MLI Formula: A Nearly Scale-Invariant Solution with Noise Perturbation

Bowen Tao [*] [1] [2] **Xin-Chun Li** [*] [1] [2] **De-Chuan Zhan** [1] [2]

Abstract

Monotonic Linear Interpolation (MLI) refers to the peculiar phenomenon that the error between the initial and converged model monotonically decreases along the linear interpolation, i.e., $(1-\alpha)\boldsymbol{\theta}_0 + \alpha\boldsymbol{\theta}_F$. Previous works focus on paired initial and converged points, relating MLI to the smoothness of the optimization trajectory. In this paper, we find a shocking fact that the error curves still exhibit a monotonic decrease when $\boldsymbol{\theta}_0$ is replaced with noise or even zero values, implying that the decreasing curve may be primarily related to the property of the converged model rather than the optimization trajectory. We further explore the relationship between $\alpha\boldsymbol{\theta}_F$ and $\boldsymbol{\theta}_F$ and propose scale invariance properties in various cases, including Generalized Scale Invariance (GSI), Rectified Scale Invariance (RSI), and Normalized Scale Invariance (NSI). From an inverse perspective, *the MLI formula is essentially an equation that adds varying levels of noise (i.e., $(1-\alpha)\boldsymbol{\epsilon}$) to a nearly scale-invariant network (i.e., $\alpha\boldsymbol{\theta}_F$), resulting in a monotonically increasing error as the noise level rises*. MLI is a special case where $\boldsymbol{\epsilon}$ is equal to $\boldsymbol{\theta}_0$.

1. Introduction

Deep neural networks (DNNs) are generally considered non-convex models known for their challenging optimization. The stochastic gradient descent (SGD) optimizer is widely utilized in training complex networks (Tao et al., 2023; Li et al., 2022a;b). Typically, the loss values or test errors exhibit oscillations during training, indicating a rugged loss landscape and a winding optimization trajectory. However, Goodfellow & Vinyals (2015) observes a phenomenon called Monotonic Linear Interpolation (MLI), suggesting that DNNs may not be as complex as previously thought. Specifically, Goodfellow & Vinyals (2015) introduces the linear interpolation between the parameters $\boldsymbol{\theta}_0$ at initialization and the parameters $\boldsymbol{\theta}_F$ at the local minima found after training with SGD. The interpolation, denoted as $(1-\alpha)\boldsymbol{\theta}_0 + \alpha\boldsymbol{\theta}_F$, demonstrates a monotonic decrease in loss or error as α ranges from 0 to 1. Goodfellow & Vinyals (2015) attributes MLI to the relative ease of these tasks from an optimization perspective. However, Frankle (2020) observes the MLI for DNNs on more complicated tasks, despite the presence of long plateaus in the loss and error curves. Wang et al. (2023) concludes that the plateau is caused by the bias term and network depth, rather than the difficulty of optimization. These studies *excessively focus on investigating the correlation between MLI and the training trajectory, ignoring the formula of MLI itself.*

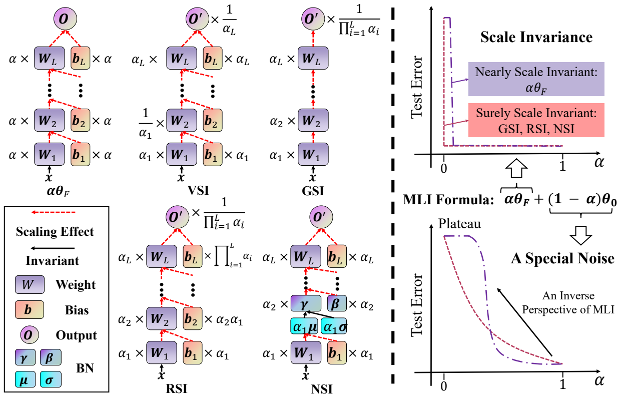
O
$\alpha \times$ W_L b_L $\times \alpha$
$\alpha \times$ W_2 b_2 $\times \alpha$
$\alpha \times$ W_1 b_1 $\times \alpha$
x
$\alpha\theta_F$
$O' \times \frac{1}{\alpha_L}$
$\alpha_L \times$ W_L b_L $\times \alpha_L$
$\frac{1}{\alpha_1} \times$ W_2 b_2
$\alpha_1 \times$ W_1 b_1 $\times \alpha_1$
VSI
$O' \times \frac{1}{\prod_{i=1}^{L} \alpha_i}$
$\alpha_L \times$ W_L
$\alpha_2 \times$ W_2
$\alpha_1 \times$ W_1
GSI
Scaling Effect
Invariant
W Weight
b Bias
O Output
γ β μ σ BN
$O' \times \frac{1}{\prod_{i=1}^{L} \alpha_i}$
$\alpha_L \times$ W_L b_L $\times \prod_{i=1}^{L} \alpha_i$
$\alpha_2 \times$ W_2 b_2 $\times \alpha_2\alpha_1$
$\alpha_1 \times$ W_1 b_1 $\times \alpha_1$
RSI
O'
$\alpha_L \times$ W_L b_L $\times \alpha_L$
$\alpha_2 \times$ γ β $\times \alpha_2$
$\alpha_1\mu$ $\alpha_1\sigma$
$\alpha_1 \times$ W_1 b_1 $\times \alpha_1$
NSI
Scale Invariance
Test Error
Nearly Scale Invariant: $\alpha\theta_F$
Surely Scale Invariant: GSI, RSI, NSI
0
1
α
MLI Formula: $\alpha\theta_F + (1-\alpha)\theta_0$
Plateau
A Special Noise
An Inverse Perspective of MLI
Test Error
0
1
α

Figure 1. The illustration of the MLI formula. The left shows several types of scale invariance property for various DNN cases. The right explains the MLI formula from a novel perspective, i.e., adding diverse levels of noise perturbation to a nearly scale-invariant network. The MLI phenomenon emerges from the combined influence of scale invariance and noise robustness.

In this paper, we extend the experiments in Goodfellow & Vinyals (2015) to modern settings, including VGG-style networks (Simonyan & Zisserman, 2015), ResNets (He et al., 2016) and Batch Normalization (BN) (Ioffe & Szegedy, 2015). We discover that substituting the original initialization $\boldsymbol{\theta}_0$ with an unrelated random initialization $\boldsymbol{\theta}'_0$ still leads to monotonic decreasing loss curves. Surprisingly, when we replace $\boldsymbol{\theta}_0$ with zero values, the test error curve along

[*] Equal contribution [1] School of Artificial Intelligence, Nanjing University, China [2] National Key Laboratory for Novel Software Technology, Nanjing University, China. Correspondence to: De-Chuan Zhan <zhandc@nju.edu.cn>.

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

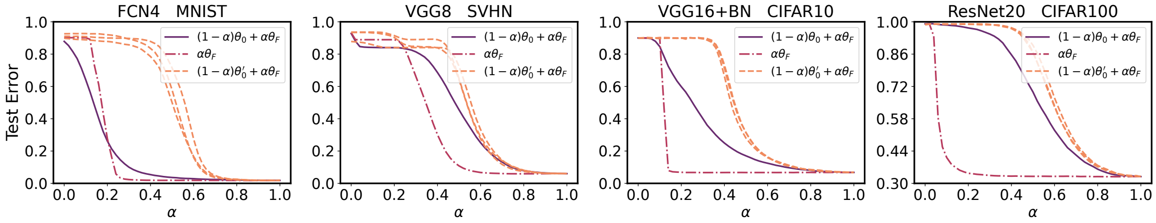

Figure 2. Test error over the linear interpolation between initialization and final parameters. The left endpoint represents various initialized parameters and the right endpoint corresponds to the same converged parameters for each curve in a sub-figure. The monotonic decreasing property is observed not only for original initialization but also for an unrelated random initialization and even for zero initialization.

$\alpha\boldsymbol{\theta}_F$ also exhibits the monotonic decreasing phenomenon, as depicted in Figure 2. Notably, the test error of $\alpha\boldsymbol{\theta}_F$ decreases rapidly and converges to the performance of final solution $\boldsymbol{\theta}_F$, which inspires us that *this observation may be a manifestation of scale invariance*. Consequently, we propose various types of scale invariance that take the bias parameters and batch normalization into account, including Generalized Scale Invariance (GSI), Rectified Scale Invariance (RSI), and Normalized Scale Invariance (NSI), as illustrated in Figure 1.

Based on these types of scale invariance, $\alpha\boldsymbol{\theta}$ approximates the behavior of an invariantly scaled network, while with varying levels of noise robustness across different α. Hence, we interpret the MLI formula as introducing noise to a nearly scale-invariant network. Subsequently, we propose the following hypothesis: *for an arbitrary model $\boldsymbol{\theta}_F$ and a specific noise $\boldsymbol{\epsilon}$, $\alpha\boldsymbol{\theta}_F$ exhibits similar performance to $\boldsymbol{\theta}_F$, and as α decreases, the error of $\alpha\boldsymbol{\theta}_F + (1-\alpha)\boldsymbol{\epsilon}$ monotonically increases*. If the hypothesis holds for $\boldsymbol{\epsilon} = \boldsymbol{\theta}_0$, it leads to the MLI phenomenon, where the loss monotonically increases as α decreases from 1 to 0. In contrast to previous works, our proposed explanation of MLI decouples the paired initialization and converged models, solely focusing on the properties of $\boldsymbol{\theta}_F$. To validate our interpretation, we provide empirical analyses on multiple datasets and perform experiments to explain the phenomenon of violating the MLI property under specific mechanisms.

In summary, our contributions are as follows: (1) We observe that substituting the initialization with *unrelated initialization or zero values* in the MLI formula still leads to the monotonic decreasing phenomenon. (2) We propose *several types of scale invariance* for cases of network architectures, including GSI, RSI, and NSI. (3) We provide an explanation for the *MLI formula as a nearly scale-invariant solution that adds varying levels of noise*. (4) We offer an *alternative perspective* to comprehend the MLI phenomenon, revealing that $\boldsymbol{\theta}_F$ is nearly scale-invariant and perturbing it with $(1-\alpha)\boldsymbol{\theta}_0$ results in a monotonic increase in the loss curve as α decreases from 1 to 0. (5) We provide explanations for the violation of MLI under specific settings, addressing cases that are *previously reported without comprehensive explanations*.

2. Related Works

The formula of MLI is essentially a parameter interpolation that reflects the loss landscape properties of DNNs. Hence, our work is closely related to parameter interpolation and the loss landscape of DNNs.

2.1. Parameter Interpolation of DNNs

Goodfellow & Vinyals (2015) proposes the MLI phenomenon and points out that the MLI persists on various network architectures, activation functions and loss functions. Subsequently, Frankle (2020) studies MLI on contemporary networks and discovers the plateau phenomenon, wherein the loss and error curve remains high until close to the converged model. Notably, this work reveals that substituting the initialization with intermediate checkpoints can disrupt the MLI property. Wang et al. (2023) attributes the plateau to the bias term. Vlaar & Frankle (2022) investigates the effects of initialization, data, optimizer and architectures on MLI, while Lucas et al. (2021) identifies specific cases that violate the MLI property. *Different from these studies, our work originates from the MLI formula and presents a novel comprehension of MLI. We also provide an explanation for the scenarios that violate the MLI property*.

Similar to MLI, studies on Linear Mode Connectivity (LMC) focus on the interpolation between two final solutions (modes). Prior works reveal insights into loss barrier (Goodfellow & Vinyals, 2015), simple non-linear curves with low loss (Draxler et al., 2018; Garipov et al., 2018), aligned interpolation with permutation invariance (Entezari et al., 2022; Ainsworth et al., 2023), and the improvement of model soups based on pre-trained models (Neyshabur et al., 2020; Wortsman et al., 2022), which contribute to a deeper understanding of DNNs' properties. *We focus on comprehending MLI and anticipate the proposed methods can be effectively applied to LMC research in the future*.

2.2. Loss Landscape of DNNs

Visualizing the loss landscape of DNNs can provide insights into their properties, such as generalization (Im et al., 2016), optimization trajectory (Lorch, 2016), architectural choices (Li et al., 2018; Fort & Jastrzebski, 2019), and the sharpness/flatness of different minima (Keskar et al., 2017). Due to the high dimension of parameter space, conventional visualization methods can only portray the loss landscape in one or two-dimensional subspace. *We also employ 1d and 2d visualization method to present our findings.*

Among the properties of DNNs, the sharpness/flatness is highly relevant to our work. The concept of flat minima is first proposed by Hochreiter & Schmidhuber (1997). Keskar et al. (2017) finds that using large or small batch size can lead to minima with varying degrees of flatness. Neyshabur et al. (2017); Jiang et al. (2020) establish a positive correlation between flatness and generalization of DNNs, while Dinh et al. (2017); Andriushchenko et al. (2023) argue that sharp minima can also exhibit good generalization. Furthermore, several optimization algorithms are developed to enhance generalization ability by seeking flatter minima (Pittorino et al., 2021; Chaudhari et al., 2017; Foret et al., 2021; Zhao et al., 2022; Kwon et al., 2021).

In all of these studies, sharpness/flatness can be interpreted as a measure of noise robustness, denoted as $\boldsymbol{\theta}_F + \boldsymbol{\epsilon}$. The most common trend is that *as the magnitude of $\boldsymbol{\epsilon}$ increases, the loss and error also increases* (Keskar et al., 2017; Pittorino et al., 2021; Foret et al., 2021). *We connect the MLI formula with this equation, allowing us to understand MLI from an alternative perspective.*

3. Preliminaries

In this section, we introduce the concept of monotonic linear interpolation (MLI) and subsequently present our findings and inspirations.

3.1. The Monotonic Linear Interpolation Property

The MLI property signifies that the linear interpolation path between initialized parameters and converged parameters monotonically decreases in loss and error. Thereby, we assert that a network has the MLI property if for any α_1, $\alpha_2 \in [0, 1]$ and $\alpha_1 < \alpha_2$:

$$\mathcal{J}(\boldsymbol{\theta}_{\alpha_1}) \geq \mathcal{J}(\boldsymbol{\theta}_{\alpha_2}), \text{where } \boldsymbol{\theta}_{\alpha_i} = (1-\alpha_i)\boldsymbol{\theta}_0 + \alpha_i\boldsymbol{\theta}_F. \quad (1)$$

Here, $\boldsymbol{\theta}_0$, $\boldsymbol{\theta}_F$ denote the parameters at initialization and convergence respectively. $\mathcal{J}(\boldsymbol{\theta})$ is an evaluation function that reflects the performance (*e.g.*, loss and error) of the model with parameters $\boldsymbol{\theta}$.

Prior to the discovery of MLI, the loss landscape of DNNs is perceived as rugged and the optimization trajectory is considered to be convoluted. However, the presence of MLI in both simple and complex tasks suggests that the landscape may be smoother than previously thought. Previous studies *overly emphasize the association between MLI and task hardness or the optimization trajectory*, whereas our work *begins with elucidating the MLI formula and subsequently provides a novel understanding of MLI by integrating scale invariance and noise perturbation*.

3.2. Findings and Inspirations

We study four image classification settings including a 4-layer fully-connected network (FCN4) for MNIST (LeCun & Cortes, 2010), VGG8 without Batch Normalization (Ioffe & Szegedy, 2015) for SVHN (Netzer et al., 2011), VGG16 with Batch Normalization (Ioffe & Szegedy, 2015) for CIFAR10 (Krizhevsky, 2009), and ResNet20 for CIFAR100 (Krizhevsky, 2009). To further understand the MLI property, we devise two variants of interpolation path in Equation (1). The first is defined as:

$$\boldsymbol{\theta}_{\alpha_i} = (1-\alpha_i)\boldsymbol{\theta}_0' + \alpha_i\boldsymbol{\theta}_F, \quad (2)$$

where $\boldsymbol{\theta}_0'$ represents an unrelated random initialization. Figure 2 reveals that the linear interpolation between $\boldsymbol{\theta}_0'$ and $\boldsymbol{\theta}_F$ also displays a monotonic decrease in error, which has also been shown in Lucas et al. (2021). Notably, we sample multiple groups of $\boldsymbol{\theta}_0'$, and all of them conform to the monotonic decreasing result. Moreover, the test error curves of networks with BN along the interpolation path exhibit significantly similar shapes (i.e., the last two subplots). Furthermore, we provide a demonstration using "scikit-learn" [1] to showcase the monotonic decreasing property. The code is listed in Appendix C.

Next, we simplify the MLI formula by omitting the term $(1-\alpha)\boldsymbol{\theta}_F$ and introduce the second variant of interpolation:

$$\boldsymbol{\theta}_{\alpha_i} = \alpha_i\boldsymbol{\theta}_F. \quad (3)$$

The simplified equation also shows monotonic decreasing results, as shown in Figure 2. Notably, the networks with BN exhibit a test error similar to that of networks $\boldsymbol{\theta}_F$ when α is relatively small (i.e., the last two subplots in Figure 2).

Based on the results presented in Figure 2, we summarize our findings as follows: (1) Changing the initialization parameter in MLI formula *does not alter the monotonic decreasing tendency of error curves*, particularly for networks with BN. (2) The network $\alpha\boldsymbol{\theta}_F$ performs similarly to $\boldsymbol{\theta}_F$ even when α is relatively small, especially for networks with BN, suggesting *a nearly scale-invariant property of the network*. (3) In all three interpolation ways, networks with BN show a shorter plateau in the test error curve compared to networks without BN, indicating *the crucial role*

[1] https://scikit-learn.org/stable/index.html

of BN in preserving the function of the network. We further validate these findings on pre-trained models, including ResNet50 (He et al., 2016) on CUB (Wah et al., 2011) and RoBERTa (Liu et al., 2019) on AG News (Zhang et al., 2015), which are provided in Appendix A.1.

Inspired by the aforementioned findings, we propose a hypothesis that the network with parameters $\alpha\boldsymbol{\theta}_F$ can achieve a similar test error to networks with parameters $\boldsymbol{\theta}_F$ when α reaches a certain threshold, exhibiting a scale invariance property of DNNs. Additionally, we consider $(1-\alpha)\boldsymbol{\theta}_0$ in Equation (1) and $(1-\alpha)\boldsymbol{\theta}'_0$ in Equation (2) as forms of noise perturbation on the scaled converged parameters $\alpha\boldsymbol{\theta}_F$ due to the common phenomenon of monotonic decreasing. Therefore, we propose a novel conjecture that *the MLI formula reflects a property of DNNs, which is jointly determined by scale invariance and noise robustness.*

4. Explanations of MLI Formula

In this section, we first generalize and decompose the MLI formula into two parts as follows:

$$\underbrace{\alpha\boldsymbol{\theta}_F}_{\text{Nearly Scale-Invariant}} + \underbrace{(1-\alpha)\boldsymbol{\epsilon}}_{\text{Varying Levels of Noise}} . \tag{4}$$

Here we substitute the initialization $\boldsymbol{\theta}_0$ with a noise $\boldsymbol{\epsilon}$. The former part $\alpha\boldsymbol{\theta}_F$ is related to the scale invariance property, while the latter part is associated with noise robustness.

4.1. Scale Invariance Property

We explore the scale invariance property across diverse network architectures and introduce several variants of this property for specific networks.

4.1.1. Vanilla Scale Invariance

Consider a neural network consisting of L layers. For the l-th layer, we denote $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$ as the weight matrix and $b_l \in \mathbb{R}^{d_l}$ as the bias vector, where d_l is the number of neurons in the l-th layer for $l \in \{1, 2, \dots, L\}$. Given an input $x \in \mathbb{R}^{d_0}$, the output of the l-th layer is:

$$h_l(x) := \sigma\left(W_l h_{l-1}(x) + b_l\right), \tag{5}$$

where the activation function $\sigma(\cdot)$ can be identity or ReLU function. Notably, due to the property $\sigma(\alpha x) = \alpha\sigma(x)$ for any $\alpha > 0$, the network is invariant to the scaling of parameters in each layer by a positive factor (Pittorino et al., 2022). For the l-th layer, we scale its weight and bias by α_l. To eliminate the impact of α_l, we scale the weight parameters of its subsequent layer by multiplying $\frac{1}{\alpha_l}$. Hence, we define the Vanilla Scale Invariance (VSI) property as:

$$h'_l = \sigma\left(\alpha_l W_l h_{l-1} + \alpha_l b_l\right) = \alpha_l h_l, \tag{6}$$

$$h'_{l+1} = \sigma\left(\frac{1}{\alpha_l} W_{l+1} h'_l + b_{l+1}\right) = h_{l+1}. \tag{7}$$

We use h_l to denote $h_l(x)$ for simplicity. α_l is the scaling factor of the l-th layer and h'_l denotes the output of the scaled l-th layer. Since $h'_{l+1} = h_{l+1}$, the output of the $(l+1)$-th layer remains unchanged. Therefore, the function implemented by the network and the associated loss remains unaffected. The VSI property is shown in Figure 1.

4.1.2. Generalized Scale Invariance

For DNNs **without bias parameters and BN layers**, we extend VSI to Generalized Scale Invariance (GSI) property for multiple layers. We denote r as the number of layers involved in the multiple-layer scale transformation. From the l-th layer to the $(l+r)$-th layer, we scale the weights by corresponding positive scaling factors $\{\alpha_i\}_{i=l}^{l+r}$ and accumulate the scaling effect value $\prod_{i=l}^{l+r} \alpha_i$ at the $(l+r)$-th layer. We introduce a scaling factor $\frac{1}{\prod_{i=l}^{l+r} \alpha_i}$ in the subsequent layer to eliminate the scaling effect. GSI is defined as:

$$h'_{l+i} = \left(\prod_{j=0}^{i} \alpha_{l+j}\right) h_{l+i}, \quad i \in [0, r], \tag{8}$$

$$h'_{l+r+1} = \sigma\left(\frac{1}{\prod_{i=l}^{l+r} \alpha_i} W_{l+r+1} \left(\prod_{i=l}^{l+r} \alpha_i\right) h_{l+r}\right) = h_{l+r+1}, \tag{9}$$

where the outputs from the l-th layer to the $(l+r)$-th layer are scaled, while the output of the $(l+r+1)$-th layer is consistent with the original network, denoted as $h'_{l+r+1} = h_{l+r+1}$. GSI extends beyond VSI to multiple layers and applies exclusively to networks without bias and BN layers.

We can also apply GSI to all layers, where $l = 1$ and $l + r = L$. According to GSI, the output logits should be re-scaled by $\frac{1}{\prod_{i=1}^{L} \alpha_i}$, which is illustrated as the GSI figure in Figure 1. Notably, for classification tasks, we can maintain the network's function without re-scaling its weights and output. The output of network scaled by $\alpha_1, \dots, \alpha_L$ is:

$$h'_L = \left(\prod_{i=1}^{L} \alpha_i\right) \sigma(W_L h_{L-1}) = \left(\prod_{i=1}^{L} \alpha_i\right) h_L. \tag{10}$$

We find that even when we do not re-scale the weights to maintain consistent outputs, the error of the network remains unchanged, as $\prod_{i=1}^{L} \alpha_i$ does not affect the class index of the highest probability among the output categories. However, a significant difference rises in the value of loss function $\mathcal{L}$ (*e.g.*, cross-entropy), which can be mitigated by adjusting the temperature t:

$$\mathcal{L} = -\log \frac{\exp\left(\left(\prod_{i=1}^{L} \alpha_i\right) h_L^c / t\right)}{\sum_{j=1}^{C} \exp\left(\left(\prod_{i=1}^{L} \alpha_i\right) h_L^j / t\right)}, \tag{11}$$

where c, h_L^j denote the correct label and the output of the j-th class respectively. C is the number of classes. The value of the loss function aligns with that of the network prior to applying the scale transformation when t equals $\prod_{i=1}^{L} \alpha_i$.

4.1.3. Rectified Scale Invariance

For networks **with bias parameters but without BN layers**, the GSI property is disrupted due to the bias terms. When we scale the network from the l-th layer, we can obtain $h_l' = \alpha_l h_l$. However, for the subsequent layer,

$$\begin{aligned} h_{l+1}' &= \sigma\left(\alpha_{l+1} W_{l+1} h_l' + \alpha_{l+1} b_{l+1}\right) \\ &= \sigma\left(\alpha_{l+1} \alpha_l W_{l+1} h_l + \alpha_{l+1} b_{l+1}\right) \neq \alpha_{l+1} \alpha_l h_{l+1}. \end{aligned} \tag{12}$$

It is evident that *the weight and bias parameters are scaled by different scaling factors, making it impossible to extract these factors simultaneously from the activation function*. Considering the influence of the bias term in Equation (12), we modify the factor of the bias in the $(l+1)$-th layer to $\alpha_{l+1}\alpha_l$ instead of α_l:

$$h_{l+1}' = \sigma(\alpha_{l+1} \alpha_l W_{l+1} h_l + \alpha_{l+1} \alpha_l b_{l+1}) = \alpha_{l+1} \alpha_l h_{l+1}, \tag{13}$$

where the bias term b_{l+1} is scaled by the product of scaling factors from the l-th layer to the current $(l+1)$-th layer. Building upon Equation (13), we propose a variant of GSI called Rectified Scale Invariance (RSI) property:

$$\begin{aligned} h_{l+i}' &= \sigma\left(\alpha_{l+i} W_{l+i} h_{l+i-1}' + \left(\prod_{j=0}^{i} \alpha_{l+j}\right) b_{l+i}\right) \\ &= \left(\prod_{j=0}^{i} \alpha_{l+j}\right) h_{l+i}, \quad i \in [0, r], \end{aligned} \tag{14}$$

$$\begin{aligned} h_{l+r+1}' &= \sigma\left(\frac{1}{\prod_{i=l}^{l+r} \alpha_i} W_{l+r+1} \left(\prod_{j=0}^{r} \alpha_{l+j}\right) h_{l+r} + b_{l+r}\right) \\ &= h_{l+r+1}. \end{aligned} \tag{15}$$

Here, we scale the weight parameters by α_{l+i} from the l-th layer to the $(l+r)$-th layer. Correspondingly, we apply $\prod_{j=0}^{i} \alpha_{l+j}$ to scale the bias parameters, as shown in Figure 1. The primary distinction between RSI and GSI lies in the scaling factor applied to the bias term. Unlike GSI, RSI does not alter the function of the network but adjusts the scaling factor of the bias term.

Therefore, GSI is suitable for networks with only weight parameters, while RSI is applicable to networks with both bias and weight parameters.

4.1.4. Normalized Scale Invariance

For **networks with BN layers**, BN demonstrates the Normalized Scale Invariance (NSI) property, a variant of the scale invariance property. BN operates in two distinct steps: a normalization step and a subsequent re-scaling step. The normalization step is defined as follows:

$$\hat{h_l} = \frac{h_l - \mathbb{E}[h_l]}{\sqrt{\mathrm{Var}[h_l] + \eta}}, \tag{16}$$

where the mean $\mathbb{E}[h_l]$ and variance $\mathrm{Var}[h_l]$ are computed channel-wise over the mini-batch. η is a small value introduced for numerical stability. The normalization step is followed by the re-scaling step:

$$h_{l+1} = \gamma * \hat{h}_l + \beta, \tag{17}$$

where scaling parameter γ and shifting parameter β are learnable and enable the network to learn a proper scale and bias for each feature. A network equipped with BN demonstrates scale invariance property, caused by the normalization step in BN, as illustrated by:

$$\begin{aligned} h_l' &= \sigma\left(\alpha_l W_l h_{l-1} + \alpha_l b_l\right) = \alpha_l h_l, \\ \hat{h_l'} &= \frac{\alpha_l h_l - \alpha_l \mathbb{E}[h_l]}{\sqrt{\alpha_l^2 \mathrm{Var}[h_l] + \eta}} = \hat{h_l}, \end{aligned} \tag{18}$$

where the second equality holds due to $\mathbb{E}[ax] = a\mathbb{E}[x]$ and $\mathrm{Var}[ax] = a^2\mathrm{Var}[x]$. We omit the influence of η. The NSI property is illustrated in Figure 1. Therefore, the updating of running statistics significantly contributes to the NSI property, which explains the short plateau in test error curves of scaled networks following Equation (3) in Figure 2.

In modern network architectures like ResNet (He et al., 2016), the BN layer is commonly employed after convolutional layers, thereby adhering to the NSI property. However, in certain instances, fully connected layers following convolution may be not accompanied by BN layers, causing the scale effects of these subsequent layers to deviate from the NSI property but still conform to the RSI property.

Based on the above analysis, we observe that *networks $\boldsymbol{\theta}$ without bias parameters or incorporating BN tend to exhibit closely aligned performance between $\alpha\boldsymbol{\theta}$ and $\boldsymbol{\theta}$*. Conversely, *networks $\boldsymbol{\theta}$ with bias parameters and without BN manifest a performance disparity between $\alpha\boldsymbol{\theta}$ and $\boldsymbol{\theta}$, with the extent of distinction determined by the magnitude of bias parameters*. A summarized analysis and empirical verification of the differences in output between a scaled network and a scale-invariant network are provided in Appendix A.2.

4.2. Noise Robustness

Considering the insight discussed in Section 3.2 that the monotonic decreasing property is independent of initialization, we can generalize $\boldsymbol{\theta}_0$ in Equation (1) and $\boldsymbol{\theta}'_0$ in Equation (2) as a noise perturbation $\boldsymbol{\epsilon}$ applied to the parameters $\alpha\boldsymbol{\theta}_F$, formally expressed as:

$$\boldsymbol{\theta}_\alpha = \alpha\boldsymbol{\theta}_F + (1-\alpha)\boldsymbol{\epsilon}. \tag{19}$$

For $\alpha\boldsymbol{\theta}_F$, the magnitude of its parameters diminishes as α decreases, while the performance remains relatively stable owing to the scale invariance property of DNNs, especially for networks with BN or without bias parameters. Concurrently, the magnitude of noise $(1-\alpha)\boldsymbol{\epsilon}$ increases, making the network $\alpha\boldsymbol{\theta}_F$ more susceptible to noise perturbation.

In terms of the noise robustness, Li et al. (2018) demonstrates that the loss and error of networks progressively increase with higher intensity of noise perturbation. It is logical to deduce that the noise $(1-\alpha)\boldsymbol{\epsilon}$ added to $\alpha\boldsymbol{\theta}_F$ amplifies as α decreases, leading to a decline in performance. Consequently, we propose that *for a given specific noise $\boldsymbol{\epsilon}$, as α decreases from 1 to 0, although $\alpha\boldsymbol{\theta}_F$ performs similarly to $\boldsymbol{\theta}_F$, the higher magnitude noise causes the scaled network to experience a monotonic increase in error*. When $\boldsymbol{\epsilon} = \boldsymbol{\theta}_0$, this proposal explains the MLI property from an inverse perspective. We assert that *the MLI property is influenced by both the scale invariance property and noise robustness of DNNs, irrespective of the optimization trajectory*.

Notably, *our proposal may not apply to arbitrary $\boldsymbol{\epsilon}$*. For instance, if $\boldsymbol{\epsilon} = -\nabla_{\boldsymbol{\theta}_F}\mathcal{L}(\boldsymbol{\theta}_F)$, the loss of $\boldsymbol{\theta}_\alpha$ in Equation (19) may either decrease or remain unchanged as α decreases from 1 to 0. However, it is challenging to prove which noise $\boldsymbol{\epsilon}$ satisfies our proposal, thus we can only empirically verify the effects of noise by several experimental studies (Section 5.1 and Appendix A.3). The results consistently demonstrate monotonic increasing error curves when α decreases from 1 to 0.

5. Experimental Results

In this section, we perform experiments aimed at comprehensively understanding the MLI formula and validating our hypotheses.

5.1. The Effect of Noise $\boldsymbol{\epsilon}$

In Section 3.2, our findings reveal that the monotonic decreasing property holds for different initialization. Subsequently, we extend the original initialization to $\boldsymbol{\epsilon}$ in Section 4.2 and study its effects in this section. Figure 2 shows that different random initialization consistently exhibit the monotonic decreasing phenomenon. Additionally, we investigate the impact of noise magnitudes. To provide a clear depiction of differences in error and loss curves, we introduce a simple measure to evaluate the descent interval of a curve. Specifically, the interpolation curves that conform to the MLI phenomenon decrease rapidly within a range of α. For instance, in the first subplot of Figure 2, the curve of $\alpha\boldsymbol{\theta}_F$ shows a rapid decline when α is in the range of $[0.1, 0.3]$. Hence, we employ the two endpoints within which the error decreases rapidly to represent the curve.

Definition 5.1. Consider an evaluation function $\mathcal{J}(\boldsymbol{\theta})$ along the linear interpolation path defined in Equation (1). The set $\mathbb{A}$ contains the values of α corresponding to the rapid descent interval of the curve:

$$\mathbb{A} := \left\{ \alpha_i \Big| \frac{\mathcal{J}(\boldsymbol{\theta}_{\alpha_i}) - \mathcal{J}(\boldsymbol{\theta}_{\alpha_{i+1}})}{\mathcal{J}(\boldsymbol{\theta}_{\alpha_0}) - \mathcal{J}(\boldsymbol{\theta}_{\alpha_1})} \geq \tau \right\}. \tag{20}$$

In order to analyze the impact of the initialization scale, we propose a metric that quantifies the relative scale between initialized parameters and converged parameters.

Definition 5.2. Consider the initialized parameters $\boldsymbol{\theta}_0$ and converged parameters $\boldsymbol{\theta}_F$, the relative scale of $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_F$ is denoted as:

$$\delta = \frac{\mathcal{M}(\boldsymbol{\theta}_0)}{\mathcal{M}(\boldsymbol{\theta}_F)}, \tag{21}$$

where $\mathcal{M}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} |\boldsymbol{\theta}_i|$ and N is the number of parameters in $\boldsymbol{\theta}$.

We denote the starting and ending points of the descent interval of the curve as α_S, α_E respectively. Therefore, the interval $[\alpha_S, \alpha_E]$ characterizes a monotonic decreasing curve and α_S indicates the length of the plateau in the curve. We set τ to 0.01 in our experiments.

5.1.1. Scaled Initialized Parameters

In contrast to training networks with varying initialization scales (Wang et al., 2023), we *multiply the initialized parameters $\boldsymbol{\theta}_0$ by different scaling factors* while keeping the converged parameters $\boldsymbol{\theta}_F$ unchanged. We present the values of α_S and α_E under various δ in Figure 3, demonstrating the monotonic decreasing error curve persists for initialized parameters across various scales. As δ increases, the plateau in error interpolation lengthens and the noise robustness gradually diminishes, as evident from Figure 3. For large values of δ, the noise $(1-\alpha)\boldsymbol{\theta}_0$ plays a prominent role in the interpolation, exerting a significant influence on the performance of interpolated networks. Conversely, $\alpha\boldsymbol{\theta}_F$ takes precedence in determining the performance of interpolated networks when δ is small. Nevertheless, the length of the rapid descent interval remains almost unchanged irrespective of the variation of δ.

5.1.2. Constant Initialized Parameters

Common initialization methods typically sample parameters from either a uniform or Gaussian distribution (Glorot & Bengio, 2010; He et al., 2015). To explore the impact of parameter distribution in initialization, we *replace the*

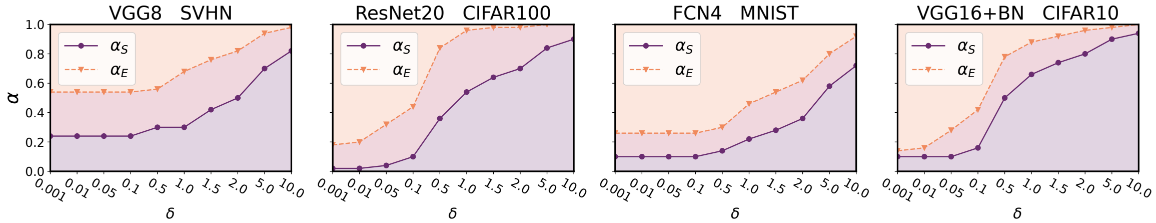

Figure 3. The α_S and α_E of test error curve along the interpolation path from initialization across various δ to the same final solution. The left two columns correspond to the scaled initialization, while the right two columns refer to the initialization with constant value. The purple, pink and orange regions respectively represent the plateau in the initial stage, the rapid descent interval and the stable phase where the performance approaches converged network.

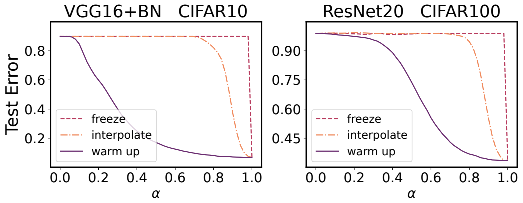

Figure 4. Test error of interpolated network where the running statistics are treated with three approaches: freeze running statistics, interpolate running statistics and reset running statistics before warming up.

initialization with a constant value, ensuring a desired δ. We find *all the results consistently exhibit the monotonic decreasing phenomenon*. Figure 3 displays α_S and α_E of the test error curve along the interpolation path between constant initialized parameters and converged parameters, which indicates our proposal about the MLI formula *holds for initialized parameters across various scales and distributions*. Furthermore, the plateau of test error curves is affected by the relative scale of noise and converged model.

5.2. The Effect of Updating BN Running Statistics

As discussed in Section 4.1.4, the updating of running statistics in BN (i.e., $\mathbb{E}[h_l]$ and $\mathrm{Var}[h_l]$ in Equation (16)) plays a crucial role in both the NSI and MLI property.

To study the effect of updating running statistics, we explore three approaches in handling these statistics in BN: (1) freezing them during interpolation; (2) interpolating them between initialized and final statistics; (3) warming up and recalculating them over the training data after resetting.

In all three methods, the learned parameters β and γ in BN are interpolated as usual. While the MLI property holds for all three ways, test error curves along the interpolation path exhibit varying lengths of plateaus, as shown in Figure 4. The plateaus of test error curves in the first two methods are longer due to the misalignment between the running statistics and activation statistics in interpolated networks.

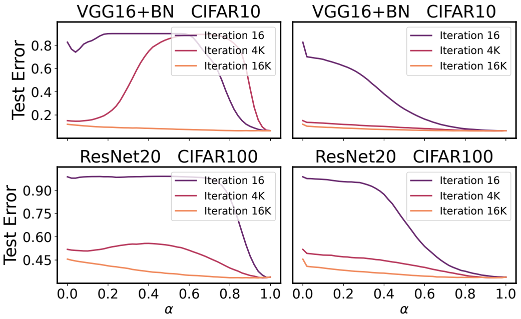

Figure 5. Test error when linearly interpolating from the state of the networks at the specified iteration to the state of the network after training. The left column corresponds to the way of interpolating the running statistics in BN, while the right column opts to reset and warm up the running statistics over an epoch of training data.

Frankle (2020) observes that linear interpolation encounters error barriers when interpolating between networks at different training iterations and the network at the end of training. However, by resetting and warming up the running statistics for each set of interpolated parameters, the test error curves along the same interpolation path satisfy the MLI property, as illustrated in Figure 5. These results highlight the importance of *handling running statistics to ensure the MLI property for networks with BN.*

5.3. Exploring the Scale Invariance Property

In our exploration of the GSI, RSI, and NSI properties, we conduct experiments using VGG16 on CIFAR10. We refer to these experiments as $GSI(\boldsymbol{\theta}_F, \alpha)$, $RSI(\boldsymbol{\theta}_F, \alpha)$ and $NSI(\boldsymbol{\theta}_F, \alpha)$ respectively. To evaluate noise robustness of $\boldsymbol{\theta}_F$, we apply noise perturbations using $(1-\alpha)\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_0$. The former noise can be regarded as the MLI formula, which applies scaled noise under different α. In contrast, the latter employs a common noise across various α. As depicted in Figure 6, the test error curves corroborate our analysis of GSI and RSI properties in Section 4.1. Notably, networks with both bias and BN display a shorter plateau in the test

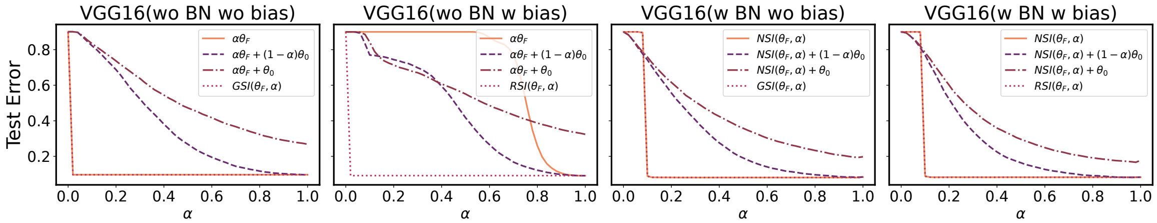

Figure 6. Test error curves of scaled networks ($\alpha\boldsymbol{\theta}_F$) and scale-invariant networks (GSI, RSI, NSI). The noise is parameterized by $(1-\alpha)\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_0$. The four groups correspond to networks wo/w BN and wo/w bias, and $NSI(\boldsymbol{\theta}_F, \alpha)$ is equivalent to $\alpha\boldsymbol{\theta}_F$ with BN.

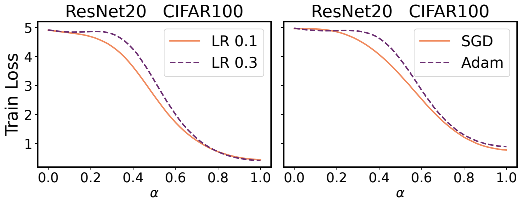

Figure 7. Train loss over linear interpolation using different learning rates and optimizers. The left column corresponds to the training configuration using SGD and the right column sets the learning rate to 0.01.

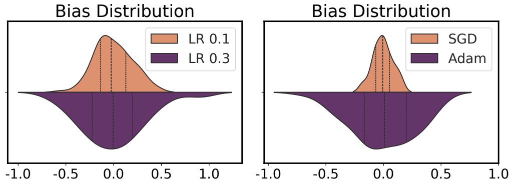

Figure 8. Comparison of bias distribution under different hyperparameters. The left column corresponds to different learning rates using SGD, while the right column corresponds to different optimizers with the same learning rate.

error curve for $\alpha\boldsymbol{\theta}_F$ compared to networks with bias, which arises from the NSI property.

Figure 6 illustrates the noise robustness of the scaled network $\alpha\boldsymbol{\theta}_F$, showing that the test error of $\alpha\boldsymbol{\theta}_F + \boldsymbol{\theta}_0$ improves as α increases, indicating improved noise robustness. Simultaneously, as α increases, the magnitude of perturbation $(1-\alpha)\boldsymbol{\theta}_0$ decreases. Hence, the performance of the network $\alpha\boldsymbol{\theta}_F + (1-\alpha)\boldsymbol{\theta}_0$ improves, exhibiting the MLI property. These results confirm our hypothesis that *the MLI property is determined jointly by the scale invariance property and noise robustness.*

5.4. Previously Unexplained MLI Violation

Lucas et al. (2021) discovers that the MLI property can be consistently broken by mechanisms that encourage the parameters to move far from initialization. Networks trained with large learning rates or with Adam optimizer (Kingma & Ba, 2014) frequently violate the MLI property. Following Lucas et al. (2021), we train ResNet20 on CIFAR100 and observe instances of MLI property violations, as shown in Figure 7. We attribute the violation of MLI property to a wider distribution of bias values centered around zero in networks trained with large learning rates or Adam compared to the distribution of bias parameters in networks that comply with the MLI property, as evident from Figure 8. The broader distribution magnifies the impact of scaled biases on the output, which can not be disregarded.

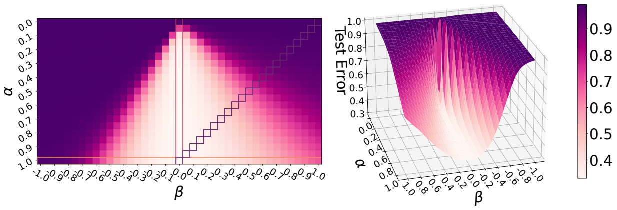

Figure 9. Test error of ResNet20 network $\boldsymbol{\theta}_F$ scaled by a range of α and perturbed by a range of β on CIFAR100.

5.5. Visualization of the Loss Landscape

We extend the one-dimensional formula of the MLI property in Equation (1) to a two-dimensional scenario: $\alpha\boldsymbol{\theta}_F + \beta\boldsymbol{\theta}_0$, where α, β represents the scaling factor and intensity of noise respectively. Figure 9 displays the test error of $\boldsymbol{\theta}_F$ at different $\alpha \in [0, 1]$ and under various $\beta \in [-1, 1]$.

Setting β to 0, we observe that $\alpha\boldsymbol{\theta}_F$ gradually achieves competitive performance with $\boldsymbol{\theta}_F$, indicating the scale invariance property, as depicted by the red rectangle. When α is set to 1, the test error of $\boldsymbol{\theta}_F + \beta\boldsymbol{\theta}_0$ reveals the noise robustness of $\boldsymbol{\theta}_F$, as shown in the orange rectangle. Notably, networks on opposite sides of $\alpha = 0$ exhibit substantial asymmetry in test error, suggesting the presence of an asymmetry local minima (He et al., 2019). The purple diagonal extending from $\alpha = 0, \beta = 1$ to $\alpha = 1, \beta = 0$ represents the test error along the interpolation path between initialized parameters $\boldsymbol{\theta}_0$ and converged parameters $\boldsymbol{\theta}_F$, which demonstrates the network satisfies the MLI property.

6. Conclusion

We begin with the MLI formula and propose an inverse perspective to explain the MLI phenomenon. Firstly, we introduce three variants of scale invariance including GSI, RSI, and NSI, which helps explain the low error of $\alpha\boldsymbol{\theta}_F$ even with small α. Next, we analyze the noise robustness of $\alpha\boldsymbol{\theta}_F$. Specifically, as α decreases from 1 to 0 in a nearly scale-invariant network $\alpha\boldsymbol{\theta}_F$, the noise robustness decreases, resulting in an increasing error when adding the noise of $(1-\alpha)\boldsymbol{\epsilon}$. Additionally, we explain the MLI phenomenon as a special case when $\boldsymbol{\epsilon} = \boldsymbol{\theta}_0$, which reveals that the MLI property is independent of initialization and is exclusively related to the scale invariance and noise robustness of DNNs.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

This work is partially supported by National Science and Technology Major Project (2022ZD0114805), NSFC (62376118, 62006112, 62250069, 61921006), Collaborative Innovation Center of Novel Software Technology and Industrialization. Professor De-Chuan Zhan is the corresponding author.

References

Ainsworth, S. K., Hayase, J., and Srinivasa, S. S. Git rebasin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023.

Andriushchenko, M., Croce, F., Müller, M., Hein, M., and Flammarion, N. A modern look at the relationship between sharpness and generalization. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 840–902, Honolulu, Hawaii, USA, 2023. PMLR.

Ba, L. J., Kiros, J. R., and Hinton, G. E. Layer normalization. *CoRR*, abs/1607.06450, 2016.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J. T., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. In *5th International Conference on Learning Representations*, Toulon, France, 2017. OpenReview.net.

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

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1019–1028, Sydney, NSW, Australia, 2017. PMLR.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. A. Essentially no barriers in neural network energy landscape. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1308–1317, Stockholmsmässan, Stockholm, Sweden, 2018. PMLR.

Entezari, R., Sedghi, H., Saukh, O., and Neyshabur, B. The role of permutation invariance in linear mode connectivity of neural networks. In *The Tenth International Conference on Learning Representations*, 2022.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *9th International Conference on Learning Representations*, Austria, 2021. OpenReview.net.

Fort, S. and Jastrzebski, S. Large scale structure of neural network loss landscapes. In *Advances in Neural Information Processing Systems 32*, pp. 6706–6714, 2019.

Frankle, J. Revisiting" qualitatively characterizing neural network optimization problems". *arXiv preprint arXiv:2012.06898*, 2020.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8803–8812, Montréal, Canada, 2018.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, D. M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 2010. JMLR.org.

Goodfellow, I. J. and Vinyals, O. Qualitatively characterizing neural network optimization problems. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015.

He, H., Huang, G., and Yuan, Y. Asymmetric valleys: Beyond sharp and flat local minima. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 2549–2560, Vancouver, BC, Canada, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision*, pp. 1026–1034, Santiago, Chile, 2015. IEEE Computer Society.

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*, pp. 770–778, Las Vegas, NV, USA, 2016. IEEE Computer Society.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Im, D. J., Tao, M., and Branson, K. An empirical analysis of the optimization of deep network loss surfaces. *arXiv preprint arXiv:1612.04010*, 2016.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456, Lille, France, 2015. JMLR.org.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020. OpenReview.net.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations*, Toulon, France, 2017. OpenReview.net.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kwon, J., Kim, J., Park, H., and Choi, I. K. ASAM: adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 5905–5914. PMLR, 2021.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 6391–6401, Montréal, Canada, 2018.

Li, X., Fan, W., Song, S., Li, Y., Li, B., Shao, Y., and Zhan, D. Asymmetric temperature scaling makes larger networks teach well again. In *Advances in Neural Information Processing Systems 35*, New Orleans, LA, USA, 2022a.

Li, X., Xu, Y., Song, S., Li, B., Li, Y., Shao, Y., and Zhan, D. Federated learning with position-aware neurons. In *Conference on Computer Vision and Pattern Recognition*, pp. 10072–10081, New Orleans, LA, USA, 2022b. IEEE.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

Lorch, E. Visualizing deep network training trajectories with pca. 2016.

Lucas, J., Bae, J., Zhang, M. R., Fort, S., Zemel, R., and Grosse, R. Analyzing monotonic linear interpolation in neural network loss landscapes. *arXiv preprint arXiv:2104.11044*, 2021.

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

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5947–5956, Long Beach, CA, USA, 2017.

Neyshabur, B., Sedghi, H., and Zhang, C. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems 33*, 2020.

Pittorino, F., Lucibello, C., Feinauer, C., Perugini, G., Baldassi, C., Demyanenko, E., and Zecchina, R. Entropic gradient descent algorithms and wide flat minima. In *9th International Conference on Learning Representations*, Austria, 2021. OpenReview.net.

Pittorino, F., Ferraro, A., Perugini, G., Feinauer, C., Baldassi, C., and Zecchina, R. Deep networks on toroids: Removing symmetries reveals the structure of flat regions in the landscape geometry. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning*, volume 162, pp. 17759–17781, Baltimore, Maryland, USA, 2022. PMLR.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015.

Tao, B., Li, L., Li, X., and Zhan, D. CLAF: contrastive learning with augmented features for imbalanced semi-supervised learning. *CoRR*, abs/2312.09598, 2023.

Vlaar, T. J. and Frankle, J. What can linear interpolation of neural network loss landscapes tell us? In *International Conference on Machine Learning*, pp. 22325–22341, 2022.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Wang, X., Wang, A. N., Zhou, M., and Ge, R. Plateau in monotonic linear interpolation - A ”biased” view of loss landscape for deep networks. In *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, 2023. OpenReview.net.

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

Zhang, X., Zhao, J. J., and LeCun, Y. Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 649–657, Montreal, Quebec, Canada, 2015.

Zhao, Y., Zhang, H., and Hu, X. Penalizing gradient norm for efficiently improving generalization in deep learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning*, volume 162, pp. 26982–26992, Baltimore, Maryland, USA, 2022. PMLR.

A. Additional Experiment Results

In our explanation of the MLI formula, we perform various additional experiments. Generally, we find that these results align well with our explanation and confirm our hypothesis that the MLI formula inherently adds various levels of noise perturbation to a nearly scale-invariant network, and MLI property is jointly determined by scale invariance property and noise robustness. In this section, we present a few additional experiments.

A.1. Findings on pre-trained models

We are interested in whether these findings in Section 3.2 hold true for the pre-trained model. To better validate our conclusion, we fine-tune ResNet50 (He et al., 2016) on CUB200 (Wah et al., 2011) and RoBERTa (Liu et al., 2019) on AG's News (Zhang et al., 2015) respectively.

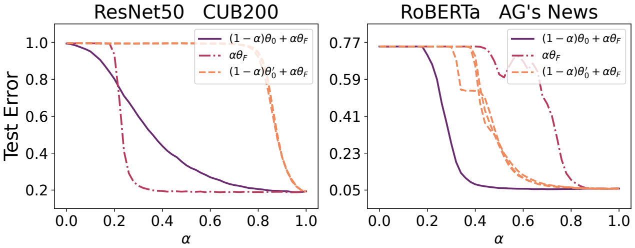

Figure 10. Test error over the linear interpolation path from pre-trained parameters or initialized parameters to fine-tuned parameters. The left endpoint corresponds to different initialization parameters and parameters in the pre-trained model, while the right endpoint corresponds to the same fine-tuned parameters in each curve.

Figure 10 plots the test error of the interpolated network following the interpolation approach in Equations (1) to (3). The pre-trained models satisfy the MLI property in the downstream task. We find that for pre-trained models, the discovery that the MLI property is independent of initialization still holds. However, RoBERTa does not exhibit significant scale invariance properties like ResNet50. We suspect that the reason lies in the fact that Layer Normalization (LN) (Ba et al., 2016) does not possess the similar scale invariance property with BN, which remains for our future work.

A.2. The differences in the outputs of $RSI(\boldsymbol{\theta}_F, \alpha)$ and $\alpha\theta_F$

This paper explores several types of scale invariance of deep neural networks. We summarize them as follows. The scale invariance property is a kind of property that deep neural networks (DNNs) have. Specifically, it refers to the property that the function of DNNs remains unchanged when definite transformations are applied to parameter weights. Several scenarios are considered:

- For networks without bias and BN layers, $\alpha\theta_F$ performs equally to θ_F (GSI in Fig. 1, Sect. 4.1.2).
- For networks with BN layers, $\alpha\theta_F$ performs equally to θ_F (NSI in Fig. 1, Sect. 4.1.4).
- For networks with bias parameters but without BN layers, $\alpha\theta_F$ performs nearly equally to θ_F and the discrepancy is determined by the bias parameters (Eq. 12). In this case, we can apply a suitable transformation to the networks to achieve the scale invariance property (RSI in Fig. 1, Section 4.1.3).

In summary, $\alpha\theta_F$ performs equally to θ_F in most cases. However, in the case of networks with bias parameters but without BN layers, $\alpha\theta_F$ performs not equally but nearly to θ_F. To address this, we propose $RSI(\theta_F, \alpha)$ to ensure that the scaled network performs equally to θ_F. Additionally, we provide empirical analysis on the differences in output between a scaled network $\alpha\theta_F$ and a scale-invariant network $RSI(\theta_F, \alpha)$ in the following.

In Section 4.1.3, we indicate that the network $\alpha\boldsymbol{\theta}$ lacks strict scale invariance property due to the scaled bias terms. Hence, we explore the difference between $\alpha\boldsymbol{\theta}_F$ and $RSI(\boldsymbol{\theta}_F, \alpha)$. We use simple networks with ReLU activation function and Mean Square Error (MSE) to measure the difference in the outputs. Our architecture consists of $128 \rightarrow 512 \rightarrow 128$ units. Figure 11 illustrates the variation of MSE with respect to α and bias. We replace the bias parameters in the last layer with

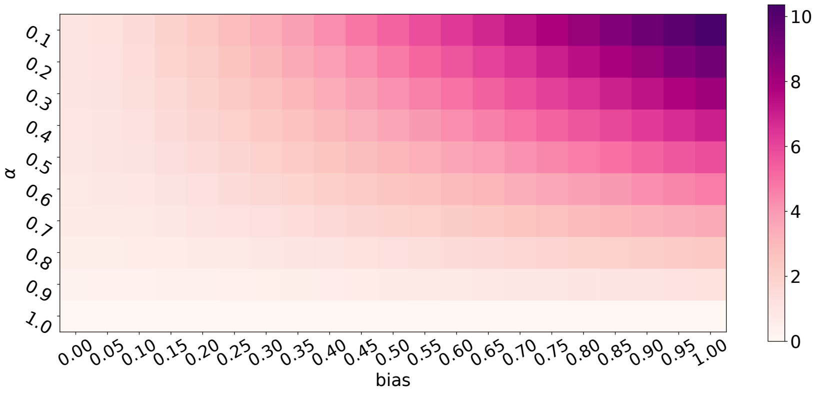

Figure 11. MSE of the outputs from $\alpha\boldsymbol{\theta}_F$ and $RSI(\boldsymbol{\theta}_F, \alpha)$

a constant value and find that MSE increases as the bias increases, which reveals that a larger value of bias has a greater impact on the output when the parameters are scaled. Additionally, the MSE between outputs decreases as α increases, which explains the test error curve of $\alpha\boldsymbol{\theta}_F$ converges to the performance of $\boldsymbol{\theta}_F$ gradually.

A.3. The noise robustness of networks

To validate the conclusion in Section 4.2, we adopt parameters obtained from different initialization methods including binary initialization, kaiming unifom, kaiming normal and orthogonal initialization as noise, and apply them to the converged parameters according to to Equation (19). The left subplot in Figure 12 contains 100 test error curves along the interpolation path between unrelated initialized parameters and the same converged parameters, 95 of which exhibit a monotonic decreasing trend. These results confirm that for a given noise, as α increases from 0 to 1, the test error of the interpolated network decreases monotonically. However, when $\boldsymbol{\epsilon}$ is set to $-\nabla_{\boldsymbol{\theta}_F}\mathcal{L}(\boldsymbol{\theta}_F)$, the test error of $\alpha\boldsymbol{\theta}_F + (1-\alpha)\boldsymbol{\theta}_0$ increases slightly at the end of interpolation, as evident from the right subplot in Figure 12.

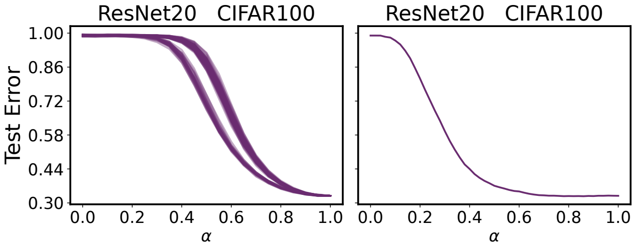

Figure 12. Test error over the interpolation path between different noise and final solution.

A.4. The effect of noise

We perform additional experiments on more networks and more datasets to explore the effect of initialization and confirm our conclusion in Section 5.1 that the scale and distribution of initialization parameters do not affect the MLI property but only impact the length of the plateau in the test error curve.

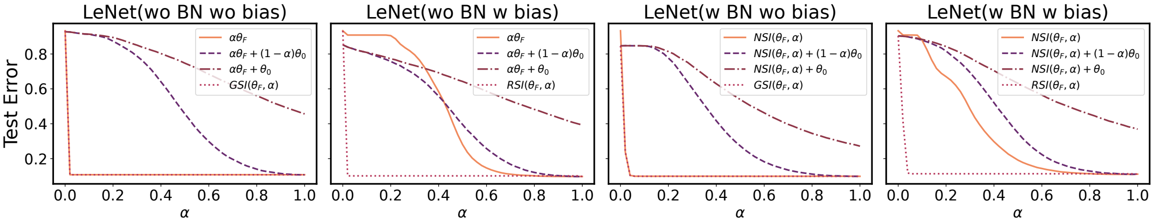

Figure 14. Test error of various scaled networks, scale-invariant networks, and perturbed networks.

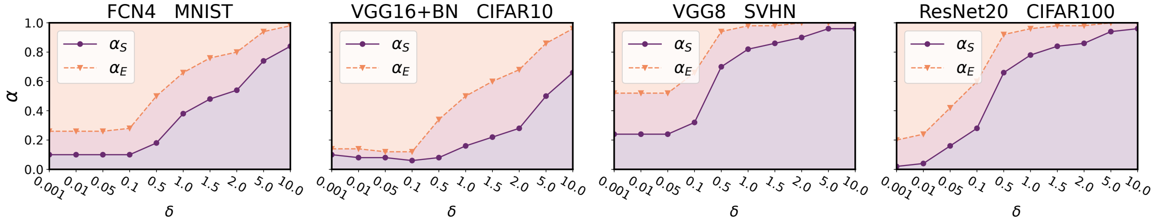

Figure 13. The α_S and α_E of test error curve along the interpolation path from initialization across various δ to the same final solution. The purple, pink, and orange regions respectively represent the plateau in the initial stage, the rapid descent interval, and the stable phase where the performance approaches converged network.

A.4.1. SCALED INITIALIZED PARAMETERS

For random initialized parameters, we scale them to satisfy the desired δ as the noise perturbation and estimate α_S, α_E of test error curve along the interpolation path between scaled initialized parameters and the original converged parameters, as shown in two sub-figures on the left of Figure 13. The results confirm the conclusion in Section 5.1 that the error plateau becomes longer with increasing in δ, while the length of rapid descent interval remains unchanged. We can infer that the value of δ determines the performance variation of the interpolated network concerning α. The noise perturbation dominates the performance when δ is large, while the converged network plays a significant role in the error of the interpolated network when δ is small.

A.4.2. CONSTANT INITIALIZED PARAMETERS

For noise perturbation consisting of a common constant value, we set the parameters to a constant value ensuring the δ interpolates them and converged parameters as MLI formula. The right two sub-figures of Figure 13 illustrate the α_S, α_E in test error curves of interpolated networks. The results demonstrate that the error of interpolated networks following the MLI formula exhibits monotonic decreasing across various distributions and scales.

A.5. Exploring the scale invariance Property

To further explore the scale invariance property of DNNs, we experiment with LeNet on the SVHN dataset and plot test error curves of networks with different compositions in Figure 14. LeNet with only weights and LeNet with bias parameters exhibit GSI property and RSI property respectively. To explore the NSI property, we add a BN layer after a convolutional layer and not modify the fully connected layer. For networks with both bias and BN, the NSI property diminishes due to the bias parameters in fully connected layers, while the RSI property still holds.

To investigate the noise robustness of scaled networks, we adopt $(1-\alpha)\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_0$ as noise perturbation. The noise robustness of the scaled converged network improves as α increases, which can be seen from the test error curve of $\alpha\boldsymbol{\theta}_F + \boldsymbol{\theta}_0$. The results align well with our hypothesis proposed in Section 5.3. The MLI property holds for the interpolated network $\alpha\boldsymbol{\theta}_F + (1-\alpha)\boldsymbol{\theta}_0$ following the MLI formula, which can be attributed to the decreasing noise perturbation and enhanced noise robustness.

A.6. Explanation of previous MLI violation

Lucas et al. (2021) shows that adaptive optimizers such as Adam consistently find final solutions that violate MLI property. We provide additional experiment results to explain the violation of MLI property. Figure 15 shows the instances of violating the MLI property. To confirm our explanation in Section 5.4, we analyze the bias distribution of the last layers in Figure 16. The networks that violate the MLI property both have a much broader distribution of bias values around 0, which leads to an increase in the impact of the scaled bias on the output of the network.

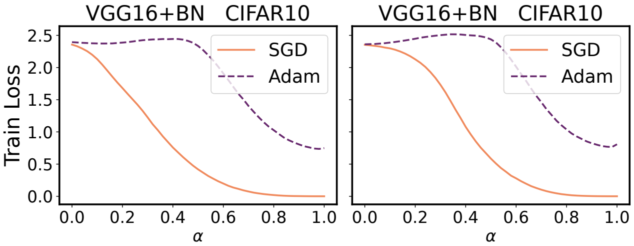

Figure 15. Train loss curves of interpolated networks that have non-monotonic interpolations from initialization to final solution.

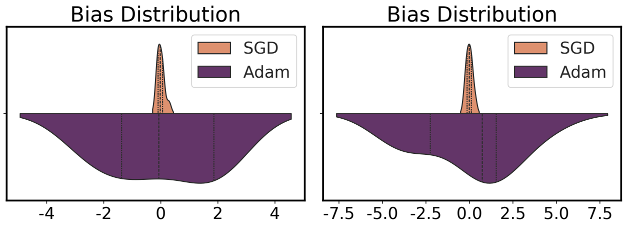

Figure 16. Comparison of bias distribution between networks satisfying the MLI property and networks that violate the MLI property.

A.7. Visualization of the loss landscape

We propose a two-dimensional visualization method and provide the visualization of the loss landscape of ResNet20 on CIFAR100 in Section 5.5. To further confirm our hypothesis about the MLI property, we visualize the test error of networks using the proposed method. The visualizations of three networks exhibit MLI property as shown in the purple rectangle of Figures 17 to 19. However, test errors of networks on opposite sides of $\alpha = 0$ exhibit substantial asymmetry, especially for networks with BN, which may confirm the conclusion in He et al. (2019) that directions on BN parameters are more asymmetric compared to non-BN parameters.

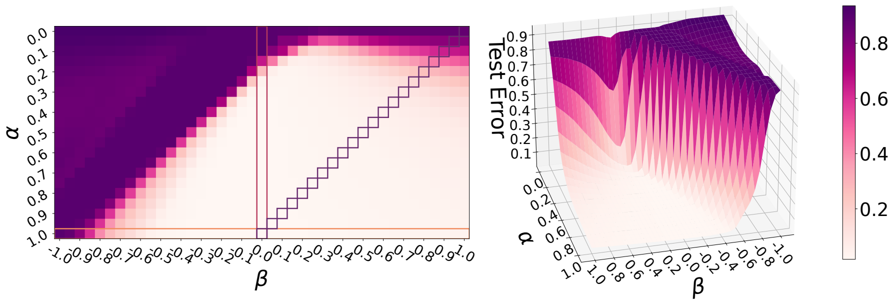

Figure 17. Test error of FCN4 scaled by a range of α and perturbed by various β on MNIST.

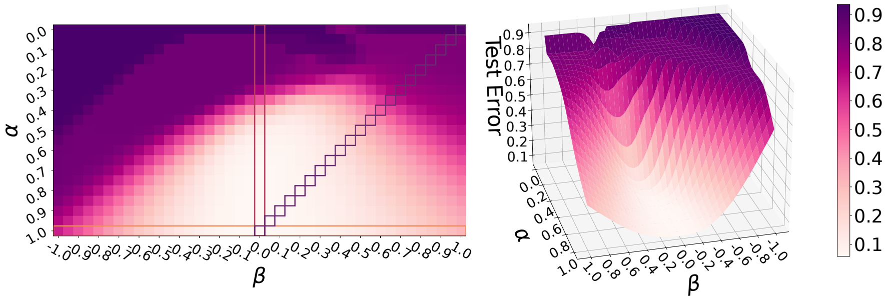

Figure 18. Test error of VGG8 without BN scaled by a range of α and perturbed by various β on SVHN.

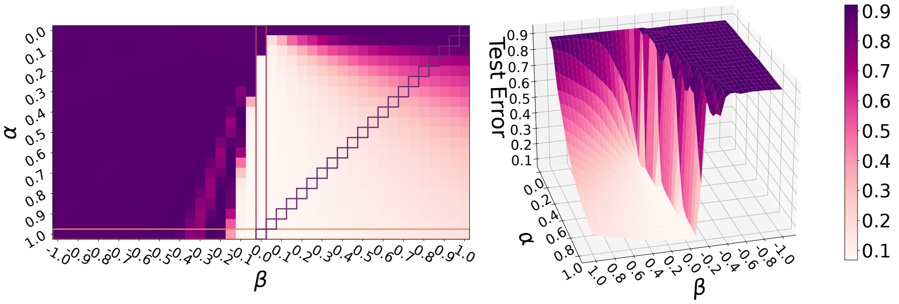

Figure 19. Test error of VGG16 with BN scaled by a range of α and perturbed by various β on CIFAR10.

B. Experiment Details

B.1. Image classification experiments

In the image classification experiments, we summarize all the settings we used below.

B.1.1. FCN4 MNIST

When training networks with the ReLU activation function, we adopt SGD optimizer with a momentum coefficient of 0.9. Each network is trained for 100 epochs using a fixed learning rate 1e-2.

B.1.2. VGG8 SVHN

We use VGG8 without batch normalization for SVHN dataset. We use Kaiming Initialization for the weight parameters and set all bias terms to zero. We train the network using SGD with momentum 0.9 and weight decay 1e-4 for 100 epochs. For the learning rate, we start from 0.01 and reduce it by a factor of 0.1 at the 60-th and 90-th epoch.

B.1.3. VGG16 CIFAR10

We train VGG16 with batch normalization on CIFAR10 dataset using SGD optimizer with momentum 0.9 and weight decay $1e-4$ for 160 epochs. We initialize the learning rate as 0.1 and decay it by a factor of 0.1 at the 80-th and 120-th epochs.

B.1.4. ResNet20 CIFAR100

We use ResNet20 for CIFAR100. We train the network using SGD with momentum 0.9 and weight decay $1e-4$ for 160 epochs. For the learning rate, we start from 0.1 and reduce it by 0.1 at the milestone of the 60-th, 90-th, and 120-th epoch.

B.2. Fine-tune pre-trained models on downstream tasks

For the image classification task, we adopt ResNet50 (He et al., 2016) pre-trained on Imagenet (Deng et al., 2009) and select CUB200 classification (Wah et al., 2011) problem as the downstream task. We fine-tune the network using SGD optimizer with momentum 0.9 and weight decay 1e-4 for 95 epochs. The learning rate is initially 1e-3 and multiplied by 0.1 for each 30 epochs.

For the sequence classification task, we adopt RoBERTa (Liu et al., 2019) pre-trained on a large corpus of English data in a self-supervised fashion and select AG's news topic classification (Zhang et al., 2015) problem as the downstream task. We fine-tune the network using Adam with weight decay 1e-4 for 5 epochs. The learning rate is initially 5e-5 and linearly reduced to 0 during training.

B.3. Violation of the MLI property

For models that violate the MLI property, we follow the hyperparameters in Lucas et al. (2021) and train it using SGD or Adam with learning rate in the set $\{0.01, 0.03, 0.1, 0.3\}$. We linearly interpolate using 50 spaced points between the network at initialization and the network at the end of training. We evaluate the error or loss on both the train set and the test set.

C. Demo Python Code

We provide a demonstration using "scikit-learn" [2] to showcase the monotonic decreasing property. The code is listed in Code 1.

Listing 1. Monotonic Loss Decreasing Presented by MLI Formula

```python
import numpy as np
from sklearn.linear_model import LogisticRegression
from matplotlib import pyplot as plt
from sklearn.datasets import load_digits
from sklearn.metrics import log_loss
from scipy.special import softmax

digits, labels = load_digits(return_X_y=True)
model = LogisticRegression(max_iter=500, fit_intercept=False)
model.fit(digits, labels)
weight = model.coef_
noise_weight = np.random.randn(*weight.shape)

betas = np.linspace(0.0, 1.0, 21)
losses = []
errors = []
for beta in betas:
    per_weight = (1.0 - beta) * noise_weight + beta * weight
    logits = np.dot(digits, per_weight.transpose())
    probas = softmax(logits, axis=1)
    loss = log_loss(y_true=labels, y_pred=probas)
    acc = np.mean(np.argmax(probas, axis=1) == labels)
    losses.append(loss)
    errors.append(1.0 - acc)

plt.figure()
plt.plot(errors / np.max(errors), color="red")
plt.plot(losses / np.max(losses), color="blue")
plt.show()
plt.close()
```

[2] https://scikit-learn.org/stable/index.html