Journal of Machine Learning **ISSN**: 2790-2048(e), 2790-203X(p)

Embedding Principle: A Hierarchical Structure of Loss Landscape of Deep Neural Networks

Yaoyu Zhang * [1], Yuqing Li † [2], Zhongwang Zhang ‡ [3], Tao Luo § [4], and Zhi-Qin John Xu ¶ [5]

[1]School of Mathematical Sciences, Institute of Natural Sciences, MOE-LSC and Qing Yuan Research Institute, Shanghai Jiao Tong University, Shanghai Center for Brain Science and Brain-Inspired Technology, Shanghai, 200240, China.
[2]School of Mathematical Sciences, CMA-Shanghai, Shanghai Jiao Tong University, Shanghai, 200240, China.
[3]School of Mathematical Sciences, Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai, 200240, China.
[4]School of Mathematical Sciences, CMA-Shanghai, Institute of Natural Sciences, MOE-LSC and Qing Yuan Research Institute, Shanghai Jiao Tong University, Shanghai, 200240, China.
[5]Institute of Natural Sciences, School of Mathematical Sciences, MOE-LSC and Qing Yuan Research Institute, Shanghai Jiao Tong University, Shanghai, 200240, China.

Abstract. We prove a general Embedding Principle of loss landscape of deep neural networks (NNs) that unravels a hierarchical structure of the loss landscape of NNs, i.e., loss landscape of an NN *contains* all critical points of all the narrower NNs. This result is obtained by constructing a class of critical embeddings which map any critical point of a narrower NN to a critical point of the target NN with the same output function. By discovering a wide class of general compatible critical embeddings, we provide a gross estimate of the dimension of critical submanifolds embedded from critical points of narrower NNs. We further prove an irreversibility property of any critical embedding that the number of negative/zero/positive eigenvalues of the Hessian matrix of a critical point may increase but never decrease as an NN becomes wider through the embedding. Using a special realization of general compatible critical embedding, we prove a stringent necessary condition for being a "truly-bad" critical point that never becomes a strict-saddle point through any critical embedding. This result implies the commonplace of strict-saddle points in wide NNs, which may be an important reason underlying the easy optimization of wide NNs widely observed in practice.

Keywords:
Neural network,
Loss landscape,
Critical point,
Embedding principle.

Article Info.:
Volume: 1
Number: 1
Pages: 60- 113
Date: March/2022
doi.org/10.4208/jml.220108

Article History:
Received: 2/12/2021
Accepted: 19/3/2022

Communicated by:
Arnulf Jentzen

1 Introduction

The loss landscape of a deep neural network (NN) is important to both its optimization dynamics and generalization performance, hence is a key issue in deep learning theory. It

*Corresponding author. zhyy.sjtu@sjtu.edu.cn.
†liyuqing_551@sjtu.edu.cn.
‡0123zzw666@sjtu.edu.cn.
§luotao41@sjtu.edu.cn.
¶Corresponding author. xuzhiqin@sjtu.edu.cn.

has been realized for a long time that it is important to quantify exactly how the loss landscape looks like [1, 2]. This problem is difficult since various visualization methods show that the NN loss landscape is very complicated [3, 4]. Moreover, its non-convexity, high dimensionality and the dependence on data, model and the specific form of loss function make it very difficult to obtain a general understanding through empirical study. Therefore, though it has been extensively studied over the years, it remains an open problem to provide a clear picture about the general structure of a DNN loss landscape, e.g., critical points/submanifolds, their output functions and other properties.

Our work is inspired by the following empirical observations. From the aspect of optimization, it is often observed that wider NNs are easier for training. This phenomenon not only holds in a neural tangent kernel (NTK) regime [5], where the gradient descent training can find the global minimum with a linear convergence rate [6–9], but also happens in highly nonlinear regimes beyond NTK [10, 11]. From the aspect of generalization, the puzzle that over-parameterized NNs often generalize well seems to contradict the conventional learning theory [12, 13]. The frequency principle [14–19] shows that NNs, over-parameterized or not, tend to fit the training data by a low-frequency function, which suggests that the learned function by an NN is often of much lower complexity than the NN's capacity. Specifically, with small initialization, e.g., in a condensed regime, weights of an NN are empirically found to condense on isolated directions resulting in an output function mimicking that of a narrower NN [11, 20]. These observations raise a question that in which sense learning of a wide NN is not drastically different from a narrower NN despite potentially huge difference in their numbers of parameters. From the aspect of pruning, empirical works propose a "lottery ticket hypothesis" that a substantially smaller sub-network can achieve the same accuracy as the original large network [21]. However, it is not yet clear about the mechanism of redundancy in a learned wide NN, which makes a drastic pruning possible in practice.

All above empirical observations, though relevant to different aspects of NN, are in essence pointing towards an intrinsic similarity between narrow and wide NNs. In this work, focusing on the loss landscape, we address the following problem: What are the relations of critical points and the corresponding output functions of loss landscape among NNs with different widths. The significance of studying the critical points and their output functions of the NN loss landscape is as follows. From the optimization perspective, NN parameters trained by gradient descent provably converge to a critical point, which in general is not necessarily a global minimum or local minimum. Moreover, even for these saddle points which can be escaped, e.g., strict-saddle points [22], they may still attract the training trajectory (points nearby with a higher loss will first come close and then move away), contributing to the implicit regularization of NNs, say towards a simpler fitting (i.e., a fitting that can be realized by a narrower NN). We are specifically interested in output functions corresponding to critical points, named as *critical functions* for convenience. Studying these critical functions that potentially attract the learning of NN is clearly important for a deeper understanding of the learning process of an NN.

Our key finding in this work is the following general principle about critical points/functions of NN loss landscape intuitively stated as follows:

***Embedding principle**: the loss landscape of an NN* contains *all critical points/functions of all the narrower NNs.*

The Embedding Principle shows that any NN loss landscape contains a hierarchical structure of critical points/functions with different complexities from NNs of different widths. Specifically, it ensures existence of "simple" critical functions that can be represented by narrow NNs. Therefore, combining with the phenomenon of Frequency Principle and condensation, we conjecture that nonlinear training of NNs may be implicitly biased towards these "simple" critical functions. We will carefully look into this conjecture in our future works.

To prove the Embedding Principle, we first construct one-step critical embeddings which map any parameters of a narrow network to that of an one-neuron wider NN preserving the output function and criticality. With these embeddings, critical points of a narrow network loss landscape is mapped to 1-d critical affine subspaces of an one-neuron wider network loss landscape with the same output function. These one-step critical embeddings are constructed by adding a null neuron or splitting an existing neuron. By composition of one-step embeddings, any critical point of a narrow network loss landscape can be mapped to a critical point of any wider network loss landscape preserving the output function. Importantly, we further propose a wide class of general compatible critical embeddings, where one-step embeddings or their composition are its special cases. Note that, all critical points of a wide NN embedded from a critical point of a narrower NN by all possible general compatible critical embeddings, form high-dimensional critical submanifolds which in general are not affine subspaces for three-layer or deeper NNs.

The critical embeddings naturally link critical points of NNs of different widths, thus providing a means to track how properties of these critical points may change as the width of the NN increases. Using these critical embeddings as a tool, we obtain rich information about the general structure of an NN loss landscape.

We show that the degeneracy of a critical point substantially increases when it is embedded to a wider network, due to the fact that a critical point can be mapped to a high-dimensional critical submanifold through a class of critical embeddings. This degeneracy of critical points arises from the neuron redundancy of the wide NN in representing certain simple critical functions from narrower NNs, which is different from over-parameterization induced degeneracy studied in [23]. We also study the property of Hessian of critical points through critical embedding, e.g., the number of its negative eigenvalues, which determines whether the corresponding critical point is a strict-saddle that enables easy optimization [22]. We prove an irreversibility of critical embedding that the number of negative eigenvalues of Hessian matrix may increase but never decrease as an NN becomes wider through critical embedding. Moreover, we introduce a notion of "truly-bad" critical point which never becomes a strict-saddle point through any critical embedding. We prove a stringent necessary condition for being a "truly-bad" critical point that requires an important ingredient of its Hessian matrix being a zero matrix. This result implies the commonplace of strict-saddle points in the high-dimensional critical submanifolds of wide NNs, which may be an important reason underlying the easy optimization of wide NNs widely observed in practice.

In summary, the following general understanding of an NN loss landscape is obtained

by the embedding principle in this work:

(i) It contains a hierarchical structure of critical points/functions with different complexities from that of all narrower NNs;

(ii) Critical functions from narrower NNs in general forms a high-dimensional critical submanifold with a gross estimate of the dimension: $K + \sum_{k\in[L]} K_l K_{l-1}$, where K_l is the difference in neuron number in layer l between the target NN and the narrower NN.

(iii) If it has critical points other than the global minima that are not strict-saddle points, they mostly can become strict-saddle points in wider NNs through critical embedding, which means the embedded critical points become more optimization-friendly in a wider NN. Remark that, other than the embedded critical points, further study is needed to better understand the optimization property of these new critical points arising in wider NNs.

2 Related works

In our previous conference paper [24], we prove the Embedding Principle inspired by experimental observations and study one-step embeddings and their multi-step composition for general deep NNs. Note that, similar results on composition embedding are studied in other works, e.g., for shallow NNs [25] and deep NNs [26, 27]. As a comprehensive extension of [24], this work mathematically formalizes the notion and the study of critical embedding, proposing the general compatible embedding for the first time, and further analyzing the transition of hessian of critical points through the critical embedding. Other than above works focusing on universal properties of the NN loss landscape, many researches study the loss landscape in detail for specific settings, e.g., shallow NNs with specific activations [28–30].

Simple gradient-descent-based optimization on the complex loss landscape of NN [1,4] often finds solutions that generalize well. Many works study the geometry of the NN loss landscape at critical points in relation to its generalization ability. For example, empirical works show that SGD [31] and dropout [32] training can find a flat minimizer, which may explain why such stochastic training can find solution that generalize better. [33] further suggest that the volume of basin of attraction of good (flat) minima may dominate over that of poor (sharp) minima in practical problems. [34] show that at a local minimum there exist many asymmetric directions such that the loss increases abruptly along one side, and slowly along the opposite side. [35] prove that for any multi-layer network with generic input data and non-linear activation functions, sub-optimal local minima can exist, no matter how wide the network is. When the network width increases towards infinity, the loss landscape may become simpler and the training can avoid spurious valleys with high probability in an over-parameterized regime [36]. In an extremely over-parameterized regime with a large initialization, i.e., the linear regime identified in [11] with the NTK regime as its special case [5], the gradient descent training can find the global minimum with a linear convergence rate [6–9, 11].

The starting point of this work originates from our work in [11], where we identify a highly nonlinear condensed regime far beyond the NTK regime that weights condense in isolated directions during the training. Moreover, neural networks of different width often exhibit similar condensed behavior, e.g., stagnating at similar loss with almost the same output function, which is illustrated in experiments in our conference paper [24]. The condensation is a highly nonlinear feature learning process important to implicit regularization and generalization of NNs. The condensation transforms a large network to a network of only a few effective neurons, leading to an output function with low complexity. Such learning process is consistent with another line of research, that is, the complexity of NN output gradually increases during the training [14–18, 37–42]. For example, the Frequency Principle [14, 15] states that NNs often fit target functions from low to high frequencies during the training. A series of works study the mechanism of condensation at an initial training stage, such as for ReLU network [20, 43] and network with continuously differentiable activation functions [44].

This work in some sense serves as our attempt to uncover the theoretical structure underlying the condensation phenomenon from the perspective of loss function by proving a general Embedding Principle. In another aspect, the condensation phenomenon also confirms the value of Embedding Principle in understanding the highly nonlinear training behavior in practice.

The Embedding Principle provides a structural mechanism underlying the degeneracy as a very common property for critical points [45, 46]. Thus it complements the understanding that global minima of NNs typically form a high dimensional manifold due to over-parameterization [23].

3 Preliminary

3.1 Deep neural networks

Consider L-layer ($L \geq 2$) fully-connected NNs with a general differentiable activation function. We regard the input as the 0-th layer and the output as the L-th layer. Let m_l be the number of neurons in the l-th layer. In particular, we also set $m_0 = d$ and $m_L = d'$. For any $i, k \in \mathbb{N}$ and $i < k$, we denote $[i : k] = \{i, i+1, \cdots, k\}$. In particular, we denote $[k] := \{1, 2, \cdots, k\}$. For a matrix $\boldsymbol{A}$, we use $(\boldsymbol{A})_{i,j}$ to denote its (i,j)-th entry. We will also define $(\boldsymbol{A})_{i,[j:k]} := \big((\boldsymbol{A})_{i,j}, (\boldsymbol{A})_{i,j+1}, \cdots, (\boldsymbol{A})_{i,k}\big)$ as part of the i-th row vector. Similarly, $(\boldsymbol{A})_{[j:k],i}$ is a part of the i-th column vector. For a vector $\boldsymbol{a}$, we use $(\boldsymbol{a})_i$ to denote its i-th entry, we also define $(\boldsymbol{a})_{[j:k]} := \big((\boldsymbol{a})_j, (\boldsymbol{a})_{j+1}, \cdots, (\boldsymbol{a})_k\big)$ as part of the vector. Given weights $\boldsymbol{W}^{[l]} \in \mathbb{R}^{m_l \times m_{l-1}}$ and bias $\boldsymbol{b}^{[l]} \in \mathbb{R}^{m_l}$ for $l \in [L]$, we define the collection of parameters $\boldsymbol{\theta}$ as a $2L$-tuple (an ordered list of $2L$ elements) whose elements are matrices or vectors

$$\boldsymbol{\theta} := \Big(\boldsymbol{\theta}|_1, \cdots, \boldsymbol{\theta}|_L\Big) := \Big(\boldsymbol{W}^{[1]}, \boldsymbol{b}^{[1]}, \cdots, \boldsymbol{W}^{[L]}, \boldsymbol{b}^{[L]}\Big), \tag{3.1}$$

where the l-th layer parameters of $\boldsymbol{\theta}$ is the ordered pair $\boldsymbol{\theta}|_l := \big(\boldsymbol{W}^{[l]}, \boldsymbol{b}^{[l]}\big)$, $l \in [L]$. We may misuse our notations and do not distinguish $\boldsymbol{\theta}$ from its vectorization $\mathrm{vec}(\boldsymbol{\theta}) \in \mathbb{R}^M$ with

$M := \sum_{l=0}^{L-1}(m_l+1)m_{l+1}$. Moreover, we call the collection of tuples of length $2L$ **the tuple class**, whose elements are matrices $\{\boldsymbol{W}^{[l]}\}_{l=1}^{L}$ with $\boldsymbol{W}^{[l]} \in \mathbb{R}^{m_l \times m_{l-1}}$, or vectors $\{\boldsymbol{b}^{[l]}\}_{l=1}^{L}$ with $\boldsymbol{b}^{[l]} \in \mathbb{R}^{m_l}$, and denoted by $\text{Tuple}_{\{m_0,\cdots,m_L\}}$, i.e.,

$$\begin{aligned}&\text{Tuple}_{\{m_0,\cdots,m_L\}}\\ :=&\left\{\boldsymbol{\theta}|\boldsymbol{\theta}=\left(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\cdots,\boldsymbol{W}^{[L]},\boldsymbol{b}^{[L]}\right),\ \boldsymbol{W}^{[l]}\in\mathbb{R}^{m_l\times m_{l-1}},\ \boldsymbol{b}^{[l]}\in\mathbb{R}^{m_l},\ l\in[L]\right\}.\end{aligned}$$

Since the tuple class inherits the structure of Euclidean spaces, obviously it is a linear space. We set $\mathbf{0}$ as the zero element in the tuple class, i.e.

$$\mathbf{0}=\left(\mathbf{0}_{m_1\times m_0},\mathbf{0}_{m_1\times 1},\cdots,\mathbf{0}_{m_L\times m_{L-1}},\mathbf{0}_{m_L\times 1}\right)\in\text{Tuple}_{\{m_0,\cdots,m_L\}},$$

and we abuse the notation $\mathbf{0}$ from time to time to denote zero elements belonging to different tuple classes.

We further define the upper bracket $[L-1]$ by limiting ourselves to the first $2L-2$ element of the tuple, i.e.,

$$\boldsymbol{\theta}^{[L-1]} := (\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\cdots,\boldsymbol{W}^{[L-2]},\boldsymbol{b}^{[L-2]},\boldsymbol{W}^{[L-1]},\boldsymbol{b}^{[L-1]})\in\text{Tuple}_{\{m_0,\cdots,m_{L-2},m_{L-1}\}}, \tag{3.2}$$

and given $M^{[L-1]} := \sum_{l=0}^{L-2}(m_l+1)m_{l+1}$, $\boldsymbol{\theta}^{[L-1]}\in\mathbb{R}^{M^{[L-1]}}$.

Given parameters $\boldsymbol{\theta}$, the neural network function $f_{\boldsymbol{\theta}}(\cdot)$ can be defined in a recursive way. First, we write $f_{\boldsymbol{\theta}}^{[0]}(\boldsymbol{x}) := \boldsymbol{x}$ for the input $\boldsymbol{x}\in\mathbb{R}^d$, then for $l\in[L-1]$, $f_{\boldsymbol{\theta}}^{[l]}$ is defined recursively as

$$f_{\boldsymbol{\theta}}^{[l]}(\boldsymbol{x}) := \sigma(\boldsymbol{W}^{[l]}f_{\boldsymbol{\theta}}^{[l-1]}(\boldsymbol{x})+\boldsymbol{b}^{[l]})$$

with $\boldsymbol{f}^{[l]}\in\mathbb{R}^{m_l}$, where $\sigma(\cdot):\ \mathbb{R}\to\mathbb{R}$ is the activation function applied coordinate-wisely, with slight abuse of notation. Finally, we denote

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) := f(\boldsymbol{x},\boldsymbol{\theta}) := f_{\boldsymbol{\theta}}^{[L]}(\boldsymbol{x}) := \boldsymbol{W}^{[L]}f_{\boldsymbol{\theta}}^{[L-1]}(\boldsymbol{x})+\boldsymbol{b}^{[L]}, \tag{3.3}$$

and for simplicity, sometimes we may drop the subscript $\boldsymbol{\theta}$ in $f_{\boldsymbol{\theta}}^{[l]}$ for $l\in[0:L]$.

3.2 Loss function

The set of training data is denoted by $S := \{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^{n}$, where $\boldsymbol{x}_i\in\mathbb{R}^d$, $\boldsymbol{y}_i\in\mathbb{R}^{d'}$. Here we assume that there exists a function $f^*(\cdot):\ \mathbb{R}^d\to\mathbb{R}^{d'}$ satisfying $f^*(\boldsymbol{x}_i)=\boldsymbol{y}_i$ for $i\in[n]$. We remark that this assumption helps simplify the notation in our work, however, it is not essential for our results. The empirical risk reads as

$$R_S(\boldsymbol{\theta}) := \frac{1}{n}\sum_{i=1}^{n}\ell(f(\boldsymbol{x}_i,\boldsymbol{\theta}),\boldsymbol{y}_i) := \mathbb{E}_S\ell(f(\boldsymbol{x},\boldsymbol{\theta}),f^*(\boldsymbol{x})), \tag{3.4}$$

where the expectation $\mathbb{E}_S$ is defined for any function $h(\cdot) : \mathbb{R}^d \to \mathbb{R}$ as

$$\mathbb{E}_S h(\boldsymbol{x}) := \frac{1}{n} \sum_{i=1}^{n} h(\boldsymbol{x}_i),$$

and the loss function $\ell(\cdot, \cdot) : \mathbb{R}^{d'} \times \mathbb{R}^{d'} \to \mathbb{R}$ in (3.4) is differentiable in both variables, and the derivative of $\ell(\cdot, \cdot)$ with respect to its first argument is denoted by $\nabla \ell(\boldsymbol{y}, \boldsymbol{y}^*)$. In this paper, we always take derivatives/gradients of $\ell(\cdot, \cdot)$ in its first argument with respect to any possible parameter.

For each $l \in [L]$, we define the error vectors as

$$\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} := \nabla_{\boldsymbol{f}^{[l]}} \ell$$

with $\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} \in \mathbb{R}^{m_l}$, and the feature gradients as

$$\boldsymbol{g}_{\boldsymbol{\theta}}^{[L]} := \mathbf{1}, \text{ and } \boldsymbol{g}_{\boldsymbol{\theta}}^{[l]} := \sigma^{(1)} \left(\boldsymbol{W}^{[l]} \boldsymbol{f}_{\boldsymbol{\theta}}^{[l-1]} + \boldsymbol{b}^{[l]} \right) \text{ for } l \in [L-1]$$

with $\boldsymbol{g}_{\boldsymbol{\theta}}^{[l]} \in \mathbb{R}^{m_l}$, where we use $\sigma^{(1)}(\cdot)$ for the first derivative of $\sigma(\cdot)$. Moreover, we call $\boldsymbol{f}_{\boldsymbol{\theta}}^{[l]}$ the feature vectors, and we denote the collections of feature vectors, feature gradients, and error vectors $\{\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]}\}_{l=1}^{L}$ respectively by

$$\boldsymbol{F}_{\boldsymbol{\theta}} := \{\boldsymbol{f}_{\boldsymbol{\theta}}^{[l]}\}_{l=1}^{L}, \ \boldsymbol{G}_{\boldsymbol{\theta}} := \{\boldsymbol{g}_{\boldsymbol{\theta}}^{[l]}\}_{l=1}^{L}, \ \boldsymbol{Z}_{\boldsymbol{\theta}} := \{\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]}\}_{l=1}^{L}.$$

Moreover, using backpropagation, we can derive the following relations concerning the above quantities

$$\boldsymbol{z}_{\boldsymbol{\theta}}^{[L]} = \nabla \ell, \tag{3.5a}$$

$$\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} = (\boldsymbol{W}^{[l+1]})^{\intercal} \left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l+1]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l+1]} \right), \quad l \in [L-1], \tag{3.5b}$$

$$\nabla_{\boldsymbol{W}^{[l]}} \ell = \left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l]} \right) (\boldsymbol{f}_{\boldsymbol{\theta}}^{[l-1]})^{\intercal}, \quad l \in [L], \tag{3.5c}$$

$$\nabla_{\boldsymbol{b}^{[l]}} \ell = \boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l]}, \quad l \in [L], \tag{3.5d}$$

where we use $\circ$ for the Hadamard product [47] of two matrices of the same dimension.

Specifically, for simplicity of gradient computation, we define another group of error vectors for $l \in [L]$

$$\boldsymbol{e}_{\boldsymbol{\theta}}^{[l]} := \boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l]}$$

with $\boldsymbol{e}_{\boldsymbol{\theta}}^{[l]} \in \mathbb{R}^{m_l}$, and we denote $\{\boldsymbol{e}_{\boldsymbol{\theta}}^{[l]}\}_{l=1}^{L}$ by $\boldsymbol{E}_{\boldsymbol{\theta}} := \{\boldsymbol{e}_{\boldsymbol{\theta}}^{[l]}\}_{l=1}^{L}$.

Directly from relation (3.5), we obtain that

$$\boldsymbol{e}_{\boldsymbol{\theta}}^{[l]} = \boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l]} = \left((\boldsymbol{W}^{[l+1]})^{\intercal} \left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l+1]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l+1]} \right) \right) \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l]} = \left((\boldsymbol{W}^{[l+1]})^{\intercal} \boldsymbol{e}_{\boldsymbol{\theta}}^{[l+1]} \right) \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l]}. \tag{3.6}$$

3.3 Hessian

Given a scalar loss function $\ell(\cdot,\cdot):\ \mathbb{R}^{d'}\times\mathbb{R}^{d'}\to\mathbb{R}$, twice differentiable in both variables, and an activation function $\sigma(\cdot):\mathbb{R}\to\mathbb{R}$, also twice differentiable, we denote that

$$R_S(\boldsymbol{\theta})=\mathbb{E}_S\ell(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}),\boldsymbol{f}^*(\boldsymbol{x})), \tag{3.7}$$

$$\boldsymbol{v}_S(\boldsymbol{\theta}):=\nabla_{\boldsymbol{\theta}}R_S(\boldsymbol{\theta})=\mathbb{E}_S\nabla\ell(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}),\boldsymbol{f}^*(\boldsymbol{x}))^{\mathsf{T}}\nabla_{\boldsymbol{\theta}}\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})=\sum_{i=1}^{m_L}\mathbb{E}_S\partial_i\ell(\boldsymbol{f}_{\boldsymbol{\theta}},\boldsymbol{f}^*)\nabla_{\boldsymbol{\theta}}(\boldsymbol{f}_{\boldsymbol{\theta}})_i, \tag{3.8}$$

where $\partial_i\ell(\boldsymbol{f}_{\boldsymbol{\theta}},\boldsymbol{f}^*)$ is the i-th element of $\nabla\ell(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}),\boldsymbol{f}^*(\boldsymbol{x}))$, and $(\boldsymbol{f}_{\boldsymbol{\theta}})_i$ is the i-th element of vector $\boldsymbol{f}_{\boldsymbol{\theta}}$. Then for the Hessian matrix $\boldsymbol{H}_S(\boldsymbol{\theta})$, we have

$$\begin{aligned}\boldsymbol{H}_S(\boldsymbol{\theta}):=&\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}}R_S(\boldsymbol{\theta})\\&=\sum_{i=1}^{m_L}\mathbb{E}_S\nabla_{\boldsymbol{\theta}}\left(\partial_i\ell(\boldsymbol{f}_{\boldsymbol{\theta}},\boldsymbol{f}^*)\right)\nabla_{\boldsymbol{\theta}}(\boldsymbol{f}_{\boldsymbol{\theta}})_i+\sum_{i=1}^{m_L}\mathbb{E}_S\partial_i\ell(\boldsymbol{f}_{\boldsymbol{\theta}},\boldsymbol{f}^*)\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}}\left((\boldsymbol{f}_{\boldsymbol{\theta}})_i\right)\\&=\sum_{i,j=1}^{m_L}\mathbb{E}_S\partial_{ij}\ell(\boldsymbol{f}_{\boldsymbol{\theta}},\boldsymbol{f}^*)\nabla_{\boldsymbol{\theta}}(\boldsymbol{f}_{\boldsymbol{\theta}})_i\left(\nabla_{\boldsymbol{\theta}}(\boldsymbol{f}_{\boldsymbol{\theta}})_j\right)^{\mathsf{T}}+\sum_{i=1}^{m_L}\mathbb{E}_S\partial_i\ell(\boldsymbol{f}_{\boldsymbol{\theta}},\boldsymbol{f}^*)\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}}\left((\boldsymbol{f}_{\boldsymbol{\theta}})_i\right),\end{aligned}$$

where $\partial_{ij}\ell(\boldsymbol{f}_{\boldsymbol{\theta}},\boldsymbol{f}^*)$ is the (i,j)-th element of $\nabla\nabla\ell(\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}),\boldsymbol{f}^*(\boldsymbol{x}))$, with $\boldsymbol{H}_S(\boldsymbol{\theta})\in\mathbb{R}^M\times\mathbb{R}^M$.

We define matrices $\boldsymbol{H}_S^{(1)}(\boldsymbol{\theta})$ and $\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta})$ as follows:

$$\boldsymbol{H}_S^{(1)}(\boldsymbol{\theta}):=\sum_{i,j=1}^{m_L}\mathbb{E}_S\partial_{ij}\ell(\boldsymbol{f}_{\boldsymbol{\theta}},\boldsymbol{f}^*)\nabla_{\boldsymbol{\theta}}(\boldsymbol{f}_{\boldsymbol{\theta}})_i\left(\nabla_{\boldsymbol{\theta}}(\boldsymbol{f}_{\boldsymbol{\theta}})_j\right)^{\mathsf{T}}, \tag{3.9}$$

$$\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}):=\sum_{i=1}^{m_L}\mathbb{E}_S\partial_i\ell(\boldsymbol{f}_{\boldsymbol{\theta}},\boldsymbol{f}^*)\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}}\left((\boldsymbol{f}_{\boldsymbol{\theta}})_i\right), \tag{3.10}$$

then obviously $\boldsymbol{H}_S^{(1)}(\boldsymbol{\theta}),\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta})\in\mathbb{R}^{M\times M}$, and

$$\boldsymbol{H}_S(\boldsymbol{\theta})=\boldsymbol{H}_S^{(1)}(\boldsymbol{\theta})+\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}).$$

3.4 Assumptions and conventions of notations

We begin this part by introducing several assumptions that will be used throughout this paper:

Assumptions.
(i) We choose the L-layer ($L\geq 2$) fully-connected deep neural networks (NNs) as our model.

(ii) Our training data is $S=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^n$, $n\in\mathbb{Z}^+$.

(iii) We use the empirical loss $R_S(\boldsymbol{\theta})=\mathbb{E}_S\ell(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}),\boldsymbol{y})$.

(iv) Loss function $\ell(\cdot,\cdot)$ and activation function $\sigma(\cdot)$ are (weakly) differentiable. (Remark: twice differentiable is required for the computation of Hessian)

After stating out the assumptions, we would also like to introduce some conventions of notations that are frequently used in this paper in the following.

We write $\mathrm{NN}(\{m_l\}_{l=0}^L)$ for a fully-connected L-layer network with width $(m_0, \cdots, m_L)$, by which the tuple class of its parameters $\boldsymbol{\theta} \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ is determined whereas its activation $\sigma(\cdot)$ is not provided. When $\sigma(\cdot)$ is given, the output of $\mathrm{NN}(\{m_l\}_{l=0}^L)$ is denoted by $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ with $\boldsymbol{\theta} \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$.

Note that, if given two NNs, $\mathrm{NN}(\{m_l\}_{l=0}^L)$ and $\mathrm{NN}(\{m_l'\}_{l=0}^L)$, their corresponding parameters belong to different tuple classes except when $m_l' = m_l$ for all $l \in [0:L]$. Therefore, in this work, $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $R_S(\boldsymbol{\theta})$ may correspond to output and loss landscape of different NNs distinguished by $\boldsymbol{\theta}$ of different tuple classes.

Given two NNs, $\mathrm{NN}(\{m_l\}_{l=0}^L)$ and $\mathrm{NN}(\{m_l'\}_{l=0}^L)$ with $m_0' = m_0$, $m_L' = m_L$, and $m_l' \geq m_l$ for any $l \in [L-1]$, then for $K = \sum_{l=1}^{L-1}(m_l' - m_l) \in \mathbb{Z}^+$, we say that $\mathrm{NN}(\{m_l'\}_{l=0}^L)$ is K-neuron **wider** than $\mathrm{NN}(\{m_l\}_{l=0}^L)$, and conversely, $\mathrm{NN}(\{m_l\}_{l=0}^L)$ is K-neuron **narrower** than $\mathrm{NN}(\{m_l'\}_{l=0}^L)$.

As long as we have two NNs, $\mathrm{NN}(\{m_l\}_{l=0}^L)$ and $\mathrm{NN}(\{m_l'\}_{l=0}^L)$, given in the context of Definitions, Theorems, Propositions, Lemmas etc., we always assume that $\mathrm{NN}(\{m_l'\}_{l=0}^L)$ is wider than $\mathrm{NN}(\{m_l\}_{l=0}^L)$, i.e., $m_0' = m_0$, $m_L' = m_L$, and $m_l' \geq m_l$ for any $l \in [L-1]$. We also denote $M = \sum_{l=0}^{L-1}(m_l+1)m_{l+1}$ and $M' = \sum_{l=0}^{L-1}(m_l'+1)m_{l+1}'$, and consequently, $M' \geq M$. We denote the parameters of a narrower network by $\boldsymbol{\theta}_{\mathrm{narr}}$, and the counterpart of a wider network by $\boldsymbol{\theta}_{\mathrm{wide}}$. Then, given the data S, loss $\ell(\cdot,\cdot)$ and activation $\sigma(\cdot)$, the collection of critical points of narrower NN and wider NN can be found respectively and denoted by $\boldsymbol{\Theta}_{\mathrm{narr}}^{\mathrm{c}} := \{\boldsymbol{\theta} | \nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = \boldsymbol{0}\}$ and $\boldsymbol{\Theta}_{\mathrm{wide}}^{\mathrm{c}} := \{\boldsymbol{\theta} | \nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \boldsymbol{0}\}$. Furthermore, $\mathcal{F}_{\mathrm{narr}}^{\mathrm{c}} := \{\boldsymbol{f}_{\boldsymbol{\theta}} | \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\mathrm{narr}}^{\mathrm{c}}\}$ and $\mathcal{F}_{\mathrm{wide}}^{\mathrm{c}} := \{\boldsymbol{f}_{\boldsymbol{\theta}} | \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\mathrm{wide}}^{\mathrm{c}}\}$ are denoted for the function spaces induced by critical points accordingly.

Finally, a neuron, say the i-th neuron in layer l, is termed a **null neuron** if its output is a constant independent of input $\boldsymbol{x}$ for any activation, i.e., $(\boldsymbol{f}^{[l]})_i(\cdot) \equiv \mathrm{Const}$ for any $\sigma(\cdot)$. Otherwise, we call this neuron an **effective neuron**.

4 Embedding Principle

In this section, we prove the Embedding Principle by constructing critical embeddings. First, we define a critical embedding operation. Then, by constructing one-step critical embeddings and their composition, we prove the Embedding Principle that critical points of the loss landscape of a narrow network can be embedded to critical affine subspaces of the loss landscape of any wider network while preserving the output function. Finally, we emphasize the importance of Embedding Principle in understanding the implicit regularization and generalization of NNs.

We begin with the concepts of embedding, affine embedding and critical embedding.

Definition 4.1 (Embedding and affine embedding). *Given an* $\mathrm{NN}(\{m_l\}_{l=0}^L)$ *and* $\mathrm{NN}(\{m_l'\}_{l=0}^L)$*, an* ***embedding*** *is an injective operator* $\mathcal{T} : \mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \to \mathrm{Tuple}_{\{m_0',\cdots,m_L'\}}$*, i.e.,* $\mathcal{T}(\boldsymbol{\theta}_1) \neq \mathcal{T}(\boldsymbol{\theta}_2)$ *for* $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ *and* $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$*. In addition,* $\mathcal{T}$ *is an* ***affine***

embedding *if* $\tilde{\mathcal{T}}(\boldsymbol{\theta}) := \mathcal{T}(\boldsymbol{\theta}) - \mathcal{T}(\mathbf{0})$ *is a linear operator, i.e.,* $\tilde{\mathcal{T}}(\boldsymbol{\theta}_1) + \tilde{\mathcal{T}}(\boldsymbol{\theta}_2) = \tilde{\mathcal{T}}(\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2)$ *and* $\tilde{\mathcal{T}}(\beta\boldsymbol{\theta}) = \beta\tilde{\mathcal{T}}(\boldsymbol{\theta})$ *for any* $\boldsymbol{\theta}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ *and* $\beta \in \mathbb{R}$.

Remark 4.1. For any given affine embedding $\mathcal{T}$, it is associated with a matrix $\boldsymbol{A} \in \mathbb{R}^{M' \times M}$ and a vector $\boldsymbol{c} \in \mathbb{R}^{M'}$ such that $\mathrm{vec}(\mathcal{T}(\boldsymbol{\theta})) = \boldsymbol{A}\mathrm{vec}(\boldsymbol{\theta}) + \boldsymbol{c}$, where $M = \sum_{l=0}^{L-1}(m_l+1)m_{l+1}$ and $M' = \sum_{l=0}^{L-1}(m'_l+1)m'_{l+1}$. As noted before, we do not distinguish tuple $\boldsymbol{\theta}$ from its vectorization $\mathrm{vec}(\boldsymbol{\theta})$ in the following. Hence, $\mathcal{T}(\boldsymbol{\theta}) = \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{c}$.

Definition 4.2 (Critical embedding). *Given an* $\mathrm{NN}(\{m_l\}_{l=0}^L)$ *and* $\mathrm{NN}(\{m'_l\}_{l=0}^L)$, *a* ***critical embedding*** *is an affine embedding* $\mathcal{T} : \mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \to \mathrm{Tuple}_{\{m'_0,\cdots,m'_L\}}$, *which maps any set of its network parameters* $\boldsymbol{\theta}_{\mathrm{narr}} \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ *to that of a wider NN* $\boldsymbol{\theta}_{\mathrm{wide}} = \mathcal{T}(\boldsymbol{\theta}_{\mathrm{narr}}) \in \mathrm{Tuple}_{\{m'_0,\cdots,m'_L\}}$ *satisfying that: For any given data* S, *loss function* $\ell(\cdot,\cdot)$, *activation function* $\sigma(\cdot)$,

(i) **output preserving:** $f_{\boldsymbol{\theta}_{\mathrm{narr}}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}_{\mathrm{wide}}}(\boldsymbol{x})$ *for any* $\boldsymbol{x} \in \mathbb{R}^d$;

(ii) **representation preserving:**

$$\begin{aligned}&\mathrm{span}\left\{\left\{\left(f^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}}(\cdot)\right)_j\right\}_{j\in[m_l]} \cup \{1\}\right\}\\ =&\mathrm{span}\left\{\left\{\left(f^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}}(\cdot)\right)_{j'}\right\}_{j'\in[m'_l]} \cup \{1\}\right\}, \quad \text{for any } l \in [L],\end{aligned}$$

where $\{f^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}}\}_{l=1}^L$ *and* $\{f^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}}\}_{l=1}^L$ *are feature vectors of* $\mathrm{NN}(\{m_l\}_{l=0}^L)$ *and* $\mathrm{NN}(\{m'_l\}_{l=0}^L)$, *and* $1 : \mathbb{R}^d \to \mathbb{R}$ *is the constant function, i.e.,* $1(\cdot) \equiv 1$;

(iii) **criticality preserving:** *If* $\boldsymbol{\theta}_{\mathrm{narr}}$ *is a critical point of* $R_S(\boldsymbol{\theta})$, *i.e.,* $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = \mathbf{0}$, *then* $\boldsymbol{\theta}_{\mathrm{wide}}$ *is also a critical point of* $R_S(\boldsymbol{\theta})$, *i.e.,* $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \mathbf{0}$.

Specifically, if an embedding is a critical embedding with $\mathrm{NN}(\{m'_l\}_{l=0}^L)$ *one-neuron wider than* $\mathrm{NN}(\{m_l\}_{l=0}^L)$, *we call it* ***one-step critical embedding***.

4.1 One-step critical embedding

In this subsection, we introduce two types of one-step critical embeddings. we start with the definition of one-step null embedding. Intuitively, a one-step null embedding adds a null neuron to the NN, whose output is a constant independent of input $\boldsymbol{x}$ for any activation.

Definition 4.3 (One-step null embedding). *Given an* $\mathrm{NN}(\{m_l\}_{l=0}^L)$ *and its parameter* $\boldsymbol{\theta} = (\boldsymbol{W}^{[1]}, \boldsymbol{b}^{[1]}, \cdots, \boldsymbol{W}^{[L]}, \boldsymbol{b}^{[L]}) \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$, *then for any* $l \in [L-1]$, *we define the operators* $\mathcal{T}_{l,0}$ *and* $\mathcal{V}_{l,0}$ *applying on* $\boldsymbol{\theta}$ *as follows*

$$\begin{aligned}&\mathcal{T}_{l,0}(\boldsymbol{\theta})|_k = \boldsymbol{\theta}|_k, \quad k \neq l,\ l+1,\\ &\mathcal{T}_{l,0}(\boldsymbol{\theta})|_l = \left(\begin{bmatrix}\boldsymbol{W}^{[l]}\\ \mathbf{0}_{1\times m_{l-1}}\end{bmatrix}, \begin{bmatrix}\boldsymbol{b}^{[l]}\\ 0\end{bmatrix}\right),\end{aligned}$$

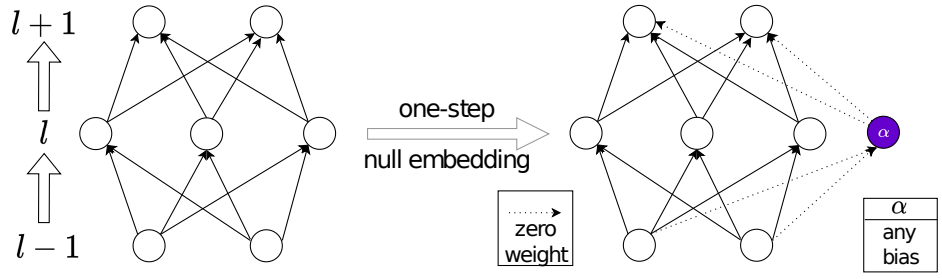

Figure 4.1: Illustration of one-step null embedding. The purple neuron is added with both input and output weights as zero and an arbitrary bias α.

$$\begin{aligned}
&\mathcal{T}_{l,0}(\boldsymbol{\theta})|_{l+1} = \left(\left[\boldsymbol{W}^{[l+1]}, \boldsymbol{0}_{m_{l+1}\times 1}\right], \boldsymbol{b}^{[l+1]}\right),\\
&\mathcal{V}_{l,0}(\boldsymbol{\theta})|_k = \left(\boldsymbol{0}_{m_k\times m_{k-1}}, \boldsymbol{0}_{m_k\times 1}\right), \quad k\neq l,\, l+1,\\
&\mathcal{V}_{l,0}(\boldsymbol{\theta})|_l = \left(\boldsymbol{0}_{(m_l+1)\times m_{l-1}}, \begin{bmatrix} \boldsymbol{0}_{m_l\times 1} \\ 1 \end{bmatrix}\right),\\
&\mathcal{V}_{l,0}(\boldsymbol{\theta})|_{l+1} = \left(\boldsymbol{0}_{m_{l+1}\times(m_l+1)}, \boldsymbol{0}_{m_{l+1}\times 1}\right).
\end{aligned}$$

We define ***one-step null embedding*** $\mathcal{T}^{\alpha}_{l,0}$ *as: For any* $\boldsymbol{\theta}\in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$,

$$\mathcal{T}^{\alpha}_{l,0}(\boldsymbol{\theta}) = (\mathcal{T}_{l,0} + \alpha\mathcal{V}_{l,0})(\boldsymbol{\theta}).$$

Note that the neuron added by the above one-step null embedding has zero output weights, zero input weights and an arbitrary bias α. An illustration is shown in Fig. 4.1.

We proceed to the definition of one-step splitting embedding.

Definition 4.4 (One-step splitting embedding). *Given an* $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ *and its parameter* $\boldsymbol{\theta} = (\boldsymbol{W}^{[1]}, \boldsymbol{b}^{[1]}, \cdots, \boldsymbol{W}^{[L]}, \boldsymbol{b}^{[L]}) \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$, *then for any* $l\in[L-1]$ *and* $s\in[m_l]$, *we define the operators* $\mathcal{T}_{l,s}$ *and* $\mathcal{V}_{l,s}$ *applying on* $\boldsymbol{\theta}$ *as follows*

$$\begin{aligned}
&\mathcal{T}_{l,s}(\boldsymbol{\theta})|_k = \boldsymbol{\theta}|_k, \quad k\neq l, l+1,\\
&\mathcal{T}_{l,s}(\boldsymbol{\theta})|_l = \left(\begin{bmatrix} \boldsymbol{W}^{[l]} \\ \boldsymbol{W}^{[l]}_{s,[1:m_{l-1}]} \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}^{[l]} \\ \boldsymbol{b}^{[l]}_{s} \end{bmatrix}\right),\\
&\mathcal{T}_{l,s}(\boldsymbol{\theta})|_{l+1} = \left(\left[\boldsymbol{W}^{[l+1]}, \boldsymbol{0}_{m_{l+1}\times 1}\right], \boldsymbol{b}^{[l+1]}\right),\\
&\mathcal{V}_{l,s}(\boldsymbol{\theta})|_k = \left(\boldsymbol{0}_{m_k\times m_{k-1}}, \boldsymbol{0}_{m_k\times 1}\right), \quad k\neq l, l+1,\\
&\mathcal{V}_{l,s}(\boldsymbol{\theta})|_l = \left(\boldsymbol{0}_{(m_l+1)\times m_{l-1}}, \boldsymbol{0}_{(m_l+1)\times 1}\right),\\
&\mathcal{V}_{l,s}(\boldsymbol{\theta})|_{l+1} = \left(\left[\boldsymbol{0}_{m_{l+1}\times(s-1)}, -\boldsymbol{W}^{[l+1]}_{[1:m_{l+1}],s}, \boldsymbol{0}_{m_{l+1}\times(m_l-s)}, \boldsymbol{W}^{[l+1]}_{[1:m_{l+1}],s}\right], \boldsymbol{0}_{m_{l+1}\times 1}\right).
\end{aligned}$$

We define ***one-step splitting embedding*** $\mathcal{T}^{\alpha}_{l,s}$ *as: For any* $\boldsymbol{\theta}\in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$,

$$\mathcal{T}^{\alpha}_{l,s}(\boldsymbol{\theta}) = (\mathcal{T}_{l,s} + \alpha\mathcal{V}_{l,s})(\boldsymbol{\theta}).$$

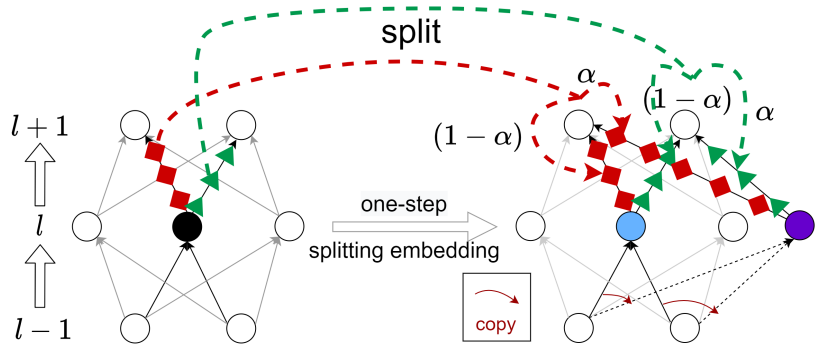

Figure 4.2: Illustration of one-step splitting embedding. The black neuron in the left network is split into the blue and purple neurons in the right network. The red (green) output weight of the black neuron in the left net is split into two red (green) weights in the right net with ratio $(1-\alpha)$ and α, respectively. This is also illustrated in [24].

An illustration of one-step splitting method is shown in Fig. 4.2.

Remark 4.2. The parameters $\mathcal{T}^{\alpha}_{l,0}(\boldsymbol{\theta})$ and $\mathcal{T}^{\alpha}_{l,s}(\boldsymbol{\theta})$ correspond to a L-layer NN with width $(m_0, \cdots, m_{l-1}, m_l+1, m_{l+1}, \cdots, m_L)$ since $\mathcal{T}^{\alpha}_{l,0}$ and $\mathcal{T}^{\alpha}_{l,s}$ are one-step embeddings.

Remark 4.3. We observe that $\mathcal{T}^{\alpha}_{l,0}$ and $\mathcal{T}^{\alpha}_{l,s}$ can be applied on the neural network parameter $\boldsymbol{\theta}$ of any given $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ of proper depth and width, hence the domain of $\mathcal{T}^{\alpha}_{l,0}$ and $\mathcal{T}^{\alpha}_{l,s}$ is not limited to a specific $\mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$. Instead, the extended domain of $\mathcal{T}^{\alpha}_{l,0}$ and $\mathcal{T}^{\alpha}_{l,s}$ assembles all possible tuple class, i.e., if we denote the extended domain of $\mathcal{T}^{\alpha}_{l,0}$ by $\mathcal{D}_{l,0}$, then

$$\mathcal{D}_{l,0} := \bigsqcup_{l<L} \mathrm{Tuple}_{\{n_0,\cdots,n_L\}}, \tag{4.1}$$

and if we denote the extended domain of $\mathcal{T}^{\alpha}_{l,s}$ by $\mathcal{D}_{l,s}$, then

$$\mathcal{D}_{l,s} := \bigsqcup_{l<L, s\le n_l} \mathrm{Tuple}_{\{n_0,\cdots,n_L\}}, \tag{4.2}$$

where $\bigsqcup$ refers to the disjoint union of different tuple classes.

By the above remark, we may extend $\mathcal{T}^{\alpha}_{l,0}$ and $\mathcal{T}^{\alpha}_{l,s}$ to their extended domains, and we identify

$$\mathcal{T}^{\alpha}_{l,0}: \mathcal{D}_{l,0} \to \mathcal{D}_{l,0}, \quad \mathcal{T}^{\alpha}_{l,s}: \mathcal{D}_{l,s} \to \mathcal{D}_{l,s},$$

with their restrictions on $\mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ for some given $\mathrm{NN}(\{m_l\}_{l=0}^{L})$.

Theorem 4.1. *One-step null embedding and one-step splitting embedding are critical embeddings.*

In order to prove Theorem 4.1, we need Lemma 4.1 and Lemma 4.2, where Lemma 4.2 has already been presented previously in our conference paper [24, Lemma 1].

Lemma 4.1. *For any one-step null embedding $\mathcal{T}^{\alpha}_{l,0}$, given any $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ and its parameters $\boldsymbol{\theta}_{\mathrm{narr}} \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ with $\mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \in \mathcal{D}_{l,0}$, we have $\boldsymbol{\theta}_{\mathrm{wide}} := \mathcal{T}^{\alpha}_{l,0}(\boldsymbol{\theta}_{\mathrm{narr}})$ satisfies the following conditions: given any data S, loss $\ell(\cdot,\cdot)$ and activation $\sigma(\cdot)$, for any $l \in [L-1]$,*

(i) feature vectors in

$$\boldsymbol{F}_{\boldsymbol{\theta}_{\mathrm{wide}}}\colon \boldsymbol{f}^{[l']}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \boldsymbol{f}^{[l']}_{\boldsymbol{\theta}_{\mathrm{narr}}}, \text{for } l' \in [L] \text{ and } l' \neq l,\ \boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \left[(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})^{\intercal}, \sigma(\alpha)\right]^{\intercal};$$

(ii) feature gradients in

$$\boldsymbol{G}_{\boldsymbol{\theta}_{\mathrm{wide}}}\colon \boldsymbol{g}^{[l']}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \boldsymbol{g}^{[l']}_{\boldsymbol{\theta}_{\mathrm{narr}}}, \text{for } l' \in [L] \text{ and } l' \neq l,\ \boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \left[(\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})^{\intercal}, \sigma^{(1)}(\alpha)\right]^{\intercal};$$

(iii) error vectors in

$$\boldsymbol{Z}_{\boldsymbol{\theta}_{\mathrm{wide}}}\colon \boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\mathrm{narr}}}, \text{for } l' \in [L] \text{ and } l' \neq l,\ \boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \left[\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}}, 0\right]^{\intercal};$$

(iv) $\mathcal{T}^{\alpha}_{l,0}$ is injective for all α;

(v) $\mathcal{T}^{\alpha}_{l,0}$ is an affine embedding for all α.

Lemma 4.2 (Lemma 1 in [24])**.** *For any one-step splitting embedding $\mathcal{T}^{\alpha}_{l,s}$, given any $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ and its parameters $\boldsymbol{\theta}_{\mathrm{narr}} \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ with $\mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \in \mathcal{D}_{l,s}$, we have $\boldsymbol{\theta}_{\mathrm{wide}} := \mathcal{T}^{\alpha}_{l,s}(\boldsymbol{\theta}_{\mathrm{narr}})$ satisfies the following conditions: given any data S, loss $\ell(\cdot,\cdot)$ and activation $\sigma(\cdot)$, for any $l \in [L-1]$,*

(i) feature vectors in

$$\boldsymbol{F}_{\boldsymbol{\theta}_{\mathrm{wide}}}\colon \boldsymbol{f}^{[l']}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \boldsymbol{f}^{[l']}_{\boldsymbol{\theta}_{\mathrm{narr}}}, \text{for } l' \in [L] \text{ and } l' \neq l,\ \boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \left[(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})^{\intercal}, (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_s\right]^{\intercal};$$

(ii) feature gradients in

$$\boldsymbol{G}_{\boldsymbol{\theta}_{\mathrm{wide}}}\colon \boldsymbol{g}^{[l']}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \boldsymbol{g}^{[l']}_{\boldsymbol{\theta}_{\mathrm{narr}}}, \text{for } l' \in [L] \text{ and } l' \neq l,\ \boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \left[(\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})^{\intercal}, (\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_s\right]^{\intercal};$$

(iii) error vectors in

$$\begin{aligned}&\boldsymbol{Z}_{\boldsymbol{\theta}_{\mathrm{wide}}}\colon \boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\mathrm{narr}}}, \text{for } l' \in [L] \text{ and } l' \neq l,\\ &\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}} = \left[\left(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}}\right)^{\intercal}_{[1:s-1]}, (1-\alpha)(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_s, \left(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}}\right)^{\intercal}_{[s+1:m_l]}, \alpha(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_s\right]^{\intercal};\end{aligned}$$

(iv) $\mathcal{T}^{\alpha}_{l,s}$ is injective for all α.

(v) $\mathcal{T}^{\alpha}_{l,s}$ is an affine embedding for all α.

Directly from Lemma 4.1 and Lemma 4.2, we obtain that both one-step null embedding (Proposition 4.1) and one-step splitting embedding (Proposition 4.2) satisfy the property of output preserving and representation preserving, and all we need is to check the property of criticality preserving. We remark that Proposition 4.2 has also been presented previously in our conference paper [24, Proposition 1].

Proposition 4.1. *For any one-step null embedding $\mathcal{T}^{\alpha}_{l,0}$, given any $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ and its parameters $\boldsymbol{\theta}_{\mathrm{narr}} \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ with $\mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \in \mathcal{D}_{l,0}$, we have $\boldsymbol{\theta}_{\mathrm{wide}} := \mathcal{T}^{\alpha}_{l,0}(\boldsymbol{\theta}_{\mathrm{narr}})$ satisfies the following conditions: given any data S, loss $\ell(\cdot,\cdot)$ and activation $\sigma(\cdot)$, if $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = \mathbf{0}$, then $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \mathbf{0}$.*

Proposition 4.2. *For any one-step splitting embedding $\mathcal{T}^{\alpha}_{l,s}$, given any $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ and its parameters $\boldsymbol{\theta}_{\mathrm{narr}} \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ with $\mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \in \mathcal{D}_{l,s}$, we have $\boldsymbol{\theta}_{\mathrm{wide}} := \mathcal{T}^{\alpha}_{l,s}(\boldsymbol{\theta}_{\mathrm{narr}})$ satisfies the following conditions: given any data S, loss $\ell(\cdot,\cdot)$ and activation $\sigma(\cdot)$, if $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = \mathbf{0}$, then $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \mathbf{0}$.*

Combining altogether Lemma 4.1, Lemma 4.2, Proposition 4.1 and Proposition 4.2, we finish our proof for Theorem 4.1.

4.2 Composition of one-step embeddings and the Embedding Principle

We firstly define the composition of two embeddings, which readily leads to the multi-step embedding operation. (A formal definition of the composition of two embeddings can be found in Appendix C.)

Definition 4.5 (Composition of two embeddings, Informal)**.** *For any two embeddings $\mathcal{T}$ and $\mathcal{T}'$, for any $\boldsymbol{\theta}$ in the domain of $\mathcal{T}$, the operator $\mathcal{T}'\mathcal{T}$ defined as $\mathcal{T}'\mathcal{T}(\boldsymbol{\theta}) := \mathcal{T}'(\mathcal{T}(\boldsymbol{\theta}))$ is also an embedding, and we term $\mathcal{T}'\mathcal{T}$ the composition of $\mathcal{T}'$ and $\mathcal{T}$.*

For simplicity, for any $K \in \mathbb{Z}^+, K \geq 2$, we denote hereafter $\prod_{l=1}^{K} \mathcal{T}_l := \mathcal{T}_K \cdots \mathcal{T}_1$ as the composition of K individual embeddings $\{\mathcal{T}_l\}_{l=1}^{K}$. For $K = 1$, $\prod_{l=1}^{1} \mathcal{T}_l := \mathcal{T}_1$.

Definition 4.6 (***K*-step (Multi-step) composition embedding**)**.** *Suppose we have two vectors $\boldsymbol{l} = (l_k)_{k=1}^{K}$, $l_k \in [L-1]$, $\boldsymbol{\alpha} = (\alpha_k)_{k=1}^{K} \subset \mathbb{R}^K$, and a sequence $\{m_l^{(0)}\}_{l=1}^{L}$ with $m_l^{(0)} := m_l$ for $l \in [L]$. Then given an $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ and its parameters $\boldsymbol{\theta}$, a **K-step composition embedding**, $\mathcal{T} : \mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \to \mathrm{Tuple}_{\{m'_0,\cdots,m'_L\}}$ with $K = \sum_{l=1}^{L-1} m'_j - \sum_{l=1}^{L-1} m_l$, is defined recursively by the composition of K one-step null embeddings or one-step splitting embeddings.*

Formally speaking, a K-step composition embedding $\mathcal{T}^{\boldsymbol{\alpha}}_{\boldsymbol{l},\boldsymbol{s}}$ is defined recursively as follows:

For $n = 1$, choose $s_1 \in [m_{l_1}^{(0)}] \cup \{0\}$, then

$$\mathcal{T}^{\boldsymbol{\alpha},(1)}_{\boldsymbol{l},\boldsymbol{s}} := \mathcal{T}^{\alpha_1}_{l_1,s_1} = \mathcal{T}_{l_1,s_1} + \alpha_1 \mathcal{V}_{l_1,s_1}.$$

Update the sequence from $\{m_l^{(0)}\}_{l=1}^{L}$ to $\{m_l^{(1)}\}_{l=1}^{L}$ following: $m_l^{(1)} = m_l^{(0)}$ for $l \in [L-1]\backslash\{l_1\}$, and $m_{l_1}^{(1)} = m_{l_1}^{(0)} + 1$;

Then inductively, for $n = k$, choose $s_k \in [m_{l_k}^{(k-1)}] \cup \{0\}$, then

$$\mathcal{T}_{\boldsymbol{l},\boldsymbol{s}}^{\boldsymbol{\alpha},(k)} := \mathcal{T}_{l_k,s_k}^{\alpha_k} \mathcal{T}_{\boldsymbol{l},\boldsymbol{s}}^{\boldsymbol{\alpha},(k-1)}.$$

Update the sequence from $\{m_l^{(k-1)}\}_{l=1}^L$ to $\{m_l^{(k)}\}_{l=1}^L$ following: $m_l^{(k)} = m_l^{(k-1)}$ for $l \in [L-1]\backslash\{l_k\}$, and $m_{l_k}^{(k)} = m_{l_k}^{(k-1)} + 1$.

Finally, $\mathcal{T} := \mathcal{T}_{\boldsymbol{l},\boldsymbol{s}}^{\boldsymbol{\alpha}} := \mathcal{T}_{\boldsymbol{l},\boldsymbol{s}}^{\boldsymbol{\alpha},(K)}$.

Remark 4.4. For each $i \in [K]$, $\mathcal{T}_{l_i,s_i}^{\alpha_i}$ is regarded as its restriction on the tuple class $\text{Tuple}_{\{m_0^{(i-1)},\cdots,m_L^{(i-1)}\}} \in \mathcal{D}_{l_i,s_i}$, hence

$$\mathcal{T}_{l_i,s_i}^{\alpha_i} : \text{Tuple}_{\{m_0^{(i-1)},\cdots,m_L^{(i-1)}\}} \to \text{Tuple}_{\{m_0^{(i)},\cdots,m_L^{(i)}\}}.$$

Theorem 4.2. *A K-step composition embedding is a critical embedding.*

The composition of one-step embeddings renders a feasible method to embedding any critical point of the loss landscape of a narrow NN to a critical point with the same output function of any wider NN, therefore, we have the following Embedding Principle.

Theorem 4.3 (Embedding Principle)**.** *Given any NN and any K-neuron wider NN, there exists a K-step composition embedding $\mathcal{T}$ satisfying that: For any given data S, loss function $\ell(\cdot,\cdot)$, activation function $\sigma(\cdot)$, given any critical point $\boldsymbol{\theta}_{\text{narr}}^{\text{c}}$ of the narrower NN, $\boldsymbol{\theta}_{\text{wide}}^{\text{c}} := \mathcal{T}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}})$ is still a critical point of the K-neuron wider NN with the same output function, i.e., $\boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}^{\text{c}}} = \boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}^{\text{c}}}$.*

Proof. Existence of a K-step composition embedding $\mathcal{T}$ can be seen from the K-step construction given in Definition 4.6, and we finish the proof. □

Moreover, we obtain a corollary from Theorem 4.3 stating the Embedding Principle for the critical functions.

Corollary 4.1 (Embedding Principle of critical functions)**.** *Given any NN and any wider NN, for any given data S, loss function $\ell(\cdot,\cdot)$ and activation function $\sigma(\cdot)$,*

$$\mathcal{F}_{\text{narr}}^{\text{c}} \subset \mathcal{F}_{\text{wide}}^{\text{c}}, \tag{4.3}$$

where $\mathcal{F}_{\text{narr}}^{\text{c}} = \{\boldsymbol{f}_{\boldsymbol{\theta}} | \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\text{narr}}^{\text{c}}\}$ and $\mathcal{F}_{\text{wide}}^{\text{c}} = \{\boldsymbol{f}_{\boldsymbol{\theta}} | \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\text{wide}}^{\text{c}}\}$ are the sets of critical functions.

4.3 Importance of Embedding Principle

Mathematically speaking, Embedding Principle is a natural result of an embedding operation that preserves output function and criticality. However, we must emphasize the importance of stating and proving it explicitly in this work. For a long time, researchers study the loss landscape of NN from an optimization perspective focusing specifically on its property in the parameter space. Lots of works make effort in tackling problems like whether bad local minima exist, whether local minima are also global minima and whether

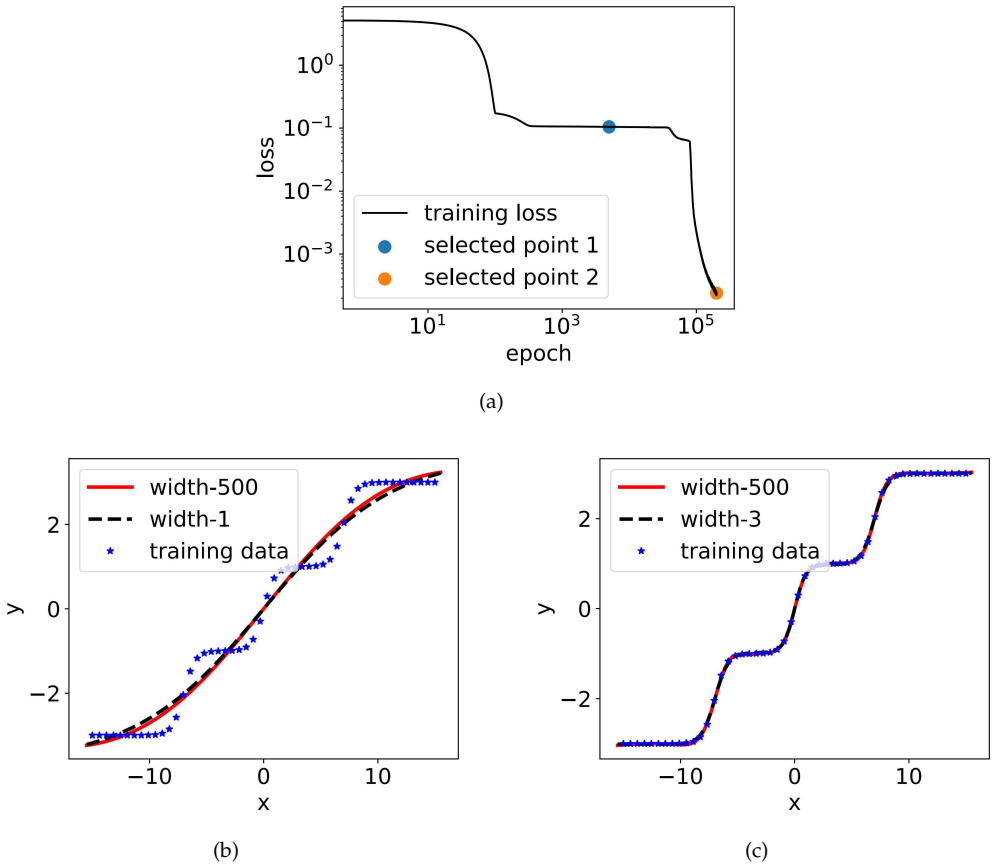

Figure 4.3: (a) The training loss of a two-layer tanh neural network with 500 hidden neurons. (b, c) Red solid: the DNN output at a training step, where the blue dot and the orange dot in (a) corresponds to (b) and (c), respectively; Black dashed: the output of the global minimum of the width-1 NN in (b) and the width-3 NN in (c), respectively; Blue dots: training data. This is also illustrated in [24].

all saddle points are strict-saddle points, etc. However, because the loss landscape of NN also has profound impact on its implicit regularization and generalization performance, it is important to look into the loss landscape from the perspective of function spaces.

Motivated by the phenomena of Frequency Principle [14–18] and condensation [11], we are very interested in the question of what are the critical functions of an NN loss landscape that may attract the training trajectory in the function space. Specifically, we care about whether there are "simple" critical functions in wide NNs that may implicitly regularize the training to help avoid overfitting. For example, in our experiments shown in Fig. 4.3, we clearly observe that the training of a width-500 two-layer tanh-NN in fitting 50 data points experiences two stages. At the first stage, it learns an output function close to the best fitting of the width-1 tanh-NN and stays for a while, seemingly that it encounters a saddle point. At the second stage, it converges to an output function close

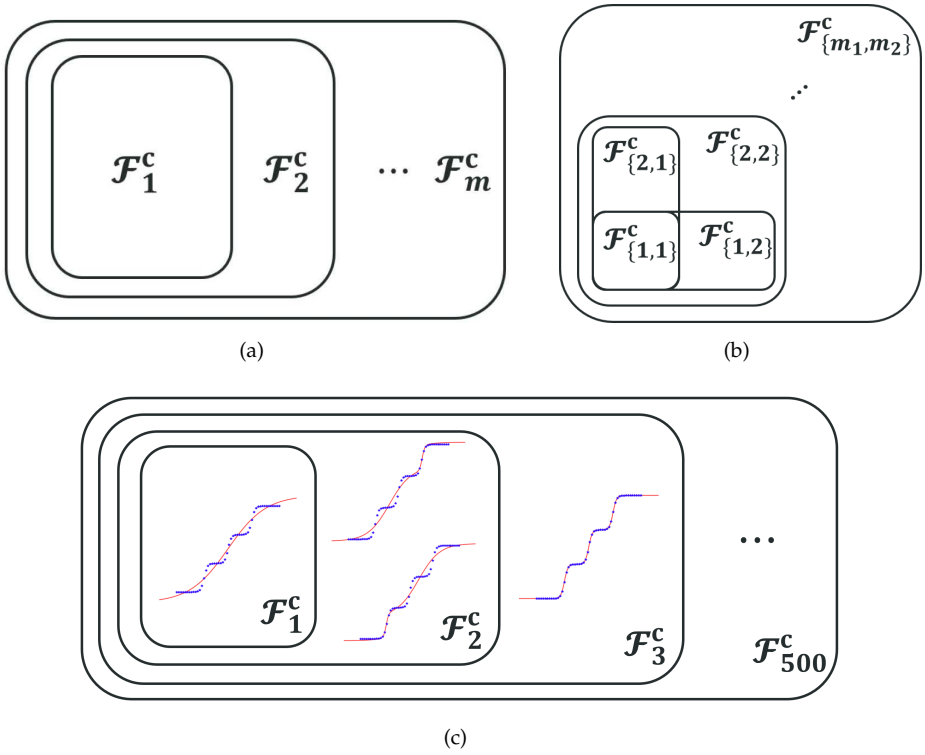

Figure 4.4: Illustration of the Embedding Principle for the critical functions of (a) two-layer width-m NN and (b) three-layer width-$\{m_1, m_2\}$ NN. (c) Some critical functions of width-500 tanh-NN predicted by the Embedding Principle for experiments in Fig. 4.3.

to the best fitting of the width-3 tanh-NN, which interpolates all the data points. Clearly, the complexity of the output function of the width-500 NN gradually increases during the training, leading to a non-overfitting interpolation of data despite of possessing overfitting capability. However, before studying the universality of such training behavior clearly relevant to generalization, it is important to have a theoretical answer to whether such "simple" critical functions always exist even in very wide NNs.

By stating and proving the Embedding Principle explicitly, we provide a clear answer of "YES" to above question. Moreover, we unravel the exact meaning of "simple" critical functions—critical functions of narrower networks. An illustration of critical functions of a two-layer width-m NN and a three-layer width-$\{m_1, m_2\}$ NN predicted by the Embedding Principle are shown in Fig. 4.4(a), (b), respectively. Critical functions of the width-500 tanh-NN for experiments in Fig. 4.3 are illustrated in Fig. 4.4(c). Note that, by Frequency Principle [14–18], we consider functions dominated by low frequency components as "simple" functions that training of an NN is implicitly biased to. Here, by Embedding Principle, "simple" functions are those critical functions of narrow NNs, which signify a hierarchical structure of fittings of training data with different complexities indicated by the narrowest NN a critical function belongs to. Combining with empirical observations of Frequency

Principle and condensation, we conjecture that nonlinear training of an NN is implicitly biased towards these "simple" critical functions. It is clearly important to look further into this conjecture in the future works.

5 General compatible critical embedding

Understanding the critical points/manifolds and their geometry of NN loss landscape is an important open problem in deep learning theory. To further advance our study along this direction, the Embedding Principle proven above inspires us to treat NN loss landscape of different widths and their critical points/manifolds as a unified object linked by the critical embeddings. These explicit critical submanifolds embedded from narrower NNs are subsets of the critical point set of a given NN. The study of them could provide lower bounds of important geometric properties of the critical point set such as dimensions of critical submanifolds (associated to the degeneracy of critical points). Clearly, the more critical embeddings we discover, the larger embedded critical submanifolds we uncover explicitly, and the tighter lower bound estimation we may obtain.

To define a general compatible embedding from a narrow NN to a wide NN, we first define mappings that establish the relation between neuron indices of the narrow NN and neuron indices of the wide NN.

Definition 5.1 (Pull-back index mapping and total index mapping). *Given an* $\mathrm{NN}(\{m_l\}_{l=0}^L)$ *and a wider* $\mathrm{NN}(\{m'_l\}_{l=0}^L)$*, a* ***pull-back index mapping*** $\mathcal{I} := \{\mathcal{I}_l\}_{l=0}^L$ *from the wider NN to the narrower NN is defined as follows: For any fixed* $l \in [L-1]$*,* $\mathcal{I}_l$ *maps a neuron index* $s' \in [m'_l]$ *of the wider* $\mathrm{NN}(\{m'_l\}_{l=0}^L)$ *in layer* l *to a neuron index* $s \in [m_l] \sqcup \{0\}$ *in the same layer of the narrower* $\mathrm{NN}(\{m_l\}_{l=0}^L)$*. As for the case of* $l = 0$ *and* $l = L$*,* $\mathcal{I}_0$ *and* $\mathcal{I}_L$ *are always the identity maps since their indices are fixed once data is given with* $m'_0 = m_0 = d$ *and* $m'_L = m_L = d'$*. To sum up*

$$\mathcal{I}_0 : [m'_0] \to [m_0], \ \mathcal{I}_l : [m'_l] \to [m_l] \sqcup \{0\} \text{ for } l \in [L-1], \ \mathcal{I}_L : [m'_L] \to [m_L]. \tag{5.1}$$

Moreover, for any nonzero index $s \neq 0$*, if* $\mathcal{I}_l^{-1}(s) \neq \emptyset$ *for all* $l \in [0:L]$*, we say that* $\mathcal{I} = \{\mathcal{I}_l\}_{l=0}^L$ *is a* ***total (pull-back) index mapping****.*

Lemma 5.1. *For any affine embedding* $\mathcal{T} : \mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \to \mathrm{Tuple}_{\{m'_0,\cdots,m'_L\}}$ *satisfying the output preserving property, if there exists a total index mapping* $\mathcal{I} = \{\mathcal{I}_l\}_{l=0}^L$ *from* $\mathrm{NN}(\{m'_l\}_{l=0}^L)$ *to* $\mathrm{NN}(\{m_l\}_{l=0}^L)$ *and auxiliary variables* $\boldsymbol{\beta} = \{\boldsymbol{\beta}_j^{[l]} \in \mathbb{R} |\ l \in [0:L], j \in [m'_l] \backslash \mathcal{I}_l^{-1}(0)\}$*, such that for any given neuron belonging to* $\mathrm{NN}(\{m'_l\}_{l=0}^L)$*, located in layer* l *with index* j*, the following two statements hold:*

(i) If $\mathcal{I}_l(j) \neq 0$*,* $(\boldsymbol{f}_{\boldsymbol{\theta}_{\mathrm{wide}}}^{[l]})_j = (\boldsymbol{f}_{\boldsymbol{\theta}_{\mathrm{narr}}}^{[l]})_{\mathcal{I}_l(j)}$ *and* $(\boldsymbol{e}_{\boldsymbol{\theta}_{\mathrm{wide}}}^{[l]})_j = \boldsymbol{\beta}_j^{[l]} (\boldsymbol{e}_{\boldsymbol{\theta}_{\mathrm{narr}}}^{[l]})_{\mathcal{I}_l(j)}$*,*

(ii) If $\mathcal{I}_l(j) = 0$*,* $(\boldsymbol{f}_{\boldsymbol{\theta}_{\mathrm{wide}}}^{[l]})_j = \mathrm{Const}$ *and* $(\boldsymbol{e}_{\boldsymbol{\theta}_{\mathrm{wide}}}^{[l]})_j = 0$*,*

then $\mathcal{T}$ *is a critical embedding.*

Next, we propose a general compatible embedding method, where above embedding methods including one-step embedding and K-step composition embedding are its special cases.

Definition 5.2 (General compatible embedding). *Given an* $\mathrm{NN}(\{m_l\}_{l=0}^L)$ *and a wider* $\mathrm{NN}(\{m'_l\}_{l=0}^L)$*, then for any total index mapping* $\mathcal{I} = \{\mathcal{I}_l\}_{l=0}^L$ *from* $\mathrm{NN}(\{m'_l\}_{l=0}^L)$ *to* $\mathrm{NN}(\{m_l\}_{l=0}^L)$*, and for any tuple* $\boldsymbol{\alpha} := \{\boldsymbol{\alpha}^{[1]}, \boldsymbol{\alpha}_{\mathrm{b}}^{[1]}, \cdots, \boldsymbol{\alpha}^{[L]}, \boldsymbol{\alpha}_{\mathrm{b}}^{[L]}\} \in \mathrm{Tuple}_{\{m'_0,\cdots,m'_L\}}$ *satisfying some* ***compatibility conditions*** *(see Condition 1 and Condition 2), we define a* ***general embedding*** $\mathcal{T}_{\mathcal{I}}^{\boldsymbol{\alpha}} : \mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \to \mathrm{Tuple}_{\{m'_0,\cdots,m'_L\}}$ *as: For any parameters* $\boldsymbol{\theta}_{\mathrm{narr}} = (\boldsymbol{W}_{\mathrm{narr}}^{[1]}, \boldsymbol{b}_{\mathrm{narr}}^{[1]}, \cdots, \boldsymbol{W}_{\mathrm{narr}}^{[L]}, \boldsymbol{b}_{\mathrm{narr}}^{[L]}) \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$*,*

$$\begin{aligned}\mathcal{T}_{\mathcal{I}}^{\boldsymbol{\alpha}}(\boldsymbol{\theta}_{\mathrm{narr}}) := \Big(&\boldsymbol{\alpha}^{[1]} \circ \boldsymbol{W}_{\mathrm{inter}}^{[1]}, \boldsymbol{\alpha}_{\mathrm{b}}^{[1]} + \boldsymbol{b}_{\mathrm{inter}}^{[1]}, \cdots, \\ &\boldsymbol{\alpha}^{[l]} \circ \boldsymbol{W}_{\mathrm{inter}}^{[l]}, \boldsymbol{\alpha}_{\mathrm{b}}^{[l]} + \boldsymbol{b}_{\mathrm{inter}}^{[l]}, \cdots, \\ &\boldsymbol{\alpha}^{[L]} \circ \boldsymbol{W}_{\mathrm{inter}}^{[L]}, \boldsymbol{\alpha}_{\mathrm{b}}^{[L]} + \boldsymbol{b}_{\mathrm{inter}}^{[L]}\Big),\end{aligned}$$

where

$$\boldsymbol{W}_{\mathrm{inter}}^{[l]} := \left[\left(\boldsymbol{W}_{\mathrm{narr}}^{[l]}\right)_{\mathcal{I}_l(i),\mathcal{I}_{l-1}(j)}\right], \quad \boldsymbol{b}_{\mathrm{inter}}^{[l]} := \left(\left(\boldsymbol{b}_{\mathrm{narr}}^{[l]}\right)_{\mathcal{I}_l(k)}\right)$$

for $l \in [L]$ *with* $i,\ k \in [m'_l]$*,* $j \in [m'_{l-1}]$*, and* $\circ$ *is the Hadamard product.*

Remark 5.1. Since $\mathcal{I}_l : [m'_l] \to [m_l] \sqcup \{0\}$ for $l \in [L-1]$, and the components in $\boldsymbol{W}_{\mathrm{narr}}^{[l]}$ and $\boldsymbol{b}_{\mathrm{narr}}^{[l]}$ are not defined for zero indices, i.e., no definitions can be found for $(\boldsymbol{W}_{\mathrm{narr}}^{[l]})_{0j}$, $(\boldsymbol{W}_{\mathrm{narr}}^{[l]})_{i0}$, $(\boldsymbol{b}_{\mathrm{narr}}^{[l]})_0$, for any $l \in [L-1]$ with $i \in [m_l]$ and $j \in [m_{l-1}] \sqcup \{0\}$, for convenience of expression, we set $(\boldsymbol{W}_{\mathrm{narr}}^{[l]})_{0j} = 1$, $(\boldsymbol{W}_{\mathrm{narr}}^{[l]})_{i0} = 1$, and $(\boldsymbol{b}_{\mathrm{narr}}^{[1]})_0 = 0$, with $i \in [m_l]$ and $j \in [m_{l-1}] \sqcup \{0\}$.

Now we proceed to state out the **certain conditions** for the tuple

$$\boldsymbol{\alpha} = \{\boldsymbol{\alpha}^{[1]}, \boldsymbol{\alpha}_{\mathrm{b}}^{[1]}, \cdots, \boldsymbol{\alpha}^{[L]}, \boldsymbol{\alpha}_{\mathrm{b}}^{[L]}\} \in \mathrm{Tuple}_{\{m'_0,\cdots,m'_L\}} \tag{5.2}$$

in Definition 5.2.

Condition 1 (Compatibility conditions I (see Fig. 5.1 for illustration)**).** The elements $\{\boldsymbol{\alpha}^{[l]}\}_{l=1}^L$ in (5.2) satisfy that: there exists a collection of auxiliary variables

$$\boldsymbol{\beta} := \left\{\boldsymbol{\beta}_j^{[l]} \in \mathbb{R} |\ l \in [0:L],\ j \in [m'_l] \backslash \mathcal{I}_l^{-1}(0)\right\}$$

such that

- $\boldsymbol{\beta}_k^{[L]} = 1$ for $k \in [m'_L]$. (Since $\mathcal{I}_L$ is the identity map, $\mathcal{I}_L^{-1}(0) = \emptyset$).
- **Forward conditions**:

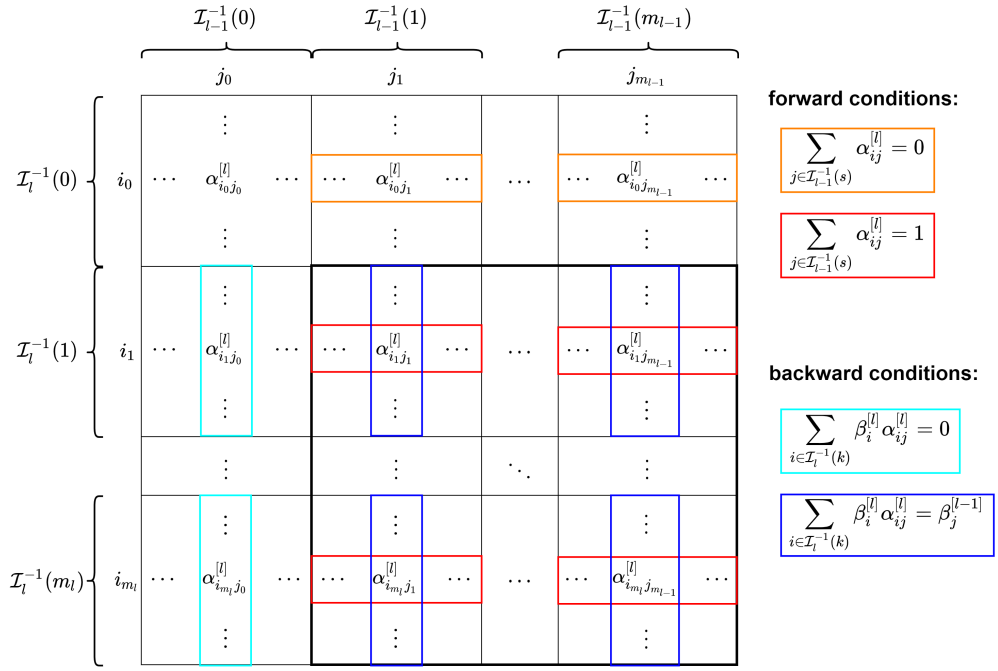

Figure 5.1: Illustration of forward and backward compatibility conditions I for $\boldsymbol{\alpha}^{[l]}$.

  - **Effective neurons** $\mathcal{I}_{l-1}^{-1}(s)$ forward to an effective neuron i: For $i \notin \mathcal{I}_l^{-1}(0)$, $s \in [m_{l-1}]$, we have $\sum_{j\in\mathcal{I}_{l-1}^{-1}(s)} \boldsymbol{\alpha}_{ij}^{[l]} = 1$;
  - **Effective neurons** $\mathcal{I}_{l-1}^{-1}(s)$ forward to a null neuron i: For $i \in \mathcal{I}_l^{-1}(0)$, $s \in [m_{l-1}]$, we have $\sum_{j\in\mathcal{I}_{l-1}^{-1}(s)} \boldsymbol{\alpha}_{ij}^{[l]} = 0$.

- **Backward conditions**:
  - **Effective neurons** $\mathcal{I}_l^{-1}(k)$ backpropagate to an effective neuron j: For $j \notin \mathcal{I}_{l-1}^{-1}(0)$, $k \in [m_l]$, we have $\sum_{i\in\mathcal{I}_l^{-1}(k)} \boldsymbol{\beta}_i^{[l]} \boldsymbol{\alpha}_{ij}^{[l]} = \boldsymbol{\beta}_j^{[l-1]}$;
  - **Effective neurons** $\mathcal{I}_l^{-1}(k)$ backpropagate to a null neuron j: For $j \in \mathcal{I}_{l-1}^{-1}(0)$, $k \in [m_l]$, we have $\sum_{i\in\mathcal{I}_l^{-1}(k)} \boldsymbol{\beta}_i^{[l]} \boldsymbol{\alpha}_{ij}^{[l]} = 0$.

The compatibility conditions for $\{\boldsymbol{\alpha}_{\mathrm{b}}^{[l]}\}_{l=1}^{L}$ is stated in the following. In order for that, we need another collection of auxiliary variables $\boldsymbol{B}_* := \{(\boldsymbol{b}_*^{[l]})_i \in \mathbb{R} |\ l \in [0:L],\ i \in \mathcal{I}_l^{-1}(0)\}$. We term $\boldsymbol{B}_*$ *the effective biases for null neurons.*

Condition 2 (**Compatibility conditions II** (see Fig. 5.2 for illustration))**.** The rest of the elements in $\boldsymbol{\alpha}$, i.e., $\{\boldsymbol{\alpha}_{\mathrm{b}}^{[l]}\}_{l=1}^{L}$ satisfy that: There also exists a collection of auxiliary variables,

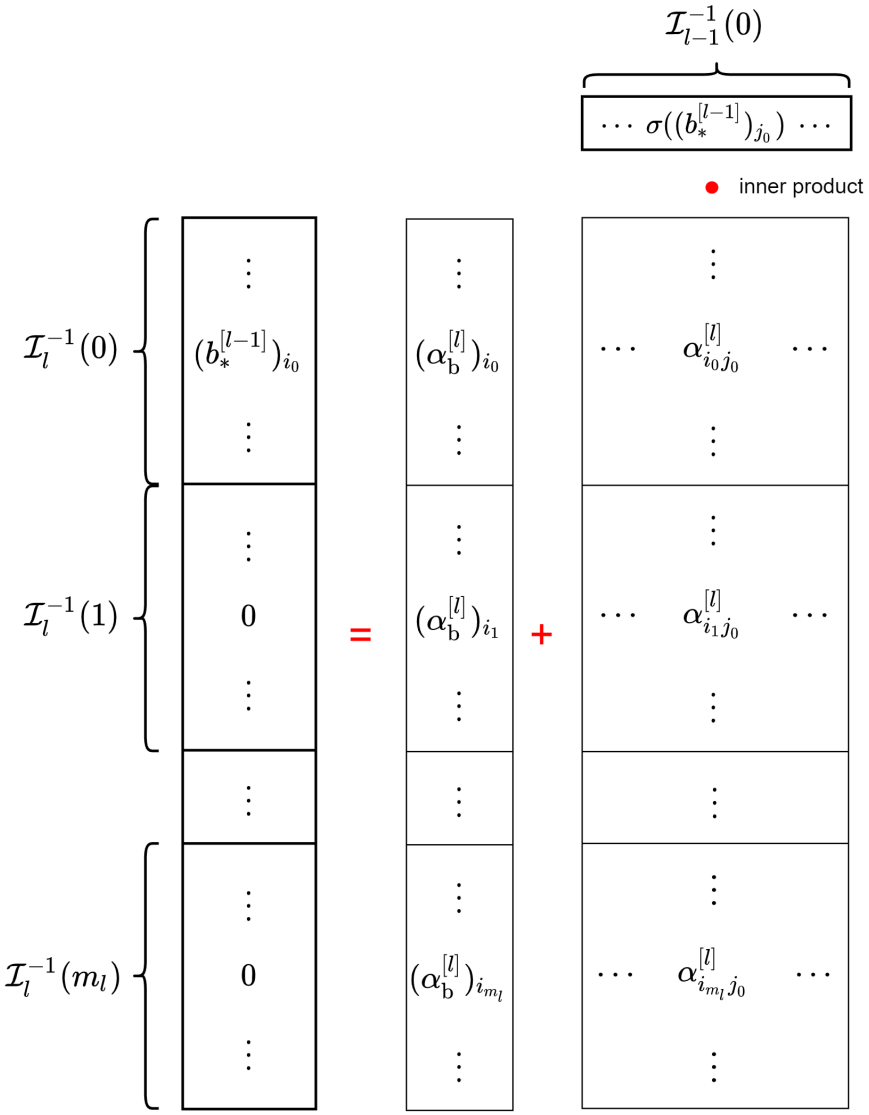

Figure 5.2: Illustration of compatibility conditions II for $\boldsymbol{\alpha}_{\mathrm{b}}^{[l]}$.

termed the *effective biases of null neurons*

$$\boldsymbol{B}_* := \left\{ (\boldsymbol{b}_*^{[l]})_i \in \mathbb{R} |\ l \in [0:L],\ i \in \mathcal{I}_l^{-1}(0) \right\},$$

we have

- **Effective neurons:** $(\boldsymbol{\alpha}_{\mathrm{b}}^{[l]})_i = 0$ for any $l \in [L]$ with $i \notin \mathcal{I}_l^{-1}(0)$;
- **Null neurons $\mathcal{I}_{l-1}^{-1}(0)$ forward** to a null neuron i: for $i \in \mathcal{I}_l^{-1}(0)$, $s = 0$, we have

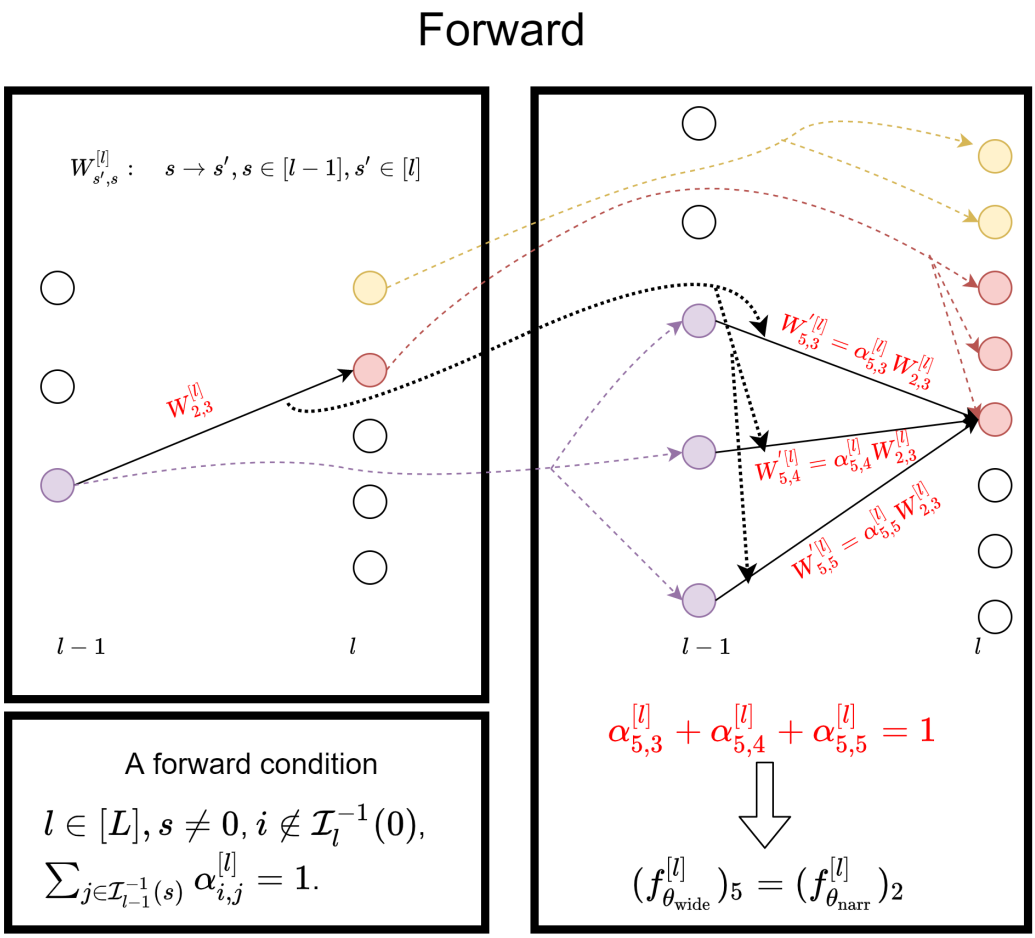

Figure 5.3: Illustration of the forward conditions for split neurons in compatibility conditions I. The sum of the input weights from neurons that are split from the same previous neuron to a post neuron should be equal to the weight from the previous to the post neuron.

$\sum_{j \in \mathcal{I}_{l-1}^{-1}(0)} \boldsymbol{\alpha}^{[l]}_{ij} \sigma((\boldsymbol{b}^{[l-1]}_*)_j) + (\boldsymbol{\alpha}^{[l]}_{\mathrm{b}})_i = (\boldsymbol{b}^{[l]}_*)_i$;

- **Null neurons** $\mathcal{I}_{l-1}^{-1}(0)$ **forward** to an effective neuron i: for $i \notin \mathcal{I}_l^{-1}(0)$, $s = 0$, we have $\sum_{j \in \mathcal{I}_{l-1}^{-1}(0)} \boldsymbol{\alpha}^{[l]}_{ij} \sigma((\boldsymbol{b}^{[l-1]}_*)_j) + (\boldsymbol{\alpha}^{[l]}_{\mathrm{b}})_i = 0$.

We then illustrate the general compatible embedding as follows. First, we illustrate an example of the general compatible embedding without null neurons through the forward conditions in Fig. 5.3 and the backward conditions in Fig. 5.4. Second, we illustrate the conditions related to null neurons in Fig. 5.5.

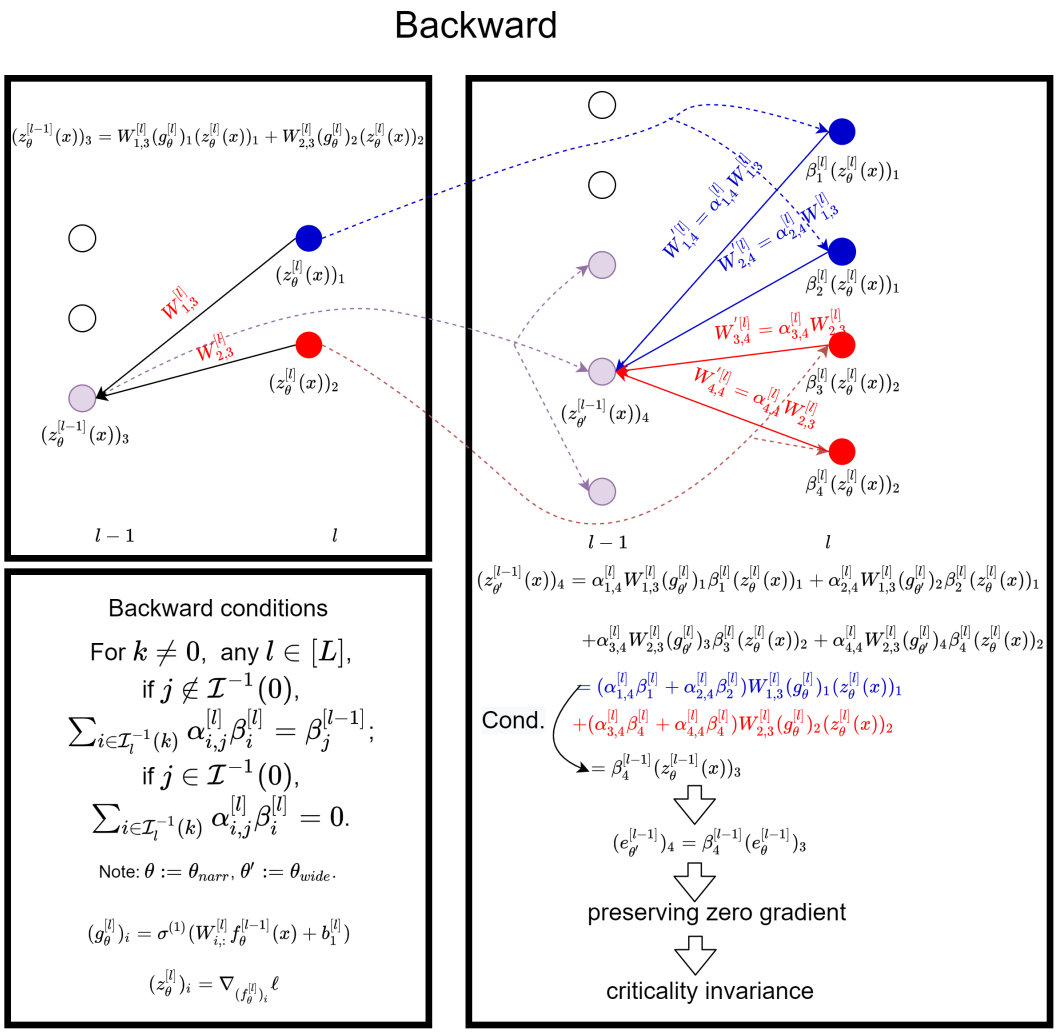

Figure 5.4: Illustration of the backward conditions for split neurons in compatibility conditions I. The gradient of the loss w.r.t. to the neuron output of the wide network is proportional to the gradient of the loss w.r.t. the corresponding neuron in the narrow network.

Theorem 5.1. *General compatible embedding is a critical embedding.*

Remark that we later name it as general compatible critical embedding in this work.

5.1 Special cases of general compatible critical embedding

In the following, we present some special cases of general compatible critical embedding. Specifically, the three-fold global splitting embedding is the key to the proof of a necessary condition of a "truly-bad" critical point in Section 6.

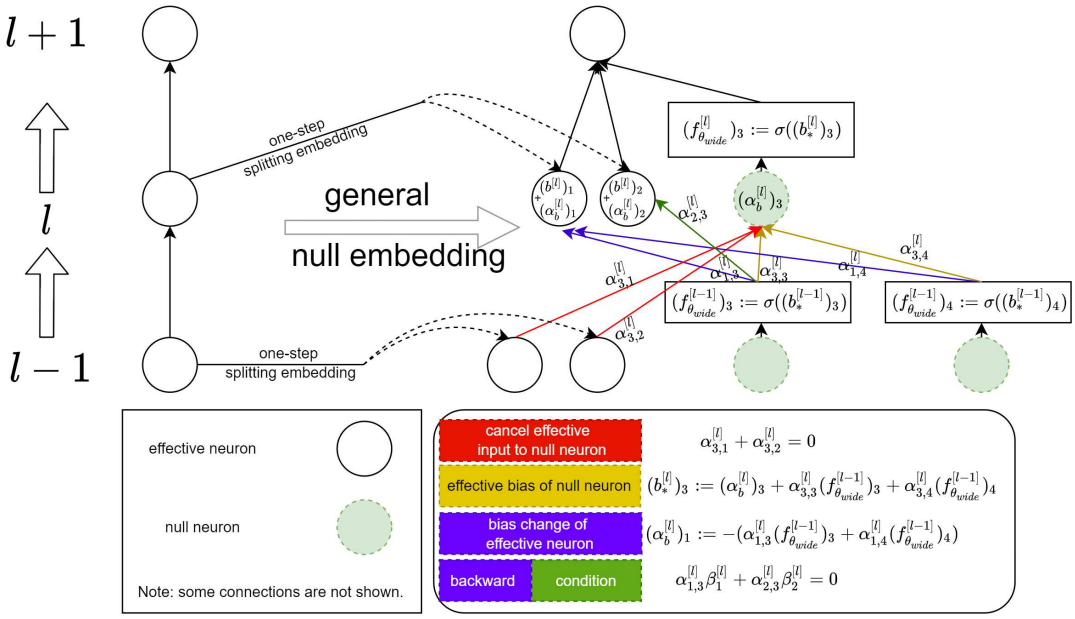

Figure 5.5: Illustration of compatibility conditions for null neurons. Red: The input from effective neurons split from the same neuron should cancel out. Yellow: the effective bias of a null neuron is the sum of its bias (can be any value) and its input from other null neurons. Purple: The added null neurons have constant input to the effective neuron, which would be added a bias correction to cancel the constant input. Purple+Green: the backward condition of a null neuron leads to the zero gradient.

Definition 5.3 (Splitting embedding). *For a general compatible critical embedding $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$, if there is no null neuron in the embedded NN, i.e., $\mathcal{I}_l^{-1}(0) = \emptyset$ for any $l \in [L-1]$, then we call $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$ a splitting embedding.*

Definition 5.4 (Null embedding). *For a general compatible critical embedding $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$, if only null neurons are added, i.e., $\#\mathcal{I}_l^{-1}(0) = m'_l - m_l$ for any $l \in [L-1]$, hence there is no neuron splitting, i.e., $\#\mathcal{I}_l^{-1}(s) = 1$ for any $l \in [L-1]$, $s \in [m_l]$, then we call $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$ a null embedding.*

Remark 5.2. One-step null embedding and its multi-step composition are special cases of null embedding. Similarly, one-step splitting embedding and its multi-step composition are special cases of splitting embedding. Multi-step embedding composed by a mixture of one-step null and splitting embedding is a special case of general compatible critical embedding.

Example 5.1 (A three-fold global splitting embedding). We define the operator $\mathcal{T}_{\text{global}}$ applying on $\boldsymbol{\theta}$ as follows

$$\mathcal{T}_{\text{global}}(\boldsymbol{\theta})|_1 = \left(\begin{bmatrix} \boldsymbol{W}^{[1]} \\ \boldsymbol{W}^{[1]} \\ \boldsymbol{W}^{[1]} \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}^{[1]} \\ \boldsymbol{b}^{[1]} \\ \boldsymbol{b}^{[1]} \end{bmatrix} \right),$$

$$\mathcal{T}_{\text{global}}(\boldsymbol{\theta})|_l = \left(\begin{bmatrix} \boldsymbol{W}^{[l]} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}^{[l]} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}^{[l]} \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}^{[l]} \\ \boldsymbol{b}^{[l]} \\ \boldsymbol{b}^{[l]} \end{bmatrix} \right), \ l \in [2 : L-1],$$
$$\mathcal{T}_{\text{global}}(\boldsymbol{\theta})|_L = \left(\left[\boldsymbol{W}^{[L]}, \boldsymbol{W}^{[L]}, -\boldsymbol{W}^{[L]} \right], \boldsymbol{b}^{[L]} \right).$$

Remark 5.3. For this global splitting embedding, $m_l' = 3m_l$ and $\mathcal{I}_l^{-1}(s) = \{s, s+m_l, s+2m_l\}$ for any $l \in [L-1]$ and $s \in [m_l]$. Moreover, for $l = 1$,

$$\boldsymbol{\alpha}^{[1]} = \begin{bmatrix} \boldsymbol{1}_{m_1 \times m_0} \\ \boldsymbol{1}_{m_1 \times m_0} \\ \boldsymbol{1}_{m_1 \times m_0} \end{bmatrix},$$
$$\boldsymbol{\alpha}_{\text{b}}^{[1]} = \boldsymbol{0}_{3m_1 \times 1},$$

and for $l \in [2 : L-1]$, we have

$$\boldsymbol{\alpha}^{[l]} = \begin{bmatrix} \boldsymbol{1}_{m_l \times m_{l-1}} & \boldsymbol{0}_{m_l \times m_{l-1}} & \boldsymbol{0}_{m_l \times m_{l-1}} \\ \boldsymbol{0}_{m_l \times m_{l-1}} & \boldsymbol{1}_{m_l \times m_{l-1}} & \boldsymbol{0}_{m_l \times m_{l-1}} \\ \boldsymbol{0}_{m_l \times m_{l-1}} & \boldsymbol{0}_{m_l \times m_{l-1}} & \boldsymbol{1}_{m_l \times m_{l-1}} \end{bmatrix},$$
$$\boldsymbol{\alpha}_{\text{b}}^{[l]} = \boldsymbol{0}_{3m_l \times 1},$$

and

$$\boldsymbol{\alpha}^{[L]} = \begin{bmatrix} \boldsymbol{1}_{m_L \times m_{L-1}} & \boldsymbol{1}_{m_L \times m_{L-1}} & -\boldsymbol{1}_{m_L \times m_{L-1}} \end{bmatrix},$$
$$\boldsymbol{\alpha}_{\text{b}}^{[L]} = \boldsymbol{0}_{m_L \times 1}.$$

Moreover, for any $l \in [L-1]$, $\boldsymbol{\beta}_j^{[l]} = 1$ when $j \in [2m_l]$ and $\boldsymbol{\beta}_j^{[l]} = -1$ when $j \in [2m_l + 1 : 3m_l]$. It is a special case of splitting embedding.

Example 5.2 (A two-fold global null embedding)**.** We define the operator $\mathcal{T}_{\text{GN}}$ applying on $\boldsymbol{\theta}$ as follows

$$\mathcal{T}_{\text{GN}}(\boldsymbol{\theta})|_1 = \left(\begin{bmatrix} \boldsymbol{W}^{[1]} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}^{[1]} \\ \boldsymbol{b}_{**}^{[1]} \end{bmatrix} \right),$$
$$\mathcal{T}_{\text{GN}}(\boldsymbol{\theta})|_l = \left(\begin{bmatrix} \boldsymbol{W}^{[l]} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}^{[l]} \\ \boldsymbol{b}_{**}^{[l]} \end{bmatrix} \right),$$
$$\mathcal{T}_{\text{GN}}(\boldsymbol{\theta})|_L = \left(\left[\boldsymbol{W}^{[L]}, \boldsymbol{0} \right], \boldsymbol{b}^{[L]} \right),$$

where $\boldsymbol{b}_{**}^{[l]}$ takes an arbitrary real vector of the same dimension as $\boldsymbol{b}^{[l]}$ for any $l \in [L-1]$.

Remark 5.4. For this global null embedding, $m_l' = 2m_l$, $\mathcal{I}_l^{-1}(s) = \{s\}$ and $\mathcal{I}_l^{-1}(0) = [m_l + 1 : 2m_l]$ for any $l \in [L-1]$ and $s \in [m_l]$. Then for $l = 1$,

$$\boldsymbol{\alpha}^{[1]} = \begin{bmatrix} \boldsymbol{1}_{m_1 \times m_0} \\ \boldsymbol{0}_{m_1 \times m_0} \end{bmatrix},$$

$$\boldsymbol{\alpha}_{\mathrm{b}}^{[1]} = \begin{bmatrix} \mathbf{0}_{m_1 \times 1} \\ \boldsymbol{b}_{**}^{[1]} \end{bmatrix},$$

and for $l \in [2 : L-1]$, we have

$$\boldsymbol{\alpha}^{[l]} = \begin{bmatrix} \mathbf{1}_{m_l \times m_{l-1}} & \mathbf{0}_{m_l \times m_{l-1}} \\ \mathbf{0}_{m_l \times m_{l-1}} & \mathbf{0}_{m_l \times m_{l-1}} \end{bmatrix},$$
$$\boldsymbol{\alpha}_{\mathrm{b}}^{[l]} = \begin{bmatrix} \mathbf{0}_{m_l \times 1} \\ \boldsymbol{b}_{**}^{[l]} \end{bmatrix},$$

and

$$\boldsymbol{\alpha}^{[L]} = \begin{bmatrix} \mathbf{1}_{m_L \times m_{L-1}} & \mathbf{0}_{m_L \times m_{L-1}} \end{bmatrix},$$
$$\boldsymbol{\alpha}_{\mathrm{b}}^{[L]} = \mathbf{0}_{m_L \times 1}.$$

Moreover, for any $l \in [L-1]$, $(\boldsymbol{b}_{*}^{[l]})_i = (\boldsymbol{b}_{**}^{[l]})_{i-m_l}$ for $i \in [m_l + 1 : 2m_l]$. It is a special case of null embedding.

6 Analysis of critical points/submanifolds by critical embeddings

6.1 Degeneracy of critical points/submanifolds

A key observation to the critical embeddings proposed above is that it is not unique. Actually, given any NN and a wider NN, there is a class of critical embeddings. Therefore, any critical point can be embedded to a set of critical points in a wider NN. By a K-step composition embedding, because embedding $\mathcal{T}_{l_i,s_i}^{\alpha_i}$ at each step i has one degree of freedom parameterized by α_i given l_i and s_i, one critical point in general can be embedded to a K-dimensional critical affine subspace, which provides a lower bound to the degeneracy of embedded critical points. Precisely, in [24], we prove the following theorem using composition of one-step splitting embeddings.

Theorem 6.1 (Theorem 2 in [24])**.** *Given an* $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ *and a* K*-neuron wider* $\mathrm{NN}(\{m_l'\}_{l=0}^{L})$*, then for any critical point* $\boldsymbol{\theta}_{\mathrm{narr}}^{\mathrm{c}} = (\boldsymbol{W}^{[1]}, \boldsymbol{b}^{[1]}, \cdots, \boldsymbol{W}^{[L]}, \boldsymbol{b}^{[L]})$ *satisfying* $\boldsymbol{W}^{[l]} \neq \mathbf{0}$ *for each* $l \in [L]$*,* $\boldsymbol{\theta}_{\mathrm{narr}}^{\mathrm{c}}$ *can be critically embedded to a* K*-dimensional critical affine subspace* $\mathcal{M}_{\mathrm{wide}} = \{\boldsymbol{\theta}_{\mathrm{wide}} + \sum_{i=1}^{K} \alpha_i \boldsymbol{\mu}_i | \alpha_i \in \mathbb{R}\}$ *of loss landscape of* $\mathrm{NN}(\{m_l'\}_{l=0}^{L})$*. Here* $\boldsymbol{\theta}_{\mathrm{wide}} := (\prod_{l=1}^{K} \mathcal{T}_{l_l,s_l})(\boldsymbol{\theta}_{\mathrm{narr}})$ *and* $\boldsymbol{\mu}_i := \mathcal{T}_{l_K,s_K} \cdots \mathcal{V}_{l_i,s_i} \cdots \mathcal{T}_{l_1,s_1}(\boldsymbol{\theta}_{\mathrm{narr}})$*, where* $s_l \neq 0$ *for all* $l \in [K]$*. (The definitions of* $\mathcal{V}_{l_l,s_l}$ *and* $\mathcal{T}_{l_l,s_l}$ *can be found in Definition* 4.4*).*

In addition, in this work, as we propose above a wider class of general compatible critical embeddings, we attempt to obtain a better estimate of the dimension of critical submanifolds corresponding to any given critical function $\boldsymbol{f}_{\boldsymbol{\theta}_{\mathrm{narr}}^{\mathrm{c}}}$ of a narrow NN. A gross estimate yields the following heuristic argument.

Heuristic argument. Given any total index mapping $\mathcal{I}$, the degree of freedom of all possible $\boldsymbol{\alpha}$ for general compatible critical embedding $\mathcal{T}_{\mathcal{I}}^{\boldsymbol{\alpha}}$ to a K-neuron wider NN is $K + \sum_{l\in[L]} K_l K_{l-1}$, where $K_l := m_l' - m_l$, and $K := \sum_{l\in[L-1]} K_l$.

Justification. By the definition of general compatible critical embedding, $\{\boldsymbol{\alpha}^{[l]}\}_{l=1}^{L}$ and auxiliary variables $\boldsymbol{\beta} = \{\boldsymbol{\beta}_j^{[l]} \in \mathbb{R} |\ l \in [L], j \in [m_l']\backslash \mathcal{I}_l^{-1}(0)\}$ with $\boldsymbol{\beta}_k^{[L]} = \mathbf{1}$ satisfy forward and backward compatibility conditions.

Step 1: We first observe that $\sum_{i\in\mathcal{I}_l^{-1}(s)} \boldsymbol{\beta}_i^{[l]} = 1$ for any $s \in [m_l]$. Thus, $\boldsymbol{\beta}$ has $K - m_{\text{null}}$ degrees of freedom, where m_{null} is the number of null neurons.

Step 2: Given any such $\boldsymbol{\beta}$, we now consider the degrees of freedom for $\{\boldsymbol{\alpha}^{[l]}\}_{l=1}^{L}$. Note that forward and backward conditions are not independent of one another because

$$\sum_{j\in\mathcal{I}_{l-1}^{-1}(s)} \sum_{i\in\mathcal{I}_l^{-1}(k)} \boldsymbol{\beta}_i^{[l]} \boldsymbol{\alpha}_{ij}^{[l]} = \sum_{i\in\mathcal{I}_l^{-1}(k)} \boldsymbol{\beta}_i^{[l]} = 1 = \sum_{j\in\mathcal{I}_{l-1}^{-1}(s)} \boldsymbol{\beta}_j^{[l-1]}$$

automatically holds for the given $\boldsymbol{\beta}$. Therefore, there are $m_l' m_{l-1} + m_l m_{l-1}' - m_l m_{l-1}$ independent linear equations for $m_l' m_{l-1}'$ parameters in $\boldsymbol{\alpha}^{[l]}$ for each layer l, resulting in $\sum_{l\in[L]} K_l K_{l-1}$ degrees of freedom in total.

Step 3: Given any effective biases $\boldsymbol{B}_*$ as auxiliary variables for all null neurons and any $\boldsymbol{\alpha}^{[l]}$ satisfying forward and backward conditions, $(\boldsymbol{\alpha}_{\text{b}}^{[l]})_i$ is uniquely determined for all neurons in the wide NN. Therefore, there are m_{null} degrees of freedom in $\boldsymbol{B}_*$.

In the end, the degrees of freedom in $\boldsymbol{\alpha}$ is the summation of degrees of freedom in auxiliary variables $\boldsymbol{\beta}$, degrees of freedom in $\{\boldsymbol{\alpha}^{[l]}\}_{l=1}^{L}$ given $\boldsymbol{\beta}$ and additional degrees of freedom in $\boldsymbol{B}_*$ for all null neurons, which add up to $K + \sum_{l\in[L]} K_l K_{l-1}$ degrees of freedom.

Intuitively, degrees of freedom in critical embedding $\mathcal{T}$ transform into dimensions of critical submanifolds, resulting in a higher estimate of degrees of degeneracy $K + \sum_{l\in[L]} K_l K_{l-1}$ in comparison to K obtained by multi-step composition embedding in [24]. We note that the nonlinear coupling between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in general results in non-affine curved critical submanifolds for three-layer or deeper NNs containing the corresponding K-dimensional critical affine subspaces identified in Theorem 6.1. A more rigorous estimate requires careful handling of the nonlinear coupling. We leave it to later works.

6.2 Irreversible transition to strict-saddle point

In this subsection, we look further into the transition between different types of critical points, e.g., local minima, saddle points, through critical embeddings. We are specifically interested in the transition of a critical point to a strict-saddle point due to its good optimization guarantee detailed in [22]. Strict-saddle point is defined as follows.

Definition 6.1 (Strict-saddle). *$\boldsymbol{\theta}$ is a strict-saddle point of loss landscape $R_S(\cdot)$ if*

(i) $\nabla R_S(\boldsymbol{\theta}) = 0$;

(ii) Hessian matrix $\boldsymbol{H}_S(\boldsymbol{\theta})$ has at least one negative eigenvalue.

Based on above definition, we prove the following irreversibility property for any critical embedding, which guarantees that a strict-saddle point always embeds to strict-saddle points.

Theorem 6.2. *Given an* $\mathrm{NN}(\{m_l\}_{l=0}^L)$ *and any of its parameters* $\boldsymbol{\theta} \in \mathbb{R}^M$*, for any critical embedding* $\mathcal{T} : \mathbb{R}^M \to \mathbb{R}^{M'}$ *to any wider* $\mathrm{NN}(\{m'_l\}_{l=0}^L)$*, the number of positive, zero, negative eigenvalues of* $\boldsymbol{H}_S(\mathcal{T}(\boldsymbol{\theta}))$ *is no less than the counterparts of* $\boldsymbol{H}_S(\boldsymbol{\theta})$*.*

6.3 "Truly-bad" critical point

The above irreversibility property of any critical embedding provokes the following thought about a conventional bad local minimum: for any bad local minimum of a given NN, if it can become a strict-saddle point in wider NNs, then it should not be a problem as we can simply use a wider NN in practice. However, there is still a "truly-bad" situation in which a critical point may never become a strict-saddle point through any critical embedding. In the following, through proving a stringent necessary condition for such a "truly-bad" critical point defined below, we justify its rarity, hence providing a potential mechanism to understand the easy optimization of wide NNs widely observed in practice.

We denote hereafter that $\boldsymbol{A} \succeq \boldsymbol{B}$ if and only if $\boldsymbol{A} - \boldsymbol{B}$ is a semi-positive definite matrix, and $\boldsymbol{A} \succ \boldsymbol{B}$ if and only if $\boldsymbol{A} - \boldsymbol{B}$ is a strictly positive definite matrix.

Definition 6.2 ("Truly-bad" critical point). *Given any data* S*, loss* $\ell(\cdot,\cdot)$ *and activation* $\sigma(\cdot)$*, for any NN, if there exists a critical point* $\boldsymbol{\theta}^{\mathrm{c}} \in \boldsymbol{\Theta}^{\mathrm{c}}$ *satisfying that:*

(i) $\boldsymbol{\theta}^{\mathrm{c}}$ *is not a strict-saddle point* [22, Definition 1]*;*

(ii) For any critical embedding $\mathcal{T}$*,* $\mathcal{T}(\boldsymbol{\theta}^{\mathrm{c}})$ *is also not a strict-saddle point,*

*then we term this critical point a "**truly-bad" critical point**.*

We would like to introduce some additional notations in order to state Lemma 6.1 and Lemma 6.2. We denote

$$\boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}) := \sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_{ij} \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \nabla_{\boldsymbol{\theta}^{[L-1]}} (\boldsymbol{f}_{\boldsymbol{\theta}})_i \left(\nabla_{\boldsymbol{\theta}^{[L-1]}} (\boldsymbol{f}_{\boldsymbol{\theta}})_j\right)^{\mathsf{T}},$$

and

$$\boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}) := \sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_i \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \boldsymbol{W}_{i,j}^{[L]} \nabla_{\boldsymbol{\theta}^{[L-1]}} \nabla_{\boldsymbol{\theta}^{[L-1]}} \left(\boldsymbol{f}_{\boldsymbol{\theta}}^{[L-1]}\right)_j.$$

We denote further that $\boldsymbol{H}_S^{[L-1]}(\boldsymbol{\theta}) := \boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}) + \boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta})$. Obviously, $\boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta})$, $\boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta})$, $\boldsymbol{H}_S^{[L-1]}(\boldsymbol{\theta}) \in \mathbb{R}^{M^{[L-1]} \times M^{[L-1]}}$, and we state Lemma 6.1 and Lemma 6.2 as follows.

Lemma 6.1. *Given any data* S*, loss* $\ell(\cdot,\cdot)$ *and activation* $\sigma(\cdot)$*, for any NN, if a critical point* $\boldsymbol{\theta}^{\mathrm{c}} \in \boldsymbol{\Theta}^{\mathrm{c}}$ *satisfies:*

(i) $\boldsymbol{H}_S(\boldsymbol{\theta}^{\mathrm{c}}) \succeq 0$*;*

(ii) $\boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}^{\rm c}) \neq \boldsymbol{0}$,

then there exists a general compatible critical embedding $\mathcal{T}$*, such that* $\mathcal{T}(\boldsymbol{\theta}^{\rm c})$ *is a strict-saddle point.*

Lemma 6.2. *Given any data* S*, loss* $\ell(\cdot,\cdot)$ *and activation* $\sigma(\cdot)$*, for any NN, if a critical point* $\boldsymbol{\theta}^{\rm c} \in \boldsymbol{\Theta}^{\rm c}$ *satisfies:*

(i) $\boldsymbol{H}_S(\boldsymbol{\theta}^{\rm c}) \succeq 0$;

(ii) $\boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}^{\rm c}) = \boldsymbol{0}$;

(iii) $\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}^{\rm c}) \neq \boldsymbol{0}$,

then there exists a general compatible critical embedding $\mathcal{T}$*, such that* $\mathcal{T}(\boldsymbol{\theta}^{\rm c})$ *is a strict-saddle point.*

Theorem 6.3. *Given any data* S*, loss* $\ell(\cdot,\cdot)$ *and activation* $\sigma(\cdot)$*, for any NN, if a critical point* $\boldsymbol{\theta}^{\rm c}$ *with* $\boldsymbol{H}_S(\boldsymbol{\theta}^{\rm c}) \succeq 0$ *satisfies* $\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}^{\rm c}) \neq \boldsymbol{0}$*, then there exists a general compatible critical embedding* $\mathcal{T}$ *such that* $\mathcal{T}(\boldsymbol{\theta}^{\rm c})$ *is a strict-saddle point, i.e.,* $\boldsymbol{H}_S(\mathcal{T}(\boldsymbol{\theta}^{\rm c}))$ *has at least one negative eigenvalue.*

Proof. This can be directly obtained by Lemmas 6.1 and 6.2. □

Theorem (short version of Theorem 6.3)**.** $\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}^{\rm c}) = \boldsymbol{0}$ is a necessary condition for a critical point $\boldsymbol{\theta}^{\rm c}$ being a "truly-bad" critical point.

7 Conclusion and discussion

In this work, we prove the Embedding Principle that loss landscape of an NN *contains* all critical points/functions of all the narrower NNs. We define the critical embedding, which serves as the key tool not only to the proof of Embedding Principle but also to the study of the general geometry of loss landscape. Importantly, we discover a wide class of general compatible embedding, by which we obtain rich understanding about the critical points/submanifolds, e.g., lower bound of their degree of degeneracy, their easy and irreversible transition to strict-saddle points.

The general compatible embedding proposed in this work unravels that the critical embedding in general is not limited to the composition of one-step embeddings, but instead, it can be a collective operation. As a consequence, all critical points embedded from a narrower NN form high-dimensional critical submanifolds, which in general are not affine subspaces for three-layer or deeper NNs. It is interesting and important. However, it remains a problem for the future study about whether the general compatible embeddings in certain sense are all the critical embeddings.

Embedding Principle provides an integrated view about loss landscapes of NNs with different widths. Specifically, it informs us that a specific bad local minimum in a NN with a fixed width may not be a big deal for optimization as long as it can become a strict-saddle point in wider NNs, i.e., not a “truly-bad” critical point. In this work, we prove a stringent necessary condition for a “truly-bad” critical point. Still, it remains a problem

about whether non-trivial "truly-bad" critical points indeed exist, and whether they are indeed a headache for optimization. We remark that complementary to the study of the "truly-bad" critical points, we also need to further study whether NNs can generate new bad local-minima forever when they become wider and wider to better understand the easy optimization of wide NNs.

We emphasize that Embedding Principle provides a function space view on the critical points /submanifolds of loss landscape, which is of great importance for studying the implicit regularization and generalization of NNs. In recent years, it has been more and more clear that optimization and generalization are heavily intertwined with one another for deep learning. Therefore, one can not expect to develop a deep learning theory with optimization theory and generalization theory established separately. Yet, for a long time, the study of loss landscape focuses mainly on the parameter space in pursuit of an optimization guarantee. On the other hand, the study of generalization focuses mainly on the function space, failing to incorporate the geometry of loss landscape in parameter space which is key to the training. Now, by the Embedding Principle, we see a hierarchical structure of critical functions of different complexities of the loss landscape, which originate from loss landscape with clear optimization implication while being amenable to the analysis of complexity with clear generalization implication. Such a critical function hierarchy reflects the degree of matching between the NN architecture and data, i.e., if critical functions of narrow NNs attain low training error, then the NN architecture well matches the target function and may obtain a well generalized solution through training like in Fig. 4.3. Even though more works need to be done to unravel the full details and implication of this critical function hierarchy, we believe it is an important piece and may be a key to the deep learning theory.

Acknowledgments

This work is sponsored by the National Key R&D Program of China Grant No. 2019YFA0709503 (Z. X.), the Shanghai Sailing Program, the Natural Science Foundation of Shanghai Grant No. 20ZR1429000 (Z. X.), the National Natural Science Foundation of China Grant No. 62002221 (Z. X.), the National Natural Science Foundation of China Grant No. 12101401 (T. L.), the National Natural Science Foundation of China Grant No. 12101402 (Y. Z.), Shanghai Municipal of Science and Technology Project Grant No. 20JC1419500 (Y.Z.), Shanghai Municipal of Science and Technology Major Project No. 2021SHZDZX0102, and the HPC of School of Mathematical Sciences and the Student Innovation Center at Shanghai Jiao Tong University.

A Theoretical details

Lemma (Lemma 4.1 in main text.)**.** For any one-step null embedding $\mathcal{T}^{\alpha}_{l,0}$, given any $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ and its parameters $\boldsymbol{\theta}_{\mathrm{narr}} \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ with $\mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \in \mathcal{D}_{l,0}$, we have $\boldsymbol{\theta}_{\mathrm{wide}} := \mathcal{T}^{\alpha}_{l,0}(\boldsymbol{\theta}_{\mathrm{narr}})$ satisfies the following conditions: given any data S, loss $\ell(\cdot,\cdot)$

and activation $\sigma(\cdot)$, for any $l \in [L-1]$,

(i) feature vectors in

$$\boldsymbol{F}_{\boldsymbol{\theta}_{\text{wide}}}\colon \boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[l']} = \boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l']}, \text{ for } l' \in [L] \text{ and } l' \neq l,\ \boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[l]} = \left[(\boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]})^\intercal, \sigma(\alpha)\right]^\intercal;$$

(ii) feature gradients in

$$\boldsymbol{G}_{\boldsymbol{\theta}_{\text{wide}}}\colon \boldsymbol{g}_{\boldsymbol{\theta}_{\text{wide}}}^{[l']} = \boldsymbol{g}_{\boldsymbol{\theta}_{\text{narr}}}^{[l']}, \text{ for } l' \in [L] \text{ and } l' \neq l,\ \boldsymbol{g}_{\boldsymbol{\theta}_{\text{wide}}}^{[l]} = \left[(\boldsymbol{g}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]})^\intercal, \sigma^{(1)}(\alpha)\right]^\intercal;$$

(iii) error vectors in

$$\boldsymbol{Z}_{\boldsymbol{\theta}_{\text{wide}}}\colon \boldsymbol{z}_{\boldsymbol{\theta}_{\text{wide}}}^{[l']} = \boldsymbol{z}_{\boldsymbol{\theta}_{\text{narr}}}^{[l']}, \text{ for } l' \in [L] \text{ and } l' \neq l,\ \boldsymbol{z}_{\boldsymbol{\theta}_{\text{wide}}}^{[l]} = \left[\boldsymbol{z}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]}, 0\right]^\intercal;$$

(iv) $\mathcal{T}_{l,0}^{\alpha}$ is injective for all α;

(v) $\mathcal{T}_{l,0}^{\alpha}$ is an affine embedding for all α.

Proof. (i) By the construction of $\boldsymbol{\theta}_{\text{wide}}$, it is clear that $\boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[l']} = \boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l']}$ for all $l' \in [l-1]$. Then

$$\boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[l]} = \sigma\left(\begin{bmatrix} \boldsymbol{W}^{[l]} \\ \boldsymbol{0}_{1\times m_{l-1}} \end{bmatrix} \boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l-1]} + \begin{bmatrix} \boldsymbol{b}^{[l]} \\ \alpha \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]} \\ \sigma(\alpha) \end{bmatrix}. \tag{A.1}$$

Note that since

$$\alpha \begin{bmatrix} \boldsymbol{0}_{m_{l+1}\times(m_l+1)} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]} \\ \sigma(\alpha) \end{bmatrix} = \boldsymbol{0}_{m_{l+1}\times 1}.$$

Thus

$$\boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[l+1]} = \sigma\left(\begin{bmatrix} \boldsymbol{W}^{[l+1]}, \boldsymbol{0}_{m_{l+1}\times 1} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]} \\ \sigma(\alpha) \end{bmatrix} + \boldsymbol{0}_{m_{l+1}\times 1} + \boldsymbol{b}^{[l+1]} + \alpha\, \boldsymbol{0}_{m_{l+1}\times 1}\right) = \boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l+1]}. \tag{A.2}$$

Next, by the construction of $\boldsymbol{\theta}_{\text{wide}}$, it is clear that $\boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[l']} = \boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l']}$ for any $l' \in [l+1:L]$.

(ii) The results for feature gradients $\boldsymbol{g}_{\boldsymbol{\theta}_{\text{wide}}}^{[l']} = \boldsymbol{g}_{\boldsymbol{\theta}_{\text{narr}}}^{[l']}$ for $l' \in [L]$ can be calculated in a similar way except by replacing $\sigma(\cdot)$ with $\sigma^{(1)}(\cdot)$.

(iii) By the backpropagation and the above facts in (i), we have

$$\boldsymbol{z}_{\boldsymbol{\theta}_{\text{wide}}}^{[L]} = \nabla\ell(\boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[L]}, \boldsymbol{y}) = \nabla\ell(\boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[L]}, \boldsymbol{y}) = \boldsymbol{z}_{\boldsymbol{\theta}_{\text{narr}}}^{[L]}.$$

Recall the recurrence relation for $l' \in [l+1:L-1]$, then we recursively obtain the following equality for l' from $L-1$ down to $l+1$:

$$\boldsymbol{z}_{\boldsymbol{\theta}_{\text{wide}}}^{[l']} = (\boldsymbol{W}^{[l'+1]})^\intercal \left(\boldsymbol{z}_{\boldsymbol{\theta}_{\text{wide}}}^{[l'+1]} \circ \boldsymbol{g}_{\boldsymbol{\theta}_{\text{wide}}}^{[l'+1]}\right) = (\boldsymbol{W}^{[l'+1]})^\intercal \left(\boldsymbol{z}_{\boldsymbol{\theta}_{\text{narr}}}^{[l'+1]} \circ \boldsymbol{g}_{\boldsymbol{\theta}_{\text{narr}}}^{[l'+1]}\right) = \boldsymbol{z}_{\boldsymbol{\theta}_{\text{narr}}}^{[l']}. \tag{A.3}$$

Next,

$$\begin{aligned} \boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}} &= \left(\left[\boldsymbol{W}^{[l+1]}, \boldsymbol{0}_{m_{l+1}\times 1}\right] + \alpha\left[\boldsymbol{0}_{m_{l+1}\times(m_l+1)}\right]\right)^{\intercal}\left(\boldsymbol{z}^{[l+1]}_{\boldsymbol{\theta}_{\text{wide}}} \circ \boldsymbol{g}^{[l+1]}_{\boldsymbol{\theta}_{\text{wide}}}\right) \\ &= \begin{bmatrix} \boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \\ 0 \end{bmatrix} + \left[\boldsymbol{0}_{(m_l+1)\times 1}\right] = \left[\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}}, 0\right]^{\intercal}. \end{aligned} \tag{A.4}$$

Finally,

$$\begin{aligned} \boldsymbol{z}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}} &= \begin{bmatrix} \boldsymbol{W}^{[l]\intercal}, \boldsymbol{0}_{m_{l-1}\times 1} \end{bmatrix}\left(\begin{bmatrix} \boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \\ 0 \end{bmatrix} \circ \begin{bmatrix} \boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \\ \sigma^{(1)}(\alpha) \end{bmatrix}\right) \\ &= (\boldsymbol{W}^{[l]})^{\intercal}\left(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \circ \boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}}\right) + \boldsymbol{0}_{m_{l-1}\times 1} = \boldsymbol{z}^{[l-1]}_{\boldsymbol{\theta}_{\text{narr}}}. \end{aligned} \tag{A.5}$$

This with the recurrence relation once again leads to $\boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\text{wide}}} = \boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\text{narr}}}$ for all $l' \in [l-1]$.

(iv) If for $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \text{Tuple}_{\{m_0,\cdots,m_L\}}$ and $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$, since $\mathcal{T}^{\alpha}_{l,0}|_k$ for $k \neq l, l+1$ is the identity map, then if there exists some $k_0 \neq l, l+1$, such that $\boldsymbol{W}^{[k_0]}_1 \neq \boldsymbol{W}^{[k_0]}_2$ or $\boldsymbol{b}^{[k_0]}_1 \neq \boldsymbol{b}^{[k_0]}_2$, then obviously $\mathcal{T}^{\alpha}_{l,0}(\boldsymbol{\theta}_1) \neq \mathcal{T}^{\alpha}_{l,0}(\boldsymbol{\theta}_2)$. If $k_0 = l$ or $k_0 = l+1$, by similar reasoning, $\mathcal{T}^{\alpha}_{l,0}(\boldsymbol{\theta}_1) \neq \mathcal{T}^{\alpha}_{l,0}(\boldsymbol{\theta}_2)$.

(v) For $\boldsymbol{\theta}_0 = (\boldsymbol{W}^{[1]}_0, \boldsymbol{b}^{[1]}_0, \cdots, \boldsymbol{W}^{[L]}_0, \boldsymbol{b}^{[L]}_0) \in \text{Tuple}_{\{m_0,\cdots,m_L\}}$, we have

$$\begin{aligned} &\tilde{\mathcal{T}}^{\alpha}_{l,0}(\boldsymbol{\theta}_0) \\ :=&\mathcal{T}^{\alpha}_{l,0}(\boldsymbol{\theta}_0) - \mathcal{T}^{\alpha}_{l,0}(\boldsymbol{0}) \\ =&\left(\boldsymbol{W}^{[1]}_0, \boldsymbol{b}^{[1]}_0, \cdots, \begin{bmatrix} \boldsymbol{W}^{[l]}_0 \\ \boldsymbol{0}_{1\times m_{l-1}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}^{[l]}_0 \\ 0 \end{bmatrix}, \left[\boldsymbol{W}^{[l+1]}_0, \boldsymbol{0}_{m_{l+1}\times 1}\right], \boldsymbol{b}^{[l+1]}_0, \cdots, \boldsymbol{W}^{[L]}_0, \boldsymbol{b}^{[L]}_0\right), \end{aligned}$$

obviously $\tilde{\mathcal{T}}^{\alpha}_{l,0}$ is a linear operator, thus $\mathcal{T}^{\alpha}_{l,0}$ is an affine operator. □

Lemma (Lemma 4.2 in main text)**.** For any one-step splitting embedding $\mathcal{T}^{\alpha}_{l,s}$, given any $\text{NN}(\{m_l\}_{l=0}^{L})$ and its parameters $\boldsymbol{\theta}_{\text{narr}} \in \text{Tuple}_{\{m_0,\cdots,m_L\}}$ with $\text{Tuple}_{\{m_0,\cdots,m_L\}} \in \mathcal{D}_{l,s}$, we have $\boldsymbol{\theta}_{\text{wide}} := \mathcal{T}^{\alpha}_{l,s}(\boldsymbol{\theta}_{\text{narr}})$ satisfies the following conditions: given any data S, loss $\ell(\cdot,\cdot)$ and activation $\sigma(\cdot)$, for any $l \in [L-1]$,

(i) feature vectors in

$$\boldsymbol{F}_{\boldsymbol{\theta}_{\text{wide}}}\colon \boldsymbol{f}^{[l']}_{\boldsymbol{\theta}_{\text{wide}}} = \boldsymbol{f}^{[l']}_{\boldsymbol{\theta}_{\text{narr}}}, \text{ for } l' \in [L] \text{ and } l' \neq l,\ \boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}} = \left[(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})^{\intercal}, (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s\right]^{\intercal};$$

(ii) feature gradients in

$$\boldsymbol{G}_{\boldsymbol{\theta}_{\text{wide}}}\colon \boldsymbol{g}^{[l']}_{\boldsymbol{\theta}_{\text{wide}}} = \boldsymbol{g}^{[l']}_{\boldsymbol{\theta}_{\text{narr}}}, \text{ for } l' \in [L] \text{ and } l' \neq l,\ \boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}} = \left[(\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})^{\intercal}, (\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s\right]^{\intercal};$$

(iii) error vectors in $\boldsymbol{Z}_{\boldsymbol{\theta}_{\text{wide}}}$: $\boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\text{wide}}} = \boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\text{narr}}}$, for $l' \in [L]$ and $l' \neq l$,

$$\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}} = \left[\left(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}}\right)^{\intercal}_{[1:s-1]}, (1-\alpha)(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s, \left(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}}\right)^{\intercal}_{[s+1:m_l]}, \alpha(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s \right]^{\intercal};$$

(iv) $\mathcal{T}^{\alpha}_{l,s}$ is injective for all α;

(v) $\mathcal{T}^{\alpha}_{l,s}$ is an affine embedding for all α.

Proof. (i) By the construction of $\boldsymbol{\theta}_{\text{wide}}$, it is clear that $\boldsymbol{f}^{[l']}_{\boldsymbol{\theta}_{\text{wide}}} = \boldsymbol{f}^{[l']}_{\boldsymbol{\theta}_{\text{narr}}}$ for all $l' \in [l-1]$. Then

$$\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}} = \sigma\left(\begin{bmatrix} \boldsymbol{W}^{[l]} \\ \boldsymbol{W}^{[l]}_{s,[1:m_{l-1}]} \end{bmatrix} \boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\text{narr}}} + \begin{bmatrix} \boldsymbol{b}^{[l]} \\ \boldsymbol{b}^{[l]}_{s} \end{bmatrix} \right) = \begin{bmatrix} \boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \\ (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s \end{bmatrix}. \tag{A.6}$$

Note that

$$\alpha \left[\boldsymbol{0}_{m_{l+1}\times(s-1)}, -\boldsymbol{W}^{[l+1]}_{[1:m_{l+1}],s}, \boldsymbol{0}_{m_{l+1}\times(m_l-s)}, \boldsymbol{W}^{[l+1]}_{[1:m_{l+1}],s} \right] \begin{bmatrix} \boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \\ (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s \end{bmatrix} = \boldsymbol{0}_{m_{l+1}\times 1}.$$

Thus

$$\boldsymbol{f}^{[l+1]}_{\boldsymbol{\theta}_{\text{wide}}} = \sigma\left(\left[\boldsymbol{W}^{[l+1]}, \boldsymbol{0}_{m_{l+1}\times 1} \right] \begin{bmatrix} \boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \\ (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s \end{bmatrix} + \boldsymbol{0}_{m_{l+1}\times 1} + \boldsymbol{b}^{[l+1]} + \alpha\, \boldsymbol{0}_{m_{l+1}\times 1} \right) = \boldsymbol{f}^{[l+1]}_{\boldsymbol{\theta}_{\text{narr}}}. \tag{A.7}$$

Next, by the construction of $\boldsymbol{\theta}_{\text{wide}}$, it is clear that $\boldsymbol{f}^{[l']}_{\boldsymbol{\theta}_{\text{wide}}} = \boldsymbol{f}^{[l']}_{\boldsymbol{\theta}_{\text{narr}}}$ for any $l' \in [l+1:L]$.

(ii) The results for feature gradients $\boldsymbol{g}^{[l']}_{\boldsymbol{\theta}_{\text{wide}}} = \boldsymbol{g}^{[l']}_{\boldsymbol{\theta}_{\text{narr}}}$ for $l' \in [L]$ can be calculated in a similar way except by replacing $\sigma(\cdot)$ with $\sigma^{(1)}(\cdot)$.

(iii) By the backpropagation and the above facts in (i), we have

$$\boldsymbol{z}^{[L]}_{\boldsymbol{\theta}_{\text{wide}}} = \nabla \ell(\boldsymbol{f}^{[L]}_{\boldsymbol{\theta}_{\text{wide}}}, \boldsymbol{y}) = \nabla \ell(\boldsymbol{f}^{[L]}_{\boldsymbol{\theta}_{\text{narr}}}, \boldsymbol{y}) = \boldsymbol{z}^{[L]}_{\boldsymbol{\theta}_{\text{narr}}}.$$

Recall the recurrence relation for $l' \in [l+1:L-1]$, then we recursively obtain the following equality for l' from $L-1$ down to $l+1$:

$$\boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\text{wide}}} = (\boldsymbol{W}^{[l'+1]})^{\intercal}\left(\boldsymbol{z}^{[l'+1]}_{\boldsymbol{\theta}_{\text{wide}}} \circ \boldsymbol{g}^{[l'+1]}_{\boldsymbol{\theta}_{\text{wide}}}\right) = (\boldsymbol{W}^{[l'+1]})^{\intercal}\left(\boldsymbol{z}^{[l'+1]}_{\boldsymbol{\theta}_{\text{narr}}} \circ \boldsymbol{g}^{[l'+1]}_{\boldsymbol{\theta}_{\text{narr}}}\right) = \boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\text{narr}}}. \tag{A.8}$$

Next,

$$\begin{aligned} \boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}} = \Big(& \left[\boldsymbol{W}^{[l+1]}, \boldsymbol{0}_{m_{l+1}\times 1} \right] + \alpha \left[\boldsymbol{0}_{m_{l+1}\times(s-1)}, -\boldsymbol{W}^{[l+1]}_{[1:m_{l+1}],s}, \boldsymbol{0}_{m_{l+1}\times(m_l-s)}, \right. \\ & \left. \boldsymbol{W}^{[l+1]}_{[1:m_{l+1}],s} \right] \Big)^{\intercal} \left(\boldsymbol{z}^{[l+1]}_{\boldsymbol{\theta}_{\text{wide}}} \circ \boldsymbol{g}^{[l+1]}_{\boldsymbol{\theta}_{\text{wide}}}\right) \end{aligned}$$

$$= \begin{bmatrix} \boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{(s-1)\times 1} \\ -\alpha(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s \\ \boldsymbol{0}_{(m_l-s)\times 1} \\ \alpha(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s \end{bmatrix}$$
$$= \left[(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})^\intercal_{[1:s-1]}, (1-\alpha)(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s, (\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})^\intercal_{[s+1:m_l]}, \alpha(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s\right]^\intercal. \tag{A.9}$$

Finally,

$$\boldsymbol{z}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}} = \left[(\boldsymbol{W}^{[l]})^\intercal, (\boldsymbol{W}^{[l]})^\intercal_{s,[1:m_{l-1}]}\right] \left(\left(\begin{bmatrix} \boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{(s-1)\times 1} \\ -\alpha(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s \\ \boldsymbol{0}_{(m_l-s)\times 1} \\ \alpha(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s \end{bmatrix} \right) \circ \begin{bmatrix} \boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \\ (\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_s \end{bmatrix} \right)$$
$$= (\boldsymbol{W}^{[l]})^\intercal \left(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}} \circ \boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}}\right) + \boldsymbol{0}_{m_{l-1}\times 1} = \boldsymbol{z}^{[l-1]}_{\boldsymbol{\theta}_{\text{narr}}}. \tag{A.10}$$

This with the recurrence relation once again leads to $\boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\text{wide}}} = \boldsymbol{z}^{[l']}_{\boldsymbol{\theta}_{\text{narr}}}$ for all $l' \in [l-1]$.

(iv) If for $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \text{Tuple}_{\{m_0,\cdots,m_L\}}$ and $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$, since $\mathcal{T}^\alpha_{l,s}|_k$ for $k \neq l, l+1$ is the identity map, then if there exists some $k_0 \neq l, l+1$, such that $\boldsymbol{W}^{[k_0]}_1 \neq \boldsymbol{W}^{[k_0]}_2$ or $\boldsymbol{b}^{[k_0]}_1 \neq \boldsymbol{b}^{[k_0]}_2$, then obviously $\mathcal{T}^\alpha_{l,s}(\boldsymbol{\theta}_1) \neq \mathcal{T}^\alpha_{l,s}(\boldsymbol{\theta}_2)$. If $k_0 = l$ or $k_0 = l+1$, by similar reasoning, $\mathcal{T}^\alpha_{l,s}(\boldsymbol{\theta}_1) \neq \mathcal{T}^\alpha_{l,s}(\boldsymbol{\theta}_2)$.

(v) For $\boldsymbol{\theta}_0 = (\boldsymbol{W}^{[1]}_0, \boldsymbol{b}^{[1]}_0, \cdots, \boldsymbol{W}^{[L]}_0, \boldsymbol{b}^{[L]}_0) \in \text{Tuple}_{\{m_0,\cdots,m_L\}}$, we have

$$\begin{aligned}\tilde{\mathcal{T}}^\alpha_{l,s}(\boldsymbol{\theta}_0) :=& \mathcal{T}^\alpha_{l,s}(\boldsymbol{\theta}_0) - \mathcal{T}^\alpha_{l,s}(\boldsymbol{0}) \\ =& \Big(\boldsymbol{W}^{[1]}_0, \boldsymbol{b}^{[1]}_0, \cdots, \begin{bmatrix} \boldsymbol{W}^{[l]}_0 \\ (\boldsymbol{W}^{[l]}_0)_{s,[1:m_{l-1}]} \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}^{[l]}_0 \\ (\boldsymbol{b}^{[l]}_0)_s \end{bmatrix}, \Big[(\boldsymbol{W}^{[l+1]}_0)_{[1:m_{l+1}],[1:s-1]}, \\ & (1-\alpha)(\boldsymbol{W}^{[l+1]}_0)_{[1:m_{l+1}],s}, (\boldsymbol{W}^{[l+1]}_0)_{[1:m_{l+1}],[s+1:m_l]}, \alpha(\boldsymbol{W}^{[l+1]}_0)_{[1:m_{l+1}],s}\Big], \\ & \boldsymbol{b}^{[l+1]}_0, \cdots, \boldsymbol{W}^{[L]}_0, \boldsymbol{b}^{[L]}_0\Big),\end{aligned}$$

obviously $\tilde{\mathcal{T}}^\alpha_{l,s}$ is a linear operator, thus $\mathcal{T}^\alpha_{l,s}$ is an affine operator. □

Directly from Lemma 4.1 and Lemma 4.2, we obtain that both one-step null embedding and one-step splitting embedding satisfy the property of output preserving and representation preserving, and all we need is to check the property of criticality preserving.

Proposition (Proposition 4.1 in main text)**.** For any one-step null embedding $\mathcal{T}^\alpha_{l,0}$, given any $\text{NN}(\{m_l\}_{l=0}^L)$ and its parameters $\boldsymbol{\theta}_{\text{narr}} \in \text{Tuple}_{\{m_0,\cdots,m_L\}}$ with $\text{Tuple}_{\{m_0,\cdots,m_L\}} \in \mathcal{D}_{l,0}$, we have $\boldsymbol{\theta}_{\text{wide}} := \mathcal{T}^\alpha_{l,0}(\boldsymbol{\theta}_{\text{narr}})$ satisfies the following conditions: given any data S, loss $\ell(\cdot,\cdot)$ and activation $\sigma(\cdot)$, if $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\text{narr}}) = \boldsymbol{0}$, then $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \boldsymbol{0}$.

Proof. Gradient of loss with respect to network parameters of each layer can be computed from $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{Z}$ as follows

$$\nabla_{\boldsymbol{W}^{[l']}} R_S(\boldsymbol{\theta}) = \nabla_{\boldsymbol{W}^{[l']}} \mathbb{E}_S \ell(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) = \mathbb{E}_S \left(\left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l']} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l']} \right) (\boldsymbol{f}_{\boldsymbol{\theta}}^{[l'-1]})^\intercal \right),$$
$$\nabla_{\boldsymbol{b}^{[l']}} R_S(\boldsymbol{\theta}) = \nabla_{\boldsymbol{b}^{[l]}} \mathbb{E}_S \ell(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) = \mathbb{E}_S \left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l']} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l']} \right).$$

Then, by Lemma 4.1, we have for $l' \neq l, l+1$,

$$\nabla_{\boldsymbol{W}^{[l']}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \nabla_{\boldsymbol{W}^{[l']}} R_S(\boldsymbol{\theta}_{\text{narr}}) = \boldsymbol{0},$$

and

$$\nabla_{\boldsymbol{b}^{[l']}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \nabla_{\boldsymbol{b}^{[l']}} R_S(\boldsymbol{\theta}_{\text{narr}}) = \boldsymbol{0}.$$

Also, for any $j \in [m_{l+1}]$, $k \in [m_l]$, since

$$(\boldsymbol{z}_{\boldsymbol{\theta}_{\text{wide}}}^{[l+1]})_j = (\boldsymbol{z}_{\boldsymbol{\theta}_{\text{narr}}}^{[l+1]})_j, \quad (\boldsymbol{g}_{\boldsymbol{\theta}_{\text{wide}}}^{[l+1]})_j = (\boldsymbol{g}_{\boldsymbol{\theta}_{\text{narr}}}^{[l+1]})_j, \quad (\boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[l]})_k = (\boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]})_k,$$

and $\boldsymbol{W}_{j,(m_l+1)}^{[l+1]} \equiv 0$, we obtain that

$$\nabla_{\boldsymbol{W}_{j,k}^{[l+1]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \nabla_{\boldsymbol{W}_{j,k}^{[l+1]}} R_S(\boldsymbol{\theta}_{\text{narr}}) = 0,$$
$$\nabla_{\boldsymbol{W}_{j,(m_l+1)}^{[l+1]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = 0,$$
$$\nabla_{\boldsymbol{b}_j^{[l+1]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \nabla_{\boldsymbol{b}_j^{[l+1]}} R_S(\boldsymbol{\theta}_{\text{narr}}) = 0.$$

Similarly, for any $j \in [m_l]$, $k \in [m_{l-1}]$, since

$$(\boldsymbol{z}_{\boldsymbol{\theta}_{\text{wide}}}^{[l]})_j = (\boldsymbol{z}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]})_j, \quad (\boldsymbol{g}_{\boldsymbol{\theta}_{\text{wide}}}^{[l]})_j = (\boldsymbol{g}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]})_j, \quad (\boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[l-1]})_k = (\boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l-1]})_k,$$

and $\boldsymbol{W}_{(m_l+1),k}^{[l]} \equiv 0$, we have

$$\nabla_{\boldsymbol{W}_{j,k}^{[l]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \nabla_{\boldsymbol{W}_{j,k}^{[l]}} R_S(\boldsymbol{\theta}_{\text{narr}}) = 0,$$
$$\nabla_{\boldsymbol{W}_{(m_l+1),k}^{[l]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = 0,$$
$$\nabla_{\boldsymbol{b}_j^{[l]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \nabla_{\boldsymbol{b}_j^{[l]}} R_S(\boldsymbol{\theta}_{\text{narr}}) = 0.$$

Moreover, by Lemma 4.1, the output function $\boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[L]} = \boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[L]}$ is independent of the hyper-parameter α, and $R_S(\boldsymbol{\theta}_{\text{wide}}) = R_S(\boldsymbol{\theta}_{\text{narr}})$, then since $\boldsymbol{b}_{(m_l+1)}^{[l]} = \alpha$, we have

$$\nabla_{\boldsymbol{b}_{(m_l+1)}^{[l]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \frac{\partial}{\partial \alpha} R_S(\boldsymbol{\theta}_{\text{narr}}) = 0.$$

Collecting all the above relations, we obtain that $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \boldsymbol{0}$. □

Proposition (Proposition 4.2 in main text)**.** For any one-step splitting embedding $\mathcal{T}^{\alpha}_{l,s}$, given any $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ and its parameters $\boldsymbol{\theta}_{\mathrm{narr}} \in \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$ with $\mathrm{Tuple}_{\{m_0,\cdots,m_L\}} \in \mathcal{D}_{l,s}$, we have $\boldsymbol{\theta}_{\mathrm{wide}} := \mathcal{T}^{\alpha}_{l,s}(\boldsymbol{\theta}_{\mathrm{narr}})$ satisfies the following conditions: given any data S, loss $\ell(\cdot,\cdot)$ and activation $\sigma(\cdot)$, if $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = \mathbf{0}$, then $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \mathbf{0}$.

Proof. Gradient of loss with respect to network parameters of each layer can be computed from $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{Z}$ as follows

$$\nabla_{\boldsymbol{W}^{[l']}} R_S(\boldsymbol{\theta}) = \nabla_{\boldsymbol{W}^{[l']}} \mathbb{E}_S \ell(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) = \mathbb{E}_S \left(\left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l']} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l']} \right) (\boldsymbol{f}_{\boldsymbol{\theta}}^{[l'-1]})^{\mathsf{T}} \right),$$
$$\nabla_{\boldsymbol{b}^{[l']}} R_S(\boldsymbol{\theta}) = \nabla_{\boldsymbol{b}^{[l]}} \mathbb{E}_S \ell(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) = \mathbb{E}_S \left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l']} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l']} \right).$$

Then, by Lemma 4.2, we have for $l' \neq l, l+1$,

$$\nabla_{\boldsymbol{W}^{[l']}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \nabla_{\boldsymbol{W}^{[l']}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = \mathbf{0},$$

and

$$\nabla_{\boldsymbol{b}^{[l']}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \nabla_{\boldsymbol{b}^{[l']}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = \mathbf{0}.$$

Also, for any $j \in [m_{l+1}]$, $k \in [m_l]$, since

$$(\boldsymbol{z}^{[l+1]}_{\boldsymbol{\theta}_{\mathrm{wide}}})_j = (\boldsymbol{z}^{[l+1]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_j, \quad (\boldsymbol{g}^{[l+1]}_{\boldsymbol{\theta}_{\mathrm{wide}}})_j = (\boldsymbol{g}^{[l+1]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_j,$$
$$(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}})_k = (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_k, \quad (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}})_{m_l+1} = (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_s,$$

we obtain

$$\nabla_{\boldsymbol{W}^{[l+1]}_{j,k}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \nabla_{\boldsymbol{W}^{[l+1]}_{j,k}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = 0,$$
$$\nabla_{\boldsymbol{W}^{[l+1]}_{j,(m_l+1)}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = 0,$$
$$\nabla_{\boldsymbol{b}^{[l+1]}_{j}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \nabla_{\boldsymbol{b}^{[l+1]}_{j}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = 0.$$

Similarly, for any $j \in [m_l] \backslash \{s\}$,

$$(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}})_j = (\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_j, \quad (\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}})_s = (1-\alpha)(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_s, \quad (\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}})_{(m_l+1)} = \alpha(\boldsymbol{z}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_s,$$

and for any $i \in [m_l]$,

$$(\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}})_i = (\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_i, \quad (\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\mathrm{wide}}})_{(m_l+1)} = (\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_s,$$

and for $k \in [m_{l-1}]$,

$$(\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\mathrm{wide}}})_k = (\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\mathrm{narr}}})_k,$$

hence for any $j \in [m_l] \backslash \{s\}$, $k \in [m_{l-1}]$:

$$\nabla_{\boldsymbol{W}^{[l]}_{j,k}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \nabla_{\boldsymbol{W}^{[l]}_{j,k}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = 0,$$
$$\nabla_{\boldsymbol{b}^{[l]}_{j}} R_S(\boldsymbol{\theta}_{\mathrm{wide}}) = \nabla_{\boldsymbol{b}^{[l]}_{j}} R_S(\boldsymbol{\theta}_{\mathrm{narr}}) = 0,$$

$$\nabla_{\boldsymbol{W}_{s,k}^{[l]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = (1-\alpha)\nabla_{\boldsymbol{W}_{s,k}^{[l]}} R_S(\boldsymbol{\theta}_{\text{narr}}) = 0,$$
$$\nabla_{\boldsymbol{W}_{(m_l+1),k}^{[l]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \alpha\nabla_{\boldsymbol{W}_{s,k}^{[l]}} R_S(\boldsymbol{\theta}_{\text{narr}}) = 0,$$
$$\nabla_{\boldsymbol{b}_s^{[l]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = (1-\alpha)\nabla_{\boldsymbol{b}_s^{[l]}} R_S(\boldsymbol{\theta}_{\text{narr}}) = 0,$$
$$\nabla_{\boldsymbol{b}_{(m_l+1)}^{[l]}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \alpha\nabla_{\boldsymbol{b}_s^{[l]}} R_S(\boldsymbol{\theta}_{\text{narr}}) = 0.$$

Collecting all the above relations, we obtain that $\nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}_{\text{wide}}) = \mathbf{0}$. □

Combining altogether Lemma 4.1, Lemma 4.2, Proposition 4.1 and Proposition 4.2, we finish our proof for Theorem 4.1.

Theorem (Theorem 4.2 in main text)**.** A K-step composition embedding is a critical embedding.

Proof. We shall prove it using induction.

For $K = 1$, Proposition 4.2 holds since both one-step null embedding and one step splitting embedding are critical embeddings.

Assume that Proposition 4.2 holds for $K = l - 1$, we want to show that it also holds for $K = l$.

From the induction hypothesis, we only need to show that if given two critical embeddings $\mathcal{T}_1$, $\mathcal{T}_2$, then $\mathcal{T}_2\mathcal{T}_1$ is also a critical embedding.

(i) $\mathcal{T}_2\mathcal{T}_1$ is injective:

For $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ belonging to a same tuple class but $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$, since $\mathcal{T}_1$ is injective, then $\mathcal{T}_1(\boldsymbol{\theta}_1) \neq \mathcal{T}_1(\boldsymbol{\theta}_2)$. Since $\mathcal{T}_2$ is injective, then $\mathcal{T}_2\mathcal{T}_1(\boldsymbol{\theta}_1) \neq \mathcal{T}_2\mathcal{T}_1(\boldsymbol{\theta}_2)$.

(ii) $\mathcal{T}_2\mathcal{T}_1$ is an affine embedding:

We use the fact that the composition of two affine operators is affine and hence we finish our proof.

(iii) $\mathcal{T}_2\mathcal{T}_1$ satisfies the property of output preserving:

Since $\mathcal{T}_1$ satisfies the property of output preserving, then for any $\boldsymbol{\theta}$, $\boldsymbol{f}_{\mathcal{T}_1(\boldsymbol{\theta})} = \boldsymbol{f}_{\boldsymbol{\theta}}$. Similarly for $\mathcal{T}_2$, we have $\boldsymbol{f}_{\mathcal{T}_2\mathcal{T}_1(\boldsymbol{\theta})} = \boldsymbol{f}_{\mathcal{T}_1(\boldsymbol{\theta})}$, hence $\boldsymbol{f}_{\mathcal{T}_2\mathcal{T}_1(\boldsymbol{\theta})} = \boldsymbol{f}_{\boldsymbol{\theta}}$.

(iv) $\mathcal{T}_2\mathcal{T}_1$ satisfies the property of representation preserving:

Similar reasoning in (iii).

(v) $\mathcal{T}_2\mathcal{T}_1$ satisfies the property of criticality preserving:

Since $\boldsymbol{\theta}$ is a critical point of $R_S(\boldsymbol{\theta})$, so is $\mathcal{T}_1(\boldsymbol{\theta})$, and $\mathcal{T}_2\mathcal{T}_1(\boldsymbol{\theta})$ as well, we finish the proof. □

Lemma (Lemma 5.1 in main text)**.** For any affine embedding $\mathcal{T} : \text{Tuple}_{\{m_0,\cdots,m_L\}} \to \text{Tuple}_{\{m_0',\cdots,m_L'\}}$ satisfying the output preserving property, if there exists a total index mapping $\mathcal{I} = \{\mathcal{I}_l\}_{l=0}^L$ from $\text{NN}(\{m_l'\}_{l=0}^L)$ to $\text{NN}(\{m_l\}_{l=0}^L)$ and auxiliary variables $\boldsymbol{\beta} = \{\boldsymbol{\beta}_j^{[l]} \in \mathbb{R} | \, l \in [0:L], j \in [m_l']\backslash\mathcal{I}_l^{-1}(0)\}$, such that for any given neuron belonging to $\text{NN}(\{m_l'\}_{l=0}^L)$, located in layer l with index j, the following two statements hold:

(i) If $\mathcal{I}_l(j) \neq 0$, $(\boldsymbol{f}_{\boldsymbol{\theta}_{\text{wide}}}^{[l]})_j = (\boldsymbol{f}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]})_{\mathcal{I}_l(j)}$ and $(\boldsymbol{e}_{\boldsymbol{\theta}_{\text{wide}}}^{[l]})_j = \boldsymbol{\beta}_j^{[l]}(\boldsymbol{e}_{\boldsymbol{\theta}_{\text{narr}}}^{[l]})_{\mathcal{I}_l(j)}$,

(ii) If $\mathcal{I}_l(j) = 0$, $(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_j = \text{Const}$ and $(\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_j = 0$,

then $\mathcal{T}$ is a critical embedding.

Proof. For any critical point $\boldsymbol{\theta}^{\mathfrak{c}}_{\text{narr}} \in \boldsymbol{\Theta}^{\mathfrak{c}}_{\text{narr}}$, we set $\boldsymbol{\theta}_{\text{wide}} := \mathcal{T}(\boldsymbol{\theta}^{\mathfrak{c}}_{\text{narr}})$. Then, since we have for any $l' \in [L]$

$$\nabla_{\boldsymbol{W}^{[l']}} R_S(\boldsymbol{\theta}) = \nabla_{\boldsymbol{W}^{[l']}} \mathbb{E}_S \ell(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) = \mathbb{E}_S \left(\left(\boldsymbol{z}^{[l']}_{\boldsymbol{\theta}} \circ \boldsymbol{g}^{[l']}_{\boldsymbol{\theta}} \right) (\boldsymbol{f}^{[l'-1]}_{\boldsymbol{\theta}})^{\intercal} \right),$$
$$\nabla_{\boldsymbol{b}^{[l']}} R_S(\boldsymbol{\theta}) = \nabla_{\boldsymbol{b}^{[l]}} \mathbb{E}_S \ell(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) = \mathbb{E}_S \left(\boldsymbol{z}^{[l']}_{\boldsymbol{\theta}} \circ \boldsymbol{g}^{[l']}_{\boldsymbol{\theta}} \right).$$

Hence for $\mathcal{I}_l(i), \mathcal{I}_{l-1}(j) \neq 0$

$$\begin{aligned}
\nabla_{(\boldsymbol{W}^{[l]}_{\text{wide}})_{ij}} R_S(\boldsymbol{\theta}_{\text{wide}}) &= \mathbb{E}_S (\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i (\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j \\
&= \mathbb{E}_S \boldsymbol{\beta}^{[l]}_i (\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}^{\mathfrak{c}}_{\text{narr}}})_{\mathcal{I}_l(i)} (\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}^{\mathfrak{c}}_{\text{narr}}})_{\mathcal{I}_{l-1}(j)} \\
&= \boldsymbol{\beta}^{[l]}_i \nabla_{(\boldsymbol{W}^{[l]}_{\text{narr}})_{\mathcal{I}_l(i), \mathcal{I}_{l-1}(j)}} R_S(\boldsymbol{\theta}^{\mathfrak{c}}_{\text{narr}}) = 0, \\
\nabla_{(\boldsymbol{b}^{[l]}_{\text{wide}})_i} R_S(\boldsymbol{\theta}_{\text{wide}}) &= \mathbb{E}_S (\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i \\
&= \mathbb{E}_S \boldsymbol{\beta}^{[l]}_i (\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}^{\mathfrak{c}}_{\text{narr}}})_{\mathcal{I}_l(i)} \\
&= \boldsymbol{\beta}^{[l]}_i \nabla_{(\boldsymbol{b}^{[l]}_{\text{narr}})_{\mathcal{I}_l(i)}} R_S(\boldsymbol{\theta}^{\mathfrak{c}}_{\text{narr}}) = 0.
\end{aligned}$$

By condition (ii), these gradients are obviously 0 for $\mathcal{I}_l(i) = 0$ or $\mathcal{I}_{l-1}(j) = 0$. Therefore, $\boldsymbol{\theta}_{\text{wide}}$ is also a critical point and $\mathcal{T}$ is criticality preserving.

Since $\mathcal{I}$ is a total index mapping, for any feature vector of $\text{NN}(\{m_l\}_{l=0}^{L})$, the component of which is also the output function of at least a neuron in the wide NN by condition (i). Moreover, any neuron output function of a neuron in the wide NN is either constant or output function of a neuron in the narrow NN by condition (i) and (ii). Therefore, $\mathcal{T}$ is representation preserving. Then $\mathcal{T}$ is a critical embedding. □

Theorem (Theorem 5.1 in main text)**.** General compatible embedding is a critical embedding.

Remark that we later name it as general compatible critical embedding in this work.

Proof. We need to prove four properties one by one.

1. $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$ is output preserving and representation preserving;
2. $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$ is an injective operator;
3. $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$ is an affine embedding;
4. $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$ is criticality preserving.

(i) We prove output preserving and representation preserving by doing induction on layers.

For the first layer, i.e., $l = 1$, we have for $i \in [m_1']$,

$$(\boldsymbol{f}^{[1]}_{\boldsymbol{\theta}_{\text{wide}}})_i = \sigma\left(\sum_{j\in[m_0']} (\boldsymbol{W}^{[1]}_{\text{wide}})_{ij}\boldsymbol{x}_j + (\boldsymbol{b}^{[1]}_{\text{wide}})_i\right)$$
$$= \sigma\left(\sum_{j\in[m_0]} \boldsymbol{\alpha}^{[1]}_{ij}(\boldsymbol{W}^{[1]}_{\text{narr}})_{\mathcal{I}_1(i),j}\boldsymbol{x}_j + (\boldsymbol{\alpha}^{[1]}_{\text{b}})_i + (\boldsymbol{b}^{[1]}_{\text{narr}})_{\mathcal{I}_1(i)}\right),$$

then for $i \notin \mathcal{I}_l^{-1}(0)$, for each $j \in [m_0]$, since $\boldsymbol{\alpha}^{[1]}_{ij} = \sum_{s\in\mathcal{I}_0^{-1}(j)} \boldsymbol{\alpha}_{is} = 1$, $(\boldsymbol{\alpha}^{[1]}_{\text{b}})_i = 0$, hence

$$(\boldsymbol{f}^{[1]}_{\boldsymbol{\theta}_{\text{wide}}})_i = (\boldsymbol{f}^{[1]}_{\boldsymbol{\theta}_{\text{narr}}})_{\mathcal{I}_1(i)}.$$

Otherwise, for $i \in \mathcal{I}_1^{-1}(0)$, we have $\boldsymbol{\alpha}^{[1]}_{ij} = 0$ and $(\boldsymbol{\alpha}^{[1]}_{\text{b}})_i = (\boldsymbol{b}^{[1]}_*)_i$, then

$$(\boldsymbol{f}^{[1]}_{\boldsymbol{\theta}_{\text{wide}}})_i = \sigma\left((\boldsymbol{b}^{[1]}_*)_i\right),$$

which is a constant function playing the same role as the bias term in the next layer.

Suppose for layer $l-1$, we have for $i \notin \mathcal{I}_{l-1}^{-1}(0)$

$$(\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_i = (\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\text{narr}}})_{\mathcal{I}_{l-1}(i)},$$

and for $i \in \mathcal{I}_{l-1}^{-1}(0)$,

$$(\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_i = \sigma\left((\boldsymbol{b}^{[l-1]}_*)_i\right).$$

Then we want to show that this is also the case for layer l.

We obtain that

$$(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i = \sigma\left(\sum_{j\in[m_{l-1}']} (\boldsymbol{W}^{[l]}_{\text{wide}})_{ij}(\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j + (\boldsymbol{b}^{[l]}_{\text{wide}})_i\right)$$
$$= \sigma\left(\sum_{s\in[m_{l-1}]\cup\{0\}}\ \sum_{j\in\mathcal{I}_{l-1}^{-1}(s)} \boldsymbol{\alpha}^{[l]}_{ij}(\boldsymbol{W}^{[l]}_{\text{narr}})_{\mathcal{I}_l(i),s}(\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j + (\boldsymbol{\alpha}^{[l]}_{\text{b}})_i + (\boldsymbol{b}^{[l]}_{\text{narr}})_{\mathcal{I}_l(i)}\right)$$
$$= \sigma\Bigg(\sum_{s\in[m_{l-1}]} (\boldsymbol{W}^{[l]}_{\text{narr}})_{\mathcal{I}_l(i),s}(\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\text{narr}}})_s \sum_{j\in\mathcal{I}_{l-1}^{-1}(s)} \boldsymbol{\alpha}^{[l]}_{ij} + \sum_{j\in\mathcal{I}_{l-1}^{-1}(0)} \boldsymbol{\alpha}^{[l]}_{ij}\sigma\left((\boldsymbol{b}^{[l-1]}_*)_j\right)$$
$$+ (\boldsymbol{\alpha}^{[l]}_{\text{b}})_i + (\boldsymbol{b}^{[l]}_{\text{narr}})_{\mathcal{I}_l(i)}\Bigg).$$

For $i \notin \mathcal{I}_l^{-1}(0)$, we have

$$(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i = \sigma\left(\sum_{s\in[m_{l-1}]} (\boldsymbol{W}^{[l]}_{\text{narr}})_{\mathcal{I}_l(i),s}(\boldsymbol{f}^{[l-1]}_{\boldsymbol{\theta}_{\text{narr}}})_s + (\boldsymbol{b}^{[l]}_{\text{narr}})_{\mathcal{I}_l(i)}\right) = (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_{\mathcal{I}_l(i)}.$$

Otherwise, for $i \in \mathcal{I}_l^{-1}(0)$, we have

$$(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i = \sigma\left((\boldsymbol{b}^{[l]}_*)_i\right).$$

Then, for any layer l, we have for $i \notin \mathcal{I}_l^{-1}(0)$,

$$(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i = (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_{\mathcal{I}_l(i)},$$

and for $i \in \mathcal{I}_l^{-1}(0)$,

$$(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i = \sigma\left((\boldsymbol{b}^{[l]}_*)_i\right).$$

Hence, we have proved already that $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$ is output preserving and representation preserving.

(ii) We prove that $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$ is injective.

If for $\boldsymbol{\theta}_{\text{narr},1}, \boldsymbol{\theta}_{\text{narr},2} \in \text{Tuple}_{\{m_0,\cdots,m_L\}}$ and $\boldsymbol{\theta}_{\text{narr},1} \neq \boldsymbol{\theta}_{\text{narr},2}$, then there exists some $l \in [L]$, such that $\boldsymbol{W}^{[l]}_{\text{narr},1} \neq \boldsymbol{W}^{[l]}_{\text{narr},2}$ or $\boldsymbol{b}^{[l]}_{\text{narr},1} \neq \boldsymbol{b}^{[l]}_{\text{narr},2}$.

Then, if $\mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}$ is not injective, there exists $\boldsymbol{\theta}_{\text{narr},1} \neq \boldsymbol{\theta}_{\text{narr},2}$, such that $\boldsymbol{\theta}_{\text{wide},1} = \boldsymbol{\theta}_{\text{wide},2}$, where $\boldsymbol{\theta}_{\text{wide},1} := \mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}(\boldsymbol{\theta}_{\text{narr},1})$ and $\boldsymbol{\theta}_{\text{wide},2} := \mathcal{T}^{\boldsymbol{\alpha}}_{\mathcal{I}}(\boldsymbol{\theta}_{\text{narr},2})$, and we want to show that this will never happen.

Since there exists $l \in [L]$, such that $\boldsymbol{W}^{[l]}_{\text{narr},1} \neq \boldsymbol{W}^{[l]}_{\text{narr},2}$ or $\boldsymbol{b}^{[l]}_{\text{narr},1} \neq \boldsymbol{b}^{[l]}_{\text{narr},2}$. For the case $\boldsymbol{W}^{[l]}_{\text{narr},1} \neq \boldsymbol{W}^{[l]}_{\text{narr},2}$, we obtain that there exists $i \in [m_{l+1}]$, $j \in [m_l]$, such that $\left(\boldsymbol{W}^{[l]}_{\text{narr},1}\right)_{i,j} \neq \left(\boldsymbol{W}^{[l]}_{\text{narr},2}\right)_{i,j}$. We observe that $\boldsymbol{W}^{[l]}_{\text{wide},1} = \boldsymbol{\alpha}^{[l]} \circ \boldsymbol{W}^{[l]}_{\text{inter},1}$, and $\boldsymbol{W}^{[l]}_{\text{wide},2} = \boldsymbol{\alpha}^{[l]} \circ \boldsymbol{W}^{[l]}_{\text{inter},2}$. Then, since $\mathcal{I}$ is a total index mapping, then for $i, j \neq 0$, $\mathcal{I}_l^{-1}(i), \mathcal{I}_{l-1}^{-1}(j) \neq \emptyset$, hence for any $k \in \mathcal{I}_l^{-1}(i)$ and $l \in \mathcal{I}_{l-1}^{-1}(j)$,

$$\left(\boldsymbol{W}^{[l]}_{\text{wide},1}\right)_{k,l} = \boldsymbol{\alpha}^{[l]}_{k,l}\left(\boldsymbol{W}^{[l]}_{\text{narr},1}\right)_{i,j},$$

and

$$\left(\boldsymbol{W}^{[l]}_{\text{wide},2}\right)_{k,l} = \boldsymbol{\alpha}^{[l]}_{k,l}\left(\boldsymbol{W}^{[l]}_{\text{narr},2}\right)_{i,j}.$$

If $\boldsymbol{W}^{[l]}_{\text{wide},1} = \boldsymbol{W}^{[l]}_{\text{wide},2}$, then

$$\sum_{s\in\mathcal{I}_{l-1}^{-1}(j)} \left(\boldsymbol{W}^{[l]}_{\text{wide},1}\right)_{k,s} = \sum_{s\in\mathcal{I}_{l-1}^{-1}(j)} \left(\boldsymbol{W}^{[l]}_{\text{wide},2}\right)_{k,s},$$

hence

$$\sum_{s\in\mathcal{I}_{l-1}^{-1}(j)} \boldsymbol{\alpha}_{k,s}^{[l]} \left(\boldsymbol{W}_{\text{narr},1}^{[l]}\right)_{i,j} = \sum_{s\in\mathcal{I}_{l-1}^{-1}(j)} \boldsymbol{\alpha}_{k,s}^{[l]} \left(\boldsymbol{W}_{\text{narr},2}^{[l]}\right)_{i,j},$$

and since $\sum_{s\in\mathcal{I}_{l-1}^{-1}(j)} \boldsymbol{\alpha}_{k,s}^{[l]} = 1$, we obtain that

$$\left(\boldsymbol{W}_{\text{narr},1}^{[l]}\right)_{i,j} = \left(\boldsymbol{W}_{\text{narr},2}^{[l]}\right)_{i,j},$$

which contradicts $\boldsymbol{\theta}_{\text{narr},1} \neq \boldsymbol{\theta}_{\text{narr},2}$.

For the case where $\boldsymbol{b}_{\text{narr},1}^{[l]} \neq \boldsymbol{b}_{\text{narr},2}^{[l]}$, since $(\boldsymbol{\alpha}_{\text{b}}^{[l]})_i = 0$ for any $l \in [L]$ with $i \notin \mathcal{I}_l^{-1}(0)$. Then for any $k \in \mathcal{I}_l^{-1}(j)$, $j \neq 0$,

$$\left(\boldsymbol{b}_{\text{wide},1}^{[l]}\right)_k = \boldsymbol{\alpha}_{\text{b}}^{[l]} + \left(\boldsymbol{b}_{\text{narr},1}^{[l]}\right)_j = \left(\boldsymbol{b}_{\text{narr},1}^{[l]}\right)_j,$$

and

$$\left(\boldsymbol{b}_{\text{wide},2}^{[l]}\right)_k = \boldsymbol{\alpha}_{\text{b}}^{[l]} + \left(\boldsymbol{b}_{\text{narr},2}^{[l]}\right)_j = \left(\boldsymbol{b}_{\text{narr},2}^{[l]}\right)_j,$$

hence $\boldsymbol{b}_{\text{wide},1}^{[l]} \neq \boldsymbol{b}_{\text{wide},2}^{[l]}$, and we finish the injection proof.

(iii) We prove that $\mathcal{T}_{\mathcal{I}}^{\boldsymbol{\alpha}}$ is affine.

It is obvious that $\mathcal{T}_{\mathcal{I}}^{\boldsymbol{\alpha}}$ is affine since for any $\boldsymbol{\theta} \in \text{Tuple}_{\{m_0,\cdots,m_L\}}$, $\tilde{\mathcal{T}}_{\mathcal{I}}^{\boldsymbol{\alpha}}(\boldsymbol{\theta}) := \mathcal{T}_{\mathcal{I}}^{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \mathcal{T}_{\mathcal{I}}^{\boldsymbol{\alpha}}(\boldsymbol{0})$ puts the weights and biases of null neuron as zero, and multiplies the weights and biases of effective by some constant, thus $\tilde{\mathcal{T}}_{\mathcal{I}}^{\boldsymbol{\alpha}}$ is a linear operator.

(iv) We prove that $\mathcal{T}_{\mathcal{I}}^{\boldsymbol{\alpha}}$ is criticality preserving.

We only need to check whether or not the conditions in Lemma 5.1 are satisfied. Moreover, we want to show that the collection of auxiliary variables

$$\boldsymbol{\beta} := \left\{\boldsymbol{\beta}_j^{[l]} \in \mathbb{R} |\ l \in [0:L],\ j \in [m_l'] \backslash \mathcal{I}_l^{-1}(0)\right\}$$

in Condition 1 are exactly the auxiliary variables

$$\boldsymbol{\beta} = \left\{\boldsymbol{\beta}_j^{[l]} \in \mathbb{R} |\ l \in [0:L], j \in [m_l'] \backslash \mathcal{I}_l^{-1}(0)\right\}$$

in Lemma 5.1. We show them using induction.

For layer L, by definition

$$\boldsymbol{e}_{\boldsymbol{\theta}_{\text{wide}}}^{[L]} = \boldsymbol{z}_{\boldsymbol{\theta}_{\text{wide}}}^{[L]} \circ \boldsymbol{g}_{\boldsymbol{\theta}_{\text{wide}}}^{[L]},$$

since $\mathcal{T}_{\mathcal{I}}^{\boldsymbol{\alpha}}$ is output preserving for any activation, then

$$\boldsymbol{e}_{\boldsymbol{\theta}_{\text{wide}}}^{[L]} = \boldsymbol{z}_{\boldsymbol{\theta}_{\text{wide}}}^{[L]} \circ \boldsymbol{g}_{\boldsymbol{\theta}_{\text{wide}}}^{[L]} = \boldsymbol{z}_{\boldsymbol{\theta}_{\text{narr}}}^{[L]} \circ \boldsymbol{g}_{\boldsymbol{\theta}_{\text{narr}}}^{[L]} = \boldsymbol{e}_{\boldsymbol{\theta}_{\text{narr}}}^{[L]},$$

hence for $i \in [m_L]$, $(\boldsymbol{e}_{\boldsymbol{\theta}_{\text{wide}}}^{[L]})_i = \boldsymbol{\beta}_i^{[L]} (\boldsymbol{e}_{\boldsymbol{\theta}_{\text{narr}}}^{[L]})_i$, since $\boldsymbol{\beta}_i^{[L]} = 1$ for $i \in [m_L]$.

For any $l \in [L]$, suppose for layer l, we have for $i \notin \mathcal{I}_l^{-1}(0)$,

$$\left(\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}}\right)_i = \boldsymbol{\beta}_i^{[l]} \left(\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}}\right)_{\mathcal{I}_l(i)},$$

and for $i \in \mathcal{I}_l^{-1}(0)$,

$$\left(\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}}\right)_i = 0.$$

Then for layer $l-1$, we have

$$\begin{aligned}
(\boldsymbol{e}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j &= (\boldsymbol{g}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j \sum_{i\in[m_l']} (\boldsymbol{W}^{[l]}_{\text{wide}})_{ij} (\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i \\
&= (\boldsymbol{g}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j \sum_{k\in[m_l]\cup\{0\}} \sum_{i\in\mathcal{I}_l^{-1}(k)} \boldsymbol{\alpha}_{ij}^{[l]} (\boldsymbol{W}^{[l]}_{\text{inter}})_{ij} (\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i \\
&= (\boldsymbol{g}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j \sum_{k\in[m_l]} (\boldsymbol{W}^{[l]}_{\text{narr}})_{k,\mathcal{I}_{l-1}(j)} (\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_k \sum_{i\in\mathcal{I}_l^{-1}(k)} \boldsymbol{\alpha}_{ij}^{[l]} \boldsymbol{\beta}_i^{[l]},
\end{aligned}$$

from Condition 1, we observe that $\boldsymbol{\beta}_j^{[l-1]} := \sum_{i\in\mathcal{I}_l^{-1}(k)} \boldsymbol{\alpha}_{ij}^{[l]} \boldsymbol{\beta}_i^{[l]}$, then

$$(\boldsymbol{e}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j = (\boldsymbol{g}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j \boldsymbol{\beta}_j^{[l-1]} \sum_{k\in[m_l]} (\boldsymbol{W}^{[l]}_{\text{narr}})_{k,\mathcal{I}_{l-1}(j)} (\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_k.$$

Since the choice of activation $\sigma(\cdot)$ is arbitrary, we may follow the same argument used for $\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}}$ and $\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}}$ in item (i), where for $i \notin \mathcal{I}_l^{-1}(0)$,

$$(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i = (\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_{\mathcal{I}_l(i)},$$

and for $i \in \mathcal{I}_l^{-1}(0)$,

$$(\boldsymbol{f}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i = \sigma\left((\boldsymbol{b}_*^{[l]})_i\right).$$

Similarly, for $i \notin \mathcal{I}_l^{-1}(0)$,

$$(\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i = (\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_{\mathcal{I}_l(i)},$$

and for $i \in \mathcal{I}_l^{-1}(0)$,

$$(\boldsymbol{g}^{[l]}_{\boldsymbol{\theta}_{\text{wide}}})_i = \sigma^{(1)}\left((\boldsymbol{b}_*^{[l]})_i\right).$$

Hence for $j \notin \mathcal{I}_{l-1}^{-1}(0)$,

$$\begin{aligned}
(\boldsymbol{e}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j &= (\boldsymbol{g}^{[l-1]}_{\boldsymbol{\theta}_{\text{wide}}})_j \boldsymbol{\beta}_j^{[l-1]} \sum_{k\in[m_l]} (\boldsymbol{W}^{[l]}_{\text{narr}})_{k,\mathcal{I}_{l-1}(j)} (\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_k \\
&= \boldsymbol{\beta}_j^{[l-1]} (\boldsymbol{g}^{[l-1]}_{\boldsymbol{\theta}_{\text{narr}}})_{\mathcal{I}_{l-1}(j)} \sum_{k\in[m_l]} (\boldsymbol{W}^{[l]}_{\text{narr}})_{k,\mathcal{I}_{l-1}(j)} (\boldsymbol{e}^{[l]}_{\boldsymbol{\theta}_{\text{narr}}})_k \\
&= \boldsymbol{\beta}_j^{[l-1]} (\boldsymbol{e}^{[l-1]}_{\boldsymbol{\theta}_{\text{narr}}})_j.
\end{aligned}$$

Otherwise, for $j \in \mathcal{I}_{l-1}^{-1}(0)$, we set $\boldsymbol{\beta}_j^{[l-1]} = 0$, then

$$(\boldsymbol{e}_{\boldsymbol{\theta}_{\text{wide}}}^{[l-1]})_j = \boldsymbol{\beta}_j^{[l-1]} \sigma^{(1)}\left((\boldsymbol{b}_*^{[l-1]})_j\right)(\boldsymbol{z}_{\boldsymbol{\theta}_{\text{narr}}}^{[l-1]})_{\mathcal{I}_{l-1}(j)} = 0.$$

From the above proof, we find out that the variables

$$\boldsymbol{\beta} := \left\{\boldsymbol{\beta}_j^{[l]} \in \mathbb{R} |\ l \in [0:L],\ j \in [m_l'] \backslash \mathcal{I}_l^{-1}(0)\right\}$$

in Condition 1 are exactly the variables in Lemma 5.1, hence we finish our proof. □

Theorem (Theorem 6.2 in main text)**.** Given an NN$(\{m_l\}_{l=0}^L)$ and any of its parameters $\boldsymbol{\theta} \in \mathbb{R}^M$, for any critical embedding $\mathcal{T} : \mathbb{R}^M \to \mathbb{R}^{M'}$ to any wider NN$(\{m_l'\}_{l=0}^L)$, the number of positive, zero, negative eigenvalues of $\boldsymbol{H}_S(\mathcal{T}(\boldsymbol{\theta}))$ is no less than the counterparts of $\boldsymbol{H}_S(\boldsymbol{\theta})$.

Proof. Because $\mathcal{T}$ is a critical embedding, therefore, it is an affine injective operator associated with $\boldsymbol{A} \in \mathbb{R}^{M' \times M}, \boldsymbol{c} \in \mathbb{R}^{M'}$, such that $\mathcal{T}(\boldsymbol{\theta}) = \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{c}$. By the output preserving property of $\mathcal{T}$, we have

$$R_S(\boldsymbol{\theta}) \equiv R_S(\boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{c}).$$

Hence,

$$\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}) \equiv \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{c}).$$

Then

$$\boldsymbol{A}^\intercal \boldsymbol{H}_S(\boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{c})\boldsymbol{A} \equiv \boldsymbol{H}_S(\boldsymbol{\theta}).$$

Given any $\boldsymbol{\theta}_0$, if $\boldsymbol{H}_S(\boldsymbol{\theta}_0)$ has k negative eigenvalues $\{\lambda_j^{\text{neg}}\}_{j=1}^k$ with associated orthonormal eigenvectors $\{\boldsymbol{e}_j^{\text{neg}}\}_{j=1}^k$, then $\{\boldsymbol{A}\boldsymbol{e}_j^{\text{neg}}\}_{j=1}^k$ satisfies, for any $\boldsymbol{e}_j^{\text{neg}}$,

$$(\boldsymbol{A}\boldsymbol{e}_j^{\text{neg}})^\intercal \boldsymbol{H}_S(\boldsymbol{A}\boldsymbol{\theta}_0 + \boldsymbol{c})\boldsymbol{A}\boldsymbol{e}_j^{\text{neg}} = \left(\boldsymbol{e}_j^{\text{neg}}\right)^\intercal \boldsymbol{H}_S(\boldsymbol{\theta})\boldsymbol{e}_j^{\text{neg}} = \lambda_j^{\text{neg}} < 0. \tag{A.11}$$

By full rankness of $\boldsymbol{A}$, we have

$$\dim\left(\text{span}\left(\left\{\boldsymbol{A}\boldsymbol{e}_j^{\text{neg}}\right\}_{j=1}^k\right)\right) = k.$$

Thus, $\boldsymbol{H}_S(\boldsymbol{A}\boldsymbol{\theta}_0 + \boldsymbol{c})$ has at least k negative eigenvalues. Similarly, we can prove this result for the number of zero and positive eigenvalues.

Hence, in particular, for any critical embedding $\mathcal{T}$, the number of negative eigenvalues of $\boldsymbol{H}_S(\boldsymbol{\theta})$ is no more than the counterpart of $\boldsymbol{H}_S(\mathcal{T}(\boldsymbol{\theta}))$. □

We would like to introduce some additional notations in order to state Lemma 6.1 and Lemma 6.2. In order to calculate the Hessian $\boldsymbol{H}_S(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta})$, we need to compute $\boldsymbol{v}_S(\boldsymbol{\theta})$:

$$\boldsymbol{v}_S(\boldsymbol{\theta}) := \mathbb{E}_S \nabla \ell(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}), \boldsymbol{f}^*(\boldsymbol{x}))^\intercal \nabla_{\boldsymbol{\theta}} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{i=1}^{m_L} \mathbb{E}_S \partial_i \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \nabla_{\boldsymbol{\theta}} (\boldsymbol{f}_{\boldsymbol{\theta}})_i,$$

where $\partial_i \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*)$ is the i-th element of $\nabla \ell(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}), \boldsymbol{f}^*(\boldsymbol{x}))$, and $(\boldsymbol{f}_{\boldsymbol{\theta}})_i$ is the i-th element of vector $\boldsymbol{f}_{\boldsymbol{\theta}}$, then for the Hessian $\boldsymbol{H}_S(\boldsymbol{\theta})$, we have

$$\begin{aligned}
\boldsymbol{H}_S(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} R_S(\boldsymbol{\theta}) \\
&= \sum_{i=1}^{m_L} \mathbb{E}_S \nabla_{\boldsymbol{\theta}} \left(\partial_i \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \right) \nabla_{\boldsymbol{\theta}} (\boldsymbol{f}_{\boldsymbol{\theta}})_i + \sum_{i=1}^{m_L} \mathbb{E}_S \partial_i \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \left((\boldsymbol{f}_{\boldsymbol{\theta}})_i \right) \\
&= \sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_{ij} \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \nabla_{\boldsymbol{\theta}} (\boldsymbol{f}_{\boldsymbol{\theta}})_i \left(\nabla_{\boldsymbol{\theta}} (\boldsymbol{f}_{\boldsymbol{\theta}})_j \right)^{\mathsf{T}} + \sum_{i=1}^{m_L} \mathbb{E}_S \partial_i \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \left((\boldsymbol{f}_{\boldsymbol{\theta}})_i \right),
\end{aligned}$$

where $\partial_{ij} \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*)$ is the (i, j)-th element of $\nabla \nabla \ell(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}), \boldsymbol{f}^*(\boldsymbol{x}))$.

Hence $\boldsymbol{H}_S^{(1)}(\boldsymbol{\theta})$ and $\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta})$ respectively becomes

$$\boldsymbol{H}_S^{(1)}(\boldsymbol{\theta}) := \sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_{ij} \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \nabla_{\boldsymbol{\theta}} (\boldsymbol{f}_{\boldsymbol{\theta}})_i \left(\nabla_{\boldsymbol{\theta}} (\boldsymbol{f}_{\boldsymbol{\theta}})_j \right)^{\mathsf{T}}, \tag{A.12}$$

and

$$\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}) := \sum_{i=1}^{m_L} \mathbb{E}_S \partial_i \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \left((\boldsymbol{f}_{\boldsymbol{\theta}})_i \right). \tag{A.13}$$

Obviously, we observe that

$$\boldsymbol{H}_S(\boldsymbol{\theta}) = \boldsymbol{H}_S^{(1)}(\boldsymbol{\theta}) + \boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}). \tag{A.14}$$

We observe that for $i \in [m_L]$, $j \in [m_{L-1}]$,

$$(\boldsymbol{f}_{\boldsymbol{\theta}})_i = \sum_{j=1}^{m_{L-1}} (\boldsymbol{W}^{[L]})_{i,j} \left(\boldsymbol{f}_{\boldsymbol{\theta}}^{[L-1]} \right)_j + (\boldsymbol{b}^{[L]})_i,$$

hence we obtain that, for any $i \in [m_L]$,

$$\begin{aligned}
\frac{\partial (\boldsymbol{f}_{\boldsymbol{\theta}})_i}{\partial \boldsymbol{W}^{[L]}} &= \begin{bmatrix} \mathbf{0}_{(i-1) \times m_{L-1}} \\ \left(\boldsymbol{f}_{\boldsymbol{\theta}}^{[L-1]} \right)_j \\ \mathbf{0}_{(m_L - i) \times m_{L-1}} \end{bmatrix}, \quad j \in [m_{L-1}], \\
\frac{\partial (\boldsymbol{f}_{\boldsymbol{\theta}})_i}{\partial \boldsymbol{b}^{[L]}} &= \begin{bmatrix} \mathbf{0}_{(i-1) \times 1} \\ 1 \\ \mathbf{0}_{(m_L - i) \times 1} \end{bmatrix}.
\end{aligned} \tag{A.15}$$

Finally, given an $\mathrm{NN}(\{m_l\}_{l=0}^L)$ and its parameter

$$\boldsymbol{\theta} = (\boldsymbol{W}^{[1]}, \boldsymbol{b}^{[1]}, \cdots, \boldsymbol{W}^{[L]}, \boldsymbol{b}^{[L]}) \in \mathrm{Tuple}_{\{m_0, \cdots, m_L\}},$$

we remind the readers once again that the collection of parameters $\boldsymbol{\theta}$ is a $2L$-tuple. We denote that the upper bracket $[L-1]$ by limiting ourselves to the first $2L-2$ element of the tuple, i.e.,

$$\boldsymbol{\theta}^{[L-1]} := (\boldsymbol{W}^{[1]}, \boldsymbol{b}^{[1]}, \cdots, \boldsymbol{W}^{[L-2]}, \boldsymbol{b}^{[L-2]}, \boldsymbol{W}^{[L-1]}, \boldsymbol{b}^{[L-1]}) \in \mathrm{Tuple}_{\{m_0, \cdots, m_{L-2}, m_{L-1}\}}, \tag{A.16}$$

and similarly, we identify $\boldsymbol{\theta}^{[L-1]}$ with its vectorization $\mathrm{vec}(\boldsymbol{\theta}^{[L-1]}) \in \mathbb{R}^{M^{[L-1]}}$ with $M^{[L-1]} = \sum_{l=0}^{L-2}(m_l+1)m_{l+1}$. Then, we denote hereafter the expressions

$$\boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}) := \sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_i \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \boldsymbol{W}_{i,j}^{[L]} \nabla_{\boldsymbol{\theta}^{[L-1]}} \nabla_{\boldsymbol{\theta}^{[L-1]}} \left(\boldsymbol{f}_{\boldsymbol{\theta}}^{[L-1]}\right)_j,$$

$$\boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}) := \sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_{ij} \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*) \nabla_{\boldsymbol{\theta}^{[L-1]}} (\boldsymbol{f}_{\boldsymbol{\theta}})_i \left(\nabla_{\boldsymbol{\theta}^{[L-1]}} (\boldsymbol{f}_{\boldsymbol{\theta}})_j\right)^\intercal,$$

and $\boldsymbol{H}_S^{[L-1]}(\boldsymbol{\theta}) := \boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}) + \boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta})$.

Lemma (Lemma 6.1 in main text)**.** Given any data S, loss $\ell(\cdot,\cdot)$ and activation $\sigma(\cdot)$, for any NN, if a critical point $\boldsymbol{\theta}^{\mathrm{c}} \in \boldsymbol{\Theta}^{\mathrm{c}}$ satisfies:

(i) $\boldsymbol{H}_S(\boldsymbol{\theta}^{\mathrm{c}}) \succeq 0$;

(ii) $\boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}^{\mathrm{c}}) \neq \boldsymbol{0}$,

then there exists a general compatible critical embedding $\mathcal{T}$, such that $\mathcal{T}(\boldsymbol{\theta}^{\mathrm{c}})$ is a strict-saddle point.

Proof. We consider the three-fold global splitting embedding $\mathcal{T}_{\mathrm{global}}$ defined in Example 5.1. Obviously, $\mathcal{T}_{\mathrm{global}}$ is a general compatible critical embedding. After choosing a critical point $\boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{narr}} \in \boldsymbol{\Theta}^{\mathrm{c}}$, and we have that $\boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{wide}} := \mathcal{T}_{\mathrm{global}}(\boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{narr}})$.

$\boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{narr}}$ and $\boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{wide}}$ are tuples, and we misuse these notations and identify them with their vectorizations, i.e., $\boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{narr}} \in \mathbb{R}^M$ and $\boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{wide}} \in \mathbb{R}^{M'}$ for some M and M'. More specifically, we set

$$\begin{aligned}\boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{narr}} = \Big(&\mathrm{vec}(\boldsymbol{W}^{[1]})^\intercal, \cdots, \mathrm{vec}(\boldsymbol{W}^{[L-2]})^\intercal, \mathrm{vec}(\boldsymbol{W}^{[L-1]})^\intercal,\\ &\boldsymbol{b}^{[1]\intercal}, \cdots, \boldsymbol{b}^{[L-1]\intercal}, \mathrm{vec}(\boldsymbol{W}^{[L]})^\intercal, \boldsymbol{b}^{[L]\intercal}\Big)^\intercal,\\ \boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{wide}} = \Big(&\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{0}_{1\times(M'-3M+2m_L)}, \mathrm{vec}(\boldsymbol{W}^{[L]})^\intercal, \mathrm{vec}(\boldsymbol{W}^{[L]})^\intercal, \mathrm{vec}(-\boldsymbol{W}^{[L]})^\intercal, \boldsymbol{b}^{[L]\intercal}\Big)^\intercal,\end{aligned}$$

with

$$\begin{aligned}\boldsymbol{\theta}_1 := \boldsymbol{\theta}_2 := \boldsymbol{\theta}_3 :=& \boldsymbol{\theta}^{[L-1]}\\ =& \Big(\mathrm{vec}(\boldsymbol{W}^{[1]})^\intercal, \cdots, \mathrm{vec}(\boldsymbol{W}^{[L-2]})^\intercal, \mathrm{vec}(\boldsymbol{W}^{[L-1]})^\intercal, \boldsymbol{b}^{[1]\intercal}, \cdots, \boldsymbol{b}^{[L-1]\intercal}\Big),\end{aligned}$$

then we observe that

$$\boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{narr}} = \Big(\boldsymbol{\theta}_1, \mathrm{vec}(\boldsymbol{W}^{[L]})^\intercal, \boldsymbol{b}^{[L]\intercal}\Big)^\intercal,$$

and we are able to do some computations:

$$\begin{aligned}\boldsymbol{H}_S^{[L-1]}(\boldsymbol{\theta}^{\mathrm{c}}_{\mathrm{wide}}) =& \sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_{ij} \ell(\boldsymbol{f}_{\boldsymbol{\theta}_{\mathrm{wide}}}, \boldsymbol{f}^*) \nabla_{\boldsymbol{\theta}^{[L-1]}_{\mathrm{wide}}} (\boldsymbol{f}_{\boldsymbol{\theta}_{\mathrm{wide}}})_i \left(\nabla_{\boldsymbol{\theta}^{[L-1]}_{\mathrm{wide}}} (\boldsymbol{f}_{\boldsymbol{\theta}_{\mathrm{wide}}})_j\right)^\intercal\\ &+ \sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_i \ell(\boldsymbol{f}_{\boldsymbol{\theta}_{\mathrm{wide}}}, \boldsymbol{f}^*) \left(\boldsymbol{W}^{[L]}_{\mathrm{wide}}\right)_{i,j} \nabla_{\boldsymbol{\theta}^{[L-1]}_{\mathrm{wide}}} \nabla_{\boldsymbol{\theta}^{[L-1]}_{\mathrm{wide}}} \left(\boldsymbol{f}^{[L-1]}_{\boldsymbol{\theta}_{\mathrm{wide}}}\right)_j, \qquad (A.17)\end{aligned}$$

Then for the first part

$$\sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_{ij} \ell(f_{\boldsymbol{\theta}_{\text{wide}}}, f^*) \nabla_{\boldsymbol{\theta}_{\text{wide}}^{[L-1]}} (f_{\boldsymbol{\theta}_{\text{wide}}})_i \left(\nabla_{\boldsymbol{\theta}_{\text{wide}}^{[L-1]}} (f_{\boldsymbol{\theta}_{\text{wide}}})_j \right)^\intercal,$$

we have

$$\sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_{ij} \ell(f_{\boldsymbol{\theta}_{\text{wide}}}, f^*) \nabla_{\boldsymbol{\theta}_{\text{wide}}^{[L-1]}} (f_{\boldsymbol{\theta}_{\text{wide}}})_i \left(\nabla_{\boldsymbol{\theta}_{\text{wide}}^{[L-1]}} (f_{\boldsymbol{\theta}_{\text{wide}}})_j \right)^\intercal = \begin{pmatrix} \boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} & \boldsymbol{A}_{1,3} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} & \boldsymbol{A}_{2,3} \\ \boldsymbol{A}_{3,1} & \boldsymbol{A}_{3,2} & \boldsymbol{A}_{3,3} \end{pmatrix},$$

with

$$\boldsymbol{A}_{p,q} := \mathbb{E}_S \sum_{i,j=1}^{m_L} \partial_{ij} \ell(f_{\boldsymbol{\theta}_{\text{wide}}}, f^*) \nabla_{\boldsymbol{\theta}_p} (f_{\boldsymbol{\theta}_{\text{wide}}})_i \left(\nabla_{\boldsymbol{\theta}_q} (f_{\boldsymbol{\theta}_{\text{wide}}})_j \right)^\intercal,$$

for $p \in [3]$ and $q \in [3]$.

For the second part

$$\sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_i \ell(f_{\boldsymbol{\theta}_{\text{wide}}}, f^*) \boldsymbol{W}_{i,j}^{[L]} \nabla_{\boldsymbol{\theta}_{\text{wide}}^{[L-1]}} \nabla_{\boldsymbol{\theta}_{\text{wide}}^{[L-1]}} \left(f_{\boldsymbol{\theta}_{\text{wide}}}^{[L-1]} \right)_j,$$

we have

$$\sum_{i,j=1}^{m_L} \mathbb{E}_S \partial_i \ell(f_{\boldsymbol{\theta}_{\text{wide}}}, f^*) \boldsymbol{W}_{i,j}^{[L]} \nabla_{\boldsymbol{\theta}_{\text{wide}}^{[L-1]}} \nabla_{\boldsymbol{\theta}_{\text{wide}}^{[L-1]}} \left(f_{\boldsymbol{\theta}_{\text{wide}}}^{[L-1]} \right)_j = \begin{pmatrix} \boldsymbol{B}_{1,1} & \boldsymbol{B}_{1,2} & \boldsymbol{B}_{1,3} \\ \boldsymbol{B}_{2,1} & \boldsymbol{B}_{2,2} & \boldsymbol{B}_{2,3} \\ \boldsymbol{B}_{3,1} & \boldsymbol{B}_{3,2} & \boldsymbol{B}_{3,3} \end{pmatrix},$$

with

$$\boldsymbol{B}_{p,q} := \mathbb{E}_S \sum_{i,j=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\text{wide}}}, f^*) \boldsymbol{W}_{i,j}^{[L]} \nabla_{\boldsymbol{\theta}_p} \nabla_{\boldsymbol{\theta}_q} \left(f_{\boldsymbol{\theta}_{\text{wide}}}^{[L-1]} \right)_j,$$

for $p \in [3]$ and $q \in [3]$.

Moreover,

$$\nabla_{\boldsymbol{\theta}_1} \left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x}) \right)_i = \sum_{j=1}^{m_{L-1}} \left(\boldsymbol{W}_{\text{wide}}^{[L]} \right)_{i,j} \nabla_{\boldsymbol{\theta}_1} \left(f_{\boldsymbol{\theta}_{\text{wide}}}^{[L-1]} \right)_j, \quad i \in [m_L], \tag{A.18a}$$

$$\nabla_{\boldsymbol{\theta}_2} \left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x}) \right)_i = \sum_{j=m_{L-1}+1}^{2m_{L-1}} \left(\boldsymbol{W}_{\text{wide}}^{[L]} \right)_{i,j} \nabla_{\boldsymbol{\theta}_2} \left(f_{\boldsymbol{\theta}_{\text{wide}}}^{[L-1]} \right)_j, \quad i \in [m_L], \tag{A.18b}$$

$$\nabla_{\boldsymbol{\theta}_3} \left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x}) \right)_i = \sum_{j=2m_{L-1}+1}^{3m_{L-1}} \left(\boldsymbol{W}_{\text{wide}}^{[L]} \right)_{i,j} \nabla_{\boldsymbol{\theta}_3} \left(f_{\boldsymbol{\theta}_{\text{wide}}}^{[L-1]} \right)_j, \quad i \in [m_L], \tag{A.18c}$$

hence by construction of $\mathcal{T}_{\text{global}}$, we obtain that

$$\nabla_{\boldsymbol{\theta}_1} \left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x}) \right)_i = \nabla_{\boldsymbol{\theta}_2} \left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x}) \right)_i = -\nabla_{\boldsymbol{\theta}_3} \left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x}) \right)_i.$$

And for the Hessian, we have

$$\nabla_{\boldsymbol{\theta}_1}\nabla_{\boldsymbol{\theta}_1}\left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x})\right)_i = \sum_{j=1}^{m_{L-1}}\left(\boldsymbol{W}_{\text{wide}}^{[L]}\right)_{i,j}\nabla_{\boldsymbol{\theta}_1}\nabla_{\boldsymbol{\theta}_1}\left(f_{\boldsymbol{\theta}_{\text{wide}}}^{[L-1]}\right)_j, \quad i\in[m_L], \tag{A.19a}$$

$$\nabla_{\boldsymbol{\theta}_2}\nabla_{\boldsymbol{\theta}_2}\left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x})\right)_i = \sum_{j=m_{L-1}+1}^{2m_{L-1}}\left(\boldsymbol{W}_{\text{wide}}^{[L]}\right)_{i,j}\nabla_{\boldsymbol{\theta}_2}\nabla_{\boldsymbol{\theta}_2}\left(f_{\boldsymbol{\theta}_{\text{wide}}}^{[L-1]}\right)_j, \quad i\in[m_L], \tag{A.19b}$$

$$\nabla_{\boldsymbol{\theta}_3}\nabla_{\boldsymbol{\theta}_3}\left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x})\right)_i = \sum_{j=2m_{L-1}+1}^{3m_{L-1}}\left(\boldsymbol{W}_{\text{wide}}^{[L]}\right)_{i,j}\nabla_{\boldsymbol{\theta}_3}\nabla_{\boldsymbol{\theta}_3}\left(f_{\boldsymbol{\theta}_{\text{wide}}}^{[L-1]}\right)_j, \quad i\in[m_L], \tag{A.19c}$$

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_1}\nabla_{\boldsymbol{\theta}_2}\left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x})\right)_i &= \nabla_{\boldsymbol{\theta}_1}\nabla_{\boldsymbol{\theta}_3}\left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x})\right)_i\\ &= \nabla_{\boldsymbol{\theta}_2}\nabla_{\boldsymbol{\theta}_3}\left(f_{\boldsymbol{\theta}_{\text{wide}}}(\boldsymbol{x})\right)_i = \boldsymbol{0}, \quad i\in[m_L],\end{aligned} \tag{A.19d}$$

Thus for $\boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) \neq \boldsymbol{0}$, then there exists a nonzero eigenvalue $\lambda \neq 0$ associated with its unit eigenvector $\boldsymbol{v}$.

For the cases where $\lambda > 0$, then $\boldsymbol{v}^\intercal \boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}})\boldsymbol{v} = \lambda$. We observe that the matrix $\widetilde{\boldsymbol{H}}(\boldsymbol{\theta}_{\text{wide}}^{\text{c}})$ below is a principle submatrix of the Hessian at $\boldsymbol{\theta}_{\text{wide}}^{\text{c}}$, i.e., by choosing the columns and rows corresponding to $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$, we obtain that

$$\begin{aligned}\widetilde{\boldsymbol{H}}(\boldsymbol{\theta}_{\text{wide}}^{\text{c}}) := &\begin{bmatrix} \boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) & \boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) & -\boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) \\ \boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) & \boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) & -\boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) \\ -\boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) & -\boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) & \boldsymbol{H}_S^{(1),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) \end{bmatrix}\\ &+ \begin{bmatrix} \boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}_{\text{narr}}^{\text{c}}) \end{bmatrix},\end{aligned}$$

Then for $\boldsymbol{u} = [\frac{1}{2}\boldsymbol{v}^\intercal, \frac{1}{2}\boldsymbol{v}^\intercal, \boldsymbol{v}^\intercal]^\intercal$,

$$\boldsymbol{u}^\intercal \widetilde{\boldsymbol{H}}(\boldsymbol{\theta}_{\text{wide}}^{\text{c}})\boldsymbol{u} = -\frac{1}{2}\lambda < 0,$$

indicating $\boldsymbol{\theta}_{\text{wide}}^{\text{c}}$ is a strict-saddle point.

Otherwise, if $\lambda < 0$, since $\boldsymbol{v}^\intercal \widetilde{\boldsymbol{H}}(\boldsymbol{\theta}_{\text{wide}}^{\text{c}})\boldsymbol{v} = \lambda$, then for $\boldsymbol{u} = [\boldsymbol{v}^\intercal, -\boldsymbol{v}^\intercal, \boldsymbol{0}]^\intercal$,

$$\boldsymbol{u}^\intercal \widetilde{\boldsymbol{H}}(\boldsymbol{\theta}_{\text{wide}}^{\text{c}})\boldsymbol{u} = 2\lambda < 0,$$

indicating $\boldsymbol{\theta}_{\text{wide}}^{\text{c}}$ is also a strict-saddle point. □

Lemma (Lemma 6.2 in main text)**.** Given any data S, loss $\ell(\cdot,\cdot)$ and activation $\sigma(\cdot)$, for any NN, if a critical point $\boldsymbol{\theta}^{\text{c}} \in \boldsymbol{\Theta}^{\text{c}}$ satisfies:

(i) $\boldsymbol{H}_S(\boldsymbol{\theta}^{\text{c}}) \succeq 0$;

(ii) $\boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}^{\text{c}}) = \boldsymbol{0}$;

(iii) $\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}^{\mathrm{c}}) \neq \mathbf{0}$.

Then there exists a general compatible critical embedding $\mathcal{T}$, such that $\mathcal{T}(\boldsymbol{\theta}^{\mathrm{c}})$ is a strict-saddle point.

Proof. Since we have

$$\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}) := \sum_{i=1}^{m_L} \mathbb{E}_S \partial_i \ell(f_{\boldsymbol{\theta}}, f^*) \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \left((f_{\boldsymbol{\theta}})_i\right), \tag{A.20}$$

then $\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}_{\mathrm{narr}}^{\mathrm{c}})$ can be written into the form of

$$\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}_{\mathrm{narr}}^{\mathrm{c}}) = \mathbb{E}_S \begin{pmatrix} \boldsymbol{C}_{1,1} & \boldsymbol{C}_{1,2} & \boldsymbol{C}_{1,3} \\ \boldsymbol{C}_{2,1} & \boldsymbol{C}_{2,2} & \boldsymbol{C}_{2,3} \\ \boldsymbol{C}_{3,1} & \boldsymbol{C}_{3,2} & \boldsymbol{C}_{3,3} \end{pmatrix},$$

where

$$\begin{aligned}
\boldsymbol{C}_{1,1} &= \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{W}^{[L]}} \nabla_{\boldsymbol{W}^{[L]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right), \\
\boldsymbol{C}_{1,2} &= \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{W}^{[L]}} \nabla_{\boldsymbol{b}^{[L]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right), \\
\boldsymbol{C}_{1,3} &= \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{W}^{[L]}} \nabla_{\boldsymbol{\theta}^{[L-1]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right), \\
\boldsymbol{C}_{2,1} &= \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{b}^{[L]}} \nabla_{\boldsymbol{W}^{[L]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right), \\
\boldsymbol{C}_{2,2} &= \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{b}^{[L]}} \nabla_{\boldsymbol{b}^{[L]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right), \\
\boldsymbol{C}_{2,3} &= \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{b}^{[L]}} \nabla_{\boldsymbol{\theta}^{[L-1]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right), \\
\boldsymbol{C}_{3,1} &= \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{\theta}^{[L-1]}} \nabla_{\boldsymbol{W}^{[L]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right), \\
\boldsymbol{C}_{3,2} &= \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{\theta}^{[L-1]}} \nabla_{\boldsymbol{b}^{[L]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right), \\
\boldsymbol{C}_{3,3} &= \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{\theta}^{[L-1]}} \nabla_{\boldsymbol{\theta}^{[L-1]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right),
\end{aligned}$$

from relation (A.15), we obtain that

$$\begin{aligned}
&\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}_{\mathrm{narr}}^{\mathrm{c}}) \\
=&\mathbb{E}_S \begin{bmatrix} \mathbf{0} & \mathbf{0} & \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{W}^{[L]}} \nabla_{\boldsymbol{\theta}^{[L-1]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \sum_{i=1}^{m_L} \partial_i \ell(f_{\boldsymbol{\theta}_{\mathrm{narr}}}, f^*) \nabla_{\boldsymbol{\theta}^{[L-1]}} \nabla_{\boldsymbol{W}^{[L]}} \left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right) & \mathbf{0} & \boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}_{\mathrm{narr}}^{\mathrm{c}}) \end{bmatrix},
\end{aligned}$$

We observe that the matrix $\widetilde{\boldsymbol{H}}_S^{(2)}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}})$ below is a principle submatrix of the Hessian at $\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}}$, i.e., by choosing the columns and rows corresponding to $\left(\boldsymbol{W}^{[L]}, \boldsymbol{\theta}^{[L-1]}\right)$, we obtain that

$$\begin{aligned}&\widetilde{\boldsymbol{H}}_S^{(2)}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}})\\ =&\mathbb{E}_S\begin{bmatrix}\mathbf{0} & \sum_{i=1}^{m_L}\partial_i\ell(f_{\boldsymbol{\theta}_{\mathrm{wide}}}, \boldsymbol{f}^*)\nabla_{\boldsymbol{W}^{[L]}}\nabla_{\boldsymbol{\theta}^{[L-1]}}\left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right)\\ \sum_{i=1}^{m_L}\partial_i\ell(f_{\boldsymbol{\theta}_{\mathrm{wide}}}, \boldsymbol{f}^*)\nabla_{\boldsymbol{\theta}^{[L-1]}}\nabla_{\boldsymbol{W}^{[L]}}\left((f_{\boldsymbol{\theta}_{\mathrm{narr}}})_i\right) & \boldsymbol{H}_S^{(2),[L-1]}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}})\end{bmatrix},\end{aligned}$$

and it satisfies $\mathrm{tr}(\widetilde{\boldsymbol{H}}_S^{(2)}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}})) = 0$.

Since $\boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}}) \neq \mathbf{0}$, then $\widetilde{\boldsymbol{H}}_S^{(2)}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}})$ has at least one negative eigenvalue, denoted by λ^{neg} associated with its unit eigenvector $\boldsymbol{v}$, i.e., $\boldsymbol{v}^\intercal\widetilde{\boldsymbol{H}}_S^{(2)}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}})\boldsymbol{v} = \lambda^{\mathrm{neg}} < 0$. Then we consider the three-fold global splitting embedding $\mathcal{T}_{\mathrm{global}}$ defined in Example 5.1, with $\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{wide}} = \mathcal{T}_{\mathrm{global}}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}})$.

From relation (A.14), we obtain that

$$\boldsymbol{H}_S(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{wide}}) = \boldsymbol{H}_S^{(1)}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{wide}}) + \boldsymbol{H}_S^{(2)}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{wide}}),$$

we recall that

$$\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \boldsymbol{\theta}^{[L-1]} = \left(\mathrm{vec}(\boldsymbol{W}^{[1]})^\intercal, \cdots, \mathrm{vec}(\boldsymbol{W}^{[L-2]})^\intercal, \mathrm{vec}(\boldsymbol{W}^{[L-1]})^\intercal, \boldsymbol{b}^{[1]\intercal}, \cdots, \boldsymbol{b}^{[L-1]\intercal}\right),$$

and if we concatenate $\boldsymbol{\theta}_1$ and $\mathrm{vec}(\boldsymbol{W}^{[L]})^\intercal$ into a new vector $\boldsymbol{d}_1 := \left(\boldsymbol{\theta}_1, \mathrm{vec}(\boldsymbol{W}^{[L]})^\intercal\right)$, and $\boldsymbol{\theta}_2$ and $\mathrm{vec}(\boldsymbol{W}^{[L]})^\intercal$ into a new vector $\boldsymbol{d}_2 := \left(\boldsymbol{\theta}_2, \mathrm{vec}(\boldsymbol{W}^{[L]})^\intercal\right)$, we observe that the matrix $\widetilde{\boldsymbol{H}}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{wide}})$ below is a principle submatrix of the Hessian $\boldsymbol{H}_S(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{wide}}$, i.e., by choosing the columns and rows corresponding to $(\boldsymbol{d}_1, \boldsymbol{d}_2)$, we obtain that

$$\widetilde{\boldsymbol{H}}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{wide}}) := \begin{bmatrix}\boldsymbol{D}_{1,1} & \boldsymbol{D}_{1,2}\\ \boldsymbol{D}_{2,1} & \boldsymbol{D}_{2,2}\end{bmatrix} + \begin{bmatrix}\widetilde{\boldsymbol{H}}_S^{(2)}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}}) & \mathbf{0}\\ \mathbf{0} & \widetilde{\boldsymbol{H}}_S^{(2)}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{narr}})\end{bmatrix},$$

with

$$\boldsymbol{D}_{p,q} := \mathbb{E}_S\sum_{i,j=1}^{m_L}\partial_{ij}\ell(f_{\boldsymbol{\theta}_{\mathrm{wide}}}, \boldsymbol{f}^*)\nabla_{\boldsymbol{d}_p}(f_{\boldsymbol{\theta}_{\mathrm{wide}}})_i\left(\nabla_{\boldsymbol{d}_q}(f_{\boldsymbol{\theta}_{\mathrm{wide}}})_j\right)^\intercal,$$

for $p \in [2]$ and $q \in [2]$. Since $\boldsymbol{d}_1 = \boldsymbol{d}_2$, we obtain that

$$\boldsymbol{D}_{1,1} = \boldsymbol{D}_{2,2} = \boldsymbol{D}_{1,2} = \boldsymbol{D}_{2,1}.$$

Then, by choosing $\boldsymbol{u} = [\boldsymbol{v}^T, -\boldsymbol{v}^T]^T$, we have

$$\boldsymbol{u}^\intercal\widetilde{\boldsymbol{H}}(\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{wide}})\boldsymbol{u} = 2\lambda^{\mathrm{neg}} < 0,$$

indicating $\boldsymbol{\theta}^{\mathsf{c}}_{\mathrm{wide}}$ is a strict-saddle point. □

B Step by step backpropagation

In this section, we derive all the relations in (3.5) concerning $\{\boldsymbol{f}_{\boldsymbol{\theta}}^{[l]}\}_{l=1}^{L}$, $\{\boldsymbol{g}_{\boldsymbol{\theta}}^{[l]}\}_{l=1}^{L}$, $\{\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]}\}_{l=1}^{L}$. We shall write out all the relations once again

$$\boldsymbol{z}_{\boldsymbol{\theta}}^{[L]} = \nabla \ell, \tag{B.1}$$

$$\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} = (\boldsymbol{W}^{[l+1]})^{\intercal} \left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l+1]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l+1]} \right), \quad l \in [L-1], \tag{B.2}$$

$$\nabla_{\boldsymbol{W}^{[l]}} \ell = \left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l]} \right) (\boldsymbol{f}_{\boldsymbol{\theta}}^{[l-1]})^{\intercal}, \quad l \in [L], \tag{B.3}$$

$$\nabla_{\boldsymbol{b}^{[l]}} \ell = \boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[l]}, \quad l \in [L]. \tag{B.4}$$

For relation (B.1), since by definition $\boldsymbol{z}_{\boldsymbol{\theta}}^{[L]} = \nabla_{\boldsymbol{f}^{[L]}} \ell = \nabla_{\boldsymbol{f}_{\boldsymbol{\theta}}} \ell$, and $\ell = \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*)$, where $\boldsymbol{f}^*$ is fixed, hence

$$\left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[L]} \right)_i = \frac{\partial \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*)}{\partial \, (\boldsymbol{f}_{\boldsymbol{\theta}})_i} = \partial_i \ell(\boldsymbol{f}_{\boldsymbol{\theta}}, \boldsymbol{f}^*).$$

We prove relation (B.2) in detail and the rest of the relations follow the same reasoning. First things first, we show that it holds for $l = L-1$. Since $\boldsymbol{z}_{\boldsymbol{\theta}}^{[L]} = \nabla \ell$, and $\boldsymbol{g}_{\boldsymbol{\theta}}^{[L]} := \mathbf{1}$, hence $\boldsymbol{z}_{\boldsymbol{\theta}}^{[L]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[L]} = \nabla \ell$. By definition,

$$\boldsymbol{z}_{\boldsymbol{\theta}}^{[L-1]} = \nabla_{\boldsymbol{f}^{[L-1]}} \ell,$$

using chain rule, since

$$(\boldsymbol{f}^{[L]})_i = \sum_{j=1}^{m_{L-1}} \boldsymbol{W}_{i,j}^{[L]} \left(\boldsymbol{f}^{[L-1]} \right)_j + (\boldsymbol{b}^{[L]})_i,$$

then

$$\left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[L-1]} \right)_j = \frac{\partial \ell}{\partial \, (\boldsymbol{f}^{[L-1]})_j} = \sum_{i=1}^{m_L} \frac{\partial \ell}{\partial \, (\boldsymbol{f}^{[L]})_i} \frac{\partial \, (\boldsymbol{f}^{[L]})_i}{\partial \, (\boldsymbol{f}^{[L-1]})_j} = \sum_{i=1}^{m_L} \partial_i \ell \, \boldsymbol{W}_{i,j}^{[L]},$$

hence

$$\boldsymbol{z}_{\boldsymbol{\theta}}^{[L-1]} = (\boldsymbol{W}^{[L]})^{\intercal} \nabla \ell = (\boldsymbol{W}^{[L]})^{\intercal} \left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[L]} \circ \boldsymbol{g}_{\boldsymbol{\theta}}^{[L]} \right).$$

Now for $l \in [L-2]$, we have that since

$$\left(\boldsymbol{f}^{[l+1]} \right)_i = \sigma \left(\left(\sum_{j=1}^{m_l} \boldsymbol{W}_{i,j}^{[l+1]} \left(\boldsymbol{f}^{[l]} \right)_j \right) + (\boldsymbol{b}^{[l+1]})_i \right),$$

then

$$\left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]} \right)_j = \frac{\partial \ell}{\partial \, (\boldsymbol{f}^{[l]})_j} = \sum_{i=1}^{m_{l+1}} \frac{\partial \ell}{\partial \, (\boldsymbol{f}^{[l+1]})_i} \frac{\partial \left(\boldsymbol{f}^{[l+1]} \right)_i}{\partial \, (\boldsymbol{f}^{[l]})_j},$$

where

$$\frac{\partial\left(\boldsymbol{f}^{[l+1]}\right)_i}{\partial\left(\boldsymbol{f}^{[l]}\right)_j}=\sigma^{(1)}\left(\left(\sum_{j=1}^{m_l}\boldsymbol{W}_{i,j}^{[l+1]}\left(\boldsymbol{f}^{[l]}\right)_j\right)+(\boldsymbol{b}^{[l+1]})_i\right)\boldsymbol{W}_{i,j}^{[l+1]},$$

hence

$$\begin{aligned}\left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l]}\right)_j&=\frac{\partial\ell}{\partial\boldsymbol{f}_j^{[l]}}=\sum_{i=1}^{m_{l+1}}\frac{\partial\ell}{\partial\boldsymbol{f}_i^{[l+1]}}\sigma^{(1)}\left(\left(\sum_{j=1}^{m_l}\boldsymbol{W}_{i,j}^{[l+1]}\boldsymbol{f}_j^{[l]}\right)+\boldsymbol{b}_i^{[l+1]}\right)\boldsymbol{W}_{i,j}^{[l+1]}\\&=(\boldsymbol{W}^{[l+1]})^{\intercal}\left(\boldsymbol{z}_{\boldsymbol{\theta}}^{[l+1]}\circ\boldsymbol{g}_{\boldsymbol{\theta}}^{[l+1]}\right).\end{aligned}$$

C Formal definition of the composition of two embeddings, Definition 4.5

Definition (Composition of two embeddings)**.** Suppose we have an $\mathrm{NN}(\{m_l\}_{l=0}^{L})$ and its parameters $\boldsymbol{\theta}\in\mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$, and we have two embeddings $\mathcal{T}$ and $\mathcal{T}'$ satisfying

$$\mathcal{T}:\ \mathrm{Tuple}_{\{m_0,\cdots,m_L\}}\to\mathrm{Tuple}_{\{m_0',\cdots,m_L'\}},\quad \mathcal{T}':\ \mathrm{Tuple}_{\{m_0',\cdots,m_L'\}}\to\mathrm{Tuple}_{\{m_0'',\cdots,m_L''\}},$$

with $\mathcal{T}'$ maps the range of $\mathcal{T}$ into $\mathrm{Tuple}_{\{m_0'',\cdots,m_L''\}}$, where $m_0''=m_0'=m_0$, $m_L''=m_L'=m_L$, and for any $l\in[L-1]$, $m_l''\geq m_l'\geq m_l$. Since $\mathcal{T}'\mathcal{T}$ is obviously an injective operator, then $\mathcal{T}'\mathcal{T}$ is an embedding $\mathcal{T}'\mathcal{T}:\mathrm{Tuple}_{\{m_0,\cdots,m_L\}}\to\mathrm{Tuple}_{\{m_0'',\cdots,m_L''\}}$, and we term $\mathcal{T}'\mathcal{T}$ the composition of $\mathcal{T}'$ and $\mathcal{T}$,i.e., for any $\boldsymbol{\theta}\in\mathrm{Tuple}_{\{m_0,\cdots,m_L\}}$,

$$\mathcal{T}'\mathcal{T}(\boldsymbol{\theta}):=\mathcal{T}'(\mathcal{T}(\boldsymbol{\theta})).$$

References

[1] Weinan E, Chao Ma, Lei Wu, and Stephan Wojtowytsch. Towards a mathematical understanding of neural network-based machine learning: What we know and what we don't. *CSIAM Transactions on Applied Mathematics*, 1(4):561–615, 2020.

[2] Ruoyu Sun, Dawei Li, Shiyu Liang, Tian Ding, and Rayadurgam Srikant. The global landscape of neural networks: An overview. *IEEE Signal Processing Magazine*, 37(5):95–108, 2020.

[3] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6391–6401, 2018.

[4] Ivan Skorokhodov and Mikhail Burtsev. Loss landscape sightseeing with multi-point optimization. *arXiv preprint arXiv:1910.03867*, 2019.

[5] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8580–8589, 2018.

[6] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.

[7] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 8141–8150, 2019.

[8] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018.

[10] Lénaïc Chizat and Francis R. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems*, 2018.

[11] Tao Luo, Zhi-Qin John Xu, Zheng Ma, and Yaoyu Zhang. Phase diagram for two-layer relu neural networks at infinite-width limit. *Journal of Machine Learning Research*, 22(71):1–47, 2021.

[12] Leo Breiman. Reflections after refereeing papers for nips. *The Mathematics of Generalization*, XX:11–15, 1995.

[13] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017*.

[14] Zhi-Qin J Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. *International Conference on Neural Information Processing*, pages 264–274, 2019.

[15] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767, 2020.

[16] Nasim Rahaman, Devansh Arpit, Aristide Baratin, Felix Draxler, Min Lin, Fred A Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of deep neural networks. *International Conference on Machine Learning*, 2019.

[17] Yaoyu Zhang, Tao Luo, Zheng Ma, and Zhi-Qin John Xu. A linear frequency principle model to understand the absence of overfitting in neural networks. *Chinese Physics Letters*, 38(3):038701, 2021.

[18] Tao Luo, Zheng Ma, Zhi-Qin John Xu, and Yaoyu Zhang. Theory of the frequency principle for general deep neural networks. *CSIAM Transactions on Applied Mathematics*, 2(3):484–507, 2021.

[19] Zhi-Qin John Xu, Yaoyu Zhang, and Tao Luo. Overview frequency principle/spectral bias in deep learning. *arXiv preprint arXiv:2201.07395*, 2022.

[20] Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient descent quantizes relu network features. *arXiv preprint arXiv:1803.08367*, 2018.

[21] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

[22] Jason D Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I Jordan, and Benjamin Recht. First-order methods almost always avoid strict saddle points. *Mathematical Programming*, 176(1):311–337, 2019.

[23] Yaim Cooper. Global minima of overparameterized neural networks. *SIAM Journal on Mathematics of Data Science*, 3(2):676–691, 2021.

[24] Yaoyu Zhang, Zhongwang Zhang, Tao Luo, and Zhi-Qin John Xu. Embedding principle of loss landscape of deep neural networks. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2021.

[25] Kenji Fukumizu and Shun-ichi Amari. Local minima and plateaus in hierarchical structures of multilayer perceptrons. *Neural Networks*, 13(3):317–327, 2000.

[26] Kenji Fukumizu, Shoichiro Yamaguchi, Yoh-ichi Mototake, and Mirai Tanaka. Semi-flat minima and saddle points by embedding neural networks to overparameterization. *Advances in Neural Information Processing Systems*, 32:13868–13876, 2019.

[27] Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Clement Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9722–9732. PMLR, 2021.

[28] Simon Du and Jason Lee. On the power of over-parametrization in neural networks with quadratic activation. In *International Conference on Machine Learning*, pages 1329–1338. PMLR, 2018.

[29] Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2018.

[30] Patrick Cheridito, Arnulf Jentzen, and Florian Rossmannek. Landscape analysis for shallow relu neural networks: Complete classification of critical points for affine target functions. *arXiv preprint arXiv:2103.10922*, 2021.

[31] Nitish Shirish Keskar, Jorge Nocedal, Ping Tak Peter Tang, Dheevatsa Mudigere, and Mikhail Smelyanskiy. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[32] Zhongwang Zhang, Hanxu Zhou, and Zhi-Qin John Xu. A variance principle explains why dropout finds flatter minima. *arXiv preprint arXiv:2111.01022*, 2021.

[33] Lei Wu, Zhanxing Zhu, and Weinan E. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

[34] Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: beyond sharp and flat local minima. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 2553–2564, 2019.

[35] Tian Ding, Dawei Li, and Ruoyu Sun. Sub-optimal local minima exist for neural networks with almost all non-linear activations. *arXiv preprint arXiv:1911.01413*, 2019.

[36] Luca Venturi, Afonso S Bandeira, and Joan Bruna. Spurious valleys in one-hidden-layer neural network optimization landscapes. *Journal of Machine Learning Research*, 20:133, 2019.

[37] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.

[38] Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. Sgd on neural networks learns functions of increasing complexity. *Advances in Neural Information Processing Systems*, 32:3496–3506, 2019.

[39] Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X*, 10(4):041044, 2020.

[40] Shilin He, Xing Wang, Shuming Shi, Michael R Lyu, and Zhaopeng Tu. Assessing the bilingual knowledge learned by neural machine translation models. *arXiv preprint arXiv:2004.13270*, 2020.

[41] Chris Mingard, Joar Skalse, Guillermo Valle-Pérez, David Martínez-Rubio, Vladimir Mikulik, and Ard A Louis. Neural networks are a priori biased towards boolean functions with low entropy. *arXiv preprint arXiv:1909.11522*, 2019.

[42] Pengzhan Jin, Lu Lu, Yifa Tang, and George Em Karniadakis. Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness. *Neural Networks*, 130:85–99, 2020.

[43] Franco Pellegrini and Giulio Biroli. An analytic theory of shallow networks dynamics for hinge loss classification. *Advances in Neural Information Processing Systems*, 33, 2020.

[44] Zhi-Qin John Xu, Hanxu Zhou, Tao Luo, and Yaoyu Zhang. Towards understanding the condensation of two-layer neural networks at initial training. *arXiv preprint arXiv:2105.11686*, 2021.

[45] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204. PMLR, 2015.

[46] Levent Sagun, Léon Bottou, and Yann LeCun. Singularity of the Hessian in deep learning. *arXiv preprint arXiv:1611.07476*, 2016.

[47] Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge University Press, 2012.