification depends on the relative order of the entries in the output of the final layer. As a consequence, for each data point in the dataset $\mathcal{D}$, the linear interpolation of the outputs of the models makes the correct classification if both models make the correct classification. Therefore, only if one of the model makes the incorrect classification, the linear interpolation of the outputs of the models would possibly make the incorrect classification, i.e,

$$\mathrm{Err}_{\mathcal{D}}(\alpha f(\boldsymbol{\theta}_A) + (1-\alpha) f(\boldsymbol{\theta}_B)) \leq \mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_A) + \mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_B).$$

Since $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ satisfy LLFC, then at last layer we have

$$f(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B) = \alpha f(\boldsymbol{\theta}_A) + (1-\alpha) f(\boldsymbol{\theta}_B),$$

then have

$$\mathrm{Err}_{\mathcal{D}}(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B) \leq \mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_A) + \mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_B).$$

According to the condition that

$$\max\{\mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_A), \mathrm{Err}_{\mathcal{D}}(\boldsymbol{\theta}_B)\} \leq \epsilon,$$

which indicates

$$\mathrm{Err}_{\mathcal{D}}(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B) \leq 2\epsilon,$$

and this finishes the proof. □

## A.2 Proof of Theorem 1

In this section, we prove Theorem 1 in the main paper. Theorem 1 indicates that we can derive LLFC from two simple conditions: weak additivity for ReLU activations (Definition 3) and commutativity (Definition 4). Note that though we consider a multi-layer perceptron (MLP) for convenience, our proof and results can be easily adopted to any feed-forward structure, e.g., a convolutional neural network (CNN).

**Definition 3 (Weak Additivity for ReLU Activations).** *Given a dataset $\mathcal{D}$, two modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ are said to satisfy weak additivity for ReLU activations if*

$$\forall \ell \in [L], \quad \sigma\left(\tilde{\boldsymbol{H}}_A^{(\ell)} + \tilde{\boldsymbol{H}}_B^{(\ell)}\right) = \sigma\left(\tilde{\boldsymbol{H}}_A^{(\ell)}\right) + \sigma\left(\tilde{\boldsymbol{H}}_B^{(\ell)}\right).$$

**Definition 4 (Commutativity).** *Given a dataset $\mathcal{D}$, two modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ are said to satisfy commutativity if*

$$\forall \ell \in [L],\ \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} + \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} = \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} + \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_A^{(\ell-1)}.$$

**Theorem 1.** *Given a dataset $\mathcal{D}$, if two modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ satisfy weak additivity for ReLU activations (Definition 3) and commutativity (Definition 4), then*

$$\forall \alpha \in [0,1], \forall \ell \in [L],\ f^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B\right) = \alpha f^{(\ell)}\left(\boldsymbol{\theta}_A\right) + (1-\alpha)\, f^{(\ell)}\left(\boldsymbol{\theta}_B\right).$$

*Proof.* Before delving into the proof, let us denote the forward propagation in each layer $\ell$ by

$$\begin{aligned}
\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}; \boldsymbol{H}^{(\ell-1)}\right) &= \boldsymbol{W}^{(\ell)} \boldsymbol{H}^{(\ell-1)} + \boldsymbol{b}^{(\ell)} \mathbf{1}_{d_\ell}^\top \\
g^{(\ell)}\left(\boldsymbol{\theta}; \boldsymbol{H}^{(\ell-1)}\right) &= \sigma\left(\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}; \boldsymbol{H}^{(\ell-1)}\right)\right) = \boldsymbol{H}^{(\ell)}
\end{aligned}$$

Given $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ that satisfy the commutativity property, then $\forall \ell \in [L]$ and $\forall \alpha \in [0,1]$, we have

$$\begin{aligned}
\boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} + \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} =& \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} + \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} \\
\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}_A^{(\ell-1)}\right) + \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}_B^{(\ell-1)}\right) =& \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}_B^{(\ell-1)}\right) + \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}_A^{(\ell-1)}\right) \\
\alpha\,(1-\alpha)\left(\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}_A^{(\ell-1)}\right) + \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}_B^{(\ell-1)}\right)\right) =& \alpha\,(1-\alpha)\left(\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}_B^{(\ell-1)}\right) + \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}_A^{(\ell-1)}\right)\right) \\
\alpha \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}_A^{(\ell-1)}\right) + (1-\alpha)\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}_B^{(\ell-1)}\right) =& \alpha^2 \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}_A^{(\ell-1)}\right) + (1-\alpha)^2 \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}_B^{(\ell-1)}\right) \\
&+ \alpha\,(1-\alpha)\left(\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}_B^{(\ell-1)}\right) + \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}_A^{(\ell-1)}\right)\right)
\end{aligned}$$

Additionally, we can easily verify that

$$\begin{aligned}
\tilde{g}^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \boldsymbol{H}^{(\ell)}\right) &= \alpha \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}^{(\ell)}\right) + (1-\alpha)\,\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}^{(\ell)}\right) \\
\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}; \alpha \boldsymbol{H}_A^{(\ell)} + (1-\alpha)\,\boldsymbol{H}_B^{(\ell)}\right) &= \alpha \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}; \boldsymbol{H}_A^{(\ell)}\right) + (1-\alpha)\,\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}; \boldsymbol{H}_B^{(\ell)}\right)
\end{aligned}$$

Subsequently,

$$\begin{aligned}
\alpha \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}_A^{(\ell-1)}\right) + (1-\alpha)\,\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}_B^{(\ell-1)}\right) =& \alpha \tilde{g}^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \boldsymbol{H}_A^{(\ell-1)}\right) \\
&+ (1-\alpha)\,\tilde{g}^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \boldsymbol{H}_B^{(\ell-1)}\right) \\
=& \tilde{g}^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \alpha \boldsymbol{H}_A^{(\ell-1)} + (1-\alpha)\,\boldsymbol{H}_B^{(\ell-1)}\right).
\end{aligned}$$

Given the weak additivity for ReLU activation is satisfied for $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$, then we have

$$\begin{aligned}
\sigma\left(\alpha \tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}_A^{(\ell-1)}\right) + (1-\alpha)\,\tilde{g}^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}_B^{(\ell-1)}\right)\right) &= \sigma\left(\tilde{g}^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \alpha \boldsymbol{H}_A^{(\ell-1)} + (1-\alpha)\,\boldsymbol{H}_B^{(\ell-1)}\right)\right) \\
\alpha g^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{H}_A^{(\ell-1)}\right) + (1-\alpha)\,g^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{H}_B^{(\ell-1)}\right) &= g^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \alpha \boldsymbol{H}_A^{(\ell-1)} + (1-\alpha)\,\boldsymbol{H}_B^{(\ell-1)}\right)
\end{aligned}$$

To conclude, $\forall \ell \in [L]$ and $\forall \alpha \in [0,1]$, we have

$$\alpha \boldsymbol{H}_A^{(\ell)} + (1-\alpha)\,\boldsymbol{H}_B^{(\ell)} = g^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \alpha \boldsymbol{H}_A^{(\ell-1)} + (1-\alpha)\,\boldsymbol{H}_B^{(\ell-1)}\right) \tag{10}$$

For the right hand side of Equation (10), recursively, we can have

$$\begin{aligned}
& g^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \alpha \boldsymbol{H}_A^{(\ell-1)} + (1-\alpha)\boldsymbol{H}_B^{(\ell-1)}\right) \\
=& g^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; g^{(\ell-1)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \alpha \boldsymbol{H}_A^{(\ell-2)} + (1-\alpha)\,\boldsymbol{H}_B^{(\ell-2)}\right)\right) \\
=& \left(g^{(\ell)} \circ g^{(\ell-1)}\right)\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \alpha \boldsymbol{H}_A^{(\ell-2)} + (1-\alpha)\,\boldsymbol{H}_B^{(\ell-2)}\right) \\
=& \cdots \\
=& \left(g^{(\ell)} \circ g^{(\ell-1)} \cdots \circ g^{(1)}\right)\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \boldsymbol{X}\right) \\
=& f^{(\ell)}\left(\alpha \boldsymbol{\theta}_A + (1-\alpha)\,\boldsymbol{\theta}_B; \boldsymbol{X}\right),
\end{aligned}$$

where $\boldsymbol{X}$ denotes the input data matrix.

Recall we denote $\boldsymbol{H}^{(\ell)} = f^{(\ell)}\left(\boldsymbol{\theta}; \boldsymbol{X}\right)$ which indicates

$$\alpha f^{(\ell)}\left(\boldsymbol{\theta}_A; \boldsymbol{X}\right) + (1-\alpha) f^{(\ell)}\left(\boldsymbol{\theta}_B; \boldsymbol{X}\right) = f^{(\ell)}\left(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B; \boldsymbol{X}\right),$$

and this finishes the proof. □

### A.3 Derivation of Quadratic Assignment Problem

In this section, we aim to show that minimizing $\sum_{\ell=1}^{L} \left\| \left( \boldsymbol{W}_A^{(\ell)} - \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)^\top} \right) \left( \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \right) \right\|_F^2$ includes solving Quadratic Assignment Problems (QAPs), known to be NP-hard.

$$\begin{aligned}
&\underset{\pi=\{\boldsymbol{P}^{(\ell)}\}}{\arg\min} \sum_{\ell=1}^{L} \left\| \left( \boldsymbol{W}_A^{(\ell)} - \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)^\top} \right) \left( \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \right) \right\|_F^2 \\
\iff &\underset{\pi=\{\boldsymbol{P}^{(\ell)}\}}{\arg\min} \sum_{\ell=1}^{L} \left\| \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} + \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \right\|_F^2 \\
\iff &\underset{\pi=\{\boldsymbol{P}^{(\ell)}\}}{\arg\min} \sum_{\ell=1}^{L} \Bigg( \left\| \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \right\|_F^2 + \left\| \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \right\|_F^2 \\
&\quad + \left\langle \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)}, \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \right\rangle_F \Bigg).
\end{aligned}$$

Consider its first term, i.e.,

$$\begin{aligned}
&\underset{\pi=\{\boldsymbol{P}^{(\ell)}\}}{\arg\min} \sum_{\ell=1}^{L} \left\| \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \right\|_F^2 \\
\iff &\underset{\pi=\{\boldsymbol{P}^{(\ell)}\}}{\arg\min} \sum_{\ell=1}^{L} \operatorname{tr}\left( \left( \boldsymbol{H}_A^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} - \boldsymbol{H}_B^{(\ell-1)^\top} \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \right) \left( \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \right) \right) \\
\iff &\underset{\pi=\{\boldsymbol{P}^{(\ell)}\}}{\arg\min} \sum_{\ell=1}^{L} \operatorname{tr}\Big( \boldsymbol{H}_A^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{H}_B^{(\ell-1)^\top} \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} \\
&\qquad\qquad - \boldsymbol{H}_A^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} + \boldsymbol{H}_B^{(\ell-1)^\top} \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \Big) \\
\iff &\underset{\pi=\{\boldsymbol{P}^{(\ell)}\}}{\arg\min} \sum_{\ell=1}^{L} \operatorname{tr}\left( -2\boldsymbol{H}_B^{(\ell-1)^\top} \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} + \boldsymbol{H}_B^{(\ell-1)^\top} \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \right) \\
\iff &\underset{\pi=\{\boldsymbol{P}^{(\ell)}\}}{\arg\min} \sum_{\ell=1}^{L} \operatorname{tr}\left( -2\boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)^\top} + \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)^\top} \right) \\
\iff &\underset{\pi=\{\boldsymbol{P}^{(\ell)}\}}{\arg\min} \sum_{\ell=1}^{L} \left( \operatorname{tr}\left( \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)^\top} \right) - 2 \operatorname{tr}\left( \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)^\top} \right) \right)
\end{aligned}$$

where $\operatorname{tr}\left( \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)^\top} \right) - 2\operatorname{tr}\left( \boldsymbol{P}^{(\ell-1)^\top} \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)^\top} \right)$ is in the form of Koopmans-Beckmann's QAP [14] for each $P^{(\ell-1)}$ and known as NP-hard. Thus, solving $\underset{\pi=\{\boldsymbol{P}^{(\ell)}\}}{\arg\min} \sum_{\ell=1}^{L} \left\| \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \right\|_F^2$ is to solve $L-1$ QAPs in parallel.

Similarly, consider the second term, i.e,

$$
\begin{aligned}
&\arg\min_{\pi=\{\boldsymbol{P}^{(\ell)}\}} \sum_{\ell=1}^{L} \left\| \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)\top} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \right\|_F^2 \\
\iff &\arg\min_{\pi=\{\boldsymbol{P}^{(\ell)}\}} \sum_{\ell=1}^{L} \left\| \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)\top} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \right\|_F^2 \\
\iff &\arg\min_{\pi=\{\boldsymbol{P}^{(\ell)}\}} \sum_{\ell=1}^{L} \operatorname{tr}\left( \left( \boldsymbol{H}_A^{(\ell-1)\top} \boldsymbol{P}^{(\ell-1)} \boldsymbol{W}_B^{(\ell)\top} - \boldsymbol{H}_B^{(\ell-1)\top} \boldsymbol{W}_B^{(\ell)\top} \right) \left( \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)\top} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \right) \right) \\
\iff &\arg\min_{\pi=\{\boldsymbol{P}^{(\ell)}\}} \sum_{\ell=1}^{L} \operatorname{tr}\Big( \boldsymbol{H}_A^{(\ell-1)\top} \boldsymbol{P}^{(\ell-1)} \boldsymbol{W}_B^{(\ell)\top} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)\top} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{H}_A^{(\ell-1)\top} \boldsymbol{P}^{(\ell-1)} \boldsymbol{W}_B^{(\ell)\top} \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \\
&\qquad\qquad - \boldsymbol{H}_B^{(\ell-1)\top} \boldsymbol{W}_B^{(\ell)\top} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)\top} \boldsymbol{H}_A^{(\ell-1)} + \boldsymbol{H}_B^{(\ell-1)\top} \boldsymbol{W}_B^{(\ell)\top} \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \Big) \\
\iff &\arg\min_{\pi=\{\boldsymbol{P}^{(\ell)}\}} \sum_{\ell=1}^{L} \operatorname{tr}\left( \boldsymbol{H}_A^{(\ell-1)\top} \boldsymbol{P}^{(\ell-1)} \boldsymbol{W}_B^{(\ell)\top} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)\top} \boldsymbol{H}_A^{(\ell-1)} - 2 \boldsymbol{H}_A^{(\ell-1)\top} \boldsymbol{P}^{(\ell-1)} \boldsymbol{W}_B^{(\ell)\top} \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \right) \\
\iff &\arg\min_{\pi=\{\boldsymbol{P}^{(\ell)}\}} \sum_{\ell=1}^{L} \operatorname{tr}\left( \boldsymbol{P}^{(\ell-1)} \boldsymbol{W}_B^{(\ell)\top} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)\top} \boldsymbol{H}_A^{(\ell-1)} \boldsymbol{H}_A^{(\ell-1)\top} - 2 \boldsymbol{P}^{(\ell-1)} \boldsymbol{W}_B^{(\ell)\top} \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \boldsymbol{H}_A^{(\ell-1)\top} \right) \\
\iff &\arg\min_{\pi=\{\boldsymbol{P}^{(\ell)}\}} \sum_{\ell=1}^{L} \left( \operatorname{tr}\left( \boldsymbol{P}^{(\ell-1)} \boldsymbol{W}_B^{(\ell)\top} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)\top} \boldsymbol{H}_A^{(\ell-1)} \boldsymbol{H}_A^{(\ell-1)\top} \right) - 2 \operatorname{tr}\left( \boldsymbol{P}^{(\ell-1)} \boldsymbol{W}_B^{(\ell)\top} \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \boldsymbol{H}_A^{(\ell-1)\top} \right) \right),
\end{aligned}
$$

which also gives rise to Koopmans-Beckmann's QAPs.

For the last term, i.e,

$$
\begin{aligned}
&\arg\min_{\pi=\{\boldsymbol{P}^{(\ell)}\}} \sum_{\ell=1}^{L} \left\langle \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)}, \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)\top} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \right\rangle_F \\
\iff &\arg\min_{\pi=\{\boldsymbol{P}^{(\ell)}\}} \sum_{\ell=1}^{L} \left\langle \left( \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_A^{(\ell)} \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \right) \left( \boldsymbol{W}_B^{(\ell)} \boldsymbol{P}^{(\ell-1)\top} \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \right)^\top, \boldsymbol{P}^{(\ell)} \right\rangle_F,
\end{aligned}
$$

which entails solving bi-level matching problems.

Therefore, the objective can be rewritten as the summation of QAPs and bi-level matching problems and cannot be further simplified, which is NP-hard.

# B More Experimental Details and Results

## B.1 Detailed Experimental Settings

In this section, we introduce the detailed experimental setup. Before delving into details, recall that unless otherwise specified, in this paper we consider models trained on a training set, and then all the investigations are evaluated on a test set.

### B.1.1 Spawning Method

**Multi-Layer Perceptrons on the MNIST Dataset.** In accordance with the settings outlined by Ainsworth et al. [1], we train multi-layer perceptron networks with three hidden layers, each consisting of 512 units, on the MNIST dataset. We adopt the ReLU activation between layers. Optimization is done with the Adam algorithm and a learning rate of $1.2 \times 10^{-4}$. The batch size is set to 60 and the total number of training epochs is 30. To find the modes that satisfy LMC, we start spawning from a common initialization $\boldsymbol{\theta}^{(0)}$.

**VGG-16 and ResNet-20 on the CIFAR-10 Dataset.** In accordance with the settings outlined by Frankle et al. [9], we train the VGG-16 architecture [29] and the ResNet-20 architecture [12] on the CIFAR-10 dataset. Data augmentation techniques include random horizontal flips and random $32 \times 32$ pixel crops. Optimization is done using SGD with momentum (momentum set to 0.9). A weight decay of $1 \times 10^{-4}$ is applied. The learning rate is initialized at $0.1$ and is dropped by $10$ times at $80$ and $120$ epochs. The total number of epochs is $160$. To find the modes that satisfy LMC, we start spawning after training $5$ epochs for both VGG-16 and ResNet-20.

**ResNet-50 on the Tiny-ImageNet Dataset.** In accordance with the settings outlined by Frankle et al. [9], we train the ResNet-50 architecture [12] on the Tiny-ImageNet dataset. Data augmentation techniques include random horizontal flips and random $32 \times 32$ pixel crops. Optimization is done using SGD with momentum (momentum set to 0.9). A weight decay of $1 \times 10^{-4}$ is applied. The learning rate is set to $0.4$ and warmed up for $5$ epochs and then is dropped by $10$ times at $30$, $60$ and $80$ epochs. The total number of epochs is $90$. To find the modes that satisfy LMC, we start spawning after training $14$ epochs.

### B.1.2 Permutation Method

For the permutation method, we follow the experimental settings of Ainsworth et al. [1] strictly, which are described below.

**Multi-Layer Perceptrons on MNIST and CIFAR-10.** Similar to the spawning method, we use multi-layer perceptron (MLP) networks with three hidden layers, each consisting of 512 units. For MNIST, optimization is performed using Adam with a learning rate of $1 \times 10^{-3}$. For CIFAR-10, optimization is performed using SGD with a learning rate of $0.1$. Both activation matching and weight matching are used to identify modes that satisfy LMC.

**ResNet-20 on CIFAR-10.** To achieve LMC, we modify the ResNet-20 architecture by incorporating LayerNorms in place of BatchNorms. Furthermore, we increase the width of ResNet-20 by a factor of 32. Data augmentation techniques include random horizontal flips, random $32 \times 32$ pixel crops, random resizes of the image between $0.8\times$ and $1.2\times$, and random rotations between $\pm 30^\circ$. The optimization process involves using SGD with momentum (set to 0.9). A weight decay regularization term of $5 \times 10^{-4}$ is applied. A single cosine decay schedule with a linear warm-up is applied, where the learning rate is initialized to $1 \times 10^{-6}$ and gradually increased to $0.1$ over the course of an epoch, and then a single cosine decay schedule is applied for the remaining training. Only weight matching is used to identify modes that satisfy LMC.

Unlike the spawning method, VGG models are not used in the permutation method due to their inability to achieve LMC. Additionally, Ainsworth et al. [1] open-sourced their source code and pre-trained checkpoints. Therefore, we directly use the pre-trained checkpoints provided by Ainsworth et al. [1].

## B.2 Verification of LLFC Co-Occuring with LMC

In this section, we provide extensive experimental results to verify that LLFC consistently co-occurs with LMC, and conduct a new experiment to demonstrate that the constant $c$ is close to 1 in most cases. Both the spawning method and the permutation method are utilized to obtain linearly connected modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. As shown in Figures 8 to 13 and 15, we include experimental results for MLP on the MNIST dataset (spawning method, activation matching, and weight matching), MLP on the CIFAR-10 dataset (both activation matching and weight matching), VGG-16 on the CIFAR-10 dataset (spawning method), ResNet-20 on the CIFAR-10 dataset (spawning method and weight matching) and ResNet-50 on the Tiny-ImageNet dataset (spawning method). In particular, in Figure 14, we include experimental results of Straight-Trough Estimator (STE) [1]. STE method tries to learn a permutation with STE that could minimize the loss barrier between one mode and the other permuted mode.

To verify the LLFC property on each data point $\boldsymbol{x}_i$ in the test set $\mathcal{D}$, we measure $\mathrm{cosine}_\alpha(\boldsymbol{x}_i) = \cos[f^{(\ell)}(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B; \boldsymbol{x}_i), \alpha f^{(\ell)}(\boldsymbol{\theta}_A; \boldsymbol{x}_i) + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_B; \boldsymbol{x}_i)]$. We compare this to the baseline cosine similarity $\mathrm{cosine}_{A,B}(\boldsymbol{x}_i) = \cos[f^{(\ell)}(\boldsymbol{\theta}_A; \boldsymbol{x}_i), f^{(\ell)}(\boldsymbol{\theta}_B; \boldsymbol{x}_i)]$. In Figures 8 to 15, we

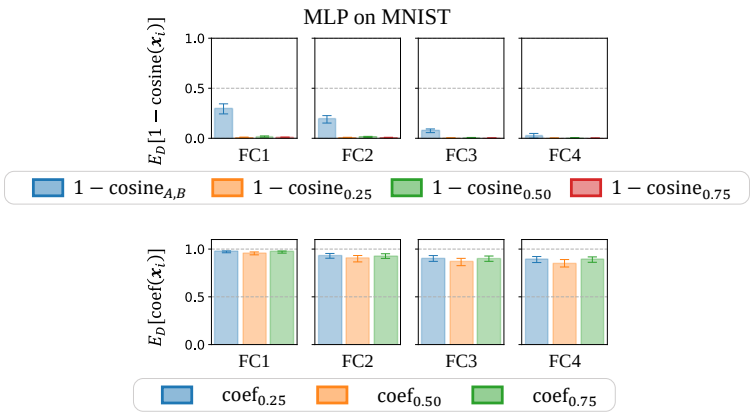


Figure 8: Comparison between $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{\alpha}(\boldsymbol{x}_i)]$ and $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{A,B}(\boldsymbol{x}_i)]$ and demonstration of $\mathbb{E}_{\mathcal{D}}[1-\text{coef}_{\alpha}(\boldsymbol{x}_i)]$. The spawning method is used to obtain two linearly connected modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. Results are presented for different layers of MLP on MNIST dataset, with $\alpha \in \{0.25, 0.5, 0.75\}$. Standard deviations across the dataset are reported by error bars.

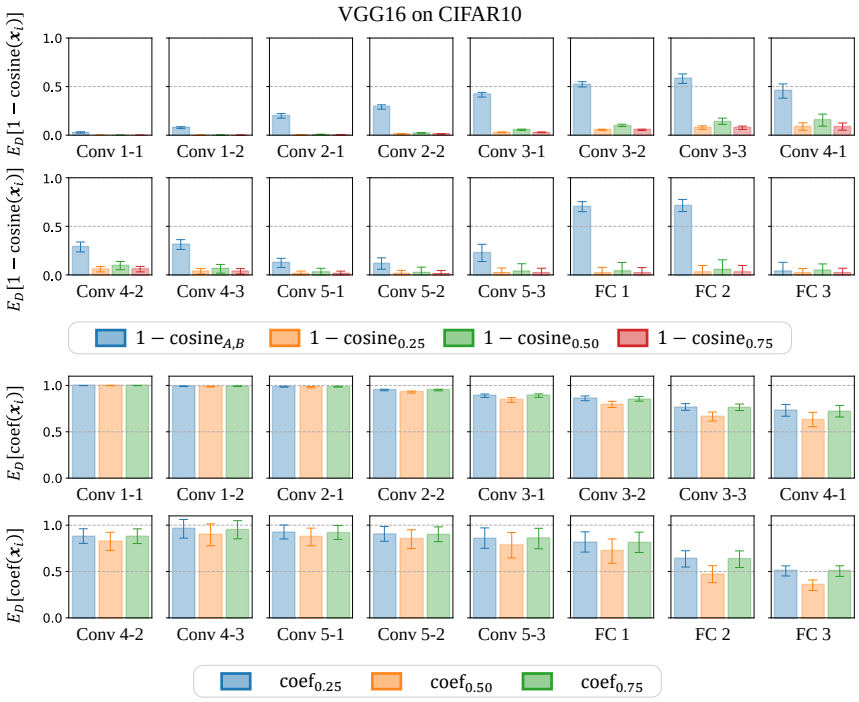


Figure 9: Comparison between $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{\alpha}(\boldsymbol{x}_i)]$ and $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{A,B}(\boldsymbol{x}_i)]$ and demonstration of $\mathbb{E}_{\mathcal{D}}[1-\text{coef}_{\alpha}(\boldsymbol{x}_i)]$. The spawning method is used to obtain two linearly connected modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. Results are presented for different layers of VGG-16 on the CIFAR-10 dataset, with $\alpha \in \{0.25, 0.5, 0.75\}$. Standard deviations across the dataset are reported by error bars.

conclude that the values of $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{\alpha}(\boldsymbol{x}_i)]$ are close to 0 compared with $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{A,B}(\boldsymbol{x}_i)]$, and thus verify our claim.

To show that the constant $c$ is close to 1 in most cases, for each data point $\boldsymbol{x}_i$ in the test set $\mathcal{D}$, we measure $\text{coef}_{\alpha}(\boldsymbol{x}_i) = \|f^{(\ell)}(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B; \boldsymbol{x}_i)\|\text{cosine}_{\alpha}(\boldsymbol{x}_i)/\|\alpha f^{(\ell)}(\boldsymbol{\theta}_A; \boldsymbol{x}_i) + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_B; \boldsymbol{x}_i)]\|$, where $\|f^{(\ell)}(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B; \boldsymbol{x}_i)\|\text{cosine}_{\alpha}(\boldsymbol{x}_i)$ denotes the length of $f^{(\ell)}(\alpha\boldsymbol{\theta}_A + (1-\alpha)\boldsymbol{\theta}_B; \boldsymbol{x}_i)$ projected on $\alpha f^{(\ell)}(\boldsymbol{\theta}_A; \boldsymbol{x}_i) + (1-\alpha) f^{(\ell)}(\boldsymbol{\theta}_B; \boldsymbol{x}_i)]$. In Figures 8 to 15, we conclude that the values of $\mathbb{E}_{\mathcal{D}}[\text{coef}_{\alpha}(\boldsymbol{x}_i)]$ are close to 1 in most cases, and thus verify our claim.

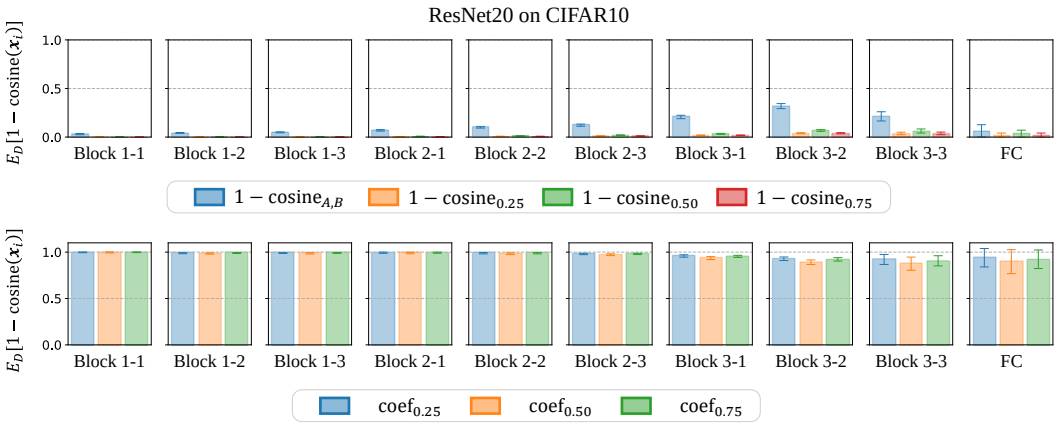


Figure 10: Comparison between $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{\alpha}(\boldsymbol{x}_i)]$ and $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{A,B}(\boldsymbol{x}_i)]$ and demonstration of $\mathbb{E}_{\mathcal{D}}[1-\text{coef}_{\alpha}(\boldsymbol{x}_i)]$. The spawning method is used to obtain two linearly connected modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. Results are presented for different layers of ResNet-20 on the CIFAR-10 dataset, with $\alpha \in \{0.25, 0.5, 0.75\}$. Standard deviations across the dataset are reported by error bars.

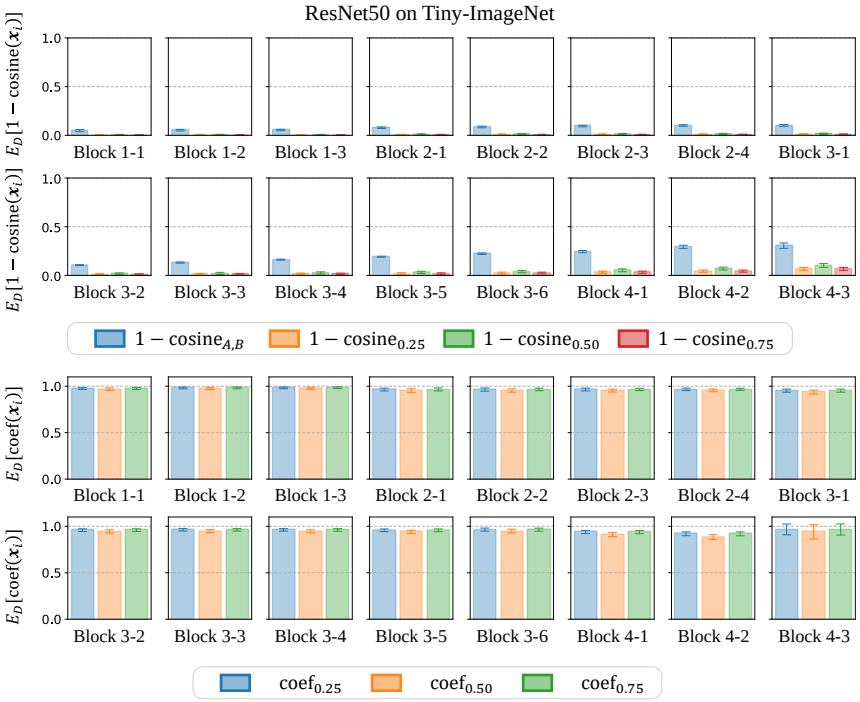


Figure 11: Comparison between $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{\alpha}(\boldsymbol{x}_i)]$ and $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{A,B}(\boldsymbol{x}_i)]$ and demonstration of $\mathbb{E}_{\mathcal{D}}[1-\text{coef}_{\alpha}(\boldsymbol{x}_i)]$. The spawning method is used to obtain two linearly connected modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. Results are presented for different layers of ResNet-50 on the Tiny-ImageNet dataset, with $\alpha \in \{0.25, 0.5, 0.75\}$. Standard deviations across the dataset are reported by error bars.

### B.3 Verification of Commutativity

In this section, we provide more experimental results on various datasets and model architectures to verify the commutativity property for modes that satisfy LLFC. As shown in Figures 16 to 18, we include more experiments results for VGG-16 on the CIFAR-10 dataset (spawning method), MLP on the MNIST dataset (activation matching) and MLP on the CIFAR-10 dataset (both activation matching and weight matching).

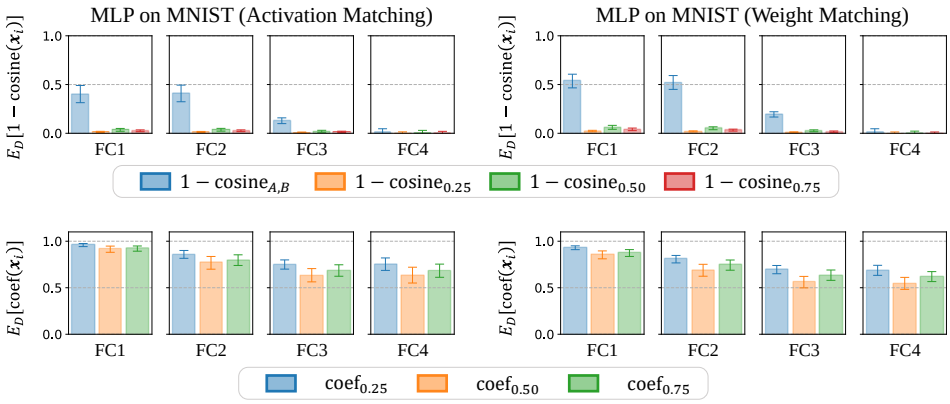


Figure 12: Comparison between $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{\alpha}(\boldsymbol{x}_i)]$ and $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{A,B}(\boldsymbol{x}_i)]$ and demonstration of $\mathbb{E}_{\mathcal{D}}[1-\text{coef}_{\alpha}(\boldsymbol{x}_i)]$. The activation matching and the weight matching are used to obtain two linearly connected modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. Results are presented for different layers of MLP on the MNIST dataset, with $\alpha \in \{0.25, 0.5, 0.75\}$. Standard deviations across the dataset are reported by error bars.

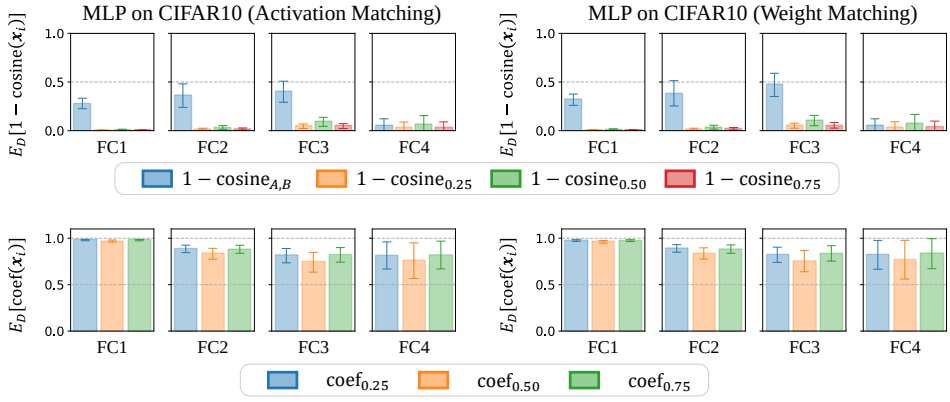


Figure 13: Comparison between $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{\alpha}(\boldsymbol{x}_i)]$ and $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{A,B}(\boldsymbol{x}_i)]$ and demonstration of $\mathbb{E}_{\mathcal{D}}[1-\text{coef}_{\alpha}(\boldsymbol{x}_i)]$. The activation matching and the weight matching are used to obtain two linearly connected modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. Results are presented for different layers of MLP on the CIFAR-10 dataset, with $\alpha \in \{0.25, 0.5, 0.75\}$. Standard deviations across the dataset are reported by error bars.

To verify the commutativity generally holds for modes that satisfy LLFC, for test set $\mathcal{D}$, we compute $\text{Dist}_{com} = \text{dist}\left(\text{vec}(\boldsymbol{W}_A^{(\ell)}\boldsymbol{H}_A^{(\ell-1)} + \boldsymbol{W}_B^{(\ell)}\boldsymbol{H}_B^{(\ell-1)}), \text{vec}(\boldsymbol{W}_A^{(\ell)}\boldsymbol{H}_B^{(\ell-1)} + \boldsymbol{W}_B^{(\ell)}\boldsymbol{H}_A^{(\ell-1)})\right)$[8]. Furthermore, we compare $\text{Dist}_{com}$ with $\text{Dist}_W = \text{dist}\left(\text{vec}(\boldsymbol{W}_A^{(\ell)}), \text{vec}(\boldsymbol{W}_B^{(\ell)})\right)$ and $\text{Dist}_H = \text{dist}\left(\text{vec}(\boldsymbol{H}_A^{(\ell-1)}), \text{vec}(\boldsymbol{H}_B^{(\ell-1)})\right)$, respectively. In Figures 16 to 18, $\text{Dist}_{com}$ is negligible compared with $\text{Dist}_W$ and $\text{Dist}_H$, confirming the commutativity condition.

Furthermore, we add baselines of models that are not linearly connected to further validate the commutativity condition. In Figure 19, we include experimental results for ResNet-20 on CIFAR-10 dataset (both spawning and weight matching method). Specifically, we measure $\text{Dist}_{com,LMC}$ of two linearly connected modes and $\text{Dist}_{com,not\ LMC}$ of two independently trained modes. In Figure 19, the values of $\text{Dist}_{com,LMC}$ are negligible compared with $\text{Dist}_{com,not\ LMC}$, which confirms the commutativity condition.

[8]We also conduct experiments on CNNs. For a Conv layer, the forward propagation will be denoted as $\boldsymbol{W}\boldsymbol{H}$ similar to a linear layer.

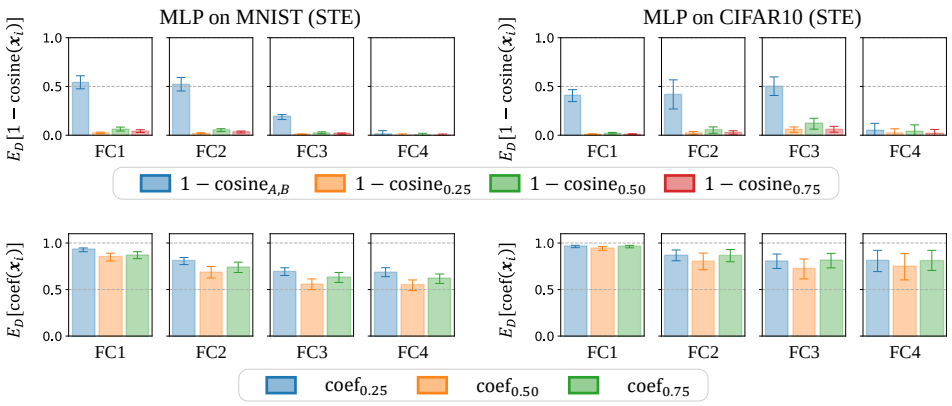


Figure 14: Comparison between $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{\alpha}(\boldsymbol{x}_i)]$ and $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{A,B}(\boldsymbol{x}_i)]$ and demonstration of $\mathbb{E}_{\mathcal{D}}[1-\text{coef}_{\alpha}(\boldsymbol{x}_i)]$. The Straight-Through Estimator (STE) [1] are used to obtain two linearly connected modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. Results are presented for different layers of MLP on both MNIST and CIFAR-10 dataset, with $\alpha \in \{0.25, 0.5, 0.75\}$. Standard deviations across the dataset are reported by error bars.

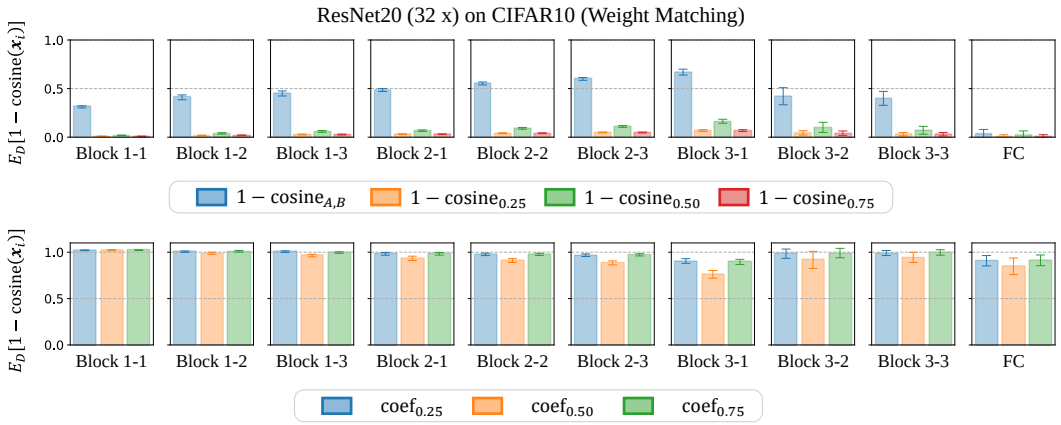


Figure 15: Comparison between $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{\alpha}(\boldsymbol{x}_i)]$ and $\mathbb{E}_{\mathcal{D}}[1-\text{cosine}_{A,B}(\boldsymbol{x}_i)]$ and demonstration of $\mathbb{E}_{\mathcal{D}}[1-\text{coef}_{\alpha}(\boldsymbol{x}_i)]$. The weight matching is used to obtain two linearly connected modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. Results are presented for different layers of ResNet-20 (32x) on the CIFAR-10 dataset, with $\alpha \in \{0.25, 0.5, 0.75\}$. Standard deviations across the dataset are reported by error bars.

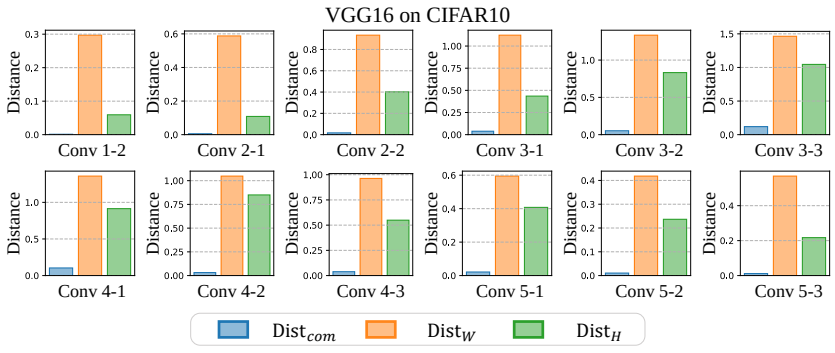


Figure 16: Comparison of $\text{Dist}_{com}$, $\text{Dist}_{W}$, and $\text{Dist}_{H}$. The spawning method is used to obtain two modes that satisfy LLFC, $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. The results are presented for different layers of VGG-16 on the CIFAR-10 dataset.

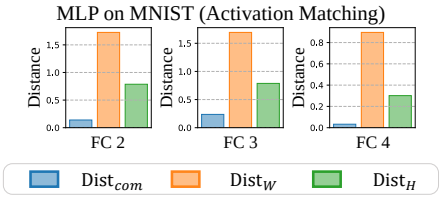


Figure 17: Comparison of Dist$_{com}$, Dist$_W$, and Dist$_H$. The activation matching is used to obtain two modes that satisfy LLFC, $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. The results are presented for different layers of MLP on the MNIST dataset.

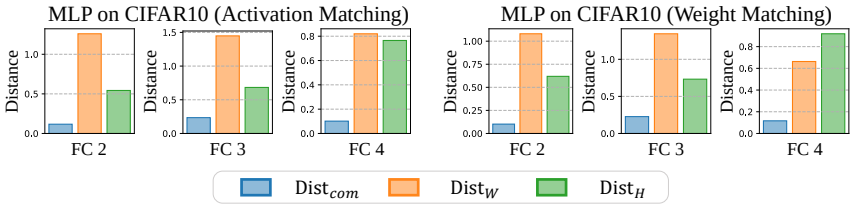


Figure 18: Comparison of Dist$_{com}$, Dist$_W$, and Dist$_H$. Both the activation matching and weight matching are used to obtain two modes that satisfy LLFC, $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. The results are presented for different layers of MLP on the CIFAR10 dataset.

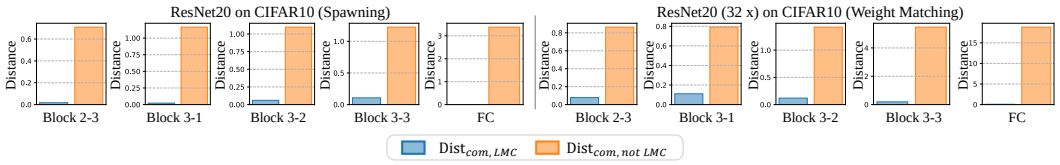


Figure 19: Comparison between Dist$_{com,LMC}$ and Dist$_{com,not\ LMC}$. Both the spawning and permutation methods are used to obtain two linearly connected modes.

| Layer $\ell$ | FC 1 | FC 2 | FC 3 |
|---|---|---|---|
| Err$_{\mathcal{D}(B_{>\ell} \circ A_{\leq \ell})}$ | 2.69 | 2.11 | 1.92 |

Table 1: Error rates (%) of stitched MLP on the MNIST test set. The model stitching is employed in different layers. The spawning method is used to obtain two neural networks that satisfy LLFC, i.e., $A$ and $B$. Error rates (%) of $A$ and $B$ are 1.9 and 1.77, respectively.

Notably, the experiments are not conducted on the first Conv/Linear layer of the model because the commutativity condition is naturally satisfied for the first layer where $\boldsymbol{H}_A^{(0)} = \boldsymbol{H}_B^{(0)} = \boldsymbol{X}$ where $\boldsymbol{X}$ is the input data matrix.

### B.4 Experiments on Model Stitching

Model stitching [19, 3] is commonly employed to analyze neural networks' internal representations. Let $A$ and $B$ represent neural networks with identical architectures. Given a loss function $\mathcal{L}$, model stitching involves finding a stitching layer $s$ (e.g., a linear $1 \times 1$ convolutional layer) such that the minimization of $\mathcal{L}(B_{>\ell} \circ s \circ A_{\leq \ell})$ is achieved. Here, $B_{>\ell}$ denotes the mapping from the activations of the $\ell$-th layer of network $B$ to the final output, $A_{\leq \ell}$ denotes the mapping from the input to the activations of the $\ell$-th layer of network $A$, and $\circ$ represents function composition.

In this section, we explore a stronger form of model stitching. Specifically, given two neural networks $A$ and $B$ that satisfy LLFC, we evaluate the arruacy of $B_{>\ell} \circ A_{\leq \ell}$ over the test set $\mathcal{D}$ without finding a stitching layer, i.e., Err$_{\mathcal{D}(B_{>\ell} \circ A_{\leq \ell})}$. As shown in Tables 1 to 3, we include experimental results for MLP on the MNIST dataset, VGG-16 on CIFAR-10 the dataset and ResNet-20 on the CIFAR-10 dataset. Only the spawning method is utilized to find modes that satisfy LLFC. The results depicted

| Layer $\ell$ | Conv 1-1 | Conv 1-2 | Conv 2-1 | Conv 2-2 | Conv 3-1 |
|---|---|---|---|---|---|
| $\mathrm{Err}_{\mathcal{D}(B_{>\ell} \circ A_{\leq \ell})}$ | 7.2 | 8.43 | 8.39 | 9.91 | 11.84 |
| Layer $\ell$ | Conv 3-2 | Conv 3-3 | Conv 4-1 | Conv 4-2 | Conv 4-3 |
| $\mathrm{Err}_{\mathcal{D}(B_{>\ell} \circ A_{\leq \ell})}$ | 9.55 | 8.22 | 7.61 | 6.99 | 7.05 |
| Layer $\ell$ | Conv 5-1 | Conv 5-2 | Conv 5-3 | FC 1 | FC 2 |
| $\mathrm{Err}_{\mathcal{D}(B_{>\ell} \circ A_{\leq \ell})}$ | 6.91 | 6.88 | 6.88 | 7.07 | 6.92 |

Table 2: Error rates (%) of stitched VGG-16 on the CIFAR-10 test set. The model stitching is employed in different layers. The spawning method is used to obtain two neural networks that satisfy LLFC, i.e., $A$ and $B$. Error rates (%) of $A$ and $B$ are $6.87$ and $7.1$, respectively.

| Layer $\ell$ | Block 1-1 | Block 1-2 | Block 1-3 | Block 2-1 | Block 2-2 | Block 2-3 | Block 3-1 | Block 3-2 | Block 3-3 |
|---|---|---|---|---|---|---|---|---|---|
| $\mathrm{Err}_{\mathcal{D}(B_{>\ell} \circ A_{\leq \ell})}$ | 10.88 | 10.57 | 13.35 | 10.64 | 10.74 | 10.55 | 12.27 | 11.8 | 8.99 |

Table 3: Error rates (%) of stitched ResNet-20 on the CIFAR-10 test set. The model stitching is employed in different layers. The spawning method is used to obtain two neural networks that satisfy LLFC, i.e., $A$ and $B$. Error rates (%) of $A$ and $B$ are $8.69$ and $8.58$, respectively.

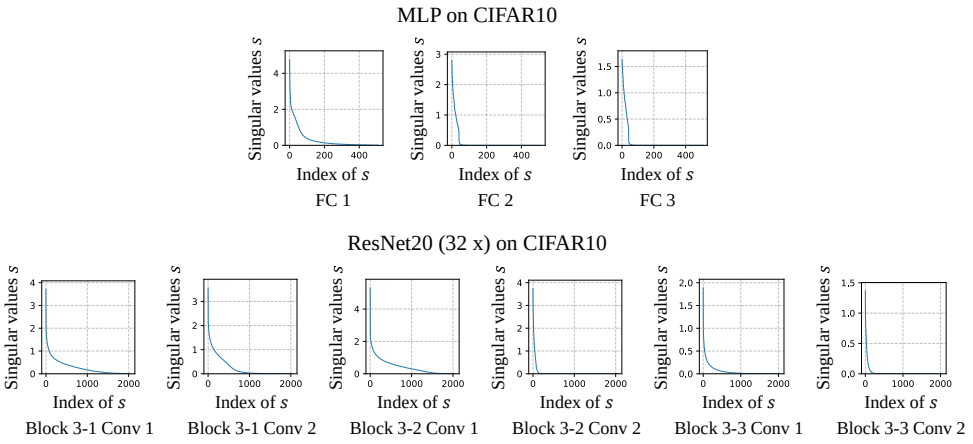


Figure 20: Singular values of weight matrix $\boldsymbol{W}^{(\ell)}$ of $\ell$-th layer of $\boldsymbol{\theta}$ in a descending order. Here, $\boldsymbol{\theta}$ can be used to achieve LMC with weight matching.The results are presented for different layers of various model architectures and datasets.

in Tables 1 to 3 demonstrate that the error rates of the stitched model on the test set closely resemble the error rates of the original models $A$ and $B$, regardless of the dataset or model architecture. This observation suggests that models that satisfy LLFC encode similar information, which can be decoded across different models. Subsequently, the experiments of model stitching provides new insights towards the commutativity property, i.e, $\forall \ell \in [L], \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} \approx \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)}$.

### B.5 Discussion on Git Re-basin [1]

In this section, we investigate the ability of permutation methods to achieve LMC. While we have interpreted the activation matching and weight matching methods proposed by Ainsworth et al. [1] as guaranteeing the commutativity property, we have yet to address why permutation methods can ensure the satisfaction of this property. Thus, in order to delve into the capability of permutation methods, we must address the question of why these methods are capable of ensuring the satisfaction of the commutativity property.

Low-rank model weights and activations contribute to ensure the commutativity property. We now consider a stronger form of the commutativity property, where given two modes $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ and a

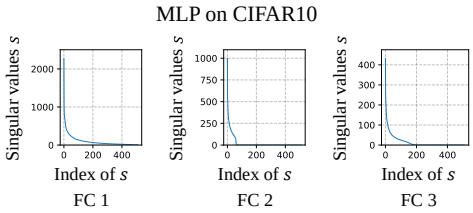


Figure 21: Singular values of post-activations $\boldsymbol{H}^{(\ell)}$ of $\ell$-th layer of $\boldsymbol{\theta}$ over the whole test set $\mathcal{D}$ in a descending order. Here, $\boldsymbol{\theta}$ can be used to achieve LMC with activation matching.The results are presented for different layers of MLP on the CIFAR-10 dataset.

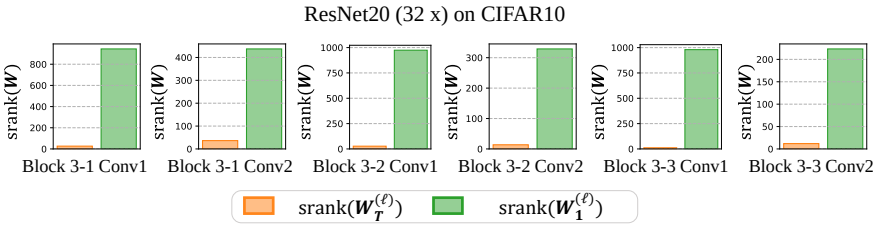


Figure 22: Comparion between the stable rank $\mathrm{srank}(\boldsymbol{W}_T^{(\ell)})$ and $\mathrm{srank}(\boldsymbol{W}_1^{(\ell)})$. Here, $\boldsymbol{W}_T^{(\ell)}$ denotes the weight matrix of the $\ell$-th layer of the model $\boldsymbol{\theta}_T$ in the terminal phase of training. Similarly, $\boldsymbol{W}_1^{(\ell)}$ denotes the weight matrix of the $\ell$-th layer of the model $\boldsymbol{\theta}_1$ in the early stage of training (1 epoch indeed). Also, the stable rank can be calculated as $\mathrm{srank}(\boldsymbol{W}) = \frac{\|\boldsymbol{W}\|_F^2}{\|\boldsymbol{W}\|_2^2}$. The results are presented for different layers of ResNet-20 (32x) on the CIFAR-10 dataset.

dataset $\mathcal{D}$, we have:

$$\forall \ell \in [L], \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} = \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} \wedge \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} = \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_A^{(\ell-1)}.$$

Thus, to satisfy the commutativity property for a given layer $\ell$, we can employ the permutation method to find a permutation matrix $\boldsymbol{P}^{(\ell-1)}$ such that:

$$\boldsymbol{W}_A^{(\ell)} \left( \boldsymbol{H}_A^{(\ell-1)} - \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} \right) = 0 \wedge \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} \left( \boldsymbol{H}_B^{(\ell-1)} - {\boldsymbol{P}^{(\ell-1)}}^\top \boldsymbol{H}_A^{(\ell-1)} \right) = 0.$$

In a homogeneous linear system $\boldsymbol{W}\boldsymbol{X} = 0$, a low-rank matrix $\boldsymbol{W}$ allows for a larger solution space for $\boldsymbol{X}$. Therefore, if the ranks of $\boldsymbol{W}_A^{(\ell)}$ and $\boldsymbol{W}_B^{(\ell)}$ are low, it becomes easier to find a permutation matrix $\boldsymbol{P}^{(\ell-1)}$ that satisfies the commutativity property. Similarly, if we consider another form of commutativity property:

$$\forall \ell \in [L], \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} = \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_A^{(\ell-1)} \wedge \boldsymbol{W}_B^{(\ell)} \boldsymbol{H}_B^{(\ell-1)} = \boldsymbol{W}_A^{(\ell)} \boldsymbol{H}_B^{(\ell-1)}.$$

Then, to ensure the commutativity property, we need to find $\boldsymbol{P}^{(\ell-1)}$ and $\boldsymbol{P}^{(\ell)}$ such that

$$\left( \boldsymbol{W}_A^{(\ell)} - \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} {\boldsymbol{P}^{(\ell-1)}}^\top \right) \boldsymbol{H}_A^{(\ell-1)} = 0 \wedge \left( \boldsymbol{P}^{(\ell)} \boldsymbol{W}_B^{(\ell)} {\boldsymbol{P}^{(\ell-1)}}^\top - \boldsymbol{W}_A^{(\ell)} \right) \boldsymbol{P}^{(\ell-1)} \boldsymbol{H}_B^{(\ell-1)} = 0.$$

Then, if the ranks of $\boldsymbol{H}_A^{(\ell-1)}$ and $\boldsymbol{H}_B^{(\ell-1)}$ are low, it is easier to find the permutation matrices to satisfy the condition. In real scenarios, both model weights (see Figure 20) and activations (see Figure 21) are approximately low-rank, which helps the permutation methods satisfy the commutativity property.

Additionally, Ainsworth et al. [1] mentioned two instances where permutation methods can fail: models with insufficient width and models in the early stages of training. In both cases, the model weights often fail to satisfy the low-rank model weight condition. In the first scenario, when the model lacks sufficient width, meaning that the dimension of the weight matrix approaches the rank of the weight matrix, the low-rank condition may not be met. For example, compared the singular values of ResNet-20 (32x) (see Figure 20) with singular values of ResNet-20 (1x) (see **??**), it is evident that in the wider architecture, the proportion of salient singular values is smaller. In the second scenario, during the initial stages of training, the weight matrices resemble random matrices and may not

exhibit low-rank characteristics. For example, as shown in Figure 22, the stable ranks of weight matrices of the model after convergence are significantly smaller than those of the model in the early stage of training. Consequently, permutation methods may struggle to find suitable permutations that fulfill the commutativity property, resulting in the inability to obtain modes that satisfy LMC.