On the Nonlinearity of Layer Normalization

Yunhao Ni [1] **Yuxin Guo** [1] **Junlong Jia** [1] **Lei Huang*** [1]

Abstract

Layer normalization (LN) is a ubiquitous technique in deep learning but our theoretical understanding to it remains elusive. This paper investigates a new theoretical direction for LN, regarding to its nonlinearity and representation capacity. We investigate the representation capacity of a network with layerwise composition of linear and LN transformations, referred to as LN-Net. We theoretically show that, given m samples with any label assignment, an LN-Net with only 3 neurons in each layer and $O(m)$ LN layers can correctly classify them. We further show the lower bound of the VC dimension of an LN-Net. The nonlinearity of LN can be amplified by group partition, which is also theoretically demonstrated with mild assumption and empirically supported by our experiments. Based on our analyses, we consider to design neural architecture by exploiting and amplifying the nonlinearity of LN, and the effectiveness is supported by our experiments.

1. Introduction

Layer normalization (LN) (Ba et al., 2016) is a ubiquitous technique in deep learning, enabling varies neural networks to train effectively. It was initially proposed to address the train-inference inconsistency problem of Batch Normalization (BN) (Ioffe & Szegedy, 2015) applied in the recurrent neural networks for Natural Language Processing (NLP) tasks. It then became the key component of Transformer (Vaswani et al., 2017) and its variants (Dai et al., 2019; Xiong et al., 2020; Dosovitskiy et al., 2021), spreading from NLP (Radford et al., 2021; Devlin et al., 2019; Raffel et al., 2020) to Computer Vision (CV) (Dosovitskiy et al., 2021; Carion et al., 2020; Cheng et al., 2022) communities. LN has got its firm position (Huang et al., 2023) in the evolution of neural architectures and is currently a basic layer in almost all the foundation models (Brown et al., 2020; Alayrac et al., 2022; Kirillov et al., 2023).

While LN is extensively used in practice, our theoretical understanding to it remains elusive. One main theoretical work for LN is its scale-invariant property, which is initially discussed in (Ba et al., 2016) to illustrate its ability in stabilizing training and is further extended in (Hoffer et al., 2018; Arora et al., 2019; Li & Arora, 2020) to consider its potential affects in optimization dynamics. Different from the previous work focusing on theoretical analyses of LN from the perspective of optimization, this paper investigates a new theoretical direction for LN, regarding to its nonlinearity and representation capacity.

We mathematically demonstrate that LN is a nonlinear transformation. We highlight that LN might be a nonlinear transformation by intuition, but there is no work, to our best knowledge, demonstrating it. Our demonstration is based on the defined lower bound named LSSR (Definition 2). The LSSR will not be broken under any linear transformation by definition, but we show that a linear neural network combined with LN can break the LSSR. Therefore, LN has nonlinearity. We also show that an LN-Net, which is a layerwise composition of linear and LN transformations, has nonlinearity.

One interesting question is that how powerful the nonlinear of an LN-Net is in theory. We theoretically show that, given m samples with any label assignment, an LN-Net with only 3 neurons in each layer and $O(m)$ LN layers can correctly classify them. We further show the lower bound of the VC dimension of an LN-Net. In particular, given an LN-Net with width only 3 neurons in each layer and L LN layers, its VC dimension is lower bounded by $L + 2$. These results show that LN-Net has great representation capacity in theory, implying the possibility that a network with linear and LN layer only can work well in practice.

We further investigate how to amplify and exploit the nonlinearity of LN. We find that Group based LN (LN-G)—which divides neurons of a layer into groups and perform LN in each group in parallel—has stronger nonlinearity than the naive LN countpart. This is also theoretically demonstrated with mild assumption and empirically supported by our comprehensive experiments. We also consider practical scenario, where we replace LN with LN-G on Transformer and ViT,

[1]SKLCCSE, Institute of Artificial Intelligence, Beihang University, Beijing, China. Correspondence to: Lei Huang <huangleiAI@buaa.edu.cn>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

since we believe the amplified nonlinearity can benefits the models. The preliminary results show the potentiality of this design in neural architecture.

2. Preliminary and Notation

We use a lowercase letter $x \in \mathbb{R}$ to denote a scalar, boldface lowercase letter $\mathbf{x} \in \mathbb{R}^d$ for a vector and boldface uppercase letter for a matrix $\boldsymbol{X} \in \mathbb{R}^{d\times m}$, where $\mathbb{R}$ is the set of real-valued numbers, and d, m are positive integers.

Neural Network. Given the input $\mathbf{x}$, a classical neural network $f_\theta(\mathbf{x})$ is typically represented as a layer-wise linear[1] and nonlinear transformation:

$$\boldsymbol{h}^{(l)} = \boldsymbol{W}^{(l)}\mathbf{x}^{(l-1)} + \boldsymbol{b}^{(l)}, \quad (1)$$

$$\mathbf{x}^{(l)} = \phi(\boldsymbol{h}^{(l)}), \quad l = 1, ..., L, \quad (2)$$

where $\theta = \{(\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}), l = 1, \cdots, L\}$ are learnable parameters, $\mathbf{x}^{(0)} = \mathbf{x}$, $\boldsymbol{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$, $\boldsymbol{b}^{(l)} \in \mathbb{R}^{d_l}$ and d_l indicates the number of neurons in the l-th layer. We set $\mathbf{x}^{(L)} = \boldsymbol{h}^{(L)}$ as the output of the network $f_\theta(\mathbf{x})$ to simplify denotations. A neural network without nonlinear transformation $\phi(\cdot)$ (Eqn. 2) is referred to as a *linear neural network*, which is still a linear transformation in native.

Layer Normalization. Layer Normalization (LN) is an essential layer in modern deep neural networks mainly for stabilizing training. Given a single sample of layer input $\mathbf{x} = [x^{(1)}, x^{(2)}, \cdots, x^{(d)}] \in \mathbb{R}^d$ with d neurons in a neural network, LN standardizes $\mathbf{x}$ within the neurons as [2]:

$$\hat{x}^{(j)} = LN(x^{(j)}) = \frac{x^{(j)} - \mu}{\sigma}, \quad j = 1, 2, \cdots, d, \quad (3)$$

where $\mu = \frac{1}{d}\sum_{i=1}^{d} x^{(j)}$ and $\sigma = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(x^{(j)} - \mu)^2}$ are the mean and variance for each sample, respectively. The standardization operation can be viewed as a combination of centering and scaling operations. Centering projects $\mathbf{x}$ onto the hyperplane $\{\mathbf{x} \in \mathbb{R}^d : x^{(1)} + \cdots + x^{(d)} = 0\}$, by $\tilde{\mathbf{x}} = (\boldsymbol{I} - \frac{1}{d}\mathbf{1}_d\mathbf{1}_d^\top)\mathbf{x}$. Scaling projects $\tilde{\mathbf{x}}$ onto the sphere $\{\mathbf{x} \in \mathbb{R}^d : [x^{(1)}]^2 + \cdots + [x^{(d)}]^2 = d\}$, by $\hat{\mathbf{x}} = \sqrt{d}\tilde{\mathbf{x}}/\|\tilde{\mathbf{x}}\|_2$. We thus also call scaling as Spherical Projection (SP), from the geometric perspective. Note that SP is the only operation for normalization in RMSNorm (Zhang & Sennrich, 2019).

Sum of Squares. Sum of Squares (SS) (Fisher, 1970) is a statistical concept that measures the variability or dispersion within a set of data. Denote m samples from class c as $\mathbf{x}_{c1}, \cdots, \mathbf{x}_{cm} \in \mathbb{R}^d$, represented as a matrix $\boldsymbol{X}_c = [\mathbf{x}_{c1}, \cdots, \mathbf{x}_{cm}]$, then SS of $\boldsymbol{X}_c$ is defined as

$$SS(\boldsymbol{X}_c) = \sum_{i=1}^{m} \|\mathbf{x}_{ci} - \bar{\mathbf{x}}_c\|^2, \quad (4)$$

where $\bar{\mathbf{x}}_c = (\mathbf{x}_{c1} + \cdots + \mathbf{x}_{cm})/m$.

3. The Existence of Nonlinearity in LN

In this section, we define Sum of Squares Ratio (SSR) and its linear invariant lower bound named LSSR. We then show that LN can break the boundary of SSR and plays a role in nonlinear representation.

3.1. Linear Invariant Lower Bound

We take binary classification for simplifying discussion. Let $\boldsymbol{X}_c = [\mathbf{x}_{c1}, \cdots, \mathbf{x}_{cm}]$ represents m samples[3] in $\mathbb{R}^d$ from the corresponding class $c \in \{1, 2\}$, and $[\boldsymbol{X}_1, \boldsymbol{X}_2] \in \mathbb{R}^{d\times 2m}$ represents all the samples together.

Definition 1. *(SSR.) Given $SS([\boldsymbol{X}_1, \boldsymbol{X}_2]) \neq 0$, the Sum of Squares Ratio (SSR) between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ is defined as*

$$SSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = \frac{SS(\boldsymbol{X}_1) + SS(\boldsymbol{X}_2)}{SS([\boldsymbol{X}_1, \boldsymbol{X}_2])}. \quad (5)$$

It is easy to demonstrate that $SSR(\boldsymbol{X}_1, \boldsymbol{X}_2) \in [0, 1]$. SSR can be an indicator to show how easy the samples in the Euclidean space from different classes can be separated. I.e., the smaller SSR is, the more easily $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are to be separated with Eulcidean distance as a measurement in most cases. Based on SSR, we further define its lower bound under any linear transformation as follows.

Definition 2. *(LSSR.) The Linear SSR (LSSR) between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ is defined as*

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = \inf_{\varphi \in \mathbb{D}_\varphi(d)} SSR(\varphi(\boldsymbol{X}_1), \varphi(\boldsymbol{X}_2)), \quad (6)$$

where $\mathbb{D}_\varphi(d)$ is the set of all linear functions defined on $\mathbb{R}^d$.

By definition, LSSR is the lower bound of SSR under any linear transformation. LSSR can be an indicator to show how easy the samples from different classes can be linearly separated. We provide illustrative examples in *Appendix A* for details. In the following proposition, we show a linear neural network can not break LSSR.

Proposition 1. *Given $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbb{R}^{d_0\times m}$ and a linear neural network represented as $\tilde{\varphi} = \varphi_1 \circ \cdots \circ \varphi_L$, where $\varphi_l : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}, (l = 1, \cdots, L)$ are all linear transformations as shown in Eqn. 1, we have that*

$$SSR(\tilde{\varphi}(\boldsymbol{X}_1), \tilde{\varphi}(\boldsymbol{X}_2)) \geq LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2). \quad (7)$$

[1] We follow the convention in deep learning community, and do not differentiate between the linear and affine transformation.

[2] LN usually uses extra learnable scale and shift parameters (Ioffe & Szegedy, 2015), and we omit them for simplifying discussion as they are affine transformation in native

[3] We use the same number (m) of samples in each class for simplifying notation, and our subsequent definition and conclusion are also apply to different number for different classes.

Proposition 1 is easily proved by the definition of LSSR, since we have $\tilde{\varphi} \in \mathbb{D}_{\varphi}(d_0)$. Proposition 1 implies that the SSR will not break the lower bound if we use an arbitrary *linear neural network* as a representation transformation over the samples. One interesting question is that whether a *linear neural network* combined with LN can break the lower bound of SSR. If Yes, we can show that LN has nonlinearity.

3.2. Break the Lower Bound of SSR with LN

Here, we focus on the *linear neural network* combined with LN. To state more precisely, we denote LN-Net as follows.

Definition 3. *(LN-Net.) The LN-Net $f_\theta(\mathbf{x})$ is defined as layer-wise composition of linear and LN transformation:*

$$\begin{aligned} \boldsymbol{h}^{(l)} &= \boldsymbol{W}^{(l)}\mathbf{x}^{(l-1)} + \boldsymbol{b}^{(l)}, \quad l = 1, ..., L, \\ \mathbf{x}^{(l)} &= LN(\boldsymbol{h}^{(l)}), \quad l = 1, ..., L-1, \end{aligned} \tag{8}$$

where $\theta = \{(\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}), l = 1, ..., L\}$ are learnable parameters, $\mathbf{x}^{(0)} = \mathbf{x}$ and $LN(\cdot)$ denotes the LN operation. We set $\mathbf{x}^{(L)} = \boldsymbol{h}^{(L)}$ as the output of the network $f_\theta(\mathbf{x})$ to simplify denotations.

We first provide a tractable method to calculate LSSR, stated by the following proposition.

Proposition 2. *Given $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbb{R}^{d\times m}$, we denote $\boldsymbol{M} = \sum_{c=1}^{2}\sum_{i=1}^{m}(\mathbf{x}_{ci} - \bar{\mathbf{x}}_c)(\mathbf{x}_{ci} - \bar{\mathbf{x}}_c)^\top$, and $\boldsymbol{N} = \sum_{c=1}^{2}\sum_{i=1}^{m}(\mathbf{x}_{ci} - \bar{\mathbf{x}})(\mathbf{x}_{ci} - \bar{\mathbf{x}})^\top$, where $\bar{\mathbf{x}} = (\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2)/2$. Supposing that $\boldsymbol{N}$ is reversible, we have*

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = \lambda^*, \tag{9}$$

and correspondingly,

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = SSR((\boldsymbol{u}^*)^\top\boldsymbol{X}_1, (\boldsymbol{u}^*)^\top\boldsymbol{X}_2), \tag{10}$$

where λ^* and $\boldsymbol{u}^*$ are the minimum eigenvalue and corresponding eigenvector of $\boldsymbol{N}^{-1}\boldsymbol{M}$.

The proof of Proposition 2 are shown in *Appendix B*. Based on Proposition 2, we further define $f_{SSR}(t)$ as

$$f_{SSR}(t) = \begin{cases} LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2), & t = 0, \\ SSR(\bar{\psi}(t; \boldsymbol{X}_1), \bar{\psi}(t; \boldsymbol{X}_2)), & t \neq 0, \end{cases} \tag{11}$$

where $\bar{\psi}(t; \mathbf{x}_{ci}) = \mathbf{1}^\top\bar{\varphi}(t; \mathbf{x}_{ci})/\|\bar{\varphi}(t; \mathbf{x}_{ci})\|_2$, $\bar{\varphi}(t; \mathbf{x}_{ci}) = [(\boldsymbol{u}^*)^\top\mathbf{x}_{ci}t, 1]^\top$ and $t \in \mathbb{R}$. We point out that $f_{SSR}(t)$ is derivable at $t = 0$, and $f'_{SSR}(0)$ is only decided by $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, which is proved in *Appendix C*.

Based on the definition of $f_{SSR}(t)$, we show that LN-Net can decrease the LSSR as stated by the following theorem.

Theorem 1. *Let $\psi = \varphi_1 \circ LN(\cdot) \circ \varphi_2$, performing over the input $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbb{R}^{d\times m}$. If $f'_{SSR}(0) \neq 0$, we can always find suitable linear functions φ_1 and φ_2, such that*

$$SSR(\psi(\boldsymbol{X}_1), \psi(\boldsymbol{X}_2)) < LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2). \tag{12}$$

The proof of Theorem 1 requires complicated derivation. Please refer to *Appendix C* for details. Note that LN-Net is a more general form of ψ in Theorem 1, which implies that LN-Net can break the lower bound of SSR.

Based on Theorem 1, we can obtain the following statement. We deduce that LN is a nonlinear transformation.

Corollary 1. *LN is a nonlinear transformation.*

Proof. We assume that $LN(\cdot)$ is a linear transformation. We thus have LN-Net is also a linear transformation. Based on Proposition 1, we have LN-Net can not break LSSR. This contradicts Theorem 1. Therefore, $LN(\cdot)$ must be a nonlinear transformation. □

Summary. In this section, we mathematically show that LN is a nonlinear transformation, and LN-Net is a network with nonlinearity. One interesting question is that how powerful the nonlinearity of an LN-Net is in theory. We will discuss about it in the following section.

4. Capacity of a Network with LN

In this section, we apply LN-Net to classify m samples with any label assignment. To prove the existence of such LN-Net, we propose Projection Merge Algorithm (PMA) and Parallelization Breaking Algorithm (PBA) to help find the parameters of the LN-Net.

4.1. LN for Xor Classification

To understand PMA intuitively, we use Spherical Projection (SP) rather than LN at the beginning. But we replace SP with LN and linear layers back in the end, according to the lemma as follows.

Lemma 1. *Denote $LN(\cdot)$ as the LN operation in $\mathbb{R}^d (d \geq 3)$, and $SP(\cdot)$ as the SP operation[4] in $\mathbb{R}^{d-1}$. We can find some linear transformations $\hat{\varphi}_1$ and $\hat{\varphi}_2$, such that*

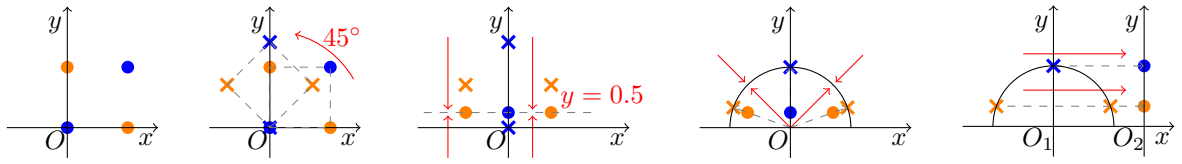
$$SP(\cdot) = \hat{\varphi}_1 \circ LN(\cdot) \circ \hat{\varphi}_2. \tag{13}$$

The proof of Lemma 1 is shown in *Appendix C*. And we can easily obtain the following corollary.

Corollary 2. *$SP(\cdot)$ can be represented by an LN-Net.*

Taking xor classification as an example, we primarily show how we use LN-Net to classify linearly inseparable samples.

[4] If there are no special instructions, we denote SP projects the sample on to the unit circle, namely $\mathbf{x} \mapsto \mathbf{x}/\|\mathbf{x}\|_2$.



(a) Initial input. (b) Rotation. (c) Linear projection. (d) Spherical projection. (e) Linear projection.

Figure 1. Solution to the Xor Classification. To begin with, we rotate them by $45°$, as shown in Figure 1(b). Then we vertically project them onto $y = 0.5$, as shown in Figure 1(c). Next, we spherically project them onto the circle $x^2 + y^2 = 1$, as shown in Figure 1(d). Finally, we horizontally project them onto $x = 0$, as shown in Figure 1(e). Now we have classified the two classes.

As shown in Figure 1(a), $(0,0),(1,1)$ and $(0,1),(1,0)$ belong to different classes. Obviously, the two classes are not linearly separable. We can classify them with SP and linear transformations only, please refer to the demonstration in Figure 1 for details.

By Lemma 1, replace SP with LN-Net. Therefore, we can construct an LN-Net according to the operations in Figure 1, and then classify the xor samples.

More generally, we discuss binary classification in Section 4.2 and multi-class classification in Section 4.3.

4.2. LN for Binary Classification

Theorem 2. *Given m samples with any binary label assignment in $\{0,1\}$, there always exists an LN-Net with only 3 neurons per layer and $O(m)$ LN layers can correctly classify them.*

To prove Theorem 2, we represent the LN-Net with SP and linear layers. Then we design an algorithm to help compute the parameters according to the input. We hence get an LN-Net with proper parameters to classify the samples. The proof is shown as follows.

We represent an LN-Net as

$$f_\theta(\cdot) = \varphi_1 \circ LN(\cdot) \circ \varphi_2 \circ \cdots \circ \varphi_{L-1} \circ LN(\cdot) \circ \varphi_L, \quad (14)$$

where $\varphi_1, \cdots, \varphi_L$ denote the linear layers, and $LN(\cdot)$ denotes the LN layers. For convenience, we replace LN with SP temporarily.

Proposition 3. *The LN-Net $f_\theta(\cdot)$ in Eqn.14 can be represented by SP and linear layers equivalently.*

Proof. Since each $LN(\cdot)$ acts on $\mathbb{R}^3$, by Lemma 1, we can construct a 2-dimensional $SP(\cdot) = \hat{\varphi}_1 \circ LN(\cdot) \circ \hat{\varphi}_2$. Define each φ_l in Theorem 2 as

$$\varphi_l = \begin{cases} \varphi_l^{(1)} \circ \varphi_l^{(2)} \circ \hat{\varphi}_1, & l = 1, \\ \hat{\varphi}_2 \circ \varphi_l^{(1)} \circ \varphi_l^{(2)} \circ \hat{\varphi}_1, & 1 < l < L, \\ \hat{\varphi}_2 \circ \varphi_l^{(1)}, & l = L, \end{cases} \quad (15)$$

where $\varphi_l^{(1)}$ and $\varphi_l^{(2)}$ are both linear functions. By Eqn.15 and Lemma 1, we can rewrite $f_\theta(\cdot)$ as

$$\tilde{f}_\theta(\cdot) = \varphi_1^{(1)} \circ \varphi_1^{(2)} \circ SP(\cdot) \circ \varphi_2^{(1)} \circ \cdots \circ SP(\cdot) \circ \varphi_L^{(1)}, \quad (16)$$

namely, $f_\theta(\cdot)$ can be represented by SP and linear layers equivalently. □

Hereafter, we consider to compute the parameters of $\tilde{f}_\theta(\cdot)$. Specifically, for each layer, we denote

$$\begin{cases} \varphi_l^{(1)} : \boldsymbol{X}^{(l-1)} \mapsto \boldsymbol{P}^{(l)}, & 1 \le l \le L; \\ \varphi_l^{(2)} : \boldsymbol{P}^{(l)} \mapsto \boldsymbol{H}^{(l)}, & 1 \le l \le L-1; \\ SP(\cdot) : \boldsymbol{H}^{(l)} \mapsto \boldsymbol{X}^{(l)}, & 1 \le l \le L-1. \end{cases} \quad (17)$$

Besides, the input of $\tilde{f}_\theta(\cdot)$ is $\boldsymbol{X}^{(0)} = [\mathbf{x}_1^{(0)}, \cdots, \mathbf{x}_m^{(0)}]$, and the output is $\boldsymbol{P}^{(L)}$. Now we construct $\tilde{f}_\theta(\cdot)$ step by step.

We denote that for each $\boldsymbol{P}^{(l)}, (l = 1, \cdots, L)$, these points are on the x-axis, namely $\boldsymbol{p}_k^{(l)} = [p_k^{(l)}, 0]^\top, (k = 1, \cdots, m)$. To get $\boldsymbol{P}^{(1)}$, we apply $\varphi_1^{(1)}$ for initialization as below.

Proposition 4. *For any input $\boldsymbol{X}^{(0)}$, we can find some $\boldsymbol{u}$, such that*

$$\varphi_1^{(1)} : \mathbf{x}_k^{(0)} \mapsto \boldsymbol{p}_k^{(1)} = [\boldsymbol{u}^\top \mathbf{x}_k^{(0)}, 0]^\top, \quad (18)$$

where $\boldsymbol{p}_i^{(1)} \neq \boldsymbol{p}_j^{(1)}$ if $\mathbf{x}_i^{(0)} \neq \mathbf{x}_j^{(0)}$.

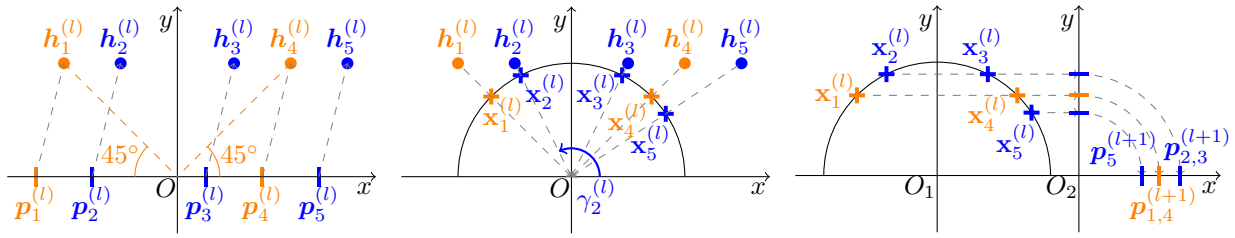
Proposition 4 parameterizes $\varphi_1^{(1)}$ and initializes $\boldsymbol{P}^{(1)}$ onto the x-axis, without merging different points[5]. Please refer to *Appendix D* for the proof.

As for other linear functions, the suitable parameters are generated from the Projection Merge Algorithm, as shown in Algorithm 1.

In Algorithm 1, $\boldsymbol{P}^{(L)}$ is the output, as well as that of $\tilde{f}_\theta(\cdot)$. Factually, by Algorithm 1, we get each $\boldsymbol{P}^{(l)}$ in a recursive way. For the case $\mathbb{J}_i \neq \emptyset$, we take 5 points as an example to show how we get $\boldsymbol{P}^{(l+1)}$ from $\boldsymbol{P}^{(l)}$ in Figure 2.

As for the case $\mathbb{J}_i = \emptyset$, it indicates that all points with the same label as $\boldsymbol{p}_i^{(l)}$ are merged together. Therefore, we

[5] In this paper, we claim that $\boldsymbol{p}_i^{(l)}$ and $\boldsymbol{p}_j^{(l)}$ are "different points" means $\boldsymbol{p}_i^{(l)} \neq \boldsymbol{p}_j^{(l)}$ rather than $i \neq j$, for each hidden layer (applies to $\mathbf{x}$ and $\boldsymbol{h}$ as well).



(a) Translation. (b) Spherical projection. (c) Linear projection.

Figure 2. Get $\boldsymbol{P}^{(l+1)}$ from $\boldsymbol{P}^{(l)}$ geometrically. In Figure 2(a), $\boldsymbol{P}^{(l)}$ is shown as the bars on the x-axis. At first, find the leftmost point, namely $\boldsymbol{p}_1^{(l)}$. Then we find another point with the same label as $\boldsymbol{p}_1^{(l)}$, but right of $\boldsymbol{p}_1^{(l)}$, choose the leftmost one, namely $\boldsymbol{p}_4^{(l)}$. Afterwards, shift all the points up by $(p_4^{(l)} - p_1^{(l)})/2$, and left by $(p_4^{(l)} + p_1^{(l)})/2$, then we get $\boldsymbol{H}^{(l)}$, as shown in Figure 2(a). Next, spherically project $\boldsymbol{H}^{(l)}$ onto the unit circle and get $\boldsymbol{X}^{(l)}$, shown as '+'s in Figure 2(b). Finally merge the points in $\boldsymbol{X}^{(l)}$ by their ordinates, as the new abscissas of $\boldsymbol{P}^{(l+1)}$, and take 0 as the new ordinates of $\boldsymbol{P}^{(l+1)}$, as shown in Figure 2(c). Now, we have $\boldsymbol{P}^{(l+1)}$.

Algorithm 1 Projection Merge Algorithm

input The initial input $\boldsymbol{P}^{(1)}$.
output The final output $\boldsymbol{P}^{(L)}$.

$l \leftarrow 1$;
$\mathbb{P} \leftarrow \{\boldsymbol{p}_1^{(l)}, \boldsymbol{p}_2^{(l)}, \cdots, \boldsymbol{p}_m^{(l)}\}$;
while $\mathbb{P} \neq \emptyset$ **do**
  $i \leftarrow \arg\min_k \{p_k^{(l)} : \boldsymbol{p}_k^{(l)} \in \mathbb{P}\}$;
  $\mathbb{J}_i \leftarrow \{\boldsymbol{p}_j^{(l)} \in \mathbb{P} : \boldsymbol{p}_j^{(l)} \neq \boldsymbol{p}_i^{(l)}, y_j = y_i\}$;
  if $\mathbb{J}_i \neq \emptyset$ **then**
    $j \leftarrow \arg\min_k \{p_k^{(l)} : \boldsymbol{p}_k^{(l)} \in \mathbb{J}_i\}$;
    for $k \leftarrow 1$ to m **do**
      $\boldsymbol{h}_k^{(l)} \leftarrow \boldsymbol{p}_k^{(l)} - \begin{bmatrix} p_i^{(l)} + p_j^{(l)} \\ p_i^{(l)} - p_j^{(l)} \end{bmatrix} /2$;
      $\mathbf{x}_k^{(l)} \leftarrow \boldsymbol{h}_k^{(l)} / \|\boldsymbol{h}_k^{(l)}\|$;
      $\boldsymbol{p}_k^{(l+1)} \leftarrow \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}_k^{(l)}$;
    end for
    $l \leftarrow l + 1$;
    $\mathbb{P} \leftarrow \{\boldsymbol{p}_1^{(l)}, \boldsymbol{p}_2^{(l)}, \cdots, \boldsymbol{p}_m^{(l)}\}$;
  else
    remove $\boldsymbol{p}_j^{(l)}$ from $\mathbb{P}$, as long as $\boldsymbol{p}_j^{(l)} = \boldsymbol{p}_i^{(l)}$;
  end if
end while
return $\boldsymbol{P}^{(l)}$;

remove them from $\mathbb{P}$, and choose the leftmost point from the remaining $\mathbb{P}$, until $\mathbb{P} = \emptyset$.

Based above, we give the properties of each layer as follows.

Proposition 5. *For each layer, $\varphi_l^{(1)} (2 \leq l \leq L)$ only merges points with the same label. Nevertheless, $\varphi_1^{(1)}, SP(\cdot)$ and $\varphi_l^{(2)} (1 \leq l \leq L-1)$ do not merge any points.*

Please refer to *Appendix D* for the proof of Proposition 5.

By Proposition 5, we figure out that Algorithm 2 will only merge points with the same label. Besides, we find that from $\boldsymbol{P}^{(l)}$ to $\boldsymbol{P}^{(l+1)}$, the number of different points will decrease at least 1. Since the input is m different points from two classes, we merge at most $m-2$ times by Algorithm 1, we thus have $L - 1 \leq m - 2$.

By Algorithm 1, we can construct other linear functions with exact parameters as follows.

$$\begin{cases} \varphi_l^{(1)} : \mathbf{x}_k^{(l-1)} \mapsto \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}_k^{(l-1)}, & 1 < l \leq L, \\ \varphi_l^{(2)} : \boldsymbol{p}_k^{(l)} \mapsto \boldsymbol{p}_k^{(l)} - \begin{bmatrix} p_i^{(l)} + p_j^{(l)} \\ p_i^{(l)} - p_j^{(l)} \end{bmatrix} /2, & 1 \leq l < L. \end{cases} \tag{19}$$

Therefore, $\tilde{f}_\theta(\cdot)$ with the parameters in Eqn.18 and Eqn.19 can classify the samples $\boldsymbol{X}^{(0)}$. Besides, the LN-Net in Eqn.14 with depth[6] $L - 1 = O(m)$ can also classify the m samples. We hence have proved Theorem 2.

Our results above are based on an LN-Net with 3 neurons each layer. Furthermore, we can generalize PMA for a wider neural network, but it is much more complex. Please refer to *Appendix D* for more details.

Based on Theorem 2, we can easily obtain the following corollary related to VC dimension (Bartlett et al., 1998) of an LN-Net.

Corollary 3. *Given an LN-Net $f_\theta(\cdot)$ with width 3 and depth L, its VC dimension $VCdim(f_\theta(\cdot))$ is lower bounded by $L + 2$.*

4.3. LN for Multi-class Classification

Theorem 3. *Given m samples with any binary label assignment, there always exists an LN-Net with only 3 neurons per layer and $O(m)$ LN layers can correctly classify them.*

Applying Algorithm 1 for a multi-class classification may

[6] We denote the number of LNs as the depth of an LN-Net.

confuse two samples with different labels. We thus introduce Parallelization Breaking Algorithm to avoid such confusion. Besides, we can also construct an LN-Net to classify the samples. The detailed analysis and proof are as below.

To begin with, we are concerned about whether Algorithm 1 applies to multi-class classification—the answer is Not. Based on Figure 2(c), we recolor $\mathbf{x}_3^{(l)}$ red, as shown in Figure 3. When we merge $\mathbf{x}_1^{(l)}$ and $\mathbf{x}_4^{(l)}$, $\mathbf{x}_2^{(l)}$ and $\mathbf{x}_3^{(l)}$ will be merged in the meanwhile. In other words, the algorithm will confuse them to be in the same class. Proposition 6 indicates the necessary condition for such confusion.

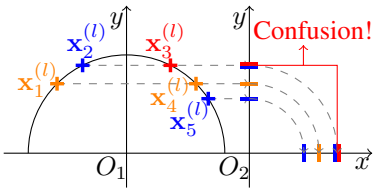


Figure 3. The case of confusion in the merging process.

Proposition 6. *Confusion refers to merging two points with different labels. If confusion happens when we project $\boldsymbol{X}^{(l+1)}$ onto the y-axis, there must be a parallelogram*[7] *consisting of four different points in $\boldsymbol{P}^{(l)}$.*

In reverse, if there is no parallelograms in $\boldsymbol{P}^{(l)}$, confusion will never happen when applying Algorithm 1. Please refer to *Appendix D* for the proof of Proposition 6.

To avoid such confusion, we propose Parallelization Breaking Algorithm (PBA) as follows.

Algorithm 2 Parallelization Breaking Algorithm

input $\boldsymbol{P}^{(l)}$, $\boldsymbol{u}_l$ (got by Proposition 7).
output $\hat{\boldsymbol{P}}^{(l)}$.
1: **for** $k \leftarrow 1$ to m **do**
2: $\quad \tilde{\boldsymbol{p}}_k^{(l)} = SP(\boldsymbol{p}_k^{(l)} + [0,1]^\top)$;
3: $\quad \hat{\boldsymbol{p}}_k^{(l)} = [\boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_k^{(l)}, 0]^\top$;
4: **end for**
5: **return** $\hat{\boldsymbol{P}}^{(l)}$;

Proposition 7. *We can always find $\boldsymbol{u}_l \in \mathbb{R}^2$ for Algorithm 2, such that there is no parallelograms in $\hat{\boldsymbol{P}}^{(l)}$, and no points merged in the algorithm.*

Please refer to *Appendix D* for the proof of Proposition 7.

PBA helps us transform $\boldsymbol{P}^{(l)}$ to $\hat{\boldsymbol{P}}^{(l)}$, based on which confusion will never happen. For multi-class classification, we insert PBA between $\varphi_l^{(1)}$ and $\varphi_l^{(2)}$ in Eqn.16, then given m samples with any label assignment, $\tilde{f}_\theta(\cdot)$ with PBA can classify them. Based above, we replace SP with LN and linear layers in $\tilde{f}_\theta(\cdot)$ with PBA, and then merge the adjacent linear layers. We figure out $\tilde{f}_\theta(\cdot)$ with PBA is also an LN-Net. We point out that the depth of this LN-Net is no more than $2m$. We hence have proved Theorem 3.

Summary. In this section, we show that LN-Net also has powerful capacity in theory. Our theoretical results show that an LN-Net with width 3 and depth $O(m)$ is able to classify given m samples with any label assignment. We see an LN-Net performing over 3 neurons can introduce nonlinearity. One question is that whether the nonlinearity of an LN-Net with $d > 3$ neurons can be amplified, if we group neurons and perform LN in each group in parallel? We answer it in the following section.

5. Amplify and Exploit the Nonlinearity of LN

5.1. Comparison of Nonlinearity

In this part, we first define a measurement over the Hessian matrix to compare the magnitude of the nonlinearity. We then show the Group based LN (LN-G)[8]—which divides neurons of a layer into groups and perform LN in each group in parallel—has stronger nonlinearity than the naive LN countpart.

Hessian of Linear Function. Given a twice differential function $f(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$, we focus on its Hessian Matrix $\nabla^2 f(\mathbf{x})$. If $f(\mathbf{x})$ is a linear function, we have $\nabla^2 f(\mathbf{x}) \equiv \boldsymbol{O}$. More generally, suppose that $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ is a linear transformation, we define $\varphi(\mathbf{x}) = \big[\varphi_1(\mathbf{x}), \cdots, \varphi_d(\mathbf{x})\big]^\top$, and each $\varphi_i(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ is a linear function, namely each Hessian matrix $\nabla^2_{\mathbf{x}} \varphi_i(\mathbf{x}) = \boldsymbol{O}$.

Measurement of Nonlinearity. Given a twice differential function[9] $f : \mathbb{R}^d \to \mathbb{R}^d$ and $\boldsymbol{y} = f(\mathbf{x})$. Denote $\boldsymbol{y} = [y_1, \cdots, y_d]^\top$ and $\mathbf{x} = [x_1, \cdots, x_d]^\top$. We define $\mathcal{H}(f; \mathbf{x})$ as an indicator to describe the Hessian information of $f : \mathbb{R}^d \to \mathbb{R}^d$ as

$$\mathcal{H}(f; \mathbf{x}) = \sum_{i=1}^{d} \left\| \frac{\partial^2 y_i}{\partial \mathbf{x}^2} \right\|_F^2, \tag{20}$$

where each $\dfrac{\partial^2 y_i}{\partial \mathbf{x}^2}$ is a Hessian matrix.

[7] The parallelogram may be degenerate. Given four points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$, if the sum of two points is the same with that of the other two, we regard they form a parallelogram.

[8] We use the new defined term LN-G rather than Group Normalization (GN) (Wu & He, 2018), considering that: 1) GN is defined on the convolutional input $\boldsymbol{X} \in \mathbb{R}^{d \times h \times w}$ but not on the input $\mathrm{x} \in \mathbb{R}^d$; 2) Given the sequential input (e.g., text) $\boldsymbol{X} \in \mathbb{R}^{d \times T}$ in Transformer/ViT, GN will share statistics over T by definition while LN-G will have no inter-sequence dependence and use separate statistics over T, like LN.

[9] For $\boldsymbol{y} = f(\mathbf{x})$, we require each $y_i (i = 1, \cdots, d)$ is twice differential about $\mathbf{x}$.

We use the Frobenius norm rather than the operator norm, For easier calculations. Note that $\mathcal{H}(f;\mathbf{x}) \geq 0$, and $\mathcal{H}(f;\mathbf{x}) = 0$ if and only if f is a linear function. We thus **assume** that the larger $\mathcal{H}(f;\mathbf{x})$ is, the more nonlinearity f contains.

Amplifying Nonliearity by Group. Denote $\psi_G(g;\cdot)$ as Group based LN (LN-G) on $\mathbb{R}^d$ with group number g, and $\psi_L(\cdot)$ as LN on $\mathbb{R}^d$. Compare LN with LN-G, the result is shown in Proposition 8.

Proposition 8. *Given $g \leq d/3$, we have*

$$\frac{\mathcal{H}(\psi_G(g;\cdot);\mathbf{x})}{\mathcal{H}(\psi_L(\cdot);\mathbf{x})} \geq 1. \tag{21}$$

Specifically, when $g = d/4$, we figure out that

$$\frac{\mathcal{H}(\psi_G(g;\cdot);\mathbf{x})}{\mathcal{H}(\psi_L(\cdot);\mathbf{x})} \geq \frac{d}{8}. \tag{22}$$

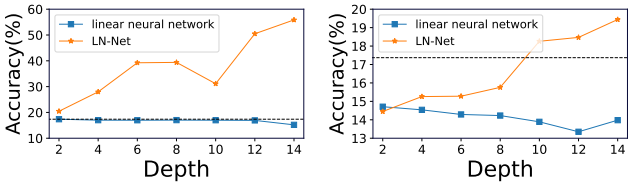
Proposition 8 shows that LN-G can amplify the nonliearity of LN by using appropriated group number. Compared with LN, when d is larger, LN-G shows more nonlineaity. Please refer to *Appendix E* for the proof. Besides, we generalize our discussion about $\mathcal{H}$ to the typical activation function ReLU, please refer to *Appendix E* for more details.

One limit of the result above is the assumption, that $\mathcal{H}(f;\mathbf{x})$ is a good indicator for measuring nonlinearity, is from the intuition and can not be well verified. In the subsequent experiments, we empirically show that LN-G indeed can amplify the nonlinearity of LN.

5.2. Comparison of Representation Capacity by Fitting Random Labels

In this part, we follow the non-parametric randomization tests fitting random labels (Zhang et al., 2017) to empirically verify the nonlinearity of LN, and to further compare the representational capacity of LN-Net with different groups for LN-G. The experiments are conducted on CIFAR-10 and MNIST with random label assigned (CIFAR-10-RL and MNIST-RL). We evaluate the classification accuracy on the training set after the model is trained, which indicates that the capacity of models in fitting dataset empirically. We only provide essential components of the experimental setup; for more details, please refer to the *Appendix F.1*.

Verify the Nonlinearity of LN. We conduct experiments on linear neural network and LN-Net with 256 neurons in each layer and various depths. We first train sufficiently a linear classifier and obtain the (nearly) upper bound accuracy (18.51 % on CIFAR-10 -RL and 15.38% on MNIST-RL). To rule out the influence in optimization difficulty, we train the linear neural network and LN-Net with various configurations, including different learning rates and (with or without) residual connection[10]. We report the best result from all configurations, as shown in Figure 4.



(a) CIFAR-10-RL. (b) MNIST-RL.

Figure 4. Results of linear neural network and LN-Net on fitting random label. The black dashed line represents the upper bound accuracy of linear classifier. (a) Results on CIFAR-10-RL; (b) Results on MNIST-RL.

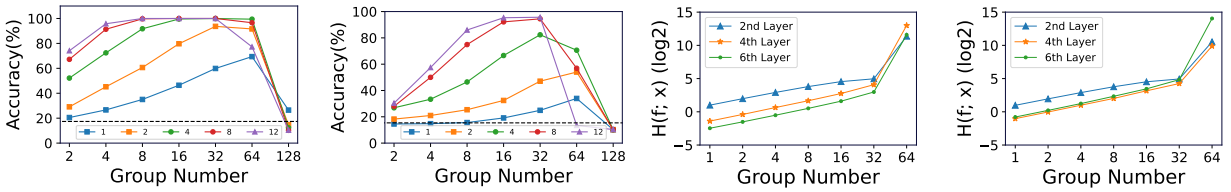
We observe that linear neural network cannot break the bound of linear classifier on all datasets, while LN-Net can reach the accuracy of 55.85% on CIFAR-10-RL and 19.44% on MNIST-RL, which is much better than the linear classifier. This result also verifies that LN has nonlinearity empirically. Besides, we observe that LN-Net obtains better performance in general as the depth increases (namely more LN layers and greater nonlinearity). We note that an LN-Net without sufficient depth does not break the bound of linear classifier on MNIST-RL. The reasons leading to this phenomenon are likely to be that: 1) MNIST-RL are more difficult to train, compare to CIFAR-10-RL; 2) LN-Nets have a non-convex optimization landscape and we cannot ensure the weight learned to be the optimal point, given fixed training epochs.

We also conduct experiments with Batch Normalization (BN) (Ioffe & Szegedy, 2015), where we replace LN with BN in LN-Net. We find that BN cannot break the bound of linear classifier on all datasets, like linear neural network. This preliminary result is interesting, which shows the potential advantage of LN over BN, in terms of the representation capacity.

Amplifying the Nonlinearity using Group. We conduct experiments on LN-Net with $d = 256$ neurons in each layer and various depths. We replace LN in LN-Net with LN-G and also vary the group number g. We train LN-Net with various learning rates and report the best training accuracy on CIFAR-10-RL and MNIST-RL, as shown in Figure 5.

We observe that some LN-Net with LN-G (e.g., depth = 8 and $g = 32$) can perfectly classify all the random labels on CIFAR-10-RL and MNIST-RL, which suggests that LN-G can amplify the nonlinearity of LN by using group, as stated in Proposition 8. We also observe that an LN-Net with appropriate group number (e.g, $g = 32$) can obtain

[10] A linear neural network with residual connection is still a linear model.



(a) Accuracy on CIFAR-10-RL. (b) Accuracy on MNIST-RL. (c) $\mathcal{H}(f;\mathbf{x})$ on CIFAR-10-RL. (d) $\mathcal{H}(f;\mathbf{x})$ on MNIST-RL.

Figure 5. Results of LN-Net using LN-G. We vary the group number g and show the training accuracy and $\mathcal{H}(f;\mathbf{x})$. (a) Training accuracy on CIFAR-10-RL; (b) Training accuracy on MNIST-RL; (c)$\mathcal{H}(f;\mathbf{x})$ on CIFAR-10-RL;(d) $\mathcal{H}(f;\mathbf{x})$ on MNIST-RL. The black dashed line in (a) and (b) has the same meaning as that in Figure 4.

better performance, as the depth increases. Besides, an LN-Net has better performance in general with larger group number, along the group number is not too much (relative to the number of neurons). E.g, An LN-Net has significantly degenerated performance when $g = 128$, due to $d/g = 2 < 3$ that go against the premise in Proposition 8.

We also calculate $\mathcal{H}(f;\mathbf{x})$ in certain layers and show how $\mathcal{H}(f;\mathbf{x})$ varies as the group number increases in Figure 5 (c) and (d). $\mathcal{H}(f;\mathbf{x})$ is calculated by averaging over 1000 samples in our experiments. We find $\mathcal{H}(f;\mathbf{x})$ increases as the group number of LN-G increases, which matches our theoretical analyses in Section 5.1.

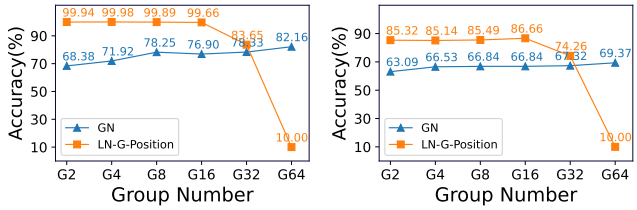
5.3. Inspiration for Neural Architecture Design

In this part, we consider designing neural networks in real scenarios, considering that LN-G can amplify the nonlinearity and have great performance in fitting the random label shown in Section 5.2. We conduct experiments on both CNN and Transformer architectures.

5.3.1. CNN WITHOUT ACTIVATION FUNCTION

To validate the representation capacity of LN-G in real scenarios further, we conducted experiments on CIFAR-10 using ResNet (He et al., 2016). To exclude the influence of other nonlinearities, we remove all nonlinear activations from the ResNet, and refer the network to ResNet-NA. We set the channel number of each layer to 128 for better ablating the group number of LN-G. We also conduct experiments on more CNNs shown in *Appendix F.2*

Investigation of LN-G. Note that LN-G may have several variants for a convolutional input $\boldsymbol{X} \in \mathbb{R}^{c\times h\times w}$, where c, h and w indicate the feature mappings' channel, height and width dimensions respectively. Following the usage of LN on CNNs, LN-G should calculate the mean/variance along all the channel, height and width dimensions, which is equivalent to Group Normalization (GN) (Wu & He, 2018). Following the usage of LN on MLP&Transformer, LN-G should calculate the mean/variance along only the channel dimension and use separate statistics over each position (a pair of height and width), and we refer to this method as LN-G-Position.



(a) Training. (b) Test.

Figure 6. Results of the variants of LN-G (GN and LN-G-Position) when using different group number. The experiments are conducted on CIFAR-10 dataset using ResNet without ReLU activation. We show (a) the training accuracy and (b) the test accuracy. In the x-axis, G2 refers to a group number of 2.

We investigate how the group number affects the performance of the variants of LN-G (GN and LN-G-Position). We vary the group number g ranging in {2, 4, 8, 16, 32, 64}. We train a total of 200 epochs using SGD with a mini-batch size of 128, momentum of 0.9 and weight decay of 0.0001. The initial learning rate is set to 0.1, and divided by 5 at the 60th, 120th, and 160th epochs. The results are shown in Figure 6. We find that GN obtains slightly better performance as the group number increases. Note that this observation does not go against the experimental results of LN-G in amplifying the nonlinearity in Section 5.2 since the 'effective samples' used to calculate the normalization statistics in each group of GN is $\frac{h*w*c}{g}$. We observe that LN-G-Position works particularly well and obtains over 85% test accuracy for multiple group number (Note that there is no ReLU activations.). We also find that LN-G-Position works particularly bad if group number is 64, because the samples used to calculate the normalization statistics in each group of LN-G-Position is $\frac{c}{g} = 2$.

Comparison to other Normalization. We also conduct experiments to train ResNet-NA by using other normalization methods, including the original Batch Normalization (BN) (Ioffe & Szegedy, 2015), Layer Normalization (LN) (Ba et al., 2016), Instance Normalization (IN) (Ulyanov et al., 2016). Besides, we also train ResNet-NA without normalization. We use the same setting up

Table 1. Comparison of different normalization methods on CIFAR-10 using ResNet-NA (ResNet without ReLU activation).

Normalization methods	Train Acc(%)	Test Acc(%)
IN	10	10
BN	36.0	39.3
LN	59.5	62.85
GN	82.16	69.37
LN-G-Position	99.66	86.66

described in previous experiments. We find that ResNet-NA without normalization is very difficult to train and shows a random guess behavior. Similarly, ResNet-NA with IN is also very difficult to train. ResNet-NA with BN can be trained normally. However, the performance of the model is relatively low, with only 39.3% test accuracy. ResNet-NA with LN obtains 62.85% test accuracy, which is significantly better than BN. Furthermore, ResNet-NA with LN-G-Position obtains the best performance, e.g., a test accuracy of 86.66% when using a group number 16 for LN-G-Position. We contribute it to the strong nonlinearity of LN-G-Position.

5.3.2. LN-G IN TRANSFORMERS

Transformer for Machine Translation. We conduct experiments to apply LN-G on Transformer (Vaswani et al., 2017) (where LN is the default normalization) for machine translation tasks using *fairseq-py* (Ott et al., 2019). We evaluate the public IWSLT14 German-to-English (De-EN) dataset using BLEU (higher is better). We use the hyperparameters recommended in *fairseq-py* (Ott et al., 2019) for Transformer and train over 50 epochs with five random seeds. The baseline LN has a BLEU score of 35.01 ± 0.10. LN-G (replacing all the LNs with LN-G) has a BLEU score of 35.23 ± 0.07.

ViT for Image Classification. We conducted experiments by applying LN-G to Tiny-VIT (with the default normalization being LN). We performed classification tests on the test set of the CIFAR-10 dataset, with hyperparameter settings referencing (Steiner et al., 2021). The classification accuracy on the test dataset was 88.81% for LN and 89.26% for LN-G (replacing all the LNs with LN-G).

These preliminary results show the potentiality of LN-G used for neural architecture design in practice.

6. Related Work

Previous theoretical analyses on normalization are mainly focused on BN, the pioneer work in normalization for deep learning. One main argument is that BN can improve the conditioning of the optimization problem (Cai et al., 2019), either by avoiding the rank collapse of pre-activation matrices (Daneshmand et al., 2020) or by alleviating the pathological sharpness of the landscape (Santurkar et al., 2018; Karakida et al., 2019; Ghorbani et al., 2019; Lyu et al., 2022). The improved conditioning enables large learning rates (Bjorck et al., 2018), thus improving the generalization (Luo et al., 2019). Another argument is that BN is scale invariant (Ba et al., 2016), enabling it to adaptively adjust the learning rate (Arora et al., 2019; Cai et al., 2019; Zhang et al., 2019; Li & Arora, 2020), which stabilizes and further accelerates training. This scale invariant analyses also applies to LN (Ba et al., 2016; Lubana et al., 2021). Some work address to understanding LN empirically through experiments, showing that the learnable parameters in LN increases the risk of over-fitting (Xu et al., 2019).

Different from these work, we investigate a new theoretical direction for LN, regarding to its nonlinearity and representation capacity. We note that there are several work (Huang et al., 2021; Labatie et al., 2021) investigating the expressive power of normalization empirically by experiments. However, their experiments are conducted on networks with activation functions, while our work focuses on analyzing the representation capacity of a network without activation functions through theory and experiment.

7. Conclusion

We mathematically demonstrated that LN is a nonlinear transformation. We also theoretically showed the representation capacity of an LN-Net in correctly classifying samples with any label assignment. We demonstrated these results by finely designing algorithms, considering the geometric property of LN. We hope that our techniques will inspire the community to reconsider the analyses of the representation capacity of a network with normalization layer, though it suffers from great challenges (Huang et al., 2023).

Limitation and Future Work. Our results in representation capacity for LN-Net is very loose currently, which is like the initial universal approximation theory in the arbitrary wide shallow neural network (Hornik et al., 1989). We believe it is interesting to extend our results along the direction as universal approximation theory is extended to the cases of arbitrary depth (Gripenberg, 2003), bounded depth and bounded width (Maiorov & Pinkus, 1999), and the question of minimal possible width (Park et al., 2020). Besides, the effectiveness of group mechanism for LN (*i.e.*, LN-G) is only verified on small-scale networks and datasets, and more results on large-scale networks and datasets are required to support the practicality of LN-G.

Acknowledgments

This work was partially supported by the National Science and Technology Major Project under Grant

2022ZD0116310, National Natural Science Foundation of China (Grant No. 62106012), the Fundamental Research Funds for the Central Universities.

Impact Statement

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

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

Arora, S., Li, Z., and Lyu, K. Theoretical analysis of auto rate-tuning by batch normalization. In *ICLR*, 2019.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization, 2016.

Bartlett, P., Maiorov, V., and Meir, R. Almost linear vc dimension bounds for piecewise polynomial networks. In *NeurIPS*, 1998.

Bjorck, N., Gomes, C. P., Selman, B., and Weinberger, K. Q. Understanding batch normalization. In *NeurIPS*, 2018.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *NeurIPS*, 2020.

Cai, Y., Li, Q., and Shen, Z. A quantitative analysis of the effect of batch normalization on gradient descent. In *ICML*, 2019.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *ECCV*, 2020.

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*, 2019.

Daneshmand, H., Kohler, J. M., Bach, F. R., Hofmann, T., and Lucchi, A. Batch normalization provably avoids ranks collapse for randomly initialised deep networks. In *NeurIPS*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Fisher, R. A. Statistical methods for research workers. In *Breakthroughs in statistics: Methodology and distribution*, pp. 66–70. Springer, 1970.

Ghorbani, B., Krishnan, S., and Xiao, Y. An investigation into neural net optimization via hessian eigenvalue density. In *ICML*, 2019.

Gripenberg, G. Approximation by neural networks with a bounded number of nodes at each level. *Journal of approximation theory*, 122(2):260–266, 2003.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hoffer, E., Banner, R., Golan, I., and Soudry, D. Norm matters: efficient and accurate normalization schemes in deep networks. In *NeurIPS*, 2018.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Huang, L., Zhou, Y., Liu, L., Zhu, F., and Shao, L. Group whitening: Balancing learning efficiency and representational capacity. In *CVPR*, 2021.

Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., and Shao, L. Normalization techniques in training dnns: Methodology, analysis and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Karakida, R., Akaho, S., and Amari, S.-i. The normalization method for alleviating pathological sharpness in wide neural networks. In *NeurIPS*, 2019.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollar, P., and Girshick, R. Segment anything. In *ICCV*, 2023.

Labatie, A., Masters, D., Eaton-Rosen, Z., and Luschi, C. Proxy-normalizing activations to match batch normalization while removing batch dependence. In *NeurIPS*, 2021.

Li, Z. and Arora, S. An exponential learning rate schedule for batch normalized networks. In *ICLR*, 2020.

Lubana, E. S., Dick, R., and Tanaka, H. Beyond batchnorm: towards a unified understanding of normalization in deep learning. In *NeurIPS*, 2021.

Luo, P., Wang, X., Shao, W., and Peng, Z. Towards understanding regularization in batch normalization. In *ICLR*, 2019.

Lyu, K., Li, Z., and Arora, S. Understanding the generalization benefit of normalization layers: Sharpness reduction. In *NeurIPS*, 2022.

Maiorov, V. and Pinkus, A. Lower bounds for approximation by mlp neural networks. *Neurocomputing*, 25(1-3):81–91, 1999.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. fairseq: A fast, extensible toolkit for sequence modeling. In *ACL*, 2019.

Park, S., Yun, C., Lee, J., and Shin, J. Minimum width for universal approximation. *arXiv preprint arXiv:2006.08859*, 2020.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. How does batch normalization help optimization? In *NeurIPS*, 2018.

Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. S. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Wu, Y. and He, K. Group normalization. In *ECCV*, 2018.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T.-Y. On layer normalization in the transformer architecture. In *ICML*, 2020.

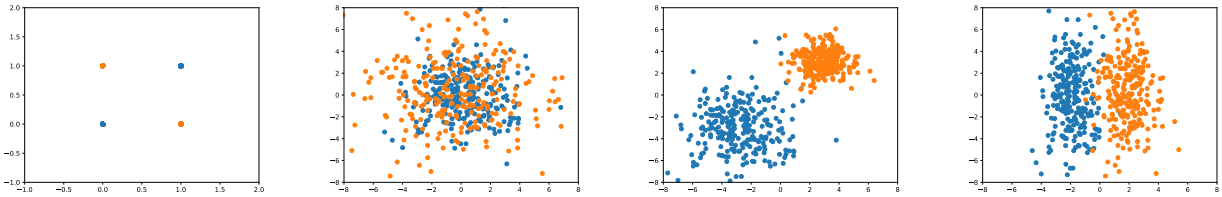
Xu, J., Sun, X., Zhang, Z., Zhao, G., and Lin, J. Understanding and improving layer normalization. In *NeurIPS*, 2019.

Zhang, B. and Sennrich, R. Root mean square layer normalization. In *NeurIPS*, 2019.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Zhang, G., Wang, C., Xu, B., and Grosse, R. B. Three mechanisms of weight decay regularization. In *ICLR*, 2019.

A. LSSR as a Linearly Separable Indicator



(a) XOR data. (b) Inseparable Gaussian data. (c) Separable Gaussian data. (d) Parallel Gaussian data.

Figure A1. We randomly sample 256 points from each class in the four different distributions of data above. The detailed data is shown in Table I.

To show how SSR and LSSR evaluate the difficulty of separating the samples from different classes linearly, we give four different distributions of data in the figure above and their details in the table below.

Table I. Detailed data of Figure A1. In Figure 1(a), the random variance X takes values 0 and 1 with probabilities $1/2$ each. In the other figures, the sign $N(\cdot,\cdot)$ denotes the Gaussian distribution.

Figure	Distribution of X_1	Distribution of X_2	SSR	LSSR
Figure 1(a)	$X_1 = \begin{bmatrix} X \\ X \end{bmatrix}$	$X_2 = \begin{bmatrix} X \\ 1-X \end{bmatrix}$	0.9963	0.9929
Figure 1(b)	$X_1 \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$	$X_2 \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}\right)$	0.9929	0.9859
Figure 1(c)	$X_1 \sim N\left(\begin{bmatrix} -3 \\ -3 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$	$X_2 \sim N\left(\begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$	0.2304	0.1312
Figure 1(d)	$X_1 \sim N\left(\begin{bmatrix} -2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 9 \end{bmatrix}\right)$	$X_2 \sim N\left(\begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 9 \end{bmatrix}\right)$	0.7536	0.2157

According to Figure A1 and Table I, we have several conclusions below. In Figure 1(a) and Figure 1(b), the classes are hard to be linearly separated, whose SSR and LSSR are both near 1. In Figure 1(c), the classes are easy to be linearly separated, whose SSR and LSSR are both near 0. However, in Figure 1(d), the classes are easy to be linearly separated, but harder to be separated if focused on the Euclidean distance. As a result, its SSR is larger, but its LSSR is near 0. We hence conclude that—LSSR is a better indicator than SSR in judging how two classes are linearly separable.

B. Proofs of Proposition 2

Proposition 2. *Given $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbb{R}^{d\times m}$, we denote $\boldsymbol{M} = \sum_{c=1}^{2}\sum_{i=1}^{m}(\mathbf{x}_{ci}-\bar{\mathbf{x}}_c)(\mathbf{x}_{ci}-\bar{\mathbf{x}}_c)^\top$, and $\boldsymbol{N} = \sum_{c=1}^{2}\sum_{i=1}^{m}(\mathbf{x}_{ci}-\bar{\mathbf{x}})(\mathbf{x}_{ci}-\bar{\mathbf{x}})^\top$, where $\bar{\mathbf{x}} = (\bar{\mathbf{x}}_1+\bar{\mathbf{x}}_2)/2$. Supposing that $\boldsymbol{N}$ is reversible, we have*

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = \lambda^*, \tag{23}$$

and correspondingly,

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = SSR((\boldsymbol{u}^*)^\top \boldsymbol{X}_1, (\boldsymbol{u}^*)^\top \boldsymbol{X}_2), \tag{24}$$

where λ^* and $\boldsymbol{u}^*$ are the minimum eigenvalue and corresponding eigenvector of $\boldsymbol{N}^{-1}\boldsymbol{M}$.

Since the definition of LSSR comes from a lower bound, we prove Proposition 2 from solving the optimization problem as follows.

$$(P_{LSSR}) \begin{cases} \min\limits_{\varphi} & LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = SSR(\varphi(\boldsymbol{X}_1), \varphi(\boldsymbol{X}_2)), \\ \text{s.t.} & \varphi(\mathbf{x}) = \boldsymbol{W}\mathbf{x} + \boldsymbol{b}, \\ & \boldsymbol{W} \in \mathbb{R}^{n\times d}, \boldsymbol{b} \in \mathbb{R}^n, n \in \mathbb{N}^*, \\ & SS(\varphi(\boldsymbol{X}_1), \varphi(\boldsymbol{X}_2)) \neq 0. \end{cases} \tag{25}$$

To solve this, we first propose four lemmas, and then use them to prove Proposition 2. Furthermore, we give the optimal $\boldsymbol{W}$ as a corollary.

B.1. Required Lemmas for the Proof

Lemma 2. *The bias $\boldsymbol{b} \in \mathbb{R}^n$ does not affect SSR, as well as LSSR.*

Proof. By the definition of SS, we obtain

$$\begin{aligned} SS(\varphi(\boldsymbol{X}_1)) &= SS(\boldsymbol{W}\boldsymbol{X}_1 + \boldsymbol{b}\mathbf{1}^\top) \\ &= \sum_{i=1}^{m} \left\| \boldsymbol{W}\mathbf{x}_{1i} + \boldsymbol{b} - \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{W}\mathbf{x}_{1i} + \boldsymbol{b}) \right\|^2 \\ &= \sum_{i=1}^{m} \left\| \boldsymbol{W}\mathbf{x}_{1i} - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{W}\mathbf{x}_{1i} \right\|^2 \\ &= \sum_{i=1}^{m} \left\| \boldsymbol{W}\mathbf{x}_{1i} - \overline{\boldsymbol{W}\mathbf{x}_1} \right\|^2 \\ &= SS(\boldsymbol{W}\boldsymbol{X}_1). \end{aligned} \tag{26}$$

Similarly, we have

$$SS(\varphi(\boldsymbol{X}_2)) = SS(\boldsymbol{W}\boldsymbol{X}_2), \tag{27}$$

and

$$SS([\varphi(\boldsymbol{X}_1), \varphi(\boldsymbol{X}_2)]) = SS([\boldsymbol{W}\boldsymbol{X}_1, \boldsymbol{W}\boldsymbol{X}_2]). \tag{28}$$

Since SSR is defined with SS, the conclusion also holds for SSR, namely

$$SSR(\varphi(\boldsymbol{X}_1), \varphi(\boldsymbol{X}_2)) = SSR(\boldsymbol{W}\boldsymbol{X}_1, \boldsymbol{W}\boldsymbol{X}_2), \tag{29}$$

where the bias $\boldsymbol{b}$ is not included. □

Lemma 3. *Suppose the eigenvalue decomposition of $\boldsymbol{W}^\top\boldsymbol{W}$ as*

$$\boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{U}\Lambda\boldsymbol{U}^\top = \sum_{i=1}^{d}\lambda_i\boldsymbol{u}_i\boldsymbol{u}_i^\top, \tag{30}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_d]$ is an orthogonal matrix, and $\Lambda = diag\{\lambda_1, \cdots, \lambda_d\}$ is a positive semi-definite and diagonal matrix. We consider to minimize $SSR(\boldsymbol{W}\boldsymbol{X}_1, \boldsymbol{W}\boldsymbol{X}_2)$ over Λ with a fixed $\boldsymbol{U}$, as:

$$\min_{\Lambda} SSR(\boldsymbol{W}\boldsymbol{X}_1, \boldsymbol{W}\boldsymbol{X}_2) = \min_{1\le j\le d} SSR(\boldsymbol{u}_j^\top\boldsymbol{X}_1, \boldsymbol{u}_j^\top\boldsymbol{X}_2). \tag{31}$$

The optimal solution is that

$$\lambda_j \begin{cases} \ge 0, & j \in \arg\min_{1\le j\le d} SSR(\boldsymbol{u}_j^\top\boldsymbol{X}_1, \boldsymbol{u}_j^\top\boldsymbol{X}_2), \\ = 0, & \textit{otherwise}, \end{cases} \tag{32}$$

for $j = 1, \cdots, d$, and $\lambda_1, \cdots, \lambda_d$ are not all zeros.

Proof. By Lemma 2, we find that

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = \min_{\boldsymbol{W}} SSR(\boldsymbol{W}\boldsymbol{X}_1, \boldsymbol{W}\boldsymbol{X}_2). \tag{33}$$

Besides, we figure out that

$$SS(\boldsymbol{W}\boldsymbol{X}_c) = \sum_{i=1}^{m}\|\boldsymbol{W}\mathbf{x}_{ci} - \boldsymbol{W}\bar{\mathbf{x}}_c\|_2^2 = \sum_{i=1}^{m}(\mathbf{x}_{ci} - \bar{\mathbf{x}}_c)^\top\boldsymbol{W}^\top\boldsymbol{W}(\mathbf{x}_{ci} - \bar{\mathbf{x}}_c), \tag{34}$$

where $\boldsymbol{X}_c = \boldsymbol{X}_1, \boldsymbol{X}_2$, or even[11] $[\boldsymbol{X}_1, \boldsymbol{X}_2]$.

[11] In this case, we choose $\bar{\mathbf{x}}$ as $\bar{\mathbf{x}}_c$ in Eqn.34

Based on the eigenvalue decomposition, we obtain that

$$\begin{aligned} SS(\boldsymbol{W}\boldsymbol{X}_1) &= \sum_{i=1}^{m}(\mathbf{x}_{1i}-\bar{\mathbf{x}}_1)^\top \boldsymbol{W}^\top \boldsymbol{W}(\mathbf{x}_{1i}-\bar{\mathbf{x}}_1) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{d}\lambda_j(\mathbf{x}_{1i}-\bar{\mathbf{x}}_1)^\top \boldsymbol{u}_j\boldsymbol{u}_j^\top(\mathbf{x}_{1i}-\bar{\mathbf{x}}_1) \\ &= \sum_{j=1}^{d}\lambda_j\sum_{i=1}^{m}(\boldsymbol{u}_j^\top\mathbf{x}_{1i}-\boldsymbol{u}_j^\top\bar{\mathbf{x}}_1)^2 \\ &= \sum_{j=1}^{d}\lambda_j SS(\boldsymbol{u}_j^\top\boldsymbol{X}_1). \end{aligned} \tag{35}$$

The term $SS(\boldsymbol{u}_j^\top\boldsymbol{X}_1)$ can be regarded as that we put a linear transformation $\boldsymbol{u}_j^\top$ on $\boldsymbol{X}_1$, and then calculate its SS.

Similarly, we have that

$$SS(\boldsymbol{W}\boldsymbol{X}_2) = \sum_{j=1}^{d}\lambda_j SS(\boldsymbol{u}_j^\top\boldsymbol{X}_2), \tag{36}$$

and

$$SS([\boldsymbol{W}\boldsymbol{X}_1,\boldsymbol{W}\boldsymbol{X}_2]) = \sum_{j=1}^{d}\lambda_j SS([\boldsymbol{u}_j^\top\boldsymbol{X}_1,\boldsymbol{u}_j^\top\boldsymbol{X}_2]). \tag{37}$$

Therefore, we obtain

$$SSR(\boldsymbol{W}\boldsymbol{X}_1,\boldsymbol{W}\boldsymbol{X}_2) = \frac{\sum_{j=1}^{d}\lambda_j[SS(\boldsymbol{u}_j^\top\boldsymbol{X}_1)+SS(\boldsymbol{u}_j^\top\boldsymbol{X}_2)]}{\sum_{j=1}^{d}\lambda_j SS([\boldsymbol{u}_j^\top\boldsymbol{X}_1,\boldsymbol{u}_j^\top\boldsymbol{X}_2])}. \tag{38}$$

By the definition of $\boldsymbol{M}$ and $\boldsymbol{N}$ in Proposition 2, we obtain that

$$\begin{aligned} SS(\boldsymbol{u}_j^\top\boldsymbol{X}_1)+SS(\boldsymbol{u}_j^\top\boldsymbol{X}_2) &= \sum_{i=1}^{m}[(\boldsymbol{u}_j^\top\mathbf{x}_{1i}-\boldsymbol{u}_j^\top\bar{\mathbf{x}}_1)^2+(\boldsymbol{u}_j^\top\mathbf{x}_{2i}-\boldsymbol{u}_j^\top\bar{\mathbf{x}}_2)^2] \\ &= \sum_{i=1}^{m}\boldsymbol{u}_j^\top[(\mathbf{x}_{1i}-\bar{\mathbf{x}}_1)(\mathbf{x}_{1i}-\bar{\mathbf{x}}_1)^\top+(\mathbf{x}_{2i}-\bar{\mathbf{x}}_2)(\mathbf{x}_{2i}-\bar{\mathbf{x}}_2)^\top]\boldsymbol{u}_j \\ &= \boldsymbol{u}_j^\top\boldsymbol{M}\boldsymbol{u}_j, \end{aligned} \tag{39}$$

and similarly, we have

$$SS([\boldsymbol{u}_j^\top\boldsymbol{X}_1,\boldsymbol{u}_j^\top\boldsymbol{X}_2]) = \boldsymbol{u}_j^\top\boldsymbol{N}\boldsymbol{u}_j. \tag{40}$$

By the hypothesis in Definition 2, we figure out that $\lambda_j(j=1,\cdots,d)$ are not all zeros, otherwise $SS([\boldsymbol{W}\boldsymbol{X}_1,\boldsymbol{W}\boldsymbol{X}_2])=0$. Besides, by the hypothesis in Proposition 2, $\boldsymbol{N}$ is reversible. We point out that $\boldsymbol{N}$ is also a positive semi-definite matrix. Furthermore, $\boldsymbol{N}$ is a positive definite matrix. When $\boldsymbol{u}_j\neq\mathbf{0}$, we find that

$$SS([\boldsymbol{u}_j^\top\boldsymbol{X}_1,\boldsymbol{u}_j^\top\boldsymbol{X}_2]) = \boldsymbol{u}_j^\top\boldsymbol{N}\boldsymbol{u}_j > 0. \tag{41}$$

Let $\eta_j = \lambda_j SS([\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2])$, we thus have $\eta_1 + \cdots + \eta_d > 0$. We obtain

$$\begin{aligned} SSR(\boldsymbol{W}\boldsymbol{X}_1, \boldsymbol{W}\boldsymbol{X}_2) =& \frac{1}{\eta_1 + \cdots + \eta_d} \sum_{j=1}^{d} \frac{\eta_j [SS(\boldsymbol{u}_j^\top \boldsymbol{X}_1) + SS(\boldsymbol{u}_j^\top \boldsymbol{X}_2)]}{SS([\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2])} \\ =& \sum_{j=1}^{d} \frac{\eta_j}{\eta_1 + \cdots + \eta_d} SSR(\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2) \\ \geq& \sum_{j=1}^{d} \frac{\eta_j}{\eta_1 + \cdots + \eta_d} \min_{1 \leq k \leq d} SSR(\boldsymbol{u}_k^\top \boldsymbol{X}_1, \boldsymbol{u}_k^\top \boldsymbol{X}_2) \\ =& \min_{1 \leq k \leq d} SSR(\boldsymbol{u}_k^\top \boldsymbol{X}_1, \boldsymbol{u}_k^\top \boldsymbol{X}_2) \\ =& \min_{1 \leq j \leq d} SSR(\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2). \end{aligned} \tag{42}$$

We figure out that the equation holds, if and only if

$$\eta_j \begin{cases} \geq 0, & j \in \arg\min_{1 \leq j \leq d} SSR(\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2); \\ = 0, & \text{otherwise.} \end{cases} \tag{43}$$

Here, $j = 1, \cdots, d$, and $\eta_1, \cdots, \eta_d$ are not all zeros.

Since $\lambda_j = \eta_j / SS([\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2])$ and $SS([\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2]) > 0$, we thus have

$$\lambda_j \begin{cases} \geq 0, & j \in \arg\min_{1 \leq j \leq d} SSR(\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2), \\ = 0, & \text{otherwise,} \end{cases} \tag{44}$$

holds for $j = 1, \cdots, d$, and $\lambda_1, \cdots, \lambda_d$ are not all zeros. □

Lemma 4. *Given $\mathbb{D}_{\boldsymbol{v}} = \{\boldsymbol{v} : \boldsymbol{v}^\top \boldsymbol{N} \boldsymbol{v} = 1\}$ and $\mathbb{D}_{\boldsymbol{u}} = \{\boldsymbol{u} : \boldsymbol{u}^\top \boldsymbol{u} = 1\}$ and the map $\psi : \mathbb{D}_{\boldsymbol{v}} \to \mathbb{D}_{\boldsymbol{u}}$, where $\boldsymbol{u} = \psi(\boldsymbol{v}) = \boldsymbol{v}/(\boldsymbol{v}^\top \boldsymbol{v})^{\frac{1}{2}}$, we have that ψ is a bijection.*

Proof. For $\boldsymbol{N}$ is a positive definite matrix , and $\boldsymbol{v}^\top \boldsymbol{N} \boldsymbol{v} = 1$, we have $\boldsymbol{v} \neq \boldsymbol{0}$. Given $\boldsymbol{u} = \psi(\boldsymbol{v})$, we obtain

$$\boldsymbol{u}^\top \boldsymbol{u} = \boldsymbol{v}^\top \boldsymbol{v} / (\boldsymbol{v}^\top \boldsymbol{v}) = 1, \tag{45}$$

for each $\boldsymbol{v}$ in $\mathbb{D}_{\boldsymbol{v}}$.

Therefore, ψ is a reflection from $\mathbb{D}_{\boldsymbol{v}}$ to $\mathbb{D}_{\boldsymbol{u}}$. Besides, we find that

$$\boldsymbol{u}^\top \boldsymbol{N} \boldsymbol{u} = \boldsymbol{v}^\top \boldsymbol{N} \boldsymbol{v} / (\boldsymbol{v}^\top \boldsymbol{v}) = 1/(\boldsymbol{v}^\top \boldsymbol{v}). \tag{46}$$

By the definition of ψ, we hence have

$$\boldsymbol{u}/(\boldsymbol{u}^\top \boldsymbol{N} \boldsymbol{u})^{\frac{1}{2}} = \psi(\boldsymbol{v})(\boldsymbol{v}^\top \boldsymbol{v})^{\frac{1}{2}} = \boldsymbol{v}. \tag{47}$$

Therefore, for each $\boldsymbol{u}$, we obtain that

$$\boldsymbol{v} = \psi^{-1}(\boldsymbol{u}) = \boldsymbol{u}/(\boldsymbol{u}^\top \boldsymbol{N} \boldsymbol{u})^{\frac{1}{2}}, \tag{48}$$

namely we find ψ^{-1} as the inverse mapping of ψ.

As a result, ψ is a bijection. □

Lemma 5. *Let the optimization problem be*

$$(P_{\boldsymbol{v}}) \begin{cases} \min\limits_{\boldsymbol{v}} & f(\boldsymbol{v}) = \dfrac{\boldsymbol{v}^\top \boldsymbol{M} \boldsymbol{v}}{\boldsymbol{v}^\top \boldsymbol{N} \boldsymbol{v}}, \\ s.t. & \boldsymbol{v}^\top \boldsymbol{N} \boldsymbol{v} = 1, \end{cases} \tag{49}$$

where $\boldsymbol{M}$ and $\boldsymbol{N}$ are defined in Proposition 2. We have that the optimal value is the minimal eigenvalue of $\boldsymbol{N}^{-1}\boldsymbol{M}$, namely λ^*. And the optimal solution is the eigenvector which belongs to λ^*.

Proof. To get the minimum, we use the Lagrange multiplier method:

$$L(\boldsymbol{v}, \alpha) = \boldsymbol{v}^\top \boldsymbol{M}\boldsymbol{v} - \alpha(\boldsymbol{v}^\top \boldsymbol{N}\boldsymbol{v} - 1). \tag{50}$$

We figure out that the KKT conditions are

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{v}} = 2\boldsymbol{M}\boldsymbol{v} - 2\alpha\boldsymbol{N}\boldsymbol{v} = \boldsymbol{0}, \\ \boldsymbol{v}^\top \boldsymbol{N}\boldsymbol{v} - 1 = 0. \end{cases} \tag{51}$$

We hence have $\boldsymbol{N}^{-1}\boldsymbol{M}\boldsymbol{v} = \alpha\boldsymbol{v}$, namely α is an eigenvalue of $\boldsymbol{N}^{-1}\boldsymbol{M}$, and $\boldsymbol{v}$ is the corresponding eigenvector. Based above, we find that

$$\boldsymbol{v}^\top \boldsymbol{M}\boldsymbol{v} = \boldsymbol{v}^\top \boldsymbol{N}\boldsymbol{N}^{-1}\boldsymbol{M}\boldsymbol{v} = \boldsymbol{v}^\top \boldsymbol{N}(\alpha\boldsymbol{v}) = \alpha. \tag{52}$$

Furthermore, the minimum of $L(\boldsymbol{v}, \alpha)$ is the minimum α, namely the minimum eigenvalue of $\boldsymbol{N}^{-1}\boldsymbol{M}$.

We hence have

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = \lambda_{min}(\boldsymbol{N}^{-1}\boldsymbol{M}) = \lambda^*, \tag{53}$$

and the optimal solution is the eigenvector which belongs to λ^*. □

B.2. Proof of Proposition 2

Based on the four lemmas above, now we give the proof of Proposition 2.

Proof. By Lemma 2 and Lemma 3, given $\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{U}\Lambda\boldsymbol{U}^\top$, we have

$$\begin{aligned} LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) &= \min_{\boldsymbol{W}} SSR(\boldsymbol{W}\boldsymbol{X}_1, \boldsymbol{W}\boldsymbol{X}_2) \\ &= \min_{\boldsymbol{U},\Lambda} SSR(\boldsymbol{W}\boldsymbol{X}_1, \boldsymbol{W}\boldsymbol{X}_2) \\ &= \min_{\boldsymbol{U}} \min_{\Lambda} SSR(\boldsymbol{W}\boldsymbol{X}_1, \boldsymbol{W}\boldsymbol{X}_2) \\ &= \min_{\boldsymbol{U}} \min_{1\le j\le d} SSR(\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2). \end{aligned} \tag{54}$$

According to Eqn.39 and Eqn.40, we define the function $f(\boldsymbol{u}) = SSR(\boldsymbol{u}^\top \boldsymbol{X}_1, \boldsymbol{u}^\top \boldsymbol{X}_2)$, namely

$$f(\boldsymbol{u}) = \frac{\boldsymbol{u}^\top \boldsymbol{M}\boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{N}\boldsymbol{u}}. \tag{55}$$

By Eqn.54, there is some $\boldsymbol{U}$, and $j^* = \arg\min_{1\le j\le d} SSR(\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2)$, such that

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = \min_{1\le j\le d} SSR(\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2) = f(\boldsymbol{u}_{j^*}). \tag{56}$$

Consider the optimization problem

$$(P_{\boldsymbol{u}}) \begin{cases} \min\limits_{\boldsymbol{u}} & f(\boldsymbol{u}) = \dfrac{\boldsymbol{u}^\top \boldsymbol{M}\boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{N}\boldsymbol{u}}, \\ s.t. & \boldsymbol{u}^\top \boldsymbol{u} = 1. \end{cases} \tag{57}$$

We denote one of the optimal solutions as $\bar{\boldsymbol{u}}$. Obviously, $\boldsymbol{u}_{j^*}$ is one of the feasible solutions above, we thus have

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = f(\boldsymbol{u}_{j^*}) \ge f(\bar{\boldsymbol{u}}). \tag{58}$$

We remind that λ^* and $\boldsymbol{u}^*$ are the minimum eigenvalue and corresponding eigenvector of $\boldsymbol{N}^{-1}\boldsymbol{M}$. On one hand, let $\boldsymbol{u}_0 = \boldsymbol{u}^*/\|\boldsymbol{u}^*\|_2$ and $\boldsymbol{v}_0 = \psi^{-1}(\boldsymbol{u}_0)$ (ψ is defined the same as that in Lemma 4), namely $\boldsymbol{v}_0 = \boldsymbol{u}_0/(\boldsymbol{u}_0^\top \boldsymbol{N}\boldsymbol{u}_0)^{\frac{1}{2}}$. We first

point out that for $k \neq 0$, we have

$$\begin{aligned} f(k\boldsymbol{u}) &= \frac{(k\boldsymbol{u})^\top \boldsymbol{M}(k\boldsymbol{u})}{(k\boldsymbol{u})^\top \boldsymbol{N}(k\boldsymbol{u})} \\ &= \frac{\boldsymbol{u}^\top \boldsymbol{M}\boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{N}\boldsymbol{u}} \\ &= f(\boldsymbol{u}). \end{aligned} \tag{59}$$

Since $1/||\boldsymbol{u}^*||_2 \neq 0$ and $1/(\boldsymbol{u}_0^\top \boldsymbol{N}\boldsymbol{u}_0)^{\frac{1}{2}} \neq 0$, we obtain

$$\begin{aligned} f(\boldsymbol{v}_0) = f(\boldsymbol{u}_0) &= f(\boldsymbol{u}^*) \\ &= \frac{(\boldsymbol{u}^*)^\top \boldsymbol{M}(\boldsymbol{u}^*)}{(\boldsymbol{u}^*)^\top \boldsymbol{N}(\boldsymbol{u}^*)} \\ &= \frac{(\boldsymbol{u}^*)^\top \boldsymbol{N}\boldsymbol{N}^{-1}\boldsymbol{M}(\boldsymbol{u}^*)}{(\boldsymbol{u}^*)^\top \boldsymbol{N}(\boldsymbol{u}^*)} \\ &= \frac{(\boldsymbol{u}^*)^\top \boldsymbol{N}(\lambda^*\boldsymbol{u}^*)}{(\boldsymbol{u}^*)^\top \boldsymbol{N}(\boldsymbol{u}^*)} \\ &= \lambda^*, \end{aligned} \tag{60}$$

where λ^* is the minimal eigenvalue of $\boldsymbol{N}^{-1}\boldsymbol{M}$, as shown in Proposition 2.

Therefore, by Lemma 5, $\boldsymbol{v}_0$ is the optimal solution of $(P_{\boldsymbol{v}})$. Furthermore, by Lemma 4, since ψ is a bijection between $\mathbb{D}_{\boldsymbol{v}}$ and $\mathbb{D}_{\boldsymbol{u}}$ and $f(\psi(\boldsymbol{v})) = f(\boldsymbol{v})$, we have that $\boldsymbol{u}_0 = \psi(\boldsymbol{v}_0)$ is also the optimal solution of $(P_{\boldsymbol{u}})$. We hence have $f(\boldsymbol{u}^*) = f(\boldsymbol{u}_0) = f(\bar{\boldsymbol{u}})$. By Eqn.58, we obtain

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) \geq f(\bar{\boldsymbol{u}}) = f(\boldsymbol{u}^*) = SSR((\boldsymbol{u}^*)^\top \boldsymbol{X}_1, (\boldsymbol{u}^*)^\top \boldsymbol{X}_2). \tag{61}$$

On the other hand, the definition of LSSR denotes the lower bound of SSR, we hence have

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) \leq SSR((\boldsymbol{u}^*)^\top \boldsymbol{X}_1, (\boldsymbol{u}^*)^\top \boldsymbol{X}_2). \tag{62}$$

By Eqn.60, Eqn.61 and Eqn.62, we obtain

$$LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) = SSR((\boldsymbol{u}^*)^\top \boldsymbol{X}_1, (\boldsymbol{u}^*)^\top \boldsymbol{X}_2) = \lambda^*. \tag{63}$$

□

B.3. Corollaries of Proposition 2

Suppose λ^* is the minimal eigenvalue of $\boldsymbol{N}^{-1}\boldsymbol{M}$, and $\boldsymbol{u}^*$ is its unique linearly independent eigenvector, we give the result in Corollary 4. If λ^* has more than one linearly independent eigenvectors, we give the result in Corollary 5.

Corollary 4. *Suppose λ^* is the minimal eigenvalue of $\boldsymbol{N}^{-1}\boldsymbol{M}$, and $\boldsymbol{u}^*$ is its unique linearly independent eigenvector, we have that the optimal $\boldsymbol{W}$ satisfies that*

$$\boldsymbol{W}^\top \boldsymbol{W} = C\boldsymbol{u}^*(\boldsymbol{u}^*)^\top, C > 0. \tag{64}$$

Proof. By Lemma 2, we can only consider the eigenvalues and eigenvectors of $\boldsymbol{W}^\top \boldsymbol{W}$. By Eqn.30, $\boldsymbol{u}_j$ will affect $\boldsymbol{W}^\top \boldsymbol{W}$ only when the corresponding eigenvalue $\lambda_j \neq 0$. Furthermore, by Lemma 3, when $\lambda_j \neq 0$, we figure out that

$$j \in \arg\min_{1 \leq j \leq d} SSR(\boldsymbol{u}_j^\top \boldsymbol{X}_1, \boldsymbol{u}_j^\top \boldsymbol{X}_2). \tag{65}$$

Since $\boldsymbol{u}_j$ is a unit vector, it must be one of the optimal solutions of $(P_{\boldsymbol{u}})$. We hence have $f(\boldsymbol{u}_j) = \lambda^*$. By Eqn.59 and Lemma 4, we have

$$f(\psi^{-1}(\boldsymbol{u}_j)) = f(\boldsymbol{u}_j) = \lambda^*, \tag{66}$$

and $\psi^{-1}(\boldsymbol{u}_j)$ is one of the optimal solutions of $(P_{\boldsymbol{v}})$. By Lemma 5, we have that $\psi^{-1}(\boldsymbol{u}_j)$ must satisfy the KKT conditions in Eqn.51, namely

$$\boldsymbol{N}^{-1}\boldsymbol{M}\psi^{-1}(\boldsymbol{u}_j) = \lambda^*\psi^{-1}(\boldsymbol{u}_j). \tag{67}$$

Therefore, $\psi^{-1}(\boldsymbol{u}_j)$ is the minimal eigenvector of $\boldsymbol{M}^{-1}\boldsymbol{N}$. For $\psi^{-1}(\boldsymbol{u}_j) = \boldsymbol{u}_j/(\boldsymbol{u}_j^\top \boldsymbol{N}\boldsymbol{u}_j)^{\frac{1}{2}}$, we obtain $\boldsymbol{u}_j$ is also the eigenvector of λ^*. For $\boldsymbol{u}^*$ is the unique linearly independent eigenvector of λ^*, we figure out that

$$\boldsymbol{u}_j = \alpha_j \boldsymbol{u}^*, \alpha_j \neq 0. \tag{68}$$

To be reminded, Eqn.68 only holds when $\lambda_j \neq 0$. Therefore, we have

$$\begin{aligned}
\boldsymbol{W}^\top\boldsymbol{W} &= \sum_{j=1}^{d} \lambda_j \boldsymbol{u}_j\boldsymbol{u}_j^\top \\
&= \sum_{\lambda_j\neq 0} \lambda_j\alpha_j^2 \boldsymbol{u}^*(\boldsymbol{u}^*)^\top \\
&= C\boldsymbol{u}^*(\boldsymbol{u}^*)^\top,
\end{aligned} \tag{69}$$

where $C = \sum\limits_{\lambda_j\neq 0} \lambda_j\alpha_j^2$.

By Lemma 3, we have $\lambda_1, \cdots, \lambda_j \geq 0$ are not all zeros. Besides, we figure out that α_j can be any non-zero real number. We thus have C can be any non-zero real number, to demonstrate $\boldsymbol{W}^\top\boldsymbol{W}$. □

Corollary 5. *Suppose that the minimal eigenvalue of $\boldsymbol{N}^{-1}\boldsymbol{M}$, namely λ^*, has k linearly independent eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k$. We denote $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k]$, then we have that the optimal $\boldsymbol{W}$ satisfies that*

$$\boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{V}\boldsymbol{C}\boldsymbol{V}^\top, \tag{70}$$

where $\boldsymbol{C}$ is a k-order semi-positive definite and non-zero matrix.

Proof. Suppose λ_j is an eigenvalue of $\boldsymbol{W}^\top\boldsymbol{W}$, and its eigenvector is $\boldsymbol{u}_j$. We can identify that $j \in \arg\min\limits_{1\leq j\leq d} SSR(\boldsymbol{u}_j^\top\boldsymbol{X}_1, \boldsymbol{u}_j^\top\boldsymbol{X}_2)$, if $\lambda_j \neq 0$. Similarly to the proof of Corollary 4, $\boldsymbol{u}_j$ must be an eigenvector of $\boldsymbol{N}^{-1}\boldsymbol{M}$, and the corresponding eigenvalue is λ^*. Accordingly, $\boldsymbol{u}_j$ is a linear combination of all the linearly independent eigenvectors of λ^*, namely

$$\boldsymbol{u}_j = \alpha_{j1}\boldsymbol{v}_1 + \alpha_{j2}\boldsymbol{v}_2 + \cdots + \alpha_{jk}\boldsymbol{v}_k = \boldsymbol{V}\boldsymbol{\alpha}_j \tag{71}$$

where $\boldsymbol{\alpha}_j = [\alpha_{j1}, \cdots, \alpha_{jk}]^\top$, and $\boldsymbol{\alpha}_j \neq \boldsymbol{0}$.

We remind that Eqn.71 only holds when $\lambda_j \neq 0$. We thus have

$$\begin{aligned}
\boldsymbol{W}^\top\boldsymbol{W} &= \sum_{j=1}^{d} \lambda_j \boldsymbol{u}_j\boldsymbol{u}_j^\top \\
&= \sum_{\lambda_j>0} \lambda_j \boldsymbol{u}_j\boldsymbol{u}_j^\top \\
&= \sum_{\lambda_j>0} \lambda_j (\boldsymbol{V}\boldsymbol{\alpha}_j)(\boldsymbol{V}\boldsymbol{\alpha}_j)^\top \\
&= \boldsymbol{V}\left(\sum_{\lambda_j>0} \lambda_j \boldsymbol{\alpha}_j\boldsymbol{\alpha}_j^\top\right)\boldsymbol{V}^\top \\
&= \boldsymbol{V}\boldsymbol{C}\boldsymbol{V}^\top,
\end{aligned} \tag{72}$$

where $\boldsymbol{C} = \sum\limits_{\lambda_j>0} \lambda_j\boldsymbol{\alpha}_j\boldsymbol{\alpha}_j^\top$.

By Lemma 3, we have $\lambda_1, \cdots, \lambda_j \geq 0$ are not all zeros. Besides, we figure out that $\boldsymbol{\alpha}_j$ can be any non-zero vector. We thus have $\boldsymbol{C}$ is any k-order semi-positive definite and non-zero matrix, to demonstrate $\boldsymbol{W}^\top\boldsymbol{W}$. □

C. Proof Related to Breaking LSSR

In this section, we prove Theorem 1 from the perspective of Taylor's expansion.

We have defined $f_{SSR}(t)$ as

$$f_{SSR}(t) = \begin{cases} LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2), & t = 0, \\ SSR(\bar{\psi}(t; \boldsymbol{X}_1), \bar{\psi}(t; \boldsymbol{X}_2)), & t \neq 0, \end{cases} \tag{73}$$

where $\bar{\psi}(t; \mathbf{x}_{ci}) = \mathbf{1}^\top \bar{\varphi}(t; \mathbf{x}_{ci}) / \|\bar{\varphi}(t; \mathbf{x}_{ci})\|_2$, $\bar{\varphi}(t; \mathbf{x}_{ci}) = [(\boldsymbol{u}^*)^\top \mathbf{x}_{ci} t, 1]^\top$ and $t \in \mathbb{R}$. We remind Theorem 1 as below.

Theorem 1. *Let $\psi = \varphi_1 \circ LN(\cdot) \circ \varphi_2$, performing over the input $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbb{R}^{d \times m}$. If $f'_{SSR}(0) \neq 0$, we can always find suitable linear functions φ_1 and φ_2, such that*

$$SSR(\psi(\boldsymbol{X}_1), \psi(\boldsymbol{X}_2)) < LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2). \tag{74}$$

We prove Lemma 1 and show three extra lemmas before the formal proof of Theorem 1.

C.1. Proof of Lemma 1

Lemma 1. *Denote $LN(\cdot)$ as the LN operation in $\mathbb{R}^d (d \geq 3)$, and $SP(\cdot)$ as the SP operation[12] in $\mathbb{R}^{d-1}$. We can find some linear transformations $\hat{\varphi}_1$ and $\hat{\varphi}_2$, such that*

$$SP(\cdot) = \hat{\varphi}_1 \circ LN(\cdot) \circ \hat{\varphi}_2. \tag{75}$$

We denote that $SP(\cdot)$ is defined on $\mathbb{R}^{d-1}$, as

$$SP(\mathbf{x}) = \mathbf{x} / \|\mathbf{x}\|_2. \tag{76}$$

While $LN(\cdot)$ is defined on $\mathbb{R}^d$, where

$$LN(\mathbf{x}) = \sqrt{d}\, (\mathbf{x} - \frac{1}{d} \mathbf{1}\mathbf{1}^\top \mathbf{x}) / \|\mathbf{x} - \frac{1}{d} \mathbf{1}\mathbf{1}^\top \mathbf{x}\|_2. \tag{77}$$

Before the proof of Lemma 1, we propose Lemma 6 as follows.

Lemma 6. *There is some orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{d \times d}$, such that $\boldsymbol{z} = \boldsymbol{Q} \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} \in \{\boldsymbol{z} \in \mathbb{R}^d : z^{(1)} + \cdots + z^{(d)} = 0\}$ (namely $\boldsymbol{z}$ is centralized), for $\mathbf{x} \in \mathbb{R}^{d-1}$,*

Proof. Suppose $\boldsymbol{Q} = \{q_{ij}\}_{d \times d} = [\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_d]$. We take $\boldsymbol{q}_d = \frac{1}{\sqrt{d}} \mathbf{1}$ specially, and $\boldsymbol{q}_1, \cdots, \boldsymbol{q}_{d-1}$ can be calculated by Schmidt orthogonalization.

Given $\mathbf{x} = [x^{(1)}, \cdots, x^{(d-1)}]^\top \in \mathbb{R}^{d-1}$, we have

$$\boldsymbol{z} = \boldsymbol{Q} \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} = [z^{(1)}, \cdots, z^{(d)}]^\top. \tag{78}$$

Since $\boldsymbol{Q}$ is an orthogonal matrix, we have

$$\boldsymbol{q}_i^\top \boldsymbol{q}_d = \frac{1}{\sqrt{d}} \sum_{k=1}^{d} q_{ki} = 0, \tag{79}$$

[12] If there are no special instructions, we denote SP projects the sample on to the unit circle, namely $\mathbf{x} \mapsto \mathbf{x} / \|\mathbf{x}\|_2$.

for $i = 1, \cdots, d-1$. Furthermore, we obtain that

$$\begin{aligned}\sum_{k=1}^{d} z^{(k)} &= \sum_{k=1}^{d}\left(\sum_{i=1}^{d} q_{ki}x^{(i)}\right)\\ &= \sum_{k=1}^{d}\left(q_{kd}\cdot 0 + \sum_{i=1}^{d-1} q_{ki}x^{(i)}\right)\\ &= \sum_{i=1}^{d-1}\left(\sum_{k=1}^{d} q_{ki}\right)x^{(i)}\\ &= 0,\end{aligned} \tag{80}$$

which shows that $\boldsymbol{z} \in \{\boldsymbol{z} \in \mathbb{R}^d : z^{(1)} + \cdots + z^{(d)} = 0\}$, namely $\boldsymbol{z}$ is centralized. □

Now we can design $\hat{\varphi}_1$ and $\hat{\varphi}_2$ based on $\boldsymbol{Q}$ in Lemma 6, and then prove Lemma 1.

Proof. Based above, we obtain

$$\|\boldsymbol{z}\|_2 = \left\|\boldsymbol{Q}\begin{bmatrix}\mathbf{x}\\0\end{bmatrix}\right\|_2 = \left\|\begin{bmatrix}\mathbf{x}\\0\end{bmatrix}\right\|_2 = \|\mathbf{x}\|_2. \tag{81}$$

By Lemma 6, we have $\mathbf{1}^\top \boldsymbol{z} = 0$, and $\boldsymbol{z} = \boldsymbol{z} - \frac{1}{d}\mathbf{1}\mathbf{1}^\top \boldsymbol{z}$. We hence find that

$$LN(\boldsymbol{z}) = \sqrt{d}(\boldsymbol{z} - \frac{1}{d}\mathbf{1}\mathbf{1}^\top \boldsymbol{z})/\|\boldsymbol{z} - \frac{1}{d}\mathbf{1}\mathbf{1}^\top \boldsymbol{z}\|_2 = \sqrt{d}\,\boldsymbol{z}/\|\boldsymbol{z}\|_2. \tag{82}$$

Let $\boldsymbol{I}_d$ denotes the identity matrix in $\mathbb{R}^{d\times d}$. We thus have

$$\begin{aligned}\frac{1}{\sqrt{d}}\begin{bmatrix}\boldsymbol{I}_{d-1} & \mathbf{0}\end{bmatrix}\boldsymbol{Q}^\top LN(\boldsymbol{Q}\begin{bmatrix}\boldsymbol{I}_{d-1} & \mathbf{0}\end{bmatrix}^\top \mathbf{x}) =& \frac{1}{\sqrt{d}}\begin{bmatrix}\boldsymbol{I}_{d-1} & \mathbf{0}\end{bmatrix}\boldsymbol{Q}^\top LN(\boldsymbol{z})\\ =& \frac{1}{\sqrt{d}}\begin{bmatrix}\boldsymbol{I}_{d-1} & \mathbf{0}\end{bmatrix}\boldsymbol{Q}^\top \sqrt{d}\,\boldsymbol{z}/\|\boldsymbol{z}\|_2\\ =& \sqrt{d}\cdot\frac{1}{\sqrt{d}}\begin{bmatrix}\boldsymbol{I}_{d-1} & \mathbf{0}\end{bmatrix}\boldsymbol{Q}^\top\boldsymbol{Q}\begin{bmatrix}\mathbf{x}\\0\end{bmatrix}/\|\mathbf{x}\|_2\\ =& \mathbf{x}/\|\mathbf{x}\|_2\\ =& SP(\mathbf{x}).\end{aligned} \tag{83}$$

Let $\hat{\varphi}_1(\mathbf{x}) = \boldsymbol{Q}\begin{bmatrix}\boldsymbol{I}_{d-1} & \mathbf{0}\end{bmatrix}^\top \mathbf{x}$, and $\hat{\varphi}_2(\mathbf{x}) = \frac{1}{\sqrt{d}}\begin{bmatrix}\boldsymbol{I}_{d-1} & \mathbf{0}\end{bmatrix}\boldsymbol{Q}^\top\mathbf{x}$. We observe that

$$SP(\cdot) = \hat{\varphi}_1 \circ LN(\cdot) \circ \hat{\varphi}_2. \tag{84}$$

□

C.2. Extra Lemmas for the Proof

Let $x_{ci} = (\boldsymbol{u}^*)^\top \mathbf{x}_{ci}, (i = 1, \cdots, m; c = 1, 2)$. We define the mean $\bar{x}_c$, the variance σ_c^2 and the third-order central moment $\overline{(x_c - \bar{x}_c)^3}$ with the equations below:

$$\begin{aligned}\bar{x}_c &= \frac{1}{m}\sum_{i=1}^{m} x_{ci},\\ \sigma_c^2 &= \frac{1}{m}\sum_{i=1}^{m}(x_{ci} - \bar{x}_c)^2,\\ \overline{(x_c - \bar{x}_c)^3} &= \frac{1}{m}\sum_{i=1}^{m}(x_{ci} - \bar{x}_c)^3.\end{aligned} \tag{85}$$

Based on $x_{ci} = (\boldsymbol{u}^*)^\top \mathbf{x}_{ci}$, we design an linear transformation $\varphi(t;\cdot) : \mathbb{R} \to \mathbb{R}^2$, where $t \in \mathbb{R}$ is a parameter:

$$\varphi(t; x_{ci}) = t \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} x_{ci} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_{ci} t \\ 1 \end{bmatrix}. \tag{86}$$

Lemma 7. *Let $\hat{\boldsymbol{X}} = SP(\varphi(t; (\boldsymbol{u}^*)^\top \boldsymbol{X}))$, and $\boldsymbol{v} = [1,1]^\top$. Besides, we define three statistics about $(\boldsymbol{u}^*)^\top \boldsymbol{X}$:*

$$\begin{cases} T_1 = (\bar{x}_1 - \bar{x}_2)^2 [\overline{(x_1 - \bar{x}_1)^3} + \overline{(x_2 - \bar{x}_2)^3}], \\ T_2 = (\bar{x}_1 - \bar{x}_2)(\sigma_1^2 - \sigma_2^2)[(\bar{x}_1 - \bar{x}_2)^2 - (\sigma_1^2 + \sigma_2^2)], \\ T_3 = [2\sigma_1^2 + 2\sigma_2^2 + (\bar{x}_1 - \bar{x}_2)^2]^2. \end{cases} \tag{87}$$

We figure out that when $t \to 0$, we have

$$SSR(\boldsymbol{v}^\top \hat{\boldsymbol{X}}_1, \boldsymbol{v}^\top \hat{\boldsymbol{X}}_2) = LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) - \frac{2(T_1 + T_2)}{T_3} t + o(t). \tag{88}$$

Proof. Denote $\hat{\mathbf{x}}_{ci} = SP(\varphi(t; x_{ci})) = [\hat{x}_{ci}^{(1)}, \hat{x}_{ci}^{(2)}]^\top$. By Newton's binomial expansion, we obtain that

$$\begin{aligned} \frac{1}{\|\varphi(t; x_{ci})\|_2} =& \frac{1}{\sqrt{1 + (x_{ci} t)^2}} \\ =& (1 + x_{ci}^2 t^2)^{-\frac{1}{2}} \\ =& 1 - \frac{1}{2}(x_{ci}^2 t^2) + \frac{3}{8}(x_{ci}^2 t^2)^2 + o((t^2)^2) \\ =& 1 - \frac{1}{2} x_{ci}^2 t^2 + o(t^3). \end{aligned} \tag{89}$$

We thus have

$$\hat{x}_{ci}^{(1)} = \frac{x_{ci} t}{\sqrt{1 + (x_{ci} t)^2}} = x_{ci} t - \frac{1}{2} x_{ci}^3 t^3 + o(t^3), \tag{90}$$

and

$$\hat{x}_{ci}^{(2)} = \frac{1}{\sqrt{1 + (x_{ci} t)^2}} = 1 - \frac{1}{2} x_{ci}^2 t^2 + o(t^3). \tag{91}$$

Let $\boldsymbol{v} = [1,1]^\top$. Then we have

$$\boldsymbol{v}^\top \hat{\mathbf{x}}_{ci} = 1 + x_{ci} t - \frac{1}{2} x_{ci}^2 t^2 - \frac{1}{2} x_{ci}^3 t^3 + o(t^3). \tag{92}$$

We denote that $a_0 = 1, a_1 = 1, a_2 = -\frac{1}{2}$ and $a_3 = -\frac{1}{2}$, therefore

$$\boldsymbol{v}^\top \hat{\mathbf{x}}_{ci} = \sum_{s=0}^{3} a_s x_{ci}^s t^s + o(t^3). \tag{93}$$

We hence obtain that

$$\begin{aligned} SS(\boldsymbol{v}^\top \hat{\boldsymbol{X}}_c) =& \sum_{i=1}^{m} (\boldsymbol{v}^\top \hat{\mathbf{x}}_{ci} - \overline{\boldsymbol{v}^\top \hat{\mathbf{x}}_c})^2 \\ =& \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{m} \boldsymbol{v}^\top \hat{\mathbf{x}}_{ci} (\boldsymbol{v}^\top \hat{\mathbf{x}}_{ci} - \boldsymbol{v}^\top \hat{\mathbf{x}}_{cj}) \\ =& \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{m} \left[\left(\sum_{r=0}^{3} a_r x_{ci}^r t^r + o(t^3) \right) \cdot \left(\sum_{s=0}^{3} a_s (x_{ci}^s - x_{cj}^s) t^s + o(t^3) \right) \right]. \end{aligned} \tag{94}$$

For $s + r > 3$, we put the multiplicative term into $o(t^3)$. Accordingly, we only consider the case $s + r \leq 3$.

For $r = 0, 1, 2, 3; s = 0$, we have

$$x_{ci}^r(x_{ci}^s - x_{cj}^s) = 0. \tag{95}$$

For $r = 0; s = 1, 2, 3$, we have

$$\sum_{i=1}^m \sum_{j=1}^m x_{ci}^r t^r \cdot (x_{ci}^s - x_{cj}^s)t^s = \sum_{i=1}^m \sum_{j=1}^m (x_{ci}^s - x_{cj}^s)t^s = 0. \tag{96}$$

For $r = 1, s = 1$, we have

$$\sum_{i=1}^m \sum_{j=1}^m x_{ci}(x_{ci} - x_{cj}) = m\sum_{i=1}^m (x_{ci} - \bar{x}_c) = m\sigma_c^2. \tag{97}$$

For $r = 2, s = 1$, we observe

$$\begin{aligned}
\sum_{i=1}^m \sum_{j=1}^m x_{ci}^2(x_{ci} - x_{cj}) &= m\sum_{i=1}^m x_{ci}^2(x_{ci} - \bar{x}_c) \\
&= m\sum_{i=1}^m [(x_{ci}^2 - 2x_{ci}\bar{x}_c + \bar{x}_c^2)(x_{ci} - \bar{x}_c) + 2x_{ci}\bar{x}_c(x_{ci} - \bar{x}_c) - \bar{x}_c^2(x_{ci} - \bar{x}_c)] \\
&= m^2[\overline{(x_c - \bar{x}_c)^3} + 2\bar{x}_c\sigma_c^2].
\end{aligned} \tag{98}$$

And for $r = 1, s = 2$, we obtain

$$\begin{aligned}
\sum_{i=1}^m \sum_{j=1}^m x_{ci}(x_{ci}^2 - x_{cj}^2) &= \sum_{i=1}^m \sum_{j=1}^m x_{ci}^3 - x_{ci}x_{cj}^2 \\
&= \sum_{i=1}^m \sum_{j=1}^m x_{ci}^3 - x_{ci}^2 x_{cj} \\
&= \sum_{i=1}^m \sum_{j=1}^m x_{ci}^2(x_{ci} - x_{cj}) \\
&= m^2[\overline{(x_c - \bar{x}_c)^3} + 2\bar{x}_c\sigma_c^2].
\end{aligned} \tag{99}$$

Therefore, we have that

$$\begin{aligned}
SS(\boldsymbol{v}^\top \hat{\boldsymbol{X}}_c) &= \frac{1}{m}\sum_{i=1}^m \sum_{j=1}^m \left[a_1^2 x_{ci}(x_{ci} - x_{cj})t^2 + a_1 a_2 x_{ci}^2(x_{ci} - x_{cj})t^3 + a_1 a_2 x_{ci}(x_{ci}^2 - x_{cj}^2)t^3 + o(t^3)\right] \\
&= ma_1^2\sigma_c^2 t^2 + 2ma_1 a_2[\overline{(x_c - \bar{x}_{ci})^3} + 2\bar{x}_c\sigma_c^2]t^3 + o(t^3) \\
&= m\sigma_c^2 t^2 - m[\overline{(x_c - \bar{x}_{ci})^3} + 2\bar{x}_c\sigma_c^2]t^3 + o(t^3) \\
&= \beta_{c2}t^2 + \beta_{c3}t^3 + o(t^3),
\end{aligned} \tag{100}$$

where $\beta_{c2} = m\sigma_c^2$, and $\beta_{c3} = -m[\overline{(x_c - \bar{x}_{ci})^3} + 2\bar{x}_c\sigma_c^2]$.

To simplify the calculation, we define

$$\begin{aligned}
SS_D(\boldsymbol{X}_1, \boldsymbol{X}_2) &= SS([\boldsymbol{X}_1, \boldsymbol{X}_2]) - SS(\boldsymbol{X}_1) - SS(\boldsymbol{X}_2) \\
&= \sum_{c=1}^2 \sum_{i=1}^m (\mathbf{x}_{ci} - \bar{\mathbf{x}})^\top(\mathbf{x}_{ci} - \bar{\mathbf{x}}) - \sum_{c=1}^2 \sum_{i=1}^m (\mathbf{x}_{ci} - \bar{\mathbf{x}}_c)^\top(\mathbf{x}_{ci} - \bar{\mathbf{x}}_c) \\
&= \sum_{c=1}^2 \sum_{i=1}^m (\mathbf{x}_{ci}^\top\mathbf{x}_{ci} - \bar{\mathbf{x}}^\top\bar{\mathbf{x}}) - \sum_{c=1}^2 \sum_{i=1}^m (\mathbf{x}_{ci}^\top\mathbf{x}_{ci} - \bar{\mathbf{x}}_c^\top\bar{\mathbf{x}}_c) \\
&= m\bar{\mathbf{x}}_1^\top\bar{\mathbf{x}}_1 + m\bar{\mathbf{x}}_2^\top\bar{\mathbf{x}}_2 - 2m\bar{\mathbf{x}}^\top\bar{\mathbf{x}} \\
&= m\bar{\mathbf{x}}_1^\top\bar{\mathbf{x}}_1 + m\bar{\mathbf{x}}_2^\top\bar{\mathbf{x}}_2 - \frac{m}{2}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2)^\top(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) \\
&= \frac{m}{2}\|\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2\|_2^2.
\end{aligned} \tag{101}$$

Similar to Eqn.100, we obtain

$$\begin{aligned}SS_D(\boldsymbol{v}^\top\hat{\boldsymbol{X}}_1,\boldsymbol{v}^\top\hat{\boldsymbol{X}}_2)=&\frac{m}{2}(\overline{\boldsymbol{v}^\top\hat{\mathbf{x}}_1}-\overline{\boldsymbol{v}^\top\hat{\mathbf{x}}_2})^2\\=&\frac{m}{2}[a_1(\bar{x}_1-\bar{x}_2)t+a_2(\overline{x_1^2}-\overline{x_2^2})t^2+o(t^2)]^2\\=&\frac{m}{2}[a_1^2(\bar{x}_1-\bar{x}_2)^2t^2+2a_1a_2(\bar{x}_1-\bar{x}_2)(\overline{x_1^2}-\overline{x_2^2})t^3+o(t^3)]\\=&\frac{m}{2}(\bar{x}_1-\bar{x}_2)^2t^2-\frac{m}{2}(\bar{x}_1-\bar{x}_2)(\overline{x_1^2}-\overline{x_2^2})t^3+o(t^3)\\=&\beta_2t^2+\beta_3t^3+o(t^3),\end{aligned}\tag{102}$$

where $\beta_2=\frac{m}{2}(\bar{x}_1-\bar{x}_2)^2$, and $\beta_3=-\frac{m}{2}(\bar{x}_1-\bar{x}_2)(\overline{x_1^2}-\overline{x_2^2})$.

We thus have

$$\begin{aligned}SSR(\boldsymbol{v}^\top\hat{\boldsymbol{X}}_1,\boldsymbol{v}^\top\hat{\boldsymbol{X}}_2)=&\frac{(\beta_{12}+\beta_{22})t^2+(\beta_{13}+\beta_{23})t^3+o(t^3)}{(\beta_{12}+\beta_{22}+\beta_2)t^2+(\beta_{13}+\beta_{23}+\beta_3)t^3+o(t^3)}\\=&\frac{(\beta_{12}+\beta_{22})t^2+(\beta_{13}+\beta_{23})t^3+o(t^3)}{(\beta_{12}+\beta_{22}+\beta_2)t^2\left[1+\frac{\beta_{13}+\beta_{23}+\beta_3}{\beta_{12}+\beta_{22}+\beta_2}t+o(t)\right]}\\=&\left[\frac{\beta_{12}+\beta_{22}}{\beta_{12}+\beta_{22}+\beta_2}+\frac{\beta_{13}+\beta_{23}}{\beta_{12}+\beta_{22}+\beta_2}t+o(t)\right]\cdot\left[1-\frac{\beta_{13}+\beta_{23}+\beta_3}{\beta_{12}+\beta_{22}+\beta_2}t+o(t)\right]\\=&\frac{\beta_{12}+\beta_{22}}{\beta_{12}+\beta_{22}+\beta_2}+\frac{(\beta_{12}+\beta_{22}+\beta_2)(\beta_{13}+\beta_{23})-(\beta_{12}+\beta_{22})(\beta_{13}+\beta_{23}+\beta_3)}{(\beta_{12}+\beta_{22}+\beta_2)^2}t+o(t)\\=&\frac{\beta_{12}+\beta_{22}}{\beta_{12}+\beta_{22}+\beta_2}+\frac{\beta_2(\beta_{13}+\beta_{23})-\beta_3(\beta_{12}+\beta_{22})}{(\beta_{12}+\beta_{22}+\beta_2)^2}t+o(t).\end{aligned}\tag{103}$$

We find that

$$\frac{\beta_{12}+\beta_{22}}{\beta_{12}+\beta_{22}+\beta_2}=SSR((\boldsymbol{u}^*)^\top\boldsymbol{X}_1,(\boldsymbol{u}^*)^\top\boldsymbol{X}_2)=LSSR(\boldsymbol{X}_1,\boldsymbol{X}_2).\tag{104}$$

On the other hand, we have

$$\begin{aligned}&\beta_2(\beta_{13}+\beta_{23})-\beta_3(\beta_{12}+\beta_{22})\\=&-\frac{1}{2}m^2(\bar{x}_1-\bar{x}_2)^2[\overline{(x_1-\bar{x}_1)^3}+2\bar{x}_1\sigma_1^2+\overline{(x_2-\bar{x}_2)^3}+2\bar{x}_2\sigma_2^2]+\frac{1}{2}m^2(\bar{x}_1-\bar{x}_2)(\overline{x_1^2}-\overline{x_2^2})(\sigma_1^2+\sigma_2^2)\\=&-\frac{1}{2}m^2(\bar{x}_1-\bar{x}_2)^2[\overline{(x_1-\bar{x}_1)^3}+\overline{(x_2-\bar{x}_2)^3}]\\&-\frac{1}{2}m^2(\bar{x}_1-\bar{x}_2)^2(2\bar{x}_1\sigma_1^2+2\bar{x}_2\sigma_2^2)+\frac{1}{2}m^2(\bar{x}_1-\bar{x}_2)(\overline{x_1^2}-\overline{x_2^2})(\sigma_1^2+\sigma_2^2).\end{aligned}\tag{105}$$

We figure out that

$$\begin{aligned}&(\bar{x}_1-\bar{x}_2)^2(2\bar{x}_1\sigma_1^2+2\bar{x}_2\sigma_2^2)-(\bar{x}_1-\bar{x}_2)(\overline{x_1^2}-\overline{x_2^2})(\sigma_1^2+\sigma_2^2)\\=&(\bar{x}_1-\bar{x}_2)[(\bar{x}_1-\bar{x}_2)(2\bar{x}_1\sigma_1^2+2\bar{x}_2\sigma_2^2)-(\bar{x}_1^2+\sigma_1^2-\bar{x}_2^2-\sigma_2^2)(\sigma_1^2+\sigma_2^2)]\\=&(\bar{x}_1-\bar{x}_2)[2\bar{x}_1(\bar{x}_1-\bar{x}_2)\sigma_1^2+2\bar{x}_2(\bar{x}_1-\bar{x}_2)\sigma_2^2-(\bar{x}_1^2-\bar{x}_2^2)\sigma_1^2-(\bar{x}_1^2-\bar{x}_2^2)\sigma_2^2-(\sigma_1^2-\sigma_2^2)(\sigma_1^2+\sigma_2^2)]\\=&(\bar{x}_1-\bar{x}_2)[(\bar{x}_1-\bar{x}_2)^2\sigma_1^2-(\bar{x}_1-\bar{x}_2)^2\sigma_2^2-(\sigma_1^2-\sigma_2^2)(\sigma_1^2+\sigma_2^2)]\\=&(\bar{x}_1-\bar{x}_2)(\sigma_1^2-\sigma_2^2)[(\bar{x}_1-\bar{x}_2)^2-(\sigma_1^2+\sigma_2^2)].\end{aligned}\tag{106}$$

By the definition of T_1,T_2 and T_3, we thus obtain

$$\begin{aligned}&\beta_2(\beta_{13}+\beta_{23})-\beta_3(\beta_{12}+\beta_{22})\\=&-\frac{1}{2}m^2(\bar{x}_1-\bar{x}_2)^2[\overline{(x_1-\bar{x}_1)^3}+\overline{(x_2-\bar{x}_2)^3}]-\frac{1}{2}m^2(\bar{x}_1-\bar{x}_2)(\sigma_1^2-\sigma_2^2)[(\bar{x}_1-\bar{x}_2)^2-(\sigma_1^2+\sigma_2^2)]\\=&-\frac{1}{2}m^2T_1-\frac{1}{2}m^2T_2.\end{aligned}\tag{107}$$

Moreover, we have

$$\begin{aligned}(\beta_{12}+\beta_{22}+\beta_2)^2 &= [m\sigma_1^2+m\sigma_2^2+\frac{m}{2}(\bar{x}_1-\bar{x}_2)^2]^2\\ &= \frac{1}{4}m^2[2\sigma_1^2+2\sigma_2^2+(\bar{x}_1-\bar{x}_2)^2]^2\\ &= \frac{1}{4}m^2T_3.\end{aligned} \tag{108}$$

As a result, we obtain that

$$SSR(\boldsymbol{v}^\top\hat{\boldsymbol{X}}_1,\boldsymbol{v}^\top\hat{\boldsymbol{X}}_2) = LSSR(\boldsymbol{X}_1,\boldsymbol{X}_2)-\frac{2(T_1+T_2)}{T_3}t+o(t). \tag{109}$$

□

Lemma 8. *For*

$$f_{SSR}(t)=\begin{cases}LSSR(\boldsymbol{X}_1,\boldsymbol{X}_2), & t=0,\\ SSR(\bar{\psi}(t;\boldsymbol{X}_1),\bar{\psi}(t;\boldsymbol{X}_2)), & t\neq 0,\end{cases} \tag{110}$$

where $\bar{\psi}(t;\mathbf{x}_{ci})=\mathbf{1}^\top\bar{\varphi}(t;\mathbf{x}_{ci})/\|\bar{\varphi}(t;\mathbf{x}_{ci})\|_2$, $\bar{\varphi}(t;\mathbf{x}_{ci})=[(\boldsymbol{u}^*)^\top\mathbf{x}_{ci}t,1]^\top$ *and* $t\in\mathbb{R}$, *we have that* $f_{SSR}(t)$ *is derivable around* $t=0$, *and* $f'_{SSR}(0)$ *is only decided by* $\boldsymbol{X}_1$ *and* $\boldsymbol{X}_2$.

Proof. It is easy to identify that $\bar{\psi}(t;\boldsymbol{X}_i)=\boldsymbol{v}^\top\hat{\boldsymbol{X}}_i$. Therefore, by Lemma 7. We have

$$\begin{aligned}SSR(\bar{\psi}(t;\boldsymbol{X}_1),\bar{\psi}(t;\boldsymbol{X}_2)) &= SSR(\boldsymbol{v}^\top\hat{\boldsymbol{X}}_1,\boldsymbol{v}^\top\hat{\boldsymbol{X}}_2)\\ &= LSSR(\boldsymbol{X}_1,\boldsymbol{X}_2)-\frac{2(T_1+T_2)}{T_3}t+o(t).\end{aligned} \tag{111}$$

We hence obtain

$$\begin{aligned}f'_{SSR}(0) &= \lim_{t\to 0}\frac{f_{SSR}(t)-f_{SSR}(0)}{t}\\ &= \lim_{t\to 0}\frac{SSR(\bar{\psi}(t;\boldsymbol{X}_1),\bar{\psi}(t;\boldsymbol{X}_2))-LSSR(\boldsymbol{X}_1,\boldsymbol{X}_2)}{t}\\ &= \frac{-\frac{2(T_1+T_2)}{T_3}t+o(t)}{t}\\ &= -\frac{2(T_1+T_2)}{T_3}.\end{aligned} \tag{112}$$

Conclusively, we have that $f_{SSR}(t)$ is derivable at $t=0$.

□

Lemma 9. *Given a differentiable function* $f(x)$, *with* $f'(0)\neq 0$, *we figure out that there is some* x^*, *such that* $f(x^*)<f(0)$.

Proof. Given that $f(0)=A$ and $f'(0)=B\neq 0$, by the definition of derivative, we have $\lim_{h\to 0}\frac{f(h)-A}{h}=B$. That is to say, $\forall\varepsilon>0$, there exists a positive $\delta>0$, whenever $0<|x|<\delta$, we have

$$\left|\frac{f(x)-A}{x}-B\right|\leq\varepsilon, \tag{113}$$

then

$$-\varepsilon|x|\leq f(x)-A-Bx\leq\varepsilon|x|. \tag{114}$$

Let $x^*=-\frac{|B|\delta}{2B}$, and $\varepsilon=\frac{|B|}{2}$. We have

$$f(x^*)\leq A+Bx+\varepsilon|x| = A-\frac{|B|\delta}{4}<A, \tag{115}$$

namely $f(x^*)<f(0)$. □

C.3. Proof of Theorem 1

Since $f'_{SSR}(0) = -2(T_1 + T_2)/T_3 \neq 0$, by Lemma 9, there is some $t = t^*$, such that

$$SSR(\bar{\psi}(t^*; \boldsymbol{X}_1), \bar{\psi}(t^*; \boldsymbol{X}_2)) < LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2). \tag{116}$$

We denote $\tilde{\varphi}_1(\mathbf{x}) = \varphi(t^*; \boldsymbol{u}^\top \mathbf{x})$ and $\tilde{\varphi}_2(\mathbf{x}) = \boldsymbol{v}^\top \mathbf{x}$. By Lemma 1, we have $SP(\cdot) = \hat{\varphi}_1 \circ LN(\cdot) \circ \hat{\varphi}_2$. We hence have

$$\boldsymbol{v}^\top \hat{\boldsymbol{X}}_c = \tilde{\varphi}_2(\hat{\varphi}_2(LN(\hat{\varphi}_1(\tilde{\varphi}_1(\boldsymbol{X}_c))))). \tag{117}$$

Let $\psi = \varphi_1 \circ LN(\cdot) \circ \varphi_2$, where $\varphi_1 = \tilde{\varphi}_1 \circ \hat{\varphi}_1$ and $\varphi_2 = \hat{\varphi}_2 \circ \tilde{\varphi}_2$. We thus have

$$SSR(\psi(\boldsymbol{X}_1), \psi(\boldsymbol{X}_2)) < LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2). \tag{118}$$

Obviously, φ_1 and φ_2 are linear functions. Consequently, we have proved Theorem 1.

C.4. A Generalized Proof of Theorem 1

To begin with, we also need to project $\boldsymbol{X}_c$ to $\boldsymbol{u}^\top \boldsymbol{X}_c$. This can reach $LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2)$, which is necessary in our discussion. More generally, we design a n-dimensional linear transformation $\varphi_n(t; \cdot) : \mathbb{R} \to \mathbb{R}^n$, instead of a 2-dimensional one in Eqn.86. Specifically, we denote

$$\varphi_n(t; x) = t \cdot \boldsymbol{w}x + \boldsymbol{b} = \begin{bmatrix} w_1 x t + b_1 \\ \cdots \\ w_n x t + b_n \end{bmatrix}. \tag{119}$$

Considering SP on $\varphi_n(t; x)$ with scaling=1, we denote

$$\hat{\mathbf{x}} = SP(\varphi_n(t; x)) = \varphi_n(t; x)/\|\varphi_n(t; x)\|. \tag{120}$$

Owing to the introduce of t, let t and $\boldsymbol{w}$ represent the direction and length of weight respectively. We thus add the constraint $\|\boldsymbol{w}\|_2 = 1$ for convenience. As for the bias, if $\boldsymbol{b} = \mathbf{0}$, $\hat{\mathbf{x}} = \boldsymbol{w}/(\|\boldsymbol{w}\|_2)$ will result in $SS(\psi(\hat{\boldsymbol{X}}_1), \psi(\hat{\boldsymbol{X}}_2)) = 0$. Therefore, we require that $\boldsymbol{b} \neq \mathbf{0}$. Now we are concerned about $LSSR(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2)$.

Factually, by Proposition 2, we need not consider all the linear functions on $\mathbb{R}^n$ to get LSSR. We figure out that there must be some $\boldsymbol{v} \in \mathbb{R}^n$, such that

$$LSSR(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2) = SSR(\boldsymbol{v}^\top \hat{\boldsymbol{X}}, \boldsymbol{v}^\top \hat{\boldsymbol{Y}}). \tag{121}$$

We give the Taylor's expansion of $\hat{\mathbf{x}}$ on its each dimension.

Let $\xi_1 = \boldsymbol{w}^\top \boldsymbol{b}$ and $\xi_2 = \boldsymbol{b}^\top \boldsymbol{b}$. We figure out that

$$\begin{aligned} \frac{1}{\|\boldsymbol{w}xt + \boldsymbol{b}\|_2} =& (1 + 2\xi_1 xt + \xi_2 x^2 t^2)^{-\frac{1}{2}} \\ =& 1 - \frac{1}{2}(2\xi_1 xt + \xi_2 x^2 t^2) + \frac{3}{8}(2\xi_1 xt + \xi_2 x^2 t^2)^2 - \frac{5}{16}(2\xi_1 xt + \xi_2 x^2 t^2)^3 + o(t^3) \\ =& 1 - \xi_1 xt + (\frac{3}{2}\xi_1^2 - \frac{1}{2}\xi_2)x^2 t^2 + (\frac{3}{2}\xi_1\xi_2 - \frac{5}{2}\xi_1^3)x^3 t^3 + o(t^3). \end{aligned} \tag{122}$$

We further obtain

$$\begin{aligned} \hat{x}^{(k)} =& \frac{w_k xt + b_k}{\|\boldsymbol{w}xt + \boldsymbol{b}\|_2} \\ =& (b_k + w_k xt)[1 - \xi_1 xt + (\frac{3}{2}\xi_1^2 - \frac{1}{2}\xi_2)x^2 t^2 + (\frac{3}{2}\xi_1\xi_2 - \frac{5}{2}\xi_1^3)x^3 t^3 + o(t^3)] \\ =& b_k + (w_k - \xi_1 b_k)xt + [(\frac{3}{2}\xi_1^2 - \frac{1}{2}\xi_2)b_k - \xi_1 w_k]x^2 t^2 + [(\frac{3}{2}\xi_1\xi_2 - \frac{5}{2}\xi_1^3)b_k + (\frac{3}{2}\xi_1^2 - \frac{1}{2}\xi_2)w_k]x^3 t^3 + o(t^3). \end{aligned} \tag{123}$$

Similarly, to simplify our calculation, we denote

$$\hat{x}_{ci}^{(k)} = a_0^{(k)} + a_1^{(k)} x_{ci} t + a_2^{(k)} x_{ci}^2 t^2 + a_3^{(k)} x_{ci}^3 t^3 + o(t^3) \tag{124}$$

where $a_0^{(k)} = b_k$, $a_1^{(k)} = w_k - \xi_1 b_k$, $a_2^{(k)} = (\frac{3}{2}\xi_1^2 - \frac{1}{2}\xi_2)b_k - \xi_1 w_k$ and $a_3^{(k)} = (\frac{3}{2}\xi_1\xi_2 - \frac{5}{2}\xi_1^3)b_k + (\frac{3}{2}\xi_1^2 - \frac{1}{2}\xi_2)w_k$. Let $\boldsymbol{v} = [v_1, \cdots, v_n]^\top$. We have

$$\begin{aligned} SS(\boldsymbol{v}^\top \hat{\boldsymbol{X}}_c) &= \sum_{i=1}^m (\boldsymbol{v}^\top \hat{\mathbf{x}}_{ci} - \overline{\boldsymbol{v}^\top \hat{\mathbf{x}}_c})^2 \\ &= \frac{1}{m}\sum_{i=1}^m \sum_{j=1}^m \boldsymbol{v}^\top \hat{\mathbf{x}}_{ci}(\boldsymbol{v}^\top \hat{\mathbf{x}}_{ci} - \boldsymbol{v}^\top \hat{\mathbf{x}}_{cj}) \\ &= \frac{1}{m}\sum_{i=1}^m \sum_{j=1}^m \left(\sum_{k=1}^n v_k \hat{x}_{ci}^{(k)}\right) \cdot \left(\sum_{l=1}^n v_l [\hat{x}_{ci}^{(l)} - \hat{x}_{cj}^{(l)}]\right) \\ &= \sum_{k=1}^n \sum_{l=1}^n v_k v_l \left(\frac{1}{m}\sum_{i=1}^m \sum_{j=1}^m \hat{x}_{ci}^{(k)}[\hat{x}_{ci}^{(l)} - \hat{x}_{cj}^{(l)}]\right). \end{aligned} \tag{125}$$

Similar to calculation when we discuss the 2-dimensional case, we have

$$\begin{aligned} &\frac{1}{m}\sum_{i=1}^m \sum_{j=1}^m \hat{x}_{ci}^{(k)}[\hat{x}_{ci}^{(l)} - \hat{x}_{cj}^{(l)}] \\ =&\frac{1}{m}\sum_{i=1}^m \sum_{j=1}^m \left[\left(\sum_{s=0}^3 a_s^{(k)} x_{ci}^s t^s + o(t^3)\right) \cdot \left(\sum_{s=0}^3 a_s^{(l)}(x_{ci}^s - x_{cj}^s)t^s + o(t^3)\right)\right] \\ =&\frac{1}{m}\sum_{i=1}^m \sum_{j=1}^m \left[a_1^{(k)}a_1^{(l)}x_{ci}(x_{ci} - x_{cj})t^2 + a_2^{(k)}a_1^{(l)}x_{ci}^2(x_{ci} - x_{cj})t^3 + a_1^{(k)}a_2^{(l)}x_{ci}(x_{ci}^2 - x_{cj}^2)t^3 + o(t^3)\right] \\ =&ma_1^{(k)}a_1^{(l)}\sigma_c^2 t^2 + m[a_1^{(k)}a_2^{(l)} + a_2^{(k)}a_1^{(l)}][\overline{(x_c - \bar{x}_{ci})^3} + 2\bar{x}_c\sigma_c^2]t^3 + o(t^3). \end{aligned} \tag{126}$$

We define

$$\theta_1 = \sum_{k=1}^n \sum_{l=1}^n v_k v_l a_1^{(k)} a_1^{(l)}, \tag{127}$$

and

$$\theta_2 = \sum_{k=1}^n \sum_{l=1}^n v_k v_l a_1^{(k)} a_2^{(l)} = \sum_{k=1}^n \sum_{l=1}^n v_k v_l a_2^{(k)} a_1^{(l)}. \tag{128}$$

We thus have that

$$\begin{aligned} SS(\boldsymbol{v}^\top \hat{\boldsymbol{X}}_c) &= \sum_{k=1}^n \sum_{l=1}^n v_k v_l \left(\frac{1}{m}\sum_{i=1}^m \sum_{j=1}^m \hat{x}_{ci}^{(k)}[\hat{x}_{ci}^{(l)} - \hat{x}_{cj}^{(l)}]\right) \\ &= \sum_{k=1}^n \sum_{l=1}^n v_k v_l \left(ma_1^{(k)}a_1^{(l)}\sigma_c^2 t^2 + m[a_1^{(k)}a_2^{(l)} + a_2^{(k)}a_1^{(l)}][\overline{(x_c - \bar{x}_{ci})^3} + 2\bar{x}_c\sigma_c^2]t^3 + o(t^3)\right) \\ &= m\theta_1\sigma_c^2 t^2 + 2m\theta_2[\overline{(x_c - \bar{x}_{ci})^3} + 2\bar{x}_c\sigma_c^2]t^3 + o(t^3) \\ &= \beta_{c2}t^2 + \beta_{c3}t^3 + o(t^3), \end{aligned} \tag{129}$$

where $\beta_{c2} = m\theta_1\sigma_c^2$ and $\beta_{c3} = 2m\theta_2[\overline{(x_c - \bar{x}_{ci})^3} + 2\bar{x}_c\sigma_c^2]$.

On the other hand, we obtain

$$\begin{aligned}
SS_D(\boldsymbol{v}^\top \hat{\boldsymbol{X}}_1, \boldsymbol{v}^\top \hat{\boldsymbol{X}}_2) =& \frac{m}{2}(\overline{\boldsymbol{v}^\top \hat{\mathbf{x}}_1} - \overline{\boldsymbol{v}^\top \hat{\mathbf{x}}_2})^2 \\
=& \frac{1}{2m}\left(\sum_{i=1}^{m}[\boldsymbol{v}^\top \hat{\mathbf{x}}_{1i} - \boldsymbol{v}^\top \hat{\mathbf{x}}_{2i}]\right)^2 \\
=& \frac{1}{2m}\left(\sum_{i=1}^{m}\sum_{k=1}^{n} v_k(\hat{x}_{1i}^{(k)} - \hat{x}_{2i}^{(k)})\right)^2 \\
=& \frac{1}{2m}\left(\sum_{i=1}^{m}\sum_{k=1}^{n} v_k[a_1^{(k)}(x_{1i} - x_{2i})t + a_2^{(k)}(x_{1i}^2 - x_{2i}^2)t^2 + o(t^2)]\right)^2 \\
=& \frac{m}{2}\left(\sum_{k=1}^{n} v_k[a_1^{(k)}(\bar{x}_1 - \bar{x}_2)t + a_2^{(k)}(\overline{x_1^2} - \overline{x_2^2})t^2 + o(t^2)]\right)^2 \\
=& \frac{m}{2}\sum_{k=1}^{n}\sum_{l=1}^{n} v_k v_l[a_1^{(k)}a_1^{(l)}(\bar{x}_1 - \bar{x}_2)^2t^2 + [a_1^{(k)}a_2^{(l)} + a_2^{(k)}a_1^{(l)}](\bar{x}_1 - \bar{x}_2)(\overline{x_1^2} - \overline{x_2^2})t^3 + o(t^3)] \\
=& \frac{m}{2}\theta_1(\bar{x}_1 - \bar{x}_2)^2t^2 + m\theta_2(\bar{x}_1 - \bar{x}_2)(\overline{x_1^2} - \overline{x_2^2})t^3 + o(t^3) \\
=& \beta_2 t^2 + \beta_3 t^3 + o(t^3),
\end{aligned} \tag{130}$$

where $\beta_2 = \frac{m}{2}\theta_1(\bar{x}_1 - \bar{x}_2)^2$ and $\beta_3 = m\theta_2(\bar{x}_1 - \bar{x}_2)(\overline{x_1^2} - \overline{x_2^2})$. Therefore, we figure out that

$$\begin{aligned}
&\beta_2(\beta_{13} + \beta_{23}) - \beta_3(\beta_{12} + \beta_{22}) \\
=& m^2\theta_1\theta_2(\bar{x}_1 - \bar{x}_2)^2[\overline{(x_1 - \bar{x}_1)^3} + 2\bar{x}_1\sigma_1^2 + \overline{(x_2 - \bar{x}_2)^3} + 2\bar{x}_2\sigma_2^2] - m^2\theta_1\theta_2(\bar{x}_1 - \bar{x}_2)(\overline{x_1^2} - \overline{x_2^2})(\sigma_1^2 + \sigma_2^2) \\
=& m^2\theta_1\theta_2(\bar{x}_1 - \bar{x}_2)^2[\overline{(x_1 - \bar{x}_1)^3} + \overline{(x_2 - \bar{x}_2)^3}] + m^2\theta_1\theta_2(\bar{x}_1 - \bar{x}_2)(\sigma_1^2 - \sigma_2^2)[(\bar{x}_1 - \bar{x}_2)^2 - (\sigma_1^2 + \sigma_2^2)] \\
=& m^2\theta_1\theta_2 T_1 + m^2\theta_1\theta_2 T_2,
\end{aligned} \tag{131}$$

and

$$\begin{aligned}
(\beta_{12} + \beta_{22} + \beta_2)^2 &= \left[m\theta_1\sigma_1^2 + m\theta_1\sigma_2^2 + \frac{m}{2}\theta_1(\bar{x}_1 - \bar{x}_2)^2\right]^2 \\
&= \frac{1}{4}m^2\theta_1^2[2\sigma_1^2 + 2\sigma_2^2 + (\bar{x}_1 - \bar{x}_2)^2]^2 \\
&= \frac{1}{4}m^2\theta_1^2 T_3.
\end{aligned} \tag{132}$$

We hence obtain

$$SSR(\boldsymbol{v}^\top \hat{\boldsymbol{X}}_1, \boldsymbol{v}^\top \hat{\boldsymbol{X}}_2) = LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2) + \frac{4\theta_2}{\theta_1}\frac{T_1 + T_2}{T_3}t + o(t). \tag{133}$$

Similarly, $f'_{SSR}(0) \neq 0$ means $T_1 + T_2 \neq 0$, we can find some t and some $\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{v}$, such that $\theta_2 \neq 0$, and $SSR(\boldsymbol{v}^\top \hat{\boldsymbol{X}}_1, \boldsymbol{v}^\top \hat{\boldsymbol{X}}_2) < LSSR(\boldsymbol{X}_1, \boldsymbol{X}_2)$. We figure out that in the n-dimensional case, $f'_{SSR}(0) \neq 0$ is also required in our proof.

Hereafter, the remaining proof is nearly the same as the 2-dimensional version.

Take our 2-dimensional $\varphi(t;\cdot)$ as an example, $\boldsymbol{w} = [1,0]^\top, \boldsymbol{b} = [0,1]^\top, \boldsymbol{v} = [1,1]^\top$. We thus have $s_1 = 0, s_2 = 1$. Furthermore, we obtain $a_1^{(1)} = 1, a_1^{(2)} = 0, a_2^{(1)} = 0, a_2^{(2)} = -\frac{1}{2}$ and then $\theta_1 = 1, \theta_2 = -\frac{1}{2}$. As a result, we have $4\theta_2/\theta_1 = -2$, which is the same as Eqn.109.

D. Proofs Related to the Algorithms

D.1. Proof of Proposition 4

Proposition 4. *For any input $\boldsymbol{X}^{(0)}$, we can find some $\mathbf{u}$, such that*

$$\varphi_1^{(1)} : \mathbf{x}_k^{(0)} \mapsto \boldsymbol{p}_k^{(1)} = [\boldsymbol{u}^\top \mathbf{x}_k^{(0)}, 0]^\top, \tag{134}$$

where $\boldsymbol{p}_i^{(1)} \neq \boldsymbol{p}_j^{(1)}$ *if* $\mathbf{x}_i^{(0)} \neq \mathbf{x}_j^{(0)}$.

Proof. In reverse, we consider to find all the $\boldsymbol{u}$, such that some two different points are coincident after the projection. Given two points $\mathbf{x}_i^{(0)} \neq \mathbf{x}_j^{(0)}$ in $\boldsymbol{X}^{(0)}$, if $\boldsymbol{u}$ project them into the same point, we have

$$\boldsymbol{u}^\top \mathbf{x}_i^{(0)} = \boldsymbol{u}^\top \mathbf{x}_j^{(0)}. \tag{135}$$

We use $\mathbb{S}_2(\mathbf{x}_i^{(0)}, \mathbf{x}_j^{(0)})$ to denote the whole solution space of Eqn.135, namely

$$\mathbb{S}_2(\mathbf{x}_i^{(0)}, \mathbf{x}_j^{(0)}) = \{\boldsymbol{u} \in \mathbb{R}^d : \boldsymbol{u}^\top (\mathbf{x}_i^{(0)} - \mathbf{x}_j^{(0)}) = 0\}. \tag{136}$$

Considering all the pairs of different points, we define

$$\hat{\mathbb{S}}_2(\boldsymbol{X}^{(0)}) = \bigcup_{\mathbf{x}_i^{(0)} \neq \mathbf{x}_j^{(0)}} \mathbb{S}_2(\mathbf{x}_i^{(0)}, \mathbf{x}_j^{(0)}) \tag{137}$$

Since $\mathbf{x}_i^{(0)} \neq \mathbf{x}_j^{(0)}$, we find each solution space $\mathbb{S}_2(\mathbf{x}_i^{(0)}, \mathbf{x}_j^{(0)})$ is $(d-1)$ dimensional, and the number of such sets is no more than m^2. Therefore, the union of these solution spaces[13] is still smaller than $\mathbb{R}^d$, namely $\hat{\mathbb{S}}_2(\boldsymbol{X}^{(0)}) \subset \mathbb{R}^d$.

We obtain that $\exists \mathbf{x}_i^{(0)} \neq \mathbf{x}_j^{(0)}, \boldsymbol{u}^\top \mathbf{x}_i^{(0)} = \boldsymbol{u}^\top \mathbf{x}_j^{(0)}$, if and only if $\boldsymbol{u} \in \hat{\mathbb{S}}_2(\boldsymbol{X}^{(0)})$. Since $\hat{\mathbb{S}}_2(\boldsymbol{X}^{(0)}) \subset \mathbb{R}^d$, we obtain

$$\mathbb{R}^d / \hat{\mathbb{S}}_2(\boldsymbol{X}^{(0)}) \neq \emptyset. \tag{138}$$

Therefore, we can always find a $\boldsymbol{u} \in \mathbb{R}^d$, such that we have $\boldsymbol{p}_i^{(1)} \neq \boldsymbol{p}_j^{(1)}$, for any $\mathbf{x}_i^{(0)} \neq \mathbf{x}_j^{(0)}$. □

D.2. Proof of Proposition 5

Proposition 5. *For each layer,* $\varphi_l^{(1)} (2 \leq l \leq L)$ *only merges points with the same label. Nevertheless,* $\varphi_1^{(1)}, SP(\cdot)$ *and* $\varphi_l^{(2)} (1 \leq l \leq L-1)$ *do not merge any points.*

Proof. 1) According to Proposition 4, we figure out that $\varphi_1^{(1)}$ does not merge any points.

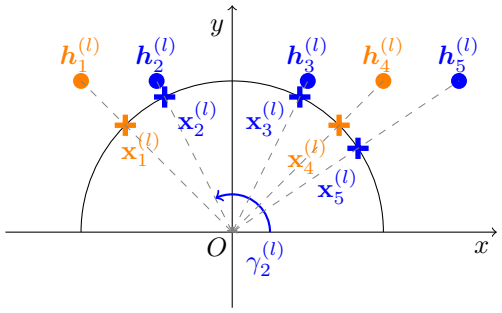


Figure A2. A copied figure from Figure 2(b).

2) Furthermore, we analyze $SP(\cdot)$. Focused on $\gamma_k^{(l)}$ (there is an example of $\gamma_2^{(l)}$ copied from Figure 2(b)), we figure out that

$$\gamma_k^{(l)} = \arctan \frac{2p_k^{(l)} - p_i^{(l)} - p_j^{(l)}}{p_j^{(l)} - p_i^{(l)}}, \tag{139}$$

where $p_i^{(l)}, p_j^{(l)}$ is defined in Algorithm 1. We can obtain that $\gamma_k^{(l)}$ is monotonically decreasing with $p_k^{(l)}$.

[13] The union of finite subspaces of $d-1$ dimensional can not cover the whole space $\mathbb{R}^d$.

When $\boldsymbol{h}_{k_i}^{(l)} \neq \boldsymbol{h}_{k_j}^{(l)}$, we have $\gamma_{k_i}^{(l)} \neq \gamma_{k_j}^{(l)}$, namely $\mathbf{x}_{k_i}^{(l)} \neq \mathbf{x}_{k_j}^{(l)}$. In other words, $SP(\cdot)$ does not merge any points.

3) We then consider $\varphi_l^{(2)}(1 \leq l \leq L-1)$. Obviously, $\varphi_l^{(2)}$ is a translation transformation, which does not merge any points.

4) Finally, we consider $\varphi_l^{(1)}(2 \leq l \leq L-1)$. We first have

$$p_k^{(l+1)} = \begin{bmatrix} 0 & 1 \end{bmatrix} \varphi_l^{(1)}(\mathbf{x}_k^{(l)}) = \sin \gamma_k^{(l)}. \tag{140}$$

Accordingly, $p_k^{(l+1)}$ is also monotonically decreasing with $p_k^{(l)}$ when $\gamma_k^{(l)} \leq \frac{\pi}{2}$.

Given two points from different classes denoted as $p_{k_1}^{(l)}, p_{k_2}^{(l)}$, we discuss them under three cases.

Case 1: If $p_{k_1}^{(l)}, p_{k_2}^{(l)} > p_j^{(l)}$, we have $\gamma_{k_1}^{(l)}, \gamma_{k_2}^{(l)} < \frac{\pi}{4}$. Therefore, $p_k^{(l+1)}$ is monotonically decreasing with $p_k^{(l)}$. We have

$$p_{k_1}^{(l)} \neq p_{k_2}^{(l)} \Leftrightarrow \boldsymbol{p}_{k_1}^{(l+1)} \neq \boldsymbol{p}_{k_2}^{(l+1)}. \tag{141}$$

Case 2: If one of $p_{k_1}^{(l)}, p_{k_2}^{(l)}$ is less than $p_j^{(l)}$ and the other is not, then one of $p_{k_1}^{(l+1)}, p_{k_2}^{(l+1)}$ is lager than $\frac{\sqrt{2}}{2}$, while the other is not. We hence have $\boldsymbol{p}_{k_1}^{(l+1)} \neq \boldsymbol{p}_{k_2}^{(l+1)}$

Case 3: If $p_{k_1}^{(l)}, p_{k_2}^{(l)}$ are both less than $p_j^{(l)}$—this case will never happen, otherwise one of them belongs to the same class with $\boldsymbol{p}_i^{(l)}$, resulting $\boldsymbol{p}_j^{(l)}$ is not the leftmost point (with the same label as $\boldsymbol{p}_i^{(l)}$), which contradicts the definition of j.

Conclusively, we find the samples from different classes will not merge by $\varphi_l^{(1)}(1 \leq l \leq L-1)$.

Based on all the discussions above, we have proved Proposition 5. □

D.3. Proof of Proposition 6

Proposition 6. *Confusion refers to merging two points with different labels. If confusion happens when we project $\boldsymbol{X}^{(l+1)}$ onto the y-axis, there must be a parallelogram[14] consisting of four different points in $\boldsymbol{P}^{(l)}$.*

Proof. If confusion happens, we will merge some two points $\mathbf{x}_s^{(l)}$ and $\mathbf{x}_t^{(l)}$ with different labels. According to Eqn.139, we find that $\sin \gamma_s^{(l)} = \sin \gamma_t^{(l)}$. Since $\mathbf{x}_s^{(l)}$ and $\mathbf{x}_t^{(l)}$ are different points on the unit circle, we have $\gamma_s^{(l)} = \pi - \gamma_t^{(l)}$, namely they are symmetric about y-axis. Furthermore, $\boldsymbol{h}_s^{(l)}$ and $\boldsymbol{h}_t^{(l)}$ are symmetric about y-axis. Besides, $\boldsymbol{h}_i^{(l)}$ and $\boldsymbol{h}_j^{(l)}$ are also symmetric about y-axis. Since the four points are on the same line, we have $\boldsymbol{h}_i^{(l)} + \boldsymbol{h}_j^{(l)} = \boldsymbol{h}_s^{(l)} + \boldsymbol{h}_t^{(l)}$. For $\boldsymbol{H}^{(l)}$ is translated from $\boldsymbol{P}^{(l)}$, we have $\boldsymbol{p}_i^{(l)} + \boldsymbol{p}_j^{(l)} = \boldsymbol{p}_s^{(l)} + \boldsymbol{p}_t^{(l)}$. We hence find a parallelogram in $\boldsymbol{P}^{(l)}$. □

D.4. Proof of Proposition 7

Proposition 7. *We can always find $\boldsymbol{u}_l \in \mathbb{R}^2$ for Algorithm 2, such that there is no parallelograms in $\hat{\boldsymbol{P}}^{(l)}$, and no points merged in the algorithm.*

Proof. By Algorithm 2, we shift the points in $\boldsymbol{P}^{(l)}$ up by 1, and then projects onto the unit circle $x^2 + y^2 = 1$, namely

$$\tilde{\boldsymbol{p}}_i^{(l)} \leftarrow SP\left(\boldsymbol{p}_i^{(l)} + \begin{bmatrix} 0 & 1 \end{bmatrix}^\top\right). \tag{142}$$

We find all points in $\tilde{\boldsymbol{P}}^{(l)}$ are on the upper half circle. Obviously, any four different points in $\tilde{\boldsymbol{P}}^{(l)}$ can not form a parallelogram, for the quadrilateral has two adjacent obtuse angles. In other words, give four different points $\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{p}}_j, \tilde{\boldsymbol{p}}_s, \tilde{\boldsymbol{p}}_t$, we have $\tilde{\boldsymbol{p}}_i + \tilde{\boldsymbol{p}}_j \neq \tilde{\boldsymbol{p}}_s + \tilde{\boldsymbol{p}}_t$. Besides, if $\boldsymbol{p}_i^{(l)} \neq \boldsymbol{p}_j^{(l)}$, we have $\tilde{\boldsymbol{p}}_i^{(l)} \neq \tilde{\boldsymbol{p}}_j^{(l)}$.

We can intuitively identify the two claims above in Figure 3.

[14]The parallelogram may be degenerate. Given four points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$, if the sum of two points is the same with that of the other two, we regard they form a parallelogram.

Similarly, consider to merge different points together by $\boldsymbol{u}_l$, we can find $\boldsymbol{u}_l$ from the set

$$\hat{\mathbb{S}}_2(\tilde{\boldsymbol{P}}^{(l)}) = \bigcup_{\tilde{\boldsymbol{p}}_i^{(l)} \neq \tilde{\boldsymbol{p}}_j^{(l)}} \mathbb{S}_2(\tilde{\boldsymbol{p}}_i^{(l)}, \tilde{\boldsymbol{p}}_j^{(l)}), \tag{143}$$

where $\mathbb{S}_2(\tilde{\boldsymbol{p}}_i^{(l)}, \tilde{\boldsymbol{p}}_j^{(l)}) = \{\boldsymbol{u}_l \in \mathbb{R}^2 : \boldsymbol{u}_l^\top (\tilde{\boldsymbol{p}}_i^{(l)} - \tilde{\boldsymbol{p}}_j^{(l)}) = 0\}$.

We then consider to form a parallelogram. We need some $\boldsymbol{u}_l$, and four different points $\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{p}}_j, \tilde{\boldsymbol{p}}_s, \tilde{\boldsymbol{p}}_t$, such that $\boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_i^{(l)} + \boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_j^{(l)} = \boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_s^{(l)} + \boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_t^{(l)}$. Obviously, we can find $\boldsymbol{u}_l$ from the set

$$\hat{\mathbb{S}}_4(\tilde{\boldsymbol{P}}^{(l)}) = \bigcup_{(i,j,s,t) \in \mathbb{I}_4(\tilde{\boldsymbol{P}}^{(l)})} \mathbb{S}_4(\tilde{\boldsymbol{p}}_i^{(l)}, \tilde{\boldsymbol{p}}_j^{(l)}, \tilde{\boldsymbol{p}}_s^{(l)}, \tilde{\boldsymbol{p}}_t^{(l)}), \tag{144}$$

where

$$\mathbb{S}_4(\tilde{\boldsymbol{p}}_i^{(l)}, \tilde{\boldsymbol{p}}_j^{(l)}, \tilde{\boldsymbol{p}}_s^{(l)}, \tilde{\boldsymbol{p}}_t^{(l)}) = \{\boldsymbol{u}_l \in \mathbb{R}^2 : \boldsymbol{u}_l^\top (\tilde{\boldsymbol{p}}_i^{(l)} + \tilde{\boldsymbol{p}}_j^{(l)} - \tilde{\boldsymbol{p}}_s^{(l)} - \tilde{\boldsymbol{p}}_t^{(l)}) = 0\}, \tag{145}$$

and the index set

$$\mathbb{I}_4(\tilde{\boldsymbol{P}}^{(l)}) = \{(i,j,s,t) : \tilde{\boldsymbol{p}}_i^{(l)}, \tilde{\boldsymbol{p}}_j^{(l)}, \tilde{\boldsymbol{p}}_s^{(l)}, \tilde{\boldsymbol{p}}_t^{(l)} \text{ are different with each other}\}. \tag{146}$$

Similarly, we point out that $\hat{\mathbb{S}}_2(\tilde{\boldsymbol{P}}^{(l)})$ consists of[15] no more than m^2 spaces of 1-dimensional. On the other hand, since $\tilde{\boldsymbol{p}}_i + \tilde{\boldsymbol{p}}_j \neq \tilde{\boldsymbol{p}}_s + \tilde{\boldsymbol{p}}_t$ holds for any four different points in $\tilde{\boldsymbol{P}}^{(l)}$, each $\mathbb{S}_4(\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{p}}_j, \tilde{\boldsymbol{p}}_s, \tilde{\boldsymbol{p}}_t)$ is a 1-dimensional space. Therefore, $\hat{\mathbb{S}}_4(\tilde{\boldsymbol{P}}^{(l)})$ consists of no more than m^4 spaces of 1-dimension. We hence obtain that $\hat{\mathbb{S}}_4(\tilde{\boldsymbol{P}}^{(l)}) \cup \hat{\mathbb{S}}_2(\tilde{\boldsymbol{P}}^{(l)})$ consists of no more than $m^2 + m^4$ spaces of 1-dimension, namely

$$[\hat{\mathbb{S}}_4(\tilde{\boldsymbol{P}}^{(l)}) \cup \hat{\mathbb{S}}_2(\tilde{\boldsymbol{P}}^{(l)})] \subset \mathbb{R}^2. \tag{147}$$

We thus have that—there exists $\tilde{\boldsymbol{p}}_i^{(l)} \neq \tilde{\boldsymbol{p}}_j^{(l)}$ subjected to $\boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_i^{(l)} = \boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_j^{(l)}$, if and only if $\boldsymbol{u}_l \in \hat{\mathbb{S}}_2(\boldsymbol{P}^{(l)})$. On the other hand, we figure out that—there exists four different points $\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{p}}_j, \tilde{\boldsymbol{p}}_s, \tilde{\boldsymbol{p}}_t$ subjected to $\boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_i^{(l)} + \boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_j^{(l)} = \boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_s^{(l)} + \boldsymbol{u}_l^\top \tilde{\boldsymbol{p}}_t^{(l)}$, if and only if $\boldsymbol{u}_l \in \hat{\mathbb{S}}_4(\boldsymbol{P}^{(l)})$. Since $[\hat{\mathbb{S}}_4(\tilde{\boldsymbol{P}}^{(l)}) \cup \hat{\mathbb{S}}_2(\tilde{\boldsymbol{P}}^{(l)})] \subset \mathbb{R}^2$, we obtain $\mathbb{R}^2 / [\hat{\mathbb{S}}_4(\tilde{\boldsymbol{P}}^{(l)}) \cup \hat{\mathbb{S}}_2(\tilde{\boldsymbol{P}}^{(l)})] \neq \emptyset$. As a result, we can always find a $\boldsymbol{u}_l \in \mathbb{R}^2 / [\hat{\mathbb{S}}_4(\tilde{\boldsymbol{P}}^{(l)}) \cup \hat{\mathbb{S}}_2(\tilde{\boldsymbol{P}}^{(l)})]$ to ensure not to merge different points, and form no parallelograms in $\hat{\boldsymbol{P}}^{(l)}$ as well. □

D.5. Discussion on a Wider LN-Net

We figure out that the algorithm here is suitable for both binary and multi-class classifications. Before giving the algorithm, we propose two lemmas as follows.

Lemma 10. *Given $\boldsymbol{X}^{(l)}$ on the unit sphere, the necessary condition of $\overline{\mathbf{x}_i^{(l)}\mathbf{x}_j^{(l)}} // \overline{\mathbf{x}_s^{(l)}\mathbf{x}_t^{(l)}}$ is that $\angle \mathbf{x}_j^{(l)} O \mathbf{x}_s^{(l)} = \angle \mathbf{x}_i^{(l)} O \mathbf{x}_t^{(l)}$, where O is origin of coordinates.*

Proof. For $\overline{\mathbf{x}_i^{(l)}\mathbf{x}_j^{(l)}} // \overline{\mathbf{x}_s^{(l)}\mathbf{x}_t^{(l)}}$, we have

$$\mathbf{x}_j^{(l)} - \mathbf{x}_i^{(l)} = k\ (\mathbf{x}_t^{(l)} - \mathbf{x}_s^{(l)}) \tag{148}$$

where $k \neq 0$.

Accordingly, we figure out that

$$\mathbf{x}_j^{(l)} + k\ \mathbf{x}_s^{(l)} = \mathbf{x}_i^{(l)} + k\ \mathbf{x}_t^{(l)}, \tag{149}$$

and furthermore,

$$(\mathbf{x}_j^{(l)})^2 + 2k\ \mathbf{x}_j^{(l)} \cdot \mathbf{x}_s^{(l)} + k^2\ (\mathbf{x}_s^{(l)})^2 = (\mathbf{x}_i^{(l)})^2 + 2k\ \mathbf{x}_i^{(l)} \cdot \mathbf{x}_t^{(l)} + k^2\ (\mathbf{x}_t^{(l)})^2. \tag{150}$$

Since $\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)}, \mathbf{x}_s^{(l)}, \mathbf{x}_t^{(l)}$ are all on the unit sphere, we have $(\mathbf{x}_j^{(l)})^2 = (\mathbf{x}_j^{(l)})^2 = (\mathbf{x}_j^{(l)})^2 = (\mathbf{x}_j^{(l)})^2 = 1$. Therefore, we have $\mathbf{x}_j^{(l)} \cdot \mathbf{x}_s^{(l)} = \mathbf{x}_i^{(l)} \cdot \mathbf{x}_t^{(l)}$

[15] $\hat{\mathbb{S}}_2(\tilde{\boldsymbol{P}}^{(l)})$ is a point set of finite lines, hence can not cover the whole $\mathbb{R}^2$.

According to the cosine theorem, we have $|\overline{\mathbf{x}_j^{(l)}\mathbf{x}_s^{(l)}}| = |\overline{\mathbf{x}_i^{(l)}\mathbf{x}_t^{(l)}}|$. Furthermore, according to the central angle theorem, we have $\angle \mathbf{x}_j^{(l)} O \mathbf{x}_s^{(l)} = \angle \mathbf{x}_i^{(l)} O \mathbf{x}_t^{(l)}$.

□

Lemma 11. *Given $\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}$ which are different from each other, the solution space*

$$\mathbb{B}_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}) = \left\{ \boldsymbol{b} \in \mathbb{R}^n : \frac{(\boldsymbol{p}_i^{(l)} + \boldsymbol{b})^\top (\boldsymbol{p}_s^{(l)} + \boldsymbol{b})}{\|\boldsymbol{p}_i^{(l)} + \boldsymbol{b}\|_2 \|\boldsymbol{p}_s^{(l)} + \boldsymbol{b}\|_2} = \frac{(\boldsymbol{p}_j^{(l)} + \boldsymbol{b})^\top (\boldsymbol{p}_t^{(l)} + \boldsymbol{b})}{\|\boldsymbol{p}_j^{(l)} + \boldsymbol{b}\|_2 \|\boldsymbol{p}_t^{(l)} + \boldsymbol{b}\|_2} \right\} \tag{151}$$

is contained in a hypersurface of $n-1$ dimension.

Proof. We first loose the equation in $\mathbb{B}_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$ to a polynomial equation.

Ignoring the case $\boldsymbol{b} \in \{-\boldsymbol{p}_i^{(l)}, -\boldsymbol{p}_j^{(l)}, -\boldsymbol{p}_s^{(l)}, -\boldsymbol{p}_t^{(l)}\}$, we can loose the equation in $\mathbb{B}_4(\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{p}_s, \boldsymbol{p}_t)$ as

$$(\boldsymbol{p}_i^{(l)} + \boldsymbol{b})^\top (\boldsymbol{p}_s^{(l)} + \boldsymbol{b}) \|\boldsymbol{p}_j^{(l)} + \boldsymbol{b}\|_2 \|\boldsymbol{p}_t^{(l)} + \boldsymbol{b}\|_2 = (\boldsymbol{p}_j^{(l)} + \boldsymbol{b})^\top (\boldsymbol{p}_t^{(l)} + \boldsymbol{b}) \|\boldsymbol{p}_i^{(l)} + \boldsymbol{b}\|_2 \|\boldsymbol{p}_s^{(l)} + \boldsymbol{b}\|_2. \tag{152}$$

Furthermore, we loose it again to

$$[(\boldsymbol{p}_i^{(l)} + \boldsymbol{b})^\top (\boldsymbol{p}_s^{(l)} + \boldsymbol{b})]^2 \|\boldsymbol{p}_j^{(l)} + \boldsymbol{b}\|_2^2 \|\boldsymbol{p}_t^{(l)} + \boldsymbol{b}\|_2^2 = [(\boldsymbol{p}_j^{(l)} + \boldsymbol{b})^\top (\boldsymbol{p}_t^{(l)} + \boldsymbol{b})]^2 \|\boldsymbol{p}_i^{(l)} + \boldsymbol{b}\|_2^2 \|\boldsymbol{p}_s^{(l)} + \boldsymbol{b}\|_2^2. \tag{153}$$

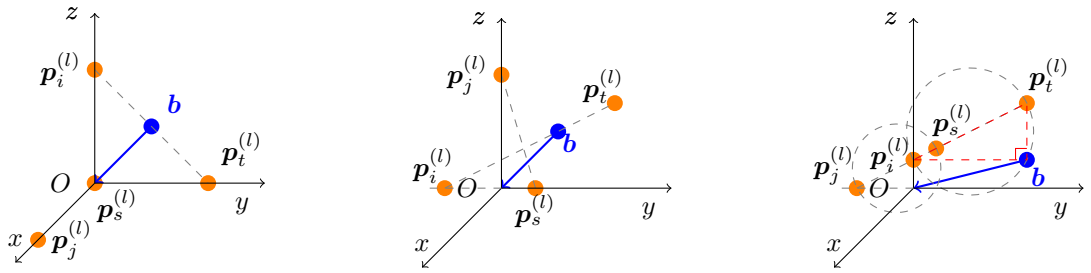
We define

$$\mathbb{B}'_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}) = \{\boldsymbol{b} : \boldsymbol{b} \text{ satisfies Eqn.153.}\}. \tag{154}$$

We find that $\boldsymbol{b} \in \mathbb{B}'_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$, for each $\boldsymbol{b} \in \mathbb{B}_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$. Since Eqn.153 is a polynomial equation about $\boldsymbol{b}$, its solution space $\mathbb{B}'_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$ is a hypersurface.

From $\mathbb{B}_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$ to $\mathbb{B}'_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$, we add the four singularities $\{-\boldsymbol{p}_i^{(l)}, -\boldsymbol{p}_j^{(l)}, -\boldsymbol{p}_s^{(l)}, -\boldsymbol{p}_t^{(l)}\}$, and we extend $\cos \angle \mathbf{x}_j^{(l)} O \mathbf{x}_s^{(l)} = \cos \angle \mathbf{x}_i^{(l)} O \mathbf{x}_t^{(l)}$ to $\cos^2 \angle \mathbf{x}_j^{(l)} O \mathbf{x}_s^{(l)} = \cos^2 \angle \mathbf{x}_i^{(l)} O \mathbf{x}_t^{(l)}$.

We then prove $\mathbb{B}'_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}) \subset \mathbb{R}^n$, to ensure it is a hypersurface of $d-1$ dimension.



(a) Four points are not on the same plane.

(b) Four points are on the same plane, but not on the same line.

(c) Four points are on the same line.

Figure A3. Three cases of the four points. We figure out that $\boldsymbol{b}$ is the shift direction and distance, and becomes the new origin when we translate $\boldsymbol{P}^{(l)}$ to $\boldsymbol{H}^{(l)}$.

Case 1: Suppose the four points $\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}$ are not on the same plane, as shown in Figure A3(a). Choose $\boldsymbol{b} = -(\boldsymbol{p}_i^{(l)} + \boldsymbol{p}_t^{(l)})/2$, we thus have $\angle \boldsymbol{h}_i^{(l)} O \boldsymbol{h}_t^{(l)} = \pi$. However, $\angle \boldsymbol{h}_j^{(l)} O \boldsymbol{h}_s^{(l)} \in (0, \pi)$, otherwise the four points will belong to the same plane. Therefore, $\boldsymbol{b} \notin \mathbb{B}'_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$.

Case 2: Suppose the four points $\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}$ are on the same plane, but not on the same line, as shown in Figure A3(b). We can always find $-\boldsymbol{b}$ on the **line segment** $\overline{\boldsymbol{p}_i^{(l)}\boldsymbol{p}_t^{(l)}}$, and ensure $-\boldsymbol{b}$ is not on the **line** $\overline{\boldsymbol{p}_j^{(l)}\boldsymbol{p}_s^{(l)}}$, otherwise the four points will be on the same line. We thus have $\angle \boldsymbol{h}_i^{(l)} O \boldsymbol{h}_t^{(l)} = \pi$, but $\angle \boldsymbol{h}_j^{(l)} O \boldsymbol{h}_s^{(l)} \in (0, \pi)$. Therefore, $\boldsymbol{b} \notin \mathbb{B}_4'(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$.

Case 3: Suppose the four points $\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}$ are on the same line, as shown in Figure A3(c). We can draw circles with $\overline{\boldsymbol{p}_i^{(l)}\boldsymbol{p}_t^{(l)}}$ and $\overline{\boldsymbol{p}_j^{(l)}\boldsymbol{p}_s^{(l)}}$, respectively. We can always find $-\boldsymbol{b}$ on the previous circle, but not on the later one, otherwise they will be not different from each other. We thus have $\angle \boldsymbol{h}_i^{(l)} O \boldsymbol{h}_t^{(l)} = \frac{\pi}{2}$, but $\angle \boldsymbol{h}_j^{(l)} O \boldsymbol{h}_s^{(l)} \neq \frac{\pi}{2}$. Therefore, $\boldsymbol{b} \notin \mathbb{B}_4'(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$.

Conclusively, we can always find some $\boldsymbol{b} \notin \mathbb{B}_4'(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$, then we have $\mathbb{B}_4'(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}) \subset \mathbb{R}^n$. Further, $\mathbb{B}_4'(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$ is a hypersurface of $d-1$ dimension, and $\mathbb{B}_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}) \subset \mathbb{B}_4'(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)})$ □

Here we propose the proposition of a wider LN-Net as follows.

Proposition 9. *A wider LN-Net can classify m samples with any label assignment.*

Proof. Similarly, we hope to merge two points from the same class, and do not merge other points meanwhile. Suppose LN acts on $\mathbb{R}^{n+1}$ by Lemma 1, we thus use SP on $\mathbb{R}^n$ for convenience. Given $\boldsymbol{P}^{(l)} \in \mathbb{R}^{n \times m}$ on a $n-1$ dimensional hyperplane, we consider to shift the points by $\boldsymbol{b} \in \mathbb{R}^n$ and get $\boldsymbol{H}^{(l)}$. After that, we spherically project $\boldsymbol{H}^{(l)}$ onto the unit sphere $\|\mathbf{x}\|_2 = 1$, represented by $\boldsymbol{X}^{(l+1)}$. Hereafter, we linearly project $\boldsymbol{X}^{(l+1)}$ onto another $n-1$ dimensional hyperplane.

Different from our method on $\mathbb{R}^2$, we can not sort the points, it is hence much harder to design a suitable algorithm in a high dimensional space. But we can consider to merge some $\boldsymbol{p}_i^{(l)}$ and $\boldsymbol{p}_j^{(l)}$ only, without merging the other points. We analyze the merging progress backward, and show how to find the projection direction and the bias $\boldsymbol{b}$.

To get $\boldsymbol{P}^{(l+1)}$ from $\boldsymbol{X}^{(l)}$, without doubt the projection direction is along $\overline{\mathbf{x}_i^{(l)}\mathbf{x}_j^{(l)}}$, and the target is some $n-1$ dimensional hyperplane. Now we need to ensure doing so will not merge other points. Obviously, its necessary and sufficient condition is that there are no other different points $\mathbf{x}_s^{(l)}, \mathbf{x}_t^{(l)}$, such that

$$\overline{\mathbf{x}_i^{(l)}\mathbf{x}_j^{(l)}} // \overline{\mathbf{x}_s^{(l)}\mathbf{x}_t^{(l)}}, \tag{155}$$

namely $\overline{\mathbf{x}_s^{(l)}\mathbf{x}_t^{(l)}}$ is parallel to the projection direction.

According to Lemma 10, for $\boldsymbol{X}^{(l)}$ is on the unit sphere, the necessary condition of $\overline{\mathbf{x}_i^{(l)}\mathbf{x}_j^{(l)}} // \overline{\mathbf{x}_s^{(l)}\mathbf{x}_t^{(l)}}$ is that—$\angle \mathbf{x}_i^{(l)} O \mathbf{x}_s^{(l)} = \angle \mathbf{x}_j^{(l)} O \mathbf{x}_t^{(l)}$, where O is the origin of coordinates.

Since $\boldsymbol{X}^{(l)} = SP(\boldsymbol{H}^{(l)})$, we have

$$\angle \mathbf{x}_i^{(l)} O \mathbf{x}_s^{(l)} = \angle \mathbf{x}_j^{(l)} O \mathbf{x}_t^{(l)} \Leftrightarrow \angle \boldsymbol{h}_i^{(l)} O \boldsymbol{h}_s^{(l)} = \angle \boldsymbol{h}_j^{(l)} O \boldsymbol{h}_t^{(l)}. \tag{156}$$

If we ensure any four different points in $\boldsymbol{H}^{(l)}$ to satisfy $\angle \boldsymbol{h}_i^{(l)} O \boldsymbol{h}_s^{(l)} \neq \angle \boldsymbol{h}_j^{(l)} O \boldsymbol{h}_t^{(l)}$, we will not merge other points when we merge $\mathbf{x}_i^{(l+1)}$ and $\mathbf{x}_j^{(l+1)}$. Since $\boldsymbol{h}_k^{(l)} = \boldsymbol{p}_k^{(l)} + \boldsymbol{b}$, according to the cosine theorem, we point out that $\angle \boldsymbol{h}_i^{(l)} O \boldsymbol{h}_s^{(l)} = \angle \boldsymbol{h}_j^{(l)} O \boldsymbol{h}_t^{(l)}$ is equivalent to

$$\frac{(\boldsymbol{p}_i^{(l)} + \boldsymbol{b})^\top (\boldsymbol{p}_s^{(l)} + \boldsymbol{b})}{\|\boldsymbol{p}_i^{(l)} + \boldsymbol{b}\|_2 \|\boldsymbol{p}_s^{(l)} + \boldsymbol{b}\|_2} = \frac{(\boldsymbol{p}_j^{(l)} + \boldsymbol{b})^\top (\boldsymbol{p}_t^{(l)} + \boldsymbol{b})}{\|\boldsymbol{p}_j^{(l)} + \boldsymbol{b}\|_2 \|\boldsymbol{p}_t^{(l)} + \boldsymbol{b}\|_2}. \tag{157}$$

We define

$$\mathbb{B}_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}) = \left\{ \boldsymbol{b} \in \mathbb{R}^n : \frac{(\boldsymbol{p}_i^{(l)} + \boldsymbol{b})^\top (\boldsymbol{p}_s^{(l)} + \boldsymbol{b})}{\|\boldsymbol{p}_i^{(l)} + \boldsymbol{b}\|_2 \|\boldsymbol{p}_s^{(l)} + \boldsymbol{b}\|_2} = \frac{(\boldsymbol{p}_j^{(l)} + \boldsymbol{b})^\top (\boldsymbol{p}_t^{(l)} + \boldsymbol{b})}{\|\boldsymbol{p}_j^{(l)} + \boldsymbol{b}\|_2 \|\boldsymbol{p}_t^{(l)} + \boldsymbol{b}\|_2} \right\}. \tag{158}$$

Since $\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}$ are different from each other, the solution space of Eqn.157 about $\boldsymbol{b}$ is contained in a hypersurface of $n-1$ dimension, by Lemma 11.

Again, we define

$$\hat{\mathbb{B}}_4(\boldsymbol{P}^{(l)}) = \bigcup_{(i,j,s,t)\in\mathbb{I}_4(\boldsymbol{P}^{(l)})} \mathbb{B}_4(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)}), \tag{159}$$

where

$$\mathbb{I}_4(\boldsymbol{P}^{(l)}) = \{(i,j,s,t) : \boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}, \boldsymbol{p}_s^{(l)}, \boldsymbol{p}_t^{(l)} \text{ are different with each other}\}. \tag{160}$$

We figure out that $\hat{\mathbb{B}}_4(\boldsymbol{P}^{(l)})$ is contained in a union of no more than m^4 hypersurfaces of $n-1$ dimension.

Besides, from $\boldsymbol{P}^{(l)}$ to $\boldsymbol{X}^{(l)}$, we can not merge any two different points. Therefore, given $\boldsymbol{p}_i^{(l)} \neq \boldsymbol{p}_j^{(l)}$, we need

$$(\boldsymbol{p}_i^{(l)} + \boldsymbol{b})/\|\boldsymbol{p}_i^{(l)} + \boldsymbol{b}\|_2 \neq (\boldsymbol{p}_j^{(l)} + \boldsymbol{b})/\|\boldsymbol{p}_j^{(l)} + \boldsymbol{b}\|_2.$$

Given two different points $\boldsymbol{p}_i, \boldsymbol{p}_j$, we define

$$\mathbb{B}_2(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}) = \left\{ \boldsymbol{b} \in \mathbb{R}^n : \frac{\boldsymbol{p}_i^{(l)} + \boldsymbol{b}}{\|\boldsymbol{p}_i^{(l)} + \boldsymbol{b}\|_2} = \frac{\boldsymbol{p}_j^{(l)} + \boldsymbol{b}}{\|\boldsymbol{p}_j^{(l)} + \boldsymbol{b}\|_2} \right\}. \tag{161}$$

Similarly, we can prove that $\mathbb{B}_2(\boldsymbol{p}_i, \boldsymbol{p}_j)$ is contained in a hypersurface of $n-1$ dimension. We find $\hat{\mathbb{B}}_2(\boldsymbol{P}^{(l)})$ is contained in the union of no more than m^2 hypersurfaces of $n-1$ dimension, where

$$\hat{\mathbb{B}}_2(\boldsymbol{P}^{(l)}) = \bigcup_{\boldsymbol{p}_i^{(l)} \neq \boldsymbol{p}_j^{(l)}} \mathbb{B}_2(\boldsymbol{p}_i^{(l)}, \boldsymbol{p}_j^{(l)}). \tag{162}$$

We figure out that $\hat{\mathbb{B}}_2(\boldsymbol{P}^{(l)}) \cup \hat{\mathbb{B}}_4(\boldsymbol{P}^{(l)})$ is contained in a union of no more than $m^2 + m^4$ hypersurfaces of $n-1$ dimension.

Therefore, we have

$$[\hat{\mathbb{B}}_2(\boldsymbol{P}^{(l)}) \cup \hat{\mathbb{B}}_4(\boldsymbol{P}^{(l)})] \subset \mathbb{R}^n \tag{163}$$

Choose some $\boldsymbol{b} \in \mathbb{R}^n / [\hat{\mathbb{B}}_2(\boldsymbol{P}^{(l)}) \cup \hat{\mathbb{B}}_4(\boldsymbol{P}^{(l)})]$, then $\angle \boldsymbol{h}_j^{(l)} O \boldsymbol{h}_s^{(l)} = \angle \boldsymbol{h}_i^{(l)} O \boldsymbol{h}_t^{(l)}$ will not holds. Furthermore, by Lemma 10, $\overline{\mathbf{x}_i^{(l)}\mathbf{x}_j^{(l)}} // \overline{\mathbf{x}_s^{(l)}\mathbf{x}_t^{(l)}}$ will not holds either. As a result, we can only merge $\boldsymbol{p}_i^{(l)}$ and $\boldsymbol{p}_j^{(l)}$ by projection.

In conclusion, we can choose to only merge two samples with the same label each step by the method above. Furthermore, we can construct an LN-Net with depth $O(m)$ to classify m samples with any label assignment. Note the width of LN-Net here is wider than 3, and we do not require the widths of each layer are equal. □

E. Proof of Proposition 8

Proposition 8. *Given $g \leq d/3$, we have*

$$\frac{\mathcal{H}(\psi_G(g;\cdot);\mathbf{x})}{\mathcal{H}(\psi_L(\cdot);\mathbf{x})} \geq 1. \tag{164}$$

Specifically, when $g = d/4$, we figure out that

$$\frac{\mathcal{H}(\psi_G(g;\cdot);\mathbf{x})}{\mathcal{H}(\psi_L(\cdot);\mathbf{x})} \geq \frac{d}{8}. \tag{165}$$

In the proof of Proposition 8, we consider a single sample only. We use x_i as the i-th ordinate of $\mathbf{x}$ instead of $x^{(i)}$ in this proof, we thus use x_i^2 to denote the squares rather than $[x^{(i)}]^2$.

E.1. Required Lemmas for the Proof

Lemma 12. *Given $\mathbf{x} \in \mathbb{R}^d$, $\mu = (x_1 + \cdots + x_d)/d$ and $\sigma^2 = [(x_1 - \mu)^2 + \cdots + (x_d - \mu)^2]/d$, we denote $LN(\mathbf{x})$ as $\hat{\mathbf{x}} = (\mathbf{x} - \mu\mathbf{1})/\sigma$. We point out that*

$$\mathcal{H}(\psi_L(\cdot);\mathbf{x}) = \frac{3}{\sigma^4} - \frac{6}{d\sigma^4} \tag{166}$$

Proof. To begin with, we regard $\hat{x}_i$ as $\psi_i(\mathbf{x})$, and then give the gradient $\nabla_{\mathbf{x}}\psi_i(\mathbf{x})$. Let $s = \frac{1}{d}\sum_{i=1}^{d}(x_i-\mu)^2$ and $\sigma = \sqrt{s}$. We have

$$\frac{\partial \mu}{\partial x_i} = \frac{1}{d}, \forall i, \tag{167}$$

$$\begin{aligned}\frac{\partial s}{\partial x_i} &= \frac{1}{d}\frac{\partial}{\partial x_i}\sum_{j=1}^{d}(x_j-\mu)^2 \\ &= \frac{1}{d}\frac{\partial}{\partial x_i}\sum_{j=1}^{d}x_j^2 - \frac{1}{d}\frac{\partial}{\partial x_i}d\mu^2 \\ &= \frac{2}{d}(x_i-\mu), \forall i,\end{aligned} \tag{168}$$

and

$$\begin{aligned}\frac{\partial \sigma}{\partial x_i} &= \frac{1}{2\sqrt{s}}\frac{\partial s}{\partial x_i} \\ &= \frac{x_i-\mu}{d\sigma} = \frac{\hat{x}_i}{d}, \forall i.\end{aligned} \tag{169}$$

We thus obtain

$$\begin{aligned}\frac{\partial \hat{x}_i}{\partial x_i} &= \frac{1}{\sigma}\frac{\partial}{\partial x_i}(x_i-\mu) + (x_i-\mu)\frac{\partial}{\partial x_i}(\frac{1}{\sigma}) \\ &= \frac{1}{\sigma}(1-\frac{1}{d}) - \frac{\hat{x}_i}{\sigma}\frac{\partial \sigma}{\partial x_i} \\ &= \frac{1}{d\sigma}(d-1-\hat{x}_i^2).\end{aligned} \tag{170}$$

While for $j \neq i$, we have

$$\begin{aligned}\frac{\partial \hat{x}_i}{\partial x_j} &= \frac{1}{\sigma}\frac{\partial}{\partial x_j}(x_i-\mu) + (x_i-\mu)\frac{\partial}{\partial x_j}(\frac{1}{\sigma}) \\ &= \frac{1}{\sigma}(0-\frac{1}{d}) - \frac{\hat{x}_i}{\sigma}\frac{\partial \sigma}{\partial x_j} \\ &= \frac{1}{d\sigma}(-1-\hat{x}_i\hat{x}_j).\end{aligned} \tag{171}$$

Based above, we calculate the Hessian matrix. For each term $\dfrac{\partial^2 \hat{x}_i}{\partial x_j \partial x_k}$, we figure out that there are four kinds of the second order derivative.

Case 1, $i = j = k$:

$$\begin{aligned}\frac{\partial^2 \hat{x}_i}{\partial x_i^2} &= -\frac{1}{d\sigma^2}(d-1-\hat{x}_i^2)\frac{\partial \sigma}{\partial x_i} - \frac{2\hat{x}_i}{d\sigma}\frac{\partial \hat{x}_i}{\partial x_i} \\ &= -\frac{1}{d^2\sigma^2}(d-1-\hat{x}_i^2)\hat{x}_i - \frac{2\hat{x}_i}{d^2\sigma^2}(d-1-\hat{x}_i^2) \\ &= \frac{1}{d^2\sigma^2}[3\hat{x}_i^3 - 3(d-1)\hat{x}_i] \\ &= \frac{1}{d^2\sigma^2}(3\hat{x}_i^3 + 3\hat{x}_i) - \frac{3\hat{x}_i}{d\sigma^2}.\end{aligned} \tag{172}$$

Case 2, only one of j, k equals to i, assume $i = k$:

$$\begin{aligned}\frac{\partial^2 \hat{x}_i}{\partial x_i \partial x_j} &= -\frac{1}{d\sigma^2}(d - 1 - \hat{x}_i^2)\frac{\partial \sigma}{\partial x_j} - \frac{2\hat{x}_i}{d\sigma}\frac{\partial \hat{x}_i}{\partial x_j} \\ &= -\frac{1}{d^2\sigma^2}(d - 1 - \hat{x}_i^2)\hat{x}_j - \frac{2\hat{x}_i}{d^2\sigma^2}(-1 - \hat{x}_i\hat{x}_j) \\ &= \frac{1}{d^2\sigma^2}[3\hat{x}_i^2\hat{x}_j + 2\hat{x}_i - (d-1)\hat{x}_j] \\ &= \frac{1}{d^2\sigma^2}(3\hat{x}_i^2\hat{x}_j + 2\hat{x}_i + \hat{x}_j) - \frac{\hat{x}_j}{d\sigma^2}.\end{aligned} \tag{173}$$

We have that $\dfrac{\partial^2 \hat{x}_i}{\partial x_j \partial x_k} = \dfrac{\partial^2 \hat{x}_i}{\partial x_k \partial x_j}$, so the result of the other case $i = j$ has the same form with that of $i = k$.

Case 3, $j = k$, but $i \neq j$:

$$\begin{aligned}\frac{\partial^2 \hat{x}_i}{\partial x_j^2} &= -\frac{1}{d\sigma^2}(-1 - \hat{x}_i\hat{x}_j)\frac{\partial \sigma}{\partial x_j} - \frac{\hat{x}_i}{d\sigma}\frac{\partial \hat{x}_j}{\partial x_j} - \frac{\hat{x}_j}{d\sigma}\frac{\partial \hat{x}_i}{\partial x_j} \\ &= -\frac{1}{d^2\sigma^2}(-1 - \hat{x}_i\hat{x}_j)\hat{x}_j - \frac{\hat{x}_i}{d^2\sigma^2}(d - 1 - \hat{x}_j^2) - \frac{\hat{x}_j}{d^2\sigma^2}(-1 - \hat{x}_i\hat{x}_j) \\ &= \frac{1}{d^2\sigma^2}[3\hat{x}_i\hat{x}_j^2 + 2\hat{x}_j - (d-1)\hat{x}_i] \\ &= \frac{1}{d^2\sigma^2}(3\hat{x}_i\hat{x}_j^2 + 2\hat{x}_j + \hat{x}_i) - \frac{\hat{x}_i}{d\sigma^2}.\end{aligned} \tag{174}$$

Case 4, i, j, k are different from each other:

$$\begin{aligned}\frac{\partial^2 \hat{x}_i}{\partial x_j \partial x_k} &= -\frac{1}{d\sigma^2}(-1 - \hat{x}_i\hat{x}_j)\frac{\partial \sigma}{\partial x_k} - \frac{\hat{x}_i}{d\sigma}\frac{\partial \hat{x}_j}{\partial x_k} - \frac{\hat{x}_j}{d\sigma}\frac{\partial \hat{x}_i}{\partial x_k} \\ &= -\frac{1}{d^2\sigma^2}(-1 - \hat{x}_i\hat{x}_j)\hat{x}_k - \frac{\hat{x}_i}{d^2\sigma^2}(-1 - \hat{x}_j\hat{x}_k) - \frac{\hat{x}_j}{d^2\sigma^2}(-1 - \hat{x}_i\hat{x}_k) \\ &= \frac{1}{d^2\sigma^2}(3\hat{x}_i\hat{x}_j\hat{x}_k + \hat{x}_i + \hat{x}_j + \hat{x}_k).\end{aligned} \tag{175}$$

It is hard to calculate the operator norm of the Hessian matrix is too difficult, so we calculate the Frobenius norm instead.

$$\begin{aligned}\left\|\frac{\partial^2 \hat{x}_i}{\partial \mathbf{x}^2}\right\|_F^2 &= \sum_{j=1}^{d}\sum_{k=1}^{d}\left(\frac{\partial^2 \hat{x}_i}{\partial x_j \partial x_k}\right)^2 \\ &= \sum_{j=1}^{d}\sum_{k=1}^{d}\frac{1}{d^4\sigma^4}(3\hat{x}_i\hat{x}_j\hat{x}_k + \hat{x}_i + \hat{x}_j + \hat{x}_k)^2 + \sum_{j\neq i}\left[\frac{\hat{x}_i^2}{d^2\sigma^4} - 2\frac{\hat{x}_i}{d^3\sigma^4}(3\hat{x}_i\hat{x}_j^2 + 2\hat{x}_j + \hat{x}_i)\right] \\ &\quad + 2\sum_{j\neq i}\left[\frac{\hat{x}_j^2}{d^2\sigma^4} - 2\frac{\hat{x}_j}{d^3\sigma^4}(3\hat{x}_i^2\hat{x}_j + 2\hat{x}_i + \hat{x}_j)\right] + \frac{9\hat{x}_i^2}{d^2\sigma^4} - 2\frac{3\hat{x}_i}{d^3\sigma^4}(3\hat{x}_i^3 + 3\hat{x}_i) \\ &= \sum_{j=1}^{d}\sum_{k=1}^{d}\frac{1}{d^4\sigma^4}(3\hat{x}_i\hat{x}_j\hat{x}_k + \hat{x}_i + \hat{x}_j + \hat{x}_k)^2 + \sum_{j=1}^{d}\left[\frac{\hat{x}_i^2}{d^2\sigma^4} - 2\frac{\hat{x}_i}{d^3\sigma^4}(3\hat{x}_i\hat{x}_j^2 + 2\hat{x}_j + \hat{x}_i)\right] \\ &\quad + 2\sum_{j=1}^{d}\left[\frac{\hat{x}_j^2}{d^2\sigma^4} - 2\frac{\hat{x}_j}{d^3\sigma^4}(3\hat{x}_i^2\hat{x}_j + 2\hat{x}_i + \hat{x}_j)\right] + \frac{6\hat{x}_i^2}{d^2\sigma^4}\end{aligned} \tag{176}$$

We note that

$$\sum_{j=1}^{d}\hat{x}_j = 0, \sum_{j=1}^{d}\hat{x}_j^2 = d. \tag{177}$$

We thus have

$$\begin{aligned}
&\sum_{j=1}^{d}\sum_{k=1}^{d}\frac{1}{d^4\sigma^4}(3\hat{x}_i\hat{x}_j\hat{x}_k+\hat{x}_i+\hat{x}_j+\hat{x}_k)^2\\
=&\sum_{j=1}^{d}\sum_{k=1}^{d}\frac{1}{d^4\sigma^4}[9\hat{x}_i^2\hat{x}_j^2\hat{x}_k^2+6\hat{x}_i\hat{x}_j\hat{x}_k(\hat{x}_i+\hat{x}_j+\hat{x}_k)+(\hat{x}_i+\hat{x}_j+\hat{x}_k)^2]\\
=&\frac{9\hat{x}_i^2}{d^2\sigma^4}+0+\sum_{j=1}^{d}\sum_{k=1}^{d}\frac{1}{d^4\sigma^4}(\hat{x}_i+\hat{x}_j+\hat{x}_k)^2\\
=&\frac{10\hat{x}_i^2}{d^2\sigma^4}+\frac{2}{d^2\sigma^4},
\end{aligned}\tag{178}$$

$$\sum_{j=1}^{d}\left[\frac{\hat{x}_i^2}{d^2\sigma^4}-2\frac{\hat{x}_i}{d^3\sigma^4}(3\hat{x}_i\hat{x}_j^2+2\hat{x}_j+\hat{x}_i)\right]=\frac{\hat{x}_i^2}{d\sigma^4}-\frac{8\hat{x}_i^2}{d^2\sigma^4},\tag{179}$$

and

$$2\sum_{j=1}^{d}\left[\frac{\hat{x}_j^2}{d^2\sigma^4}-2\frac{\hat{x}_j}{d^3\sigma^4}(3\hat{x}_i^2\hat{x}_j+2\hat{x}_i+\hat{x}_j)\right]=\frac{2}{d\sigma^4}-\frac{12\hat{x}_i^2}{d^2\sigma^4}-\frac{4}{d^2\sigma^4}.\tag{180}$$

Take Eqn.178, Eqn.179 and Eqn.180 into Eqn.176, we obtain

$$\left\|\frac{\partial^2\hat{x}_i}{\partial\mathbf{x}^2}\right\|_F^2=\frac{\hat{x}_i^2+2}{d\sigma^4}-\frac{4\hat{x}_i^2+2}{d^2\sigma^4}\tag{181}$$

Now we add up all the dimensions, as LN's information of the second order

$$\mathcal{H}(\psi_L(\cdot);\mathbf{x})=\sum_{i=1}^{d}\left\|\frac{\partial^2\hat{x}_i}{\partial\mathbf{x}^2}\right\|_F^2=\frac{3}{\sigma^4}-\frac{6}{d\sigma^4}=\frac{3}{d\sigma^4}(d-2).\tag{182}$$

When $d=2$, we have $\hat{x}_i^2=1$, and $\mathcal{H}(\psi_L(\cdot);\mathbf{x})|_{d=2}=0$ naturally.

□

Lemma 13. *Given $\mathbf{x}\in\mathbb{R}^d$, let the group number of GN be g. Suppose σ_i^2 is the variance of the i-th group, we have that*

$$\mathcal{H}(\psi_G(g;\cdot);\mathbf{x})=\sum_{i=1}^{g}\left(\frac{3}{\sigma_i^4}-\frac{6g}{d\sigma_i^4}\right)\tag{183}$$

Proof. We simplify $\psi_G(g;\cdot)$ as $\psi(\cdot)$ in the proof here. As for Group Normalization, suppose the number of groups is g, and $d=g\times c$. Let $\mathbf{x}=[\boldsymbol{z}_1^\top,\cdots,\boldsymbol{z}_g^\top]^\top$, where $\boldsymbol{z}_i=[z_{i1},\cdots,z_{ic}]^\top,(i=1,\cdots,g)$. Assume $\mathbf{x}=[x_1,\cdots,x_d]^\top$, we denote that $z_{ij}=x_{(i-1)\times c+j}$.

Let $\hat{\mathbf{x}}=GN(\mathbf{x})$, where $GN(\cdot)$ denotes the Group Normalization operation. GN can be calculated by $\mu_i=(z_{i1}+\cdots+z_{ic})/c$, $\sigma_i^2=[(z_{i1}-\mu_i)^2+\cdots+(z_{ic}-\mu_i)^2]/c$, and then $\hat{z}_{ij}=(z_{ij}-\mu_i)/\sigma_i$. Accordingly, we denote $\hat{\mathbf{x}}=[\hat{\boldsymbol{z}}_1^\top,\cdots,\hat{\boldsymbol{z}}_g^\top]^\top$, where $\hat{\boldsymbol{z}}_i=LN(\boldsymbol{z}_i),(i=1,\cdots,g)$. To begin with ,we denote $GN(\mathbf{x})$ as $\psi(\mathbf{x})=[\psi_{11}(\mathbf{x}),\psi_{12}(\mathbf{x}),\cdots,\psi_{gc}(\mathbf{x})]$. We thus have

$$\nabla_{\mathbf{x}}\psi_{ij}(\mathbf{x})=\begin{bmatrix}\nabla_{\boldsymbol{z}_1}\psi_{ij}(\mathbf{x})\\ \vdots\\ \nabla_{\boldsymbol{z}_g}\psi_{ij}(\mathbf{x})\end{bmatrix},(i=1,\cdots,g;j=1,\cdots,c).\tag{184}$$

We denote that $z_{ij} = \psi_{ij}(\mathbf{x})$. When $k \neq i$, we have $\nabla_{\boldsymbol{z}_k}\psi_{ij}(\mathbf{x}) = \mathbf{0}$. When $k \neq i$, we have $\nabla_{\boldsymbol{z}_i}\psi_{ij}(\mathbf{x})$ is a gradient of LN, for $[\psi_{i1}(\mathbf{x}), \cdots, \psi_{ic}(\mathbf{x})]^\top = LN(\boldsymbol{z}_i)$. We can give the Hessian matrix of GN, denoted as

$$\nabla^2_{\mathbf{x}}\psi_{ij}(\mathbf{x}) = \begin{bmatrix} \boldsymbol{O} & \cdots & \boldsymbol{O} \\ & \ddots & \\ \vdots & \nabla^2_{\boldsymbol{z}_i}\psi_{ij}(\mathbf{x}) & \vdots \\ & & \ddots \\ \boldsymbol{O} & \cdots & \boldsymbol{O} \end{bmatrix}, (i = 1, \cdots, g; j = 1, \cdots, c) \tag{185}$$

By the discussion about LN above, we obtain that

$$\|\nabla^2_{\boldsymbol{z}_i}\psi_{ij}(\mathbf{x})\|_F^2 = \frac{\hat{z}_{ij}^2 + 2}{c\sigma_i^4} - \frac{4\hat{z}_{ij}^2 + 2}{c^2\sigma_i^4}. \tag{186}$$

Obviously, we have $\|\nabla^2_{\mathbf{x}}\psi_{ij}(\mathbf{x})\|_F^2 = \|\nabla^2_{\boldsymbol{z}_i}\psi_{ij}(\mathbf{x})\|_F^2$. Although there are many zeros in $\nabla^2_{\mathbf{x}}\psi_{ij}(\mathbf{x})$, for $\sum_{j=1}^{c}\hat{x}_{ij}^2 = c$, we obtain

$$\begin{aligned} \mathcal{H}(\psi_G(g;\cdot);\mathbf{x}) &= \sum_{i=1}^{d}\left\|\frac{\partial^2\hat{x}_i}{\partial\mathbf{x}^2}\right\|_F^2 \\ &= \sum_{i=1}^{g}\sum_{j=1}^{c}\|\nabla^2_{\mathbf{x}}\psi_{ij}(\mathbf{x})\|_F^2 \\ &= \sum_{i=1}^{g}\sum_{j=1}^{c}\left(\frac{\hat{x}_{ij}^2 + 2}{c\sigma_i^4} - \frac{4\hat{x}_{ij}^2 + 2}{c^2\sigma_i^4}\right) \\ &= \sum_{i=1}^{g}\left(\frac{3}{\sigma_i^4} - \frac{6}{c\sigma_i^4}\right) \end{aligned} \tag{187}$$

□

Lemma 14. *In group normalization, we have*

$$\sigma^2 \geq \frac{1}{g}\sum_{i=1}^{g}\sigma_i^2. \tag{188}$$

Proof. According to the definition, we have

$$\begin{aligned} \sigma^2 - \frac{1}{g}\sum_{i=1}^{g}\sigma_i^2 &= \frac{1}{cg}\sum_{i=1}^{g}\sum_{j=1}^{c}(z_{ij} - \mu)^2 - \frac{1}{cg}\sum_{i=1}^{g}\sum_{j=1}^{c}(z_{ij} - \mu_i)^2 \\ &= \frac{1}{cg}\sum_{i=1}^{g}\sum_{j=1}^{c}(z_{ij}^2 - \mu^2) - \frac{1}{cg}\sum_{i=1}^{g}\sum_{j=1}^{c}(z_{ij}^2 - \mu_i^2) \\ &= \frac{1}{g}\sum_{i=1}^{g}\mu_i^2 - \mu^2. \end{aligned} \tag{189}$$

Since $c(\mu_1 + \cdots + \mu_g) = cg\mu$, we have

$$\sigma^2 - \frac{1}{g}\sum_{i=1}^{g}\sigma_i^2 = \frac{1}{g}\sum_{i=1}^{g}\mu_i^2 - \mu^2 = \frac{1}{g}\sum_{i=1}^{g}(\mu_i - \mu)^2 \geq 0. \tag{190}$$

□

Lemma 15. *$f(x) = \frac{1}{x^2}$ is a monotonically decreasing and convex function on $x > 0$.*

Proof. For $f(x) = \frac{1}{x^2}$, we have $f'(x) = -\frac{2}{x^3} < 0$, namely, $f(x)$ is monotonically decreasing. Furthermore, we have $f''(x) = \frac{6}{x^4} > 0$, namely, $f(x)$ is a convex function. □

Lemma 16. *Given that $\sigma_1^2, \cdots, \sigma_g^2$ and σ^2 are variances in LN-G and LN respectively, we have*

$$\sum_{i=1}^{g} \frac{1}{\sigma_i^4} \geq \frac{g}{\sigma^4}. \tag{191}$$

Proof. According to Lemma 15, we have $f(x) = \frac{1}{x^2}$ is a convex function. By Jensen's inequality, we obtain

$$\sum_{i=1}^{g} \frac{1}{g} f(\sigma_k^2) \geq f\left(\frac{1}{g}\sum_{i=1}^{g} \sigma_i^2\right) \tag{192}$$

According to Lemma 14 and Lemma 15, we have

$$\sum_{i=1}^{g} \frac{1}{g} f(\sigma_i^2) \geq f(\sigma^2), \tag{193}$$

namely

$$\frac{1}{g}\sum_{i=1}^{g} \frac{1}{\sigma_i^4} \geq \frac{1}{\sigma^4}. \tag{194}$$

□

E.2. Proof of Proposition 8

Proof. To prove

$$\frac{\mathcal{H}(\psi_G(g;\cdot);\mathbf{x})}{\mathcal{H}(\psi_L(\cdot);\mathbf{x})} \geq 1, \tag{195}$$

we can prove Eqn.196 instead:

$$\mathcal{H}(\psi_G(g;\cdot);\mathbf{x}) - \mathcal{H}(\psi_L(\cdot);\mathbf{x}) \geq 0. \tag{196}$$

According to Eqn.182, Eqn.187 and Lemma 16, we obtain

$$\begin{aligned}
\mathcal{H}(\psi_G(g;\cdot);\mathbf{x}) - \mathcal{H}(\psi_L(\cdot);\mathbf{x}) &= \sum_{i=1}^{g}\left(\frac{3}{\sigma_i^4} - \frac{6}{c\sigma_i^4}\right) - \frac{3}{d\sigma^4}(d-2) \\
&= 3\left(1-\frac{2}{c}\right)\sum_{i=1}^{g}\frac{1}{\sigma_i^4} - \frac{3}{\sigma^4}\left(1-\frac{2}{d}\right) \\
&\geq \frac{3}{\sigma^4}\left(g - \frac{2g}{c} - 1 + \frac{2}{d}\right) \\
&= \frac{3}{d\sigma^4}(d-2g-2)(g-1).
\end{aligned} \tag{197}$$

When $g \geq 2$, we have $d \geq 6$. Therefore, we obtain

$$d - 2g - 2 = d - \frac{2d}{c} - 2 \geq \frac{1}{3}d - 2 \geq 0 \tag{198}$$

According to Eqn.197, we give the necessary condition for equality in Eqn.196. One of the cases is $g = 1$ obviously. The other case is $d = 2g + 2$ — but we note that $g|d$, we hence have $g|2$. Namely $g = 2, d = 6$ is the only other case for equality.

Therefore, we have proved

$$\frac{\mathcal{H}(\psi_G(g;\cdot);\mathbf{x})}{\mathcal{H}(\psi_L(\cdot);\mathbf{x})} \geq 1. \tag{199}$$

As for the case $g = d/4$, we have that

$$\begin{aligned}\mathcal{H}(\psi_G(g;\cdot);\mathbf{x}) &\geq \frac{3}{\sigma^4}\left(g - \frac{2g}{c}\right) \\ &= \frac{3}{\sigma^4}\left(g - \frac{2g^2}{d}\right) \\ &= \frac{6}{d\sigma^4}\left(-g^2 + \frac{d}{2}g\right) \\ &= \frac{6}{d\sigma^4}\left(\frac{d^2}{16} - (g - \frac{d}{4})^2\right).\end{aligned} \tag{200}$$

When $g = d/4$, the right term reaches its maximum, where we have

$$\mathcal{H}(\psi_G(g;\cdot);\mathbf{x}) \geq \frac{3d}{8\sigma^4}. \tag{201}$$

On the other hand, we have that

$$\mathcal{H}(\psi_L(\cdot);\mathbf{x}) = \frac{3}{\sigma^4} - \frac{6}{d\sigma^4} \leq \frac{3}{\sigma^4}. \tag{202}$$

As a result, we obtain

$$\frac{\mathcal{H}(\psi_G(g;\cdot);\mathbf{x})}{\mathcal{H}(\psi_L(\cdot);\mathbf{x})} \geq \frac{d}{8}. \tag{203}$$

□

E.3. $\mathcal{H}$ about ReLU

We conduct additional analyses to compare the nonlinearity of ReLU and LN during the phase of rebuttal. ReLU is defined as $\max(0, x)$, which is not differentiable strictly. To compare ReLU with LN, we consider to introduce the Dirac function $\delta(x)$ as ReLU's second-order derivative, namely $\nabla^2 ReLU(x) = \delta(x)$. We know that $\int_I \delta(x)dx = 1$ and $\int_I f(x)\delta(x)dx = f(0)$. To apply the integral, we introduce the expectation, and assume $\boldsymbol{x} \sim N(0, \boldsymbol{I})$ is d-dimensional. Since we do not know how to calculate $\int_I f(x)\delta^2(x)dx$, we remove the square sign in $\mathcal{H}$. Specifically, we define $\bar{\mathcal{H}}(f;\boldsymbol{x})$ as

$$\bar{\mathcal{H}}(f;\boldsymbol{x}) = \sum_{i=1}^{d} \mathbb{E}_{\boldsymbol{x}} \left\| \frac{\partial^2 y_i}{\partial \boldsymbol{x}^2} \right\|_F, \tag{204}$$

like Eqn.20 in our paper, and y_i is defined similarly.

Based on the assumptions above, we have that

$$\bar{\mathcal{H}}(relu(\cdot);\boldsymbol{x}) = \frac{d}{\sqrt{2\pi}} = O(d), \tag{205}$$

and

$$\bar{\mathcal{H}}(\psi_L(\cdot);\boldsymbol{x}) = \sum_{i=1}^{d} \mathbb{E}_{\boldsymbol{x}} \frac{1}{d\sigma^2} \sqrt{d(\hat{x}_i^2 + 2) - (4\hat{x}_i^2 + 2)} = O(\sqrt{d}). \tag{206}$$

Furthermore, we have

$$\bar{\mathcal{H}}(\psi_G(g;\cdot);\boldsymbol{x}) = g \cdot O(\sqrt{c}) = O(\sqrt{dg}). \tag{207}$$

Note that we removed the square sign in $\mathcal{H}$, and there is a square root sign in $\bar{\mathcal{H}}$.

We hope the analysis above can help compare ReLU with LN, to some extent.

F. Experiments

F.1. Details of Experiments on Comparison of Representation Capacity by Fitting Random Labels.

In this section, we provide the details of experimental setup in comparing the representation capacity by fitting random labels, as stated in Section 5.2.

F.1.1. Dataset with Random Labels

We conduct the random label datasets based on CIFAR-10 and MNIST, referred to as CIFAR-10-RL and MNIST-RL. In particular, for each sample of these datasets, we randomly assign a class label to this sample and save all the samples as a dataset. Even though the labels are random, the label assignment is certain once the dataset is conducted. Therefore, it is meaningful to compare the results of different methods by fitting random labels.

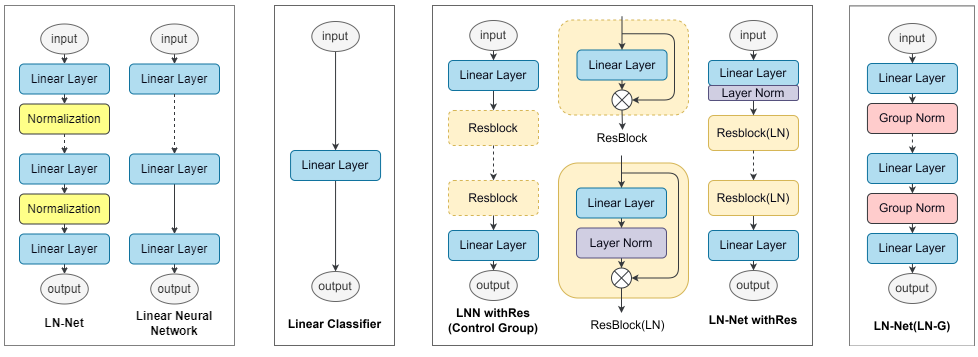
MNIST-RL is more challenging. Here, we highlight that MNIST-RL is more challenging in training a classifier for fitting the labels, compare to CIFAR-10-RL. Let X_c represents examples belong to class c. It is clear that the features in X_c are very close for the normal MNIST dataset. For example, all the digits of "0" are very similar in representation, they all have rounded curves. However, if we use the random label (the MNIST-RL dataset), the samples in X_c will have different labels. In this case, the network will need to map X_c — which is very close in representation — to different labels. As a result, we need more powerful model to fit MNIST-RL and is more difficult to train.

F.1.2. Details on Verifying Nonlinearity of LN

In this part, we use various configurations of hyper-parameters to train our models, aiming at reducing the effect from the optimization. We first sufficiently train a linear classifier (Figure A4 (b)), as the baseline, which provides the (nearly) upper bound accuracy of linear classifier. We then compare the results under linear neural network and LN-Net with residential structure for better optimization as shown in Figure A4 (c). We vary the depths ranging in $\{2, 4, 6, 8, 10, 1214\}$, and each hidden layer has a dimension of 256.

Training protocols. For the training of liner classifier, we apply both SGD optimizer with momentum (0.1) and Adam optimizer with betas $(0.9, 0.999)$. We train the model for 150 epochs and use a learning rate schedule with a decay 0.5 per 20 epochs. We search the batch sizes ranging in $\{128, 256\}$, the initial learning rates ranging in $\{0.001, 0.003, 0.005, 0.008, 0.05, 0.08, 0.1, 0.15\}$ and 5 random seeds, and report the best accuracy from these configurations of hyper-parameters. For the training of linear neural networks and LN-Nets, we follow the settings in training linear classifier, except that: 1) we use a batch size of 128 and a fixed random seed; 2) we search the initial learning rates ranging in $\{0.01, 0.03, 0.05, 0.08, 0.1\}$ for SGD and the initial learning rates ranging in $\{0.001, 0.003, 0.005, 0.008, 0.05, 0.08, 0.1, 0.15\}$ for Adam.

Results. In Figure 4 of the main paper, we show the best result of linear classifier as black dashed line, which is 18.51% on CIFAR-10-RL and 15.38% on MNIST-RL. We also provide the detailed results for linear neural network and LN-Net, shown in Table II.



(a) (b) (c) (d)

Figure A4. Schematic representation of the networks used in the experiment. (a) Original LN-Net and Linear Neural Network (LNN). (b) Linear classifier. (c) LN-Net and LNN using residual connections. (d) LN-Net using LN-G.

Table II. The result of linear neural network and LN-Net model on classification task on CIFAR-10-RL and MNIST-RL. The bold numbers refer to those outperform linear classifier. We can see layer normalization breaks the bound of linearity.

dataset	RL-CIFAR-10		RL-MNIST	
depth	Linear+Res	LN+Res	Linear+Res	LN+Res
2	17.37%	**20.45%**	14.71%	14.45%
4	17.00%	**27.97%**	14.54%	15.26%
6	16.97%	**39.24%**	14.29%	15.28%
8	17.02%	**39.39%**	14.32%	**15.76%**
10	16.98%	**31.12%**	13.89%	**18.26%**
12	16.91%	**50.48%**	13.35%	**18.47%**
14	15.19%	**55.58%**	13.98%	**19.44%**
best	17.37%	**55.58%**	14.71%	**19.44%**

F.1.3. Details on Amplifying the Nonlinearity using Group

We use the origin LN-Net and replace LN with LN-G, as shown in Figure A4 (d). For the configuration of networks, we fix the number of neurons as 256 and vary the depths ranging in $\{1, 2, 4, 6, 8, 10, 12, 14\}$. We vary the group numbers of LN-G ranging in $\{2, 4, 8, 16, 32, 64, 128\}$.

Training protocols. We use the same training protocols as the experiment above, except that we only use SGD optimizer with fixed momentum of 0.1 and search the initial learning rate ranging in $\{0.01, 0.03, 0.05, 0.1\}$.

Results. We provide the detailed results of CIFAR-10-RL in Table III and MNIST-RL in Table IV for linear neural network and LN-Net.

Table III. The performance of LN-Net with LN-G on CIFAR-10-RL. The rows of the table represent the model depth and the columns represent the group number of LN-G in the model. The percentage is the best accuracy of model under such setting. The bold number refers to the best accuracy among all group numbers under such depth.

CIFAR	1	2	4	6	8	10	12	14
2	20.51%	29.09%	52.17%	60.70%	67.21%	71.45%	74.10%	68.53%
4	26.63%	45.19%	72.41%	84.08%	91.36%	94.02%	95.76%	96.76%
8	35.02%	60.65%	91.74%	98.57%	99.72%	99.94%	99.99%	99.96%
16	46.42%	79.71%	99.58%	99.99%	**100.00%**	**100.00%**	**100.00%**	**100.00%**
32	59.89%	**93.67%**	**99.96%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
64	**69.40%**	91.62%	99.44%	99.66%	96.58%	88.20%	77.22%	44.48%
128	26.48%	14.66%	12.28%	10.38%	10.23%	10.26%	10.37%	10.22%
best	69.40%	93.67%	99.96%	100.00%	100.00%	100.00%	100.00%	100.00%

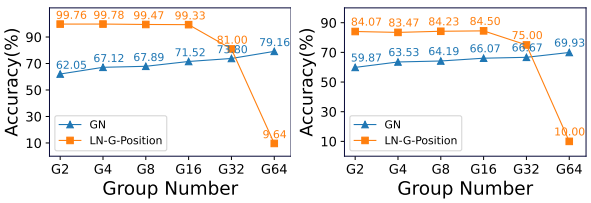
Table IV. The performance of LN-Net with LN-G on MNIST-RL. The rows of the table represent the model depth and the columns represent the group number of LN-G in the model. The percentage is the best accuracy of model under such setting. The bold number refers to the best accuracy among all group numbers under such depth.

MNIST	1	2	4	6	8	10	12	14
2	14.53%	18.25%	26.83%	27.76%	27.96%	27.56%	30.39%	30.81%
4	14.77%	20.98%	33.35%	40.67%	50.00%	53.52%	57.44%	58.78%
8	15.61%	25.38%	46.48%	64.51%	74.91%	81.34%	85.98%	89.97%
16	19.13%	32.43%	66.59%	86.20%	92.16%	94.03%	95.32%	95.25%
32	24.92%	47.08%	**82.34%**	**92.40%**	**94.47%**	**95.56%**	**95.68%**	**95.96%**
64	**33.95%**	**54.00%**	70.61%	68.63%	56.89%	42.89%	13.43%	10.21%
128	10.22%	10.17%	10.16%	10.22%	10.30%	10.25%	10.32%	10.31%
best	33.95%	54.00%	82.34%	92.40%	94.47%	95.56%	95.68%	95.96%

F.2. More Results of CNN without Activation Functions

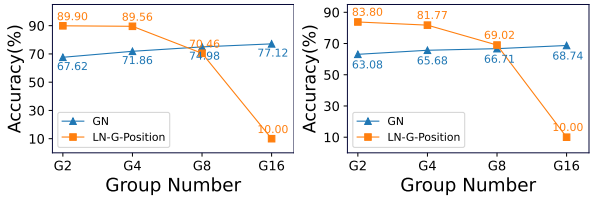
As stated in Section 5.3.1, we conduct more experiments on different networks, including the results on VGGs, and the 20-layer ResNet with the original configuration of channel number.

Results on VGGs. Following the experimental setup shown in Section 5.3.1, we also conduct experiments on CIFAR-10 classification using different normalization methods in the VGG-style networks (the network architecture used is ResNet-20, but with the residual connections removed.) with ReLU activation removed, where the group number g ranging in {2, 4, 8, 16, 32, 64}. The experimental results of different normalization methods are shown in the Table V. The results of different groups of GN and LN-G-Position are shown in the Figure A5. We have the similar observations as in the ResNet-20 shown in the main paper.



(a) Training. (b) Test.

Figure A5. Results of the variants of LN-G (GN and LN-G-Position) when using different group number. The experiments are conducted on CIFAR-10 dataset using a 20-layer VGG-style network without ReLU activation. We show (a) the training accuracy and (b) the test accuracy. In the x-axis, G2 refers to a group number of 2.



(a) Training. (b) Test.

Figure A6. Results of the variants of LN-G (GN and LN-G-Position) when using different group number. The experiments are conducted on CIFAR-10 dataset using ResNet-20-Original without ReLU activation. We show (a) the training accuracy and (b) the test accuracy.

Table V. Comparison of different normalization methods on CIFAR-10 using VGGs-NA (VGGs without ReLU activation).

Normalization methods	Train Acc(%)	Test Acc(%)
IN	9.76	10
BN	39.41	39.52
LN	51.51	51.06
GN	79.16	69.93
LN-G-Position	99.33	84.5

Results on original ResNet-20 Following the experimental setup shown in Section 5.3.1, We also conduct experiments on the original ResNet-20-NA (with ReLU removed), where the group number g ranging in {2, 4, 8, 16}. The experimental results of different normalization methods are shown in the Table VI. The results of different groups of GN and LN-G-Position are shown in the Figure A6. We have also the similar observations as in the ResNet-20 shown in the main paper.

Table VI. Comparison of different normalization methods on CIFAR-10 using ResNet-20-original-NA (the ResNet-20 using original configuration of channel numbers without ReLU activation).

Normalization methods	Train Acc(%)	Test Acc(%)
IN	10	10
BN	36.16	39.34
LN	61.12	58.69
GN	77.12	68.74
LN-G-Position	89.9	83.8