Generalization Properties of NAS under Activation and Skip Connection Search

Zhenyu Zhu, Fanghui Liu, Grigorios G Chrysos, Volkan Cevher

EPFL, Switzerland

{[first name].[surname]}@epfl.ch

Abstract

Neural Architecture Search (NAS) has fostered the automatic discovery of neural architectures, which achieve state-of-the-art accuracy in image recognition. Despite the progress achieved with NAS, so far there is little attention to theoretical guarantees on NAS. In this work, we study the generalization properties of NAS under a unifying framework enabling (deep) layer skip connection search and activation function search. To this end, we derive the lower (and upper) bounds of the minimum eigenvalue of Neural Tangent Kernel (NTK) under the (in)finite width regime using a certain search space including mixed activation functions, fully connected, and residual neural networks. We use the minimum eigenvalue to establish generalization error bounds of NAS in the stochastic gradient descent training. Importantly, we theoretically and experimentally show how the derived results can guide NAS to select the top-performing architectures, even in the case without training, leading to a training-free algorithm based on our theory. Accordingly, our numerical validation shed light on the design of computationally efficient methods for NAS. Our analysis is non-trivial due to the coupling of various architectures and activation functions under the unifying framework and has its own interest on providing the lower bound of the minimum eigenvalue of NTK in deep learning theory.

1 Introduction

Neural Architecture Search (NAS) [Zoph and Le, 2017] is a powerful technique that enables the automatic design of neural architectures. NAS defines a set of operations, that includes various activation functions and layer types, or potential connections among layers [Elsken et al., 2019, Ren et al., 2021], which are then referred to as the *search space*. Optimization over the search space returns the optimal architecture as a subset of the possible combinations of operations. NAS[1] obtains state-of-the-art results in image recognition [Liu et al., 2019a, Ding et al., 2020, Zhang et al., 2019, Chen et al., 2019] or can be used to further improve architectures defined by a human expert [Tan and Le, 2019]. The spectacular results obtained by NAS have led to a significant interest in the community to further improve the NAS algorithms, the search space etc. However, to date little focus has been provided in the following question: *Can NAS [1] achieve generalization guarantees similar to a typical neural network?*

Neural tangent kernel (NTK)-based analysis [Jacot et al., 2018] is a powerful method for analyzing the optimization and the generalization of deep networks [Allen-Zhu et al., 2019, Cao and Gu, 2019, Chen et al., 2020a, Arora et al., 2019a]. The minimum eigenvalue of NTK has been used in previous work to demonstrate the global convergence of gradient descent, such as two-layer networks [Du et al., 2019b], and deep networks with polynomially wide layers [Allen-Zhu et al., 2019]. Besides,

[1] In the sequel, we interchangeably refer to NAS as the "architecture obtained from NAS" or the framework to design the neural architecture.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

the minimum eigenvalue of NTK is also used to prove generalization bounds [Arora et al., 2019a] and memorization [Montanari and Zhong, 2020]. However, previous work mainly focuses on fixed architectures, e.g., fully-connected (FC) neural networks [Allen-Zhu et al., 2018, Bartlett et al., 2017] or residual neural networks [He et al., 2016, Huang et al., 2020], in which a single activation function is used throughout the network. These off-the-shelf theoretical results cannot be directly applied to analyze the rich search space (of NAS) that is covering various/mixed architectures and parameters. That makes the non-trivial analysis on NAS worth of study on its own right.

The recent work of Oymak et al. [2021] is the first work to provide generalization guarantees on a related problem, i.e., activation functions search. The study provides generalization results on two-layer networks relying on the minimum eigenvalue with a strictly larger than zero assumption, i.e., $\lambda_{\min}(\boldsymbol{K}) > 0$ for the NTK matrix $\boldsymbol{K}$.

In this work, we introduce the first theoretical guarantees for multilayer NAS where the search space includes activation functions and skip connections. We study the upper/lower bound of the minimum eigenvalue of NTK (in the (in)finite regime) under mixed activation functions and architectures to evade the minimum eigenvalue assumption of Oymak et al. [2021]. Then, we provide optimization and generalization guarantees of deep neural networks (DNNs) equipped with NAS. Our results indicate that the minimum eigenvalue estimation can act as a powerful metric for NAS. This method, called Eigen-NAS, is train-free, but still effective with experimental validation when compared to recent promising algorithms [Xu et al., 2021, Chen et al., 2021, Mellor et al., 2021]. Formally, our main contribution and findings are summarized below:

i) We build a general theoretical framework based on NTK for NAS with search on popular activation functions in each layer, fully-connected, and skip connections. We derive the NTK formula of these architectures in the (in)finite width regime under the unifying framework.

ii) We derive the upper and lower bound of minimum eigenvalue of NTK under the (in)finite width regime for the considered architectures. We introduce a new technique to ensure the probability of concentration inequality remains positive, addressing the negative probability issue in previous work. Our analysis highlights how the error bounds differs under activation function search and skip connection search and can guide NAS.

iii) By virtue of the NTK matrix and its minimum eigenvalue, we establish a connection between them and generalization of the searched DNN trained by stochastic gradient descent (SGD). Our theoretical results show that the generalization performance largely depends on the minimum eigenvalue of NTK for NAS, which provides theoretical guarantees for the searched architecture.

iv) Our theoretical results are supported by thorough experimental validations with the following findings: 1) our error bounds on the minimum eigenvalue largely depend on activation functions in the first layer rather than those in deeper layers. 2) The applied NAS algorithm always picks up ReLU (Rectified Linear Unit) and LeakyReLU in the optimal architecture, which coincides with our theory that predicts ReLU and LeakyReLU achieve the largest minimum eigenvalues. 3) The skip connections are required in each layer under our not very large DNNs. Furthermore, our experimental evidence on Eigen-NAS indicates that the minimum eigenvalue is a promising metric to guide NAS (without training) as suggested by our theory.

Technical challenges. The technical challenges of this paper mainly focus on how to analyze activation functions with different properties and skip connections under a unifying framework. This work is non-trivial, firstly previous works mainly focus on the ReLU activation function [Nguyen et al., 2021, Cao and Gu, 2019, Allen-Zhu et al., 2019] in optimization and generalization of a single fully-connected neural network. Their proofs heavily depend on properties of ReLU, e.g., homogeneity and $\mathrm{ReLU}(x) = x\mathrm{ReLU}'(x)$ which are invalid when other commonly-used activation functions, e.g., Tanh, Sigmoid, and Swish, are used. This problem becomes quite intractable under the consideration of mixing of activation functions and residual connections. To tackle these technical challenges, we develop the following techniques: a) to handle the non-homogeneous property of Tanh, Sigmoid, and Swish, we develop a new integral estimation approach for minimal eigenvalue estimation. b) To establish the connection between minimum eigenvalues of NTK and generalization errors, we use the Lipschitz continuity to avoid the special property of ReLU. More importantly, we introduce a new technique [Yaskov, 2014] to replace Gershgorin circle theorem for minimum eigenvalue estimation, which avoids concentration inequalities with negative probability in some certain cases [Nguyen et al., 2021].

2 Related work

Network architecture search (NAS): The idea of NAS stems from Zoph and Le [2017], while the idea of cell search, i.e., searching core building blocks and composing them together, emerged in Zoph et al. [2018]. The earlier literature used discrete optimization techniques for obtaining the architecture. DARTS Liu et al. [2019b] considers NAS as a continuous bi-level optimization task. Recent variants of DARTS [Xu et al., 2019, Wu et al., 2019] and several train-free methods [Mellor et al., 2021, Chen et al., 2021, Xu et al., 2021] have demonstrated success in reducing the search time or improving the search algorithm. However, the aforementioned works have not provided generalization guarantees for the optimal architecture.

Optimization and generalization of DNNs via NTK: In the NTK framework [Jacot et al., 2018, Du et al., 2019a, Chen et al., 2020b], the training dynamics of (in)finite-width networks can be exactly characterized by kernel tools. Leveraging NTK facilitates studies on the global convergence of GD Allen-Zhu et al. [2019], Du et al. [2019a], Nguyen [2021] in DNNs via the minimum eigenvalue of NTK. In fact, it also controls the generalization performance of DNNs [Du et al., 2019b, Cao and Gu, 2019, Allen-Zhu et al., 2018], which is further studied in Bietti and Bach [2021].

3 Problem Settings

In this section we introduce the problem setting of our NAS framework based on the search space and algorithm (search strategy) for our paper.

Let $X \subseteq \mathbb{R}^d$ be a compact metric space and $Y \subseteq \mathbb{R}$. We assume that the training set $\mathcal{D}_{tr} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ is drawn from a probability measure $\mathcal{D}$ on $X \times Y$, with its marginal data distribution denoted by $\mathcal{D}_X$. The goal of a supervised learning task is to find a hypothesis (i.e., a neural network used in this work) $f : X \to Y$ such that $f(\boldsymbol{x}; \boldsymbol{W})$ parameterized by $\boldsymbol{W}$ is a good approximation of the label $y \in Y$ corresponding to a new sample $\boldsymbol{x} \in X$. In this paper, we consider the classification task, evaluated by minimizing the expected risk

$$\min_{\boldsymbol{W}} \ \ell_{\mathcal{D}}(\boldsymbol{W}) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \, \ell[yf(\boldsymbol{x}; \boldsymbol{W})] \,,$$

where $\ell[yf(\boldsymbol{x}; \boldsymbol{W})]$ is the classification loss $\ell(\cdot)$ as a surrogate of the expected 0-1 loss $\ell_{\mathcal{D}}^{0-1}(\boldsymbol{W}) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[1\{yf(\boldsymbol{x}; \boldsymbol{W}) < 0\}]$. In this paper, we employ the cross-entropy loss, which is defined as $\ell(z) = \log[1 + \exp(-z)]$.

Notation: For an integer L, we use the shorthand $[L] = \{1, 2, \ldots, L\}$. The multivariate standard Gaussian distribution is $\mathcal{N}(\mathbf{0}, \mathbb{I}_d)$ with the zero-mean vector $\mathbf{0}$ and the identity-variance matrix $\mathbb{I}_d$. We denote the direct sum by $\oplus$. We follow the standard Bachmann–Landau notation in complexity theory e.g., $\mathcal{O}$, o, Ω, and Θ for order notation.

3.1 Neural Networks and Search Space

In this work, we consider a particular parametrization of f as a deep neural network (DNN) with depth L ($L \geq 3$)[2] which includes the fully-connected (FC) neural networks setting and the residual neural networks setting, and various activation functions in each layer. This allows for a quite general NAS setting. Formally, we define a single-output DNN with the output $\boldsymbol{f}_l(\boldsymbol{x})$ in each layer

$$\boldsymbol{f}_l(\boldsymbol{x}) = \begin{cases} \boldsymbol{x} & l = 0\,, \\ \sigma_1(\boldsymbol{W}_1 \boldsymbol{x}) & l = 1\,, \\ \sigma_l(\langle \boldsymbol{W}_l, \boldsymbol{f}_{l-1}(\boldsymbol{x})\rangle) + \alpha_{l-1} \boldsymbol{f}_{l-1}(\boldsymbol{x}) & 2 \leq l \leq L-1, \\ \langle \boldsymbol{W}_L, \boldsymbol{f}_{L-1}(\boldsymbol{x})\rangle & l = L\,, \end{cases} \tag{1}$$

where the weights of the neural networks are $\boldsymbol{W}_1 \in \mathbb{R}^{m\times d}$, $\boldsymbol{W}_l \in \mathbb{R}^{m\times m}$, $l = 2, \ldots, L-1$ and $\boldsymbol{W}_L \in \mathbb{R}^m$. The binary parameter α_l is for layer search, and the activation function is $\sigma_l(\cdot)$. The neural network output is $f(\boldsymbol{x}; \boldsymbol{W}) = f_L(\boldsymbol{x})$.

Architecture search: A binary vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_{L-2}]^\top$ represents the skip connections, where the $\alpha_l \in \{0, 1\}$ in Eq. (1) indicates whether there is a skip connection in the l-th layer. Notice that we unify FC and residual neural networks under the same framework.

[2] Our results hold for the $L = 2$ setting corresponding to one-hidden layer neural network with slight modifications on notation, so we focus on $L \geq 3$ for simplicity.

Table 1: Formula of different activation functions, definitions of relevant constants and some intermediate results.

σ_l	ReLU	LeakyReLU	Sigmoid[1]	Tanh[2]	Swish
Formula	$\max(0,x)$	$\max(\eta x, x),\ \eta \in (0,1)$	$\frac{1}{1+e^{-x}} - \frac{1}{2}$	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$\frac{x}{1+e^{-x}}$
$\beta_1(\sigma_l)$	1	$1+\eta^2$	$1/8$	2	1
$\beta_2(\sigma_l)$	1	$1+\eta^2$	$1/8$	2	1.22
$\beta_3(\sigma_l)$	1	$1+\eta^2$	$f_S(t)$	$f_T(t)$	$1/2$

[1] We consider the integral $f_{\mathrm{S}}(y) = \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi y}} e^{-\frac{x^2}{2y}} f'_{\mathrm{Sigmoid}}(x)^2 \mathrm{d}x$. We add $-1/2$ in Sigmoid to ensure $f_{\mathrm{Sigmoid}}(0) = 0$ facilitates our theoretical analysis. The parameter is $t := 3(1+\eta^2)(2+\eta^2)^{L-3}$.

[2] The definition of f_T is similar to f_S by using the Tanh function.

Activation function search: We select five representative activation functions used in Eq. (1), i.e., $\mathcal{F}_\sigma = \{\mathrm{ReLU}, \mathrm{LeaklyReLU}, \mathrm{Sigmoid}, \mathrm{Tanh}, \mathrm{Swish}\}$, that can be bounded, unbounded, smooth, non-smooth, monotonic, or non-monotonic, as reported in Table 1. We define $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \cdots, \sigma_{L-1}]^\top$ with $\sigma_l \in \mathcal{F}_\sigma$ for any $l \in [L-1]$ as the indicator to show which activation function is selected in each layer. Importantly, our NAS framework allows for a different activation function in each layer, which enlarges the search space.

In our setting, we conduct the architecture search and the skip connection search independently, and accordingly, our search space is defined as the direct sum of them

$$\mathcal{W} := \mathbb{R}^{L-2} \oplus \mathcal{F}_\sigma^{L-1} \oplus \{\mathbb{R}^{m\times d} \times (\mathbb{R}^{m\times m})^{L-2} \times \mathbb{R}^m\}, \tag{2}$$

where $\boldsymbol{W} := (\boldsymbol{\alpha}, \boldsymbol{\sigma}, \boldsymbol{W}_1, \ldots, \boldsymbol{W}_L) \in \mathcal{W}$ represents the collection of weight matrices and indicator for skips and selected activation functions for all layers.

3.2 Algorithm (Search Strategy)

The search strategy is the core part in NAS to pick up the optimal architecture from the search space. Here we build a general Algorithm 1 combining the search strategy for NAS (the first part) and the subsequent neural network training by SGD (the second part).

We firstly utilize a typical NAS algorithm, e.g., random search WS [Li and Talwalkar, 2020] or DARTS[3], to search skip connections and activation functions independently, which results in the optimal architecture $\{(\sigma_i^*)_{i=1}^{L-1}, (\alpha_i^*)_{i=1}^{L-2}\}$ with the max probability, see sec. 5.1 for details. In particular, Algorithm 1 also allows for the guidance of NAS in a train-free strategy via some specific metrics, e.g., the minimum eigenvalue of NTK (and its variant), see our Eigen-NAS method in sec. 5.2.

Then, we conduct neural network training on the selected architecture by SGD. For ease of theoretical analysis, we employ the constant step-size SGD with one epoch and randomly choose the weight parameters during all the iterations, which is commonly used in deep learning theory [Cao and Gu, 2019, Zou et al., 2019].

4 Main result

In this section, we state the main theoretical results. We present the assumptions used in our proof in sec. 4.1. Then in sec. 4.2 we provide the recursive form of NTK for DNNs defined by Eq. (1) with mixed activation functions and skip connections. The upper and lower bound of the corresponding NTK in the infinite and finite width setting is given in sec. 4.3 and 4.4, respectively. Finally, in sec. 4.5, we connect the minimum eigenvalue of NTK and the generalization error bound of DNNs under these search schemes. The proofs of our theoretical results presented in this section are deferred to Appendix B, C, and D, respectively.

4.1 Assumptions

We make the following assumptions on data and activation functions. Our assumptions are frequently employed in the literature as we highlight below.

Assumption 1. We assume that the data satisfy $\|\boldsymbol{x}\|_2 = 1$.

[3] This algorithm directly outputs the final optimal architecture and optimal parameters.

Algorithm 1: SGD for training DNNs by NAS

Input: search space $\mathcal{S}$, data $\mathcal{D}_{tr} = \{(\boldsymbol{x}_i, y_i)_{i=1}^N\}$, step size γ and $\texttt{Flag}_{\text{method}} \in \{\texttt{EigenNAS}, \texttt{DARTS}, \cdots\}$.
// `conduct NAS algorithms`
if $\texttt{Flag}_{\text{GuideNAS}} = \texttt{EigenNAS}$ **then**
  Guide NAS from $\mathcal{S}$ by our Eigen-NAS algorithm.
else if $\texttt{Flag}_{\text{GuideNAS}} = \texttt{DARTS}$ **then**
  Search neural network architectures from $\mathcal{S}$ using the DARTS algorithm.
end if
Output the optimal architecture $\{(\sigma_i^*)_{i=1}^{L-1}, (\alpha_i^*)_{i=1}^{L-2}\} \in \mathcal{S}$ with max probability.
// `do neural network training via SGD`
Gaussian initialization: $\boldsymbol{W}_l^{(1)} \sim \mathcal{N}(0, 1/m)$, $l \in [L]$
Construct the neural network $f(\boldsymbol{x}; \boldsymbol{W}_l^{(1)})$ based on $\{(\sigma_i^*)_{i=1}^{L-1}, (\alpha_i^*)_{i=1}^{L-2}\}$
for $i = 1$ **to** N **do**
  $\boldsymbol{W}^{(i+1)} = \boldsymbol{W}^{(i)} - \gamma \cdot \nabla_{\boldsymbol{W}} \ell\big(f(\boldsymbol{x}_i; \boldsymbol{W}^{(i)}) y_i\big)$.
end for
Output Randomly choose $\hat{\boldsymbol{W}}$ uniformly from $\big\{\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(N)}\big\}$.

Assumption 2. The activation function $\sigma : \mathbb{R} \to \mathbb{R}$ satisfies $\sigma \in L^2(\mathbb{R}, e^{-x^2/2}/\sqrt{2\pi})$, where $L^2(\mathbb{R}, g)$ denotes the square integrable function.

Assumption 3. We further assume that $\boldsymbol{x}$ is isotropic. i.e. $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \mathbb{I}_d/d$, where the coefficient $1/d$ is to satisfy Assumption 1 at the same time.

Remark: The first assumption on normalized data is commonly used in practice and theory on over-parameterized neural networks [Du et al., 2019b,a, Allen-Zhu et al., 2019, Oymak and Soltanolkotabi, 2020, Malach et al., 2020]. The second assumption is general as the studied activation functions in Table 1 satisfy it. The third assumption is standard in statistics and machine learning [Vershynin, 2018, Hastie et al., 2022, Klimovsky, 2012, Yaskov, 2014]. It covers Gaussian data, and data uniformly spread on the sphere, commonly used in deep learning theory [Mei et al., 2021, Ghosh et al., 2022].

4.2 Recursive NTK for DNNs defined by Eq. (1)

Recall that NTK [Jacot et al., 2018] under the infinite width setting ($m \to \infty$) is:

$$K^{(L)}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) := \mathbb{E}_{\boldsymbol{W}} \left\langle \frac{\partial f(\boldsymbol{x}; \boldsymbol{W})}{\partial \boldsymbol{W}}, \frac{\partial f(\widetilde{\boldsymbol{x}}; \boldsymbol{W})}{\partial \boldsymbol{W}} \right\rangle ,$$

where the NTK matrix for residual networks is derived by the following regular chain rule.

Lemma 1. *For any $l \in [3, L]$ and $s \in [2, L]$, denote*

$$\begin{aligned}
&\boldsymbol{G}^{(1)} = \boldsymbol{X}\boldsymbol{X}^\top, \quad \boldsymbol{A}^{(2)} = \boldsymbol{G}^{(2)} = 2\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\mathbb{I}_d)}[\sigma_1(\boldsymbol{X}\boldsymbol{w})\sigma_1(\boldsymbol{X}\boldsymbol{w})^\top], \\
&\boldsymbol{G}^{(l)} = 2\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\mathbb{I}_N)}[\sigma_{l-1}(\sqrt{\boldsymbol{A}^{(l-1)}}\boldsymbol{w})\sigma_{l-1}(\sqrt{\boldsymbol{A}^{(l-1)}}\boldsymbol{w})^\top], \quad \boldsymbol{A}^{(l)} = \boldsymbol{G}^{(l)} + \alpha_{l-2}\boldsymbol{A}^{(l-1)}, \\
&\dot{\boldsymbol{G}}^{(s)} = 2\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\mathbb{I}_N)}[\sigma'_{s-1}(\sqrt{\boldsymbol{A}^{(s-1)}}\boldsymbol{w})\sigma'_{s-1}(\sqrt{\boldsymbol{A}^{(s-1)}}\boldsymbol{w})^\top].
\end{aligned}$$

Then the NTK for residual networks defined in Eq. (1) can be written as

$$\boldsymbol{K}^{(L)} = \boldsymbol{G}^{(L)} + \sum_{l=1}^{L-1} \boldsymbol{G}^{(l)} \circ \dot{\boldsymbol{G}}^{(l+1)} \circ (\dot{\boldsymbol{G}}^{(l+2)} + \alpha_l \boldsymbol{1}_{N\times N}) \circ \cdots \circ (\dot{\boldsymbol{G}}^{(L)} + \alpha_{L-2}\boldsymbol{1}_{N\times N}).$$

Remark: (i) Our NTK formula of ResNet differs from the one of Tirer et al. [2022], Huang et al. [2020], Belfer et al. [2021] in two critical ways: 1) each skip-layer in our model skips one fully-connected layer and one activation function, as opposed to the two-layer skip of previous works, 2) our formulation does not require every layer to have a parallel skip connection, which increases the flexibility of the network. Those differences also result in a different NTK matrix.
(ii) Our NTK formulation covers different activation functions, and we adopt the same initialization (coefficient) on them to ensure fair/equal search in our NAS framework .

Lemma 1 is general to cover FC and residual neural networks, which facilitates the analysis of minimum eigenvalue of NTK under the unifying framework. If $\alpha_l = 0$ for $l \in [L-1]$, our NTK formulation for residual neural networks degenerates to that of fully-connected neural network, and $\boldsymbol{A}^l$ and $\boldsymbol{G}^l$ become equal.

4.3 Minimum Eigenvalue of NTK for infinite width

We are now ready to state the main result of infinite width neural network. We provide the error bounds of minimum eigenvalue of NTK for infinite width neural network mixed with five different activation functions. The main differences between different activation functions are illustrated in Table 1.

Theorem 1. *For a DNN defined by Eq.* (1) *and a not very large L, let $\boldsymbol{K}^{(L)}$ be the limiting NTK recursively defined in Lemma 1. Then, under Assumptions 1 and 3, when $N \geq \Omega(d^4)$, with probability at least $1 - e^{-d}$, we have*

$$\lambda_{\min}(\boldsymbol{K}^{(L)}) \geq 2\mu_1(\sigma_1)^2 \Theta(N/d) \prod_{p=3}^{L} \left(\beta_3(\sigma_{p-1}) + \alpha_{p-2} \right),$$

$$\lambda_{\min}(\boldsymbol{K}^{(L)}) \leq \frac{N}{d} \sum_{l=1}^{L} \left(\beta_1(\sigma_{l-1}) \prod_{p=2}^{l-1} \left(\beta_1(\sigma_{p-1}) + \alpha_{p-2} \right) \right) \prod_{p=l+1}^{L} \left(\beta_2(\sigma_{p-1}) + \alpha_{p-2} \right) \Bigg),$$

where $\mu_1(\sigma_1)$ is the 1-st Hermite coefficient of the first layer activation function, and $\beta_1, \beta_2, \beta_3$ are three constants on various activation functions defined in Table 1.

Remark: We consider a depth $L \leq 10$, which is often sufficient for the search phase in practical implementations [Liu et al., 2018, Dong et al., 2021]. In addition, existing NAS algorithms such as DARTS tend to have architectures with wide and shallow cell structure as suggested by Shu et al. [2020]. Theorem 1 shows the upper and lower bound of the minimum eigenvalue of NTK under the mix of activation functions and skip connections. The following conclusions can be drawn from our result:

1. The error bounds of minimum eigenvalue significantly depend on the depth of the network L, the skip connections via α_p, which makes the minimum eigenvalue increasing fast as L and the number of skip connections increase. Besides, the minimum eigenvalue is also effected by activation functions via $\beta_1, \beta_2, \beta_3$. Nevertheless, the lower bound is independent of β_1 and β_2.
2. Different activation functions lead to different tendency (increase or decrease) on $\lambda_{\min}(\boldsymbol{K}^{(L)})$. As the depth increases, the lower bound $\lambda_{\min}(\boldsymbol{K}^{(L)})$ under ReLU remains unchanged, increases under LeakyReLU, and decreases when Sigmoid, Tanh or Swish applied, which brings in new findings when compared to Nguyen et al. [2021] sorely on ReLU. For the upper bound for $\lambda_{\min}(\boldsymbol{K}^{(L)})$, we can see our results are positively correlated with the depth L.
3. One can see that $\mu_1(\sigma_1)$ is only related to the activation function of the first layer, which implies that the activation function in the first layer is very important as $\lambda_{\min}(\boldsymbol{K}^{(L)})$ largely depends on it.

4.4 Minimum Eigenvalue of NTK for finite width

To study the finite width, we firstly introduce the Jacobian of the network. Let $\boldsymbol{F} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]^T$. Then, the Jacobian $\boldsymbol{J}$ of $\boldsymbol{F}$ with respect to $\boldsymbol{W}$ is $\boldsymbol{J} = \left[\frac{\partial \boldsymbol{F}}{\partial \text{vec}(\boldsymbol{W}_1)}, \ldots, \frac{\partial \boldsymbol{F}}{\partial \text{vec}(\boldsymbol{W}_L)} \right]$, where $\boldsymbol{J}$ have dimension $\mathbb{R}^{(((L-2)\times m+d+1)\times m \times N}$. The empirical Neural Tangent Kernel (NTK) matrix can be defined as $\bar{\boldsymbol{K}}^{(L)} = \boldsymbol{J}\boldsymbol{J}^\top = \sum_{l=1}^{L} \left[\frac{\partial \boldsymbol{F}}{\partial \text{vec}(\boldsymbol{W}_l)} \right] \left[\frac{\partial \boldsymbol{F}}{\partial \text{vec}(\boldsymbol{W}_l)} \right]^\top$.

Accordingly, we generalize Theorem 1 from the infinite-width to finite-width setting below.

Theorem 2. *For an L-layer network defined by Eq.* (1)*, let $\boldsymbol{K}^{(L)} = \boldsymbol{J}\boldsymbol{J}^\top$ be the NTK matrix, and the weights of the network be initialized as $[\boldsymbol{W}_l]_{i,j} \sim \mathcal{N}(0, 1/m)$, for all $l \in [L]$. Under Assumptions 1 and 3, when $N \geq \Omega(d^4)$, then $\lambda_{\min}(\boldsymbol{J}\boldsymbol{J}^\top)$ can be bounded by*

$$\Theta\left(\frac{N}{d} \prod_{i=2}^{L-1} (\beta_3(\sigma_i) + \alpha_{i-1}) \right) \leq \lambda_{\min}(\boldsymbol{J}\boldsymbol{J}^\top) \leq \frac{N}{d} \sum_{k=0}^{L-1} \Theta \left(\prod_{i=k+2}^{L-1} (\beta_2(\sigma_i) + \alpha_{i-1}) \right),$$

where the first inequality (lower bound) holds with probability at least $1 - e^{-d} - \sum_{l=1}^{L-1} \exp(-\Omega(m)) - \exp(-\Omega(1))$ *and the second inequality (upper bound) holds with probability at least* $1 - \sum_{l=1}^{L-1} \exp(-\Omega(m)) - \exp(-\Omega(1))$*. The definitions of* β_2*, and* β_3 *are the same as those in Theorem 1.*

Remark: Theorem 2 achieves a similar result as Theorem 1 if the width m is large. The only difference in the findings is that probability that the Theorem holds increases as m increases.

4.5 Connection to Generalization Error Bound

Based on the above error bound of the minimum eigenvalue of NTK under different settings, here we establish its relationship with the generalization error of DNNs. We provide a bound on the expected 0-1 error obtained by Algorithm 1.

Theorem 3. *Given a DNN defined by Eq.* (1) *with* $\boldsymbol{y} = (y_1, \ldots, y_N)^\top$ *determined by Algorithm 1 with the step size of SGD* $\gamma = \kappa C_1 \cdot \sqrt{\boldsymbol{y}^\top (\boldsymbol{K}^{(L)})^{-1} \boldsymbol{y}}/(m\sqrt{N})$ *for some small enough absolute constant* κ*. Under Assumptions 1, 2 and 3, for any* $\delta \in \left(0, e^{-1}\right]$ *and a not very large* L*, if the width* $m \geq \hat{m}$*, where* $\hat{m}$ *depends on* $\lambda_{\min}(\boldsymbol{K}^{(L)}), \delta, N$*, and* L*, then with probability at least* $1 - \delta$ *over the randomness of* $\boldsymbol{W}^{(1)}$*, we obtain the following high probability bound:*

$$\mathbb{E}[\ell_{\mathcal{D}}^{0-1}(\hat{\boldsymbol{W}})] \leq \tilde{\mathcal{O}}\left(C_2 \sqrt{\frac{\boldsymbol{y}^\top \boldsymbol{y}}{\lambda_{\min}(\boldsymbol{K}^{(L)}) N}}\right) + \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{N}}\right),$$

where $C_1 = \sqrt{L}/(3\mathrm{Lip}_{\max} + 1)^{L-1}$ *and* $C_2 = \sqrt{L}(3\mathrm{Lip}_{\max} + 1)^{L-1}$ *are two constants only depending on* L *and* $\mathrm{Lip}_{\max}$ *is the maximum value of the Lipschitz constants of the all activation functions.*

Remark: Theorem 3 gives an algorithm-dependent generalization error bound of DNNs defined by Eq. (1) trained with SGD with different activation functions and skip connections. If m is large enough, the learning rate is infinitesimal, which means the generalization error bound mainly depends on the NTK matrix, similarly to Cao and Gu [2019], Du et al. [2019a]. Admittedly, our result is in an exponential increasing order of the depth. However, in practice, the depth L during the search phrase is smaller than 20, or even 10 [Liu et al., 2018, Dong et al., 2021]. Under this setting our results are better than the previous results, which are discussed in detail in Appendix E.

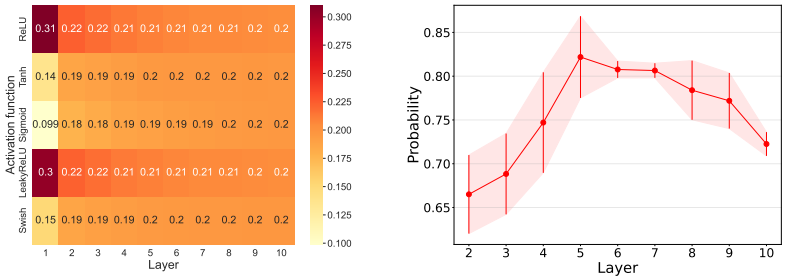
According to Theorem 3, the generalization performance of DNNs is controlled by the minimum eigenvalue of the NTK matrix, which is in turn affected by different activation functions and skip connections, as discussed in Theorem 1. Apart from the NTK matrix itself, the condition $m \geq \hat{m}$ is also effected by different activation functions, which implies that the required minimum width is different in these cases.

4.6 Proof sketch

Our work extends the proofs of Nguyen and Mondelli [2020], Cao and Gu [2019] beyond ReLU, which is critical for enabling search across activations. The extension to other activation functions and skip connections is non-trivial due to non-linearity, inhomogeneity and nonmonotonicity.

To derive the error bound on the minimum eigenvalue, we start from Lemma 1 on the NTK formula under the mixed activation functions and skip connections, and we transform the minimum eigenvalue estimation to the computation (estimation) of the bound $\boldsymbol{G}$, $\dot{\boldsymbol{G}}$ $(\lambda_{\min}(\boldsymbol{G}))$. The infinite width and finite width are included in Appendix B and C respectively. For the upper bound, we estimate the diagonal elements of $\boldsymbol{G}$ and use the minimum eigenvalue less than the mean of the diagonal elements for proof. For the lower bound, we use Hermite expansion and [Yaskov, 2014, Corollary 3.1]. Combining these results concludes the proof.

To derive the generalization error bounds, we need a series of lemmas (see Appendix D). If the input weights are close, the output of each neuron with any activation function does not change too much (see Lemma 7); If the initilizations are close, the neural network output $f(\boldsymbol{x}; \boldsymbol{W})$ is almost linear in $\boldsymbol{W}$ (see Lemma 8), and the loss function $\ell[y_i f(\boldsymbol{x}_i; \boldsymbol{W})]$ is almost a convex function of $\boldsymbol{W}$ for any $i \in [N]$ (see Lemma 9). Accordingly, the gradient and loss of the neural network can be upper bounded by Lemma 10 and 11, respectively, which concludes the proof by combining with some



(a) activation functions $\boldsymbol{\sigma}$ (b) skip connections $\boldsymbol{\alpha}$

Figure 1: Architecture search results on activation functions indicated by the probability of $\boldsymbol{\sigma}$ in (a) and skip connections indicated by $\boldsymbol{\alpha}$ in (b). We notice that for each layer, ReLU and LeakyReLU are selected with the higher probability.

relevant results [Cao and Gu, 2019, Allen-Zhu et al., 2019]. Further discussion on the differences is deferred to Appendix E.

5 Numerical Validation

To validate our theoretical results, we conduct a series of experiments on NAS. Firstly, we simulate the NTK matrices under different depths in Appendix F.4 to verify the relationship between the minimum eigenvalue of NTK and the network depth L in Theorem 1. In sec. 5.1 we use the DARTS algorithm [Liu et al., 2019b] to conduct experiments on activation function search and skip connection search under the search space of Eq. (1). Finally, we use the minimum eigenvalue of NTK to guide the training of NAS on the benchmark NAS-Bench-201 [Dong and Yang, 2020], with a comparison of recent NAS algorithms. Additional experiments on NAS-Bench-101 [Ying et al., 2019] and transfer learning are deferred to Appendix F.5 and F.6.

5.1 DARTS experiment

In this section we employ a typical NAS algorithm, DARTS Liu et al. [2019b], to assess our theoretical results on activation functions and skip connections. We select Fashion-MNIST [Xiao et al., 2017] as a standard benchmark. Details about Fashion-MNIST are shared in Appendix F.1.

Search space and search strategy: Our search space is defined by Eq. (2) on skip connections, activation functions, and weight parameters. We follow the search strategy with Liu et al. [2019b] in a two-level scheme, one is for weight parameter search $\boldsymbol{W}$ and the other is for architecture search $\{\boldsymbol{\alpha}, \boldsymbol{\sigma}\}$, which results in the final optimal architecture $\{\boldsymbol{\alpha}^*, \boldsymbol{\sigma}^*, \boldsymbol{W}^*\}$. Different from Liu et al. [2019b], the activation function search and the skip connection search in our setting is independent. To obtain $\boldsymbol{\sigma}^*$, we use the softmax function to normalize their weights and choose the specific activation function with the highest probability in each layer. To obtain $\boldsymbol{\alpha}^*$, we initialize each entry $\alpha_l = 1/2$ ($l \in [L-2]$), constrain it to $[0, 1]$ during training, and retain the skip connection when $\alpha_l^* > 1/2$.

NAS Results: We conduct the NAS experiment via DARTS on a feedforward neural network with $L = 10$ and $m = 1024$, with 5 runs. After training, the probability of these activation functions and skip connections in each layer is reported in Figure 1(a) and 1(b), respectively. We have the following findings: Firstly, after the search process, LeakyReLU and ReLU are selected as the activations with the highest probability in each layer. This coincides with our theoretical results in Theorem 1. One minor difference is that the probability of LeakyReLU is slightly inferior to ReLU in practice. The reason behind this could be the sparsity of ReLU [de Dios and Bruna, 2020]. Secondly, in the first layer, we observe the largest difference on the probability of various activation functions. As the network becomes deeper, such difference pales in importance until there is no difference among these activation functions in the last layer. This phenomenon matches our theory well. To be specific, in Theorem 1, our result on the minimum eigenvalue largely depends on the first layer and its Hermite coefficient. Besides, this result also provides a justification on omitting the high-order terms while retaining the first layer activation terms. Thirdly, for the skip search result, we find that the skip connections are required in each layer when $L \leq 10$, as suggested by our theoretical results in Theorem 1. It also verifies results of Zhou et al. [2020]. We expect that the skip connections might

Table 2: Results on CIFAR-10, CIFAR-100 and ImageNet-16 as part of NAS-Bench-201. The best performance is highlighted by **bold**. The results of NASWOT, TE-NAS and KNAS are reported from the corresponding papers. The results of ResNet, NAS-RL and DARTS are reported in [Xu et al., 2021]. The results illustrate that Eigen-NAS outperforms the prior art in CIFAR-100 and Imagenet-16.In particular, Eigen-NAS outperforms KNAS in all three cases when the same number of top-k architectures are selected, i.e., $k = 20$, and still achieves promising performance when smaller $k = 5$ used, which we attribute to the more precise minimum eigenvalue estimation.

Type	Model/Algorithm	CIFAR-10 (%)	CIFAR-100 (%)	ImageNet-16 (%)
w/o Search	ResNet [He et al., 2016]	**93.97**	70.86	42.63
Search	NAS-RL [Zoph and Le, 2017]	92.83	70.71	44.10
Gradient	DARTS [Liu et al., 2019b]	88.32	67.34	33.04
Train-free	NASWOT [Mellor et al., 2021]	92.96	70.03	44.43
Train-free	TE-NAS [Chen et al., 2021]	93.90	71.24	42.38
Train-free	KNAS [Xu et al., 2021] ($k = 20$)	93.38	70.78	44.63
Train-free	NASI (T) [Shu et al., 2022]	93.08 ± 0.24	69.51 ± 0.59	40.87 ± 0.85
Train-free	NASI (4T) [Shu et al., 2022]	93.55 ± 0.10	71.20 ± 0.14	44.84 ± 1.41
Train-free	**Eigen-NAS** ($k = 20$)	93.46 ± 0.01	$\mathbf{71.42} \pm 0.63$	$\mathbf{45.54} \pm 0.04$
Train-free	**Eigen-NAS** ($k = 5$)	93.43 ± 0.08	69.92 ± 1.82	45.53 ± 0.06

not be required in each layer for deep neural networks, since their capacity can already be enough [He et al., 2016]; but we defer the related study to a future work.

Interestingly, the search strategy favors the activation functions and the skip connections with larger minimum eigenvalue of NTK, which enjoy better generalization performance. This result also motivates us to study the following question: *can the minimum eigenvalue of NTK guide the search process in NAS?* We provide an affirmative answer in the next section with experimental validations.

5.2 NAS-Bench-201 Experiment

In this experiment, we use the minimum eigenvalue to guide NAS on NAS-Bench-201 [Dong and Yang, 2020]. Each experiment is repeated 5 times, while it can run on a single GPU in a few hours.

Benchmark and baselines: NAS-Bench-201 [Dong and Yang, 2020] is a commonly used benchmark for NAS algorithm evaluation, which includes three datasets: a) CIFAR-10 [Krizhevsky et al., 2014], b) CIFAR-100 [Krizhevsky et al., 2014] and c) ImageNet-16 [Chrabaszcz et al., 2017] for image classification. Details on the datasets exist in Appendix F.1. Apart from that, we evaluate the proposed approach with some baselines including ResNet, RL based algorithm and so on.

Algorithm procedure: Our algorithm, called Eigen-NAS, also belongs in the train-free category. Eigen-NAS follows KNAS, which leverages the minimum eigenvalue of NTK to guide NAS. However, due to the $\mathcal{O}(N^3)$ time complexity of computing these eigenvalues, KNAS instead computes $\|\boldsymbol{K}\|_{\mathrm{F}}$. However, from the expression $\lambda_{\min}(\boldsymbol{K}) \leq \frac{1}{d}\sum_{i=1}^{N} K_{ii} \leq \|\boldsymbol{K}\|_{\mathrm{F}}$ we utilize the first inequality in Eigen-NAS to obtain a tighter (and more computationally efficient) bound to $\lambda_{\min}$. The computation cost of our method is $\mathcal{O}(N)$, which is less than computing the Frobenius norm ($\mathcal{O}(N^2)$). Sequentially, the top-k best candidates architectures are chosen in KNAS and our Eigen-NAS, and then the best architecture is chosen by the validation error, refer to the results in Table 2. Due to the page limit, the algorithm is located in Appendix F.

Results: The experimental results in Table 2 verify that Eigen-NAS guided by the proposed metric above achieves the best performance on both the CIFAR-100 and ImageNet-16 datasets, and competitive performance on CIFAR-10, outperforming KNAS in all three cases when $k = 20$ for both methods. Even when we consider a smaller $k = 5$, Eigen-NAS can outperform KNAS, which we attribute to the more precise minimum eigenvalue estimation.

6 Conclusion

In this work, we explore the relationship between the minimum eigenvalue of NTK and neural architecture search. We derive upper and lower bounds on the minimum eigenvalues of NTK for (in)finite residual networks under different mixtures of activation functions, and establish a connection between the minimum eigenvalues and the generalization properties of the special search space: activation function and skip connection search of NAS. Our theoretical results on various activation

functions and mixed activation cases can also be a tool for deep learning theory researchers to prove generic results rather than studying a single architecture, e.g., ReLU networks. In addition, we use the minimum eigenvalue as a guide for the training of NAS in a train-free method, which greatly exceeds the efficiency of the classic NAS algorithm. When compared with existing train-free methods, our algorithm, called Eigen-NAS, achieves a higher accuracy. We posit that this will be useful for studying computationally efficient methods on NAS.

A core limitation is whether our proof framework can cover more general structures in NAS, such as the most commonly used convolutional neural networks (CNNs). Even though this seems possible, this is non-trivial due to the tensors that emerge. To be specific, it requires the element-recursive form of NTK matrices in Arora et al. [2019b] to be transformed into a global-recursive form (similar to Lemma 1), then analyze its minimum eigenvalue. Besides, the contraction operation of tensors, the locality and boundary effects of convolutional layer in CNNs make the analysis difficult. Therefore, we believe this is a topic on its own right. Another limitation of our work is that it does not analyze the various algorithms proposed for searching through the search space. We believe that a deeper understanding of such algorithms, such as DARTS can provide further insights into how to design improved search spaces.

Acknowledgements

We are also thankful to the reviewers for providing constructive feedback. Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-19-1-0404. This work was supported by Hasler Foundation Program: Hasler Responsible AI (project number 21043). This work was supported by SNF project – Deep Optimisation of the Swiss National Science Foundation (SNSF) under grant number 200021_205011. This work was supported by Zeiss. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 725594 - time-data). Corresponding authors: Fanghui Liu and Zhenyu Zhu.

References

Z. Allen-Zhu, Y. Li, and Y. Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning (ICML)*, 2019.

S. Arora, S. Du, W. Hu, Z. Li, and R. Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning (ICML)*, 2019a.

S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019b.

P. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Y. Belfer, A. Geifman, M. Galun, and R. Basri. Spectral analysis of the neural tangent kernel for deep residual networks, 2021.

A. Bietti and F. Bach. Deep equals shallow for relu networks in kernel regimes. In *International Conference on Learning Representations (ICLR)*, 2021.

Y. Cao and Q. Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

W. Chen, X. Gong, and Z. Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. In *International Conference on Learning Representations (ICLR)*, 2021.

Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, and J. Sun. Detnas: Backbone search for object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Z. Chen, Y. Cao, Q. Gu, and T. Zhang. A generalized neural tangent kernel analysis for two-layer neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a.

Z. Chen, Y. Cao, D. Zou, and Q. Gu. How much over-parameterization is sufficient to learn deep ReLU networks? In *International Conference on Learning Representations (ICLR)*, 2020b.

P. Chrabaszcz, I. Loshchilov, and F. Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets, 2017.

J. de Dios and J. Bruna. On sparsity in overparametrised shallow relu networks, 2020.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

S. Ding, T. Chen, X. Gong, W. Zha, and Z. Wang. Autospeech: Neural architecture search for speaker recognition, 2020.

X. Dong and Y. Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. In *International Conference on Learning Representations (ICLR)*, 2020.

X. Dong, L. Liu, K. Musial, and B. Gabrys. Nats-bench: Benchmarking nas algorithms for architecture topology and size. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2021.

S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning (ICML)*, 2019a.

S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations (ICLR)*, 2019b.

T. Elsken, J. H. Metzen, and F. Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 2019.

N. Ghosh, S. Mei, and B. Yu. The three stages of learning dynamics in high-dimensional kernel methods. In *International Conference on Learning Representations (ICLR)*, 2022.

G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. 1996.

T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of Statistics*, 2022.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

K. Huang, Y. Wang, M. Tao, and T. Zhao. Why do deep residual networks generalize better than deep feedforward networks? – a neural tangent kernel perspective. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

A. Klimovsky. High-dimensional gaussian fields with isotropic increments seen through spin glasses. *Electronic Communications in Probability*, 2012.

A. Krizhevsky, V. Nair, and G. Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

L. Li and A. Talwalkar. Random search and reproducibility for neural architecture search. In *Uncertainty in Artificial Intelligence*, 2020.

C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019a.

H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu. Hierarchical representations for efficient architecture search. In *International Conference on Learning Representations (ICLR)*, 2018.

H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations (ICLR)*, 2019b.

E. Malach, G. Yehudai, S. Shalev-Schwartz, and O. Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning (ICML)*, 2020.

S. Mei, T. Misiakiewicz, and A. Montanari. Learning with invariances in random features and kernel models. In *Conference on Learning Theory (COLT)*, 2021.

J. Mellor, J. Turner, A. Storkey, and E. J. Crowley. Neural architecture search without training. In *International Conference on Machine Learning (ICML)*, 2021.

A. Montanari and Y. Zhong. The interpolation phase transition in neural networks: Memorization and generalization under lazy training, 2020.

Q. Nguyen. On the proof of global convergence of gradient descent for deep relu networks with linear widths. In *International Conference on Machine Learning (ICML)*, 2021.

Q. Nguyen, M. Mondelli, and G. F. Montufar. Tight bounds on the smallest eigenvalue of the neural tangent kernel for deep relu networks. In *International Conference on Machine Learning (ICML)*, 2021.

Q. N. Nguyen and M. Mondelli. Global convergence of deep networks with one wide layer followed by pyramidal topology. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

S. Oymak and M. Soltanolkotabi. Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory*, 2020.

S. Oymak, M. Li, and M. Soltanolkotabi. Generalization guarantees for neural architecture search with train-validation split. In *International Conference on Machine Learning (ICML)*, 2021.

I. Radosavovic, J. Johnson, S. Xie, W.-Y. Lo, and P. Dollár. On network design spaces for visual recognition. In *International Conference on Computer Vision (ICCV)*, 2019.

P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, 2021.

J. Schur. Bemerkungen zur theorie der beschränkten bilinearformen mit unendlich vielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 1911.

Y. Shu, W. Wang, and S. Cai. Understanding architectures learnt by cell-based neural architecture search. In *International Conference on Learning Representations (ICLR)*, 2020.

Y. Shu, S. Cai, Z. Dai, B. C. Ooi, and B. K. H. Low. NASI: Label- and data-agnostic neural architecture search at initialization. In *International Conference on Learning Representations (ICLR)*, 2022.

M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

T. Tirer, J. Bruna, and R. Giryes. Kernel-based smoothness analysis of residual networks. In *Mathematical and Scientific Machine Learning*, 2022.

R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. 2018.

B. Wu, X. Dai, P. Zhang, Y. Wang, F. Sun, Y. Wu, Y. Tian, P. Vajda, Y. Jia, and K. Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

J. Xu, L. Zhao, J. Lin, R. Gao, X. Sun, and H. Yang. Knas: Green neural architecture search. In *International Conference on Machine Learning (ICML)*, 2021.

Y. Xu, L. Xie, X. Zhang, X. Chen, G.-J. Qi, Q. Tian, and H. Xiong. Pc-darts: Partial channel connections for memory-efficient architecture search. In *International Conference on Learning Representations (ICLR)*, 2019.

P. Yaskov. Lower bounds on the smallest eigenvalue of a sample covariance matrix. *Electronic Communications in Probability*, 2014.

P. Ye, B. Li, Y. Li, T. Chen, J. Fan, and W. Ouyang. β-darts: Beta-decay regularization for differentiable architecture search. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter. Nas-bench-101: Towards reproducible neural architecture search. In *International Conference on Machine Learning (ICML)*, 2019.

Y. Zhang, Z. Qiu, J. Liu, T. Yao, D. Liu, and T. Mei. Customizable architecture search for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

P. Zhou, C. Xiong, R. Socher, and S. C. Hoi. Theory-inspired path-regularized differential network architecture search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2017.

B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

D. Zou, Y. Cao, D. Zhou, and Q. Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We clearly discuss the limitation of this work in the conclusion section.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We clearly discuss the societal impact of this work in the Appendix G.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] All of the assumptions are clearly stated and are well discussed.
   (b) Did you include complete proofs of all theoretical results? [Yes] All of the proofs can be found in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The datasets we use in this work are all in the public domain and standard for image-related tasks. Thus, our setup can be reproduced by interested practitioners. The code will be open-sourced upon the acceptance of the paper.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The data splits and our comparisons follow previous works, e.g. the experiment on sec. 5 follows KNAS code.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] All our experiments are conducted on a **single** GPU in our internal cluster.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite all the corresponding papers that provide the benchmarks/assets we use.
   (b) Did you mention the license of the assets? [Yes] All the datasets used in this work are publicly available datasets; in addition, they are quite popular benchmarks for diverse tasks. All the datasets enable their use for research purposes.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The datasets we utilize are publicly available and contain tens of thousands of images. We refer to the authors original papers describing the datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The datasets used do not contain offensive content or personally identifiable information.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix introduction

The Appendix is organized as follows:

- In Appendix A, we state the introductory notations and definitions.
- We prove Theorem 1 in Appendix B. We also provide the result when the residual network only has the same activation function.
- In Appendix C, we extend the results of infinitely width to finite width and provide the proof for them.
- In Appendix D, we prove Theorem 3.
- In Appendix E, we discussion some key points of the proof and the motivation of the analysis.
- In Appendix F, we detail our experimental settings, our Eigen-NAS algorithm as used in sec. 5.2. We conduct additional numerical validations.
- Finally, in Appendix G, we discuss the societal impact of this work.

A Background

A.1 Symbols and Notation

In the paper, vectors are indicated with bold small letters, matrices with bold capital letters. To facilitate the understanding of our work, we include the some core symbols and notation in Table 3.

Table 3: Core symbols and notations used in this project.

Symbol	Dimension(s)	Definition
$\mathcal{N}(\mu, \sigma)$	-	Gaussian distribution of mean μ and variance σ
$1\{A\}$	-	Indicator function for event A
$[L]$	-	Shorthand of $\{1, 2, \ldots, L\}$
$\oplus$	-	Direct sum
$\mathcal{O}$, o, Ω and Θ	-	Standard Bachmann–Landau order notation
$\circ$	-	Element-wise hadamard product
$\|\boldsymbol{v}\|_2$	-	Euclidean norms of vectors $\boldsymbol{v}$
$\|\boldsymbol{M}\|_2$	-	Spectral norms of matrices $\boldsymbol{M}$
$\|\boldsymbol{M}\|_{\mathrm{F}}$	-	Frobenius norms of matrices $\boldsymbol{M}$
$\|\boldsymbol{M}\|_*$	-	Nuclear norms of matrices $\boldsymbol{M}$
$\lambda(\boldsymbol{M})$	-	Eigenvalues of matrices $\boldsymbol{M}$
$\boldsymbol{M}^{[l]}$	-	l-th row of matrices $\boldsymbol{M}$
$\boldsymbol{M}_{i,j}$	-	(i, j)-th element of matrices $\boldsymbol{M}$
N	-	Size of the dataset
d	-	Input size of the network
L	-	Depth of the network
m	-	Width of intermediate layer
α_l	$\mathbb{R}$	A binary variable measures whether there is a skip connection in the l-th layer
σ_l	-	The activation function of l-th layer
$\beta_1, \beta_2, \beta_3$	$\mathbb{R}, \mathbb{R}, \mathbb{R}$	Three constants defined in Table 1
$\mu_i(\sigma)$	$\mathbb{R}$	The i-th Hermite coefficient of the activation function σ
$\boldsymbol{x}_i$	$\mathbb{R}^d$	The i-th data point
y_i	$\mathbb{R}$	The i-th target vector
$\boldsymbol{W}_1$	$\mathbb{R}^{m \times d}$	Weight matrix for the input layer
$\boldsymbol{W}_l$	$\mathbb{R}^{m \times m}$	Weight matrix for the l-th hidden layer
$\boldsymbol{W}_L$	$\mathbb{R}^{1 \times m}$	Weight matrix for the output layer

A.1.1 Feature map

Here we define the core notation about feature maps that are required in the proof. Firstly, we define ω-neighborhood to describe the difference between two matrices.

For any $\boldsymbol{W} \in \mathcal{W}$, we define its ω-neighborhood as follows:

Definition 1 (ω-neighborhood)**.**

$$\mathcal{B}(\boldsymbol{W}, \omega) := \{\boldsymbol{W}' \in \mathcal{W} : \|\boldsymbol{W}_l' - \boldsymbol{W}_l\|_{\mathrm{F}} \leq \omega, \boldsymbol{\alpha}' = \boldsymbol{\alpha}, \boldsymbol{\sigma}' = \boldsymbol{\sigma}, l \in [L]\} \ .$$

Then we define $(\boldsymbol{D}_l)_{k,k} = \sigma_l{}'((\boldsymbol{W}_l \boldsymbol{f}_{l-1})_k)$ as the back-propagation matrix of the activation function. We use the notation $\widetilde{\boldsymbol{W}} \in \mathcal{B}(\boldsymbol{W}, \omega)$ to describe the relationship of the two matrices have ω-neighborhood relationship.

In addition, we define the feature map of network and its perturbing matrix as follows:

Definition 2.

$$\begin{aligned}
\widetilde{\boldsymbol{g}}_{i,1} &= \widetilde{\boldsymbol{W}}_1 \boldsymbol{x}_i \,, & \boldsymbol{g}_{i,1} &= \boldsymbol{W}_1 \boldsymbol{x}_i \,, & &for\ i \in [N], \\
\widetilde{\boldsymbol{f}}_{i,1} &= \sigma_1(\widetilde{\boldsymbol{W}}_1 \boldsymbol{x}_i) \,, & \boldsymbol{f}_{i,1} &= \sigma_1(\boldsymbol{W}_1 \boldsymbol{x}_i) \,, & &for\ i \in [N], \\
\widetilde{\boldsymbol{g}}_{i,l} &= \widetilde{\boldsymbol{W}}_l \widetilde{\boldsymbol{f}}_{i,l-1} \,, & \boldsymbol{g}_{i,l} &= \boldsymbol{W}_l \boldsymbol{f}_{i,l-1} \,, & &for\ i \in [N] \text{ and } l = 2, \ldots, L-1, \\
\widetilde{\boldsymbol{f}}_{i,l} &= \sigma_l(\widetilde{\boldsymbol{W}}_l \widetilde{\boldsymbol{f}}_{i,l-1}) + \alpha_{l-1} \widetilde{\boldsymbol{f}}_{i,l-1} \,, & \boldsymbol{f}_{i,l} &= \sigma_l(\boldsymbol{W}_l \boldsymbol{f}_{i,l-1}) + \alpha_{l-1} \boldsymbol{f}_{i,l-1} \,, & &for\ i \in [N] \text{ and } l = 2, \ldots, L-1 \,.
\end{aligned}$$

Let us define diagonal matrices $\widetilde{\boldsymbol{D}}_{i,l} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{D}_{i,l} \in \mathbb{R}^{m \times m}$ by letting $(\widetilde{\boldsymbol{D}}_{i,l})_{k,k} = \sigma_l{}'((\widetilde{\boldsymbol{g}}_{i,l})_k)$ and $(\boldsymbol{D}_{i,l})_{k,k} = \sigma_l{}'((\boldsymbol{g}_{i,l})_k), \forall k \in [m]$. Accordingly, we let $\hat{\boldsymbol{g}}_{i,l} = \widetilde{\boldsymbol{g}}_{i,l} - \boldsymbol{g}_{i,l}$, $\hat{\boldsymbol{f}}_{i,l} = \widetilde{\boldsymbol{f}}_{i,l} - \boldsymbol{f}_{i,l}$ and diagonal matrix $\hat{\boldsymbol{D}}_{i,l} = \widetilde{\boldsymbol{D}}_{i,l} - \boldsymbol{D}_{i,l}$.

A.1.2 Other notations

For the Hadamard product of the matrices $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_r$ that share the same dimensions, we use the following abbreviation:

$$\bigcirc_{i=1}^{r} (\boldsymbol{X}_i) = \boldsymbol{X}_1 \circ \boldsymbol{X}_2 \circ \cdots \circ \boldsymbol{X}_r \,.$$

B The bound of the minimum eigenvalues of NTK for infinite width

We present the details of our results on sec. 4.3 in this section. Firstly, we provide the proof of Theorem 1 in Appendix B.2. Then in Appendix B.3 we provide the result when several activation functions exist alone.

B.1 Proof of Lemma 1

Our proof mainly follows the results of Huang et al. [2020], but due to the different network structures, the proof process and results are slightly different. Moreover, we provide a matrix version results, which Huang et al. [2020] does not contain. For self-completeness, we include the proof here.

Proof. By Huang et al. [2020, Proposition 3], written as matrix form,we have:

$$\begin{aligned}
\boldsymbol{A}^{(1)} &= \boldsymbol{X}\boldsymbol{X}^\top \,, \\
\boldsymbol{A}^{(2)} &= 2\mathbb{E}_{\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I}_d)}[\sigma_1(\boldsymbol{X}\boldsymbol{w})\sigma_1(\boldsymbol{X}\boldsymbol{w})^\top] \,, \\
\boldsymbol{A}^{(l)} &= 2\mathbb{E}_{\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I}_N)}[\sigma_{l-1}(\sqrt{\boldsymbol{A}^{(l-1)}}\boldsymbol{w})\sigma_{l-1}(\sqrt{\boldsymbol{A}^{(l-1)}}\boldsymbol{w})^\top] + \alpha_{l-2}\boldsymbol{A}^{(l-1)} \,.
\end{aligned}$$

Note that, it is slightly different from the original result because the network structure is slightly different.

Let $\boldsymbol{G}^{(1)} = \boldsymbol{A}^{(1)}$, $\boldsymbol{G}^{(2)} = \boldsymbol{A}^{(2)}$ and $\boldsymbol{G}^{(l)} = 2\mathbb{E}_{\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I}_N)}[\sigma_{l-1}(\sqrt{\boldsymbol{A}^{(l-1)}}\boldsymbol{w})\sigma_{l-1}(\sqrt{\boldsymbol{A}^{(l-1)}}\boldsymbol{w})^\top]$, then we have:

$$\begin{aligned}
\boldsymbol{A}^{(1)} &= \boldsymbol{G}^{(1)} = \boldsymbol{X}\boldsymbol{X}^\top \,, \\
\boldsymbol{A}^{(2)} &= \boldsymbol{G}^{(2)} = 2\mathbb{E}_{\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I}_d)}[\sigma_1(\boldsymbol{X}\boldsymbol{w})\sigma_1(\boldsymbol{X}\boldsymbol{w})^\top] \,, \\
\boldsymbol{G}^{(l)} &= 2\mathbb{E}_{\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I}_N)}[\sigma_{l-1}(\sqrt{\boldsymbol{A}^{(l-1)}}\boldsymbol{w})\sigma_{l-1}(\sqrt{\boldsymbol{A}^{(l-1)}}\boldsymbol{w})^\top] \,, \\
\boldsymbol{A}^{(l)} &= \boldsymbol{G}^{(l)} + \alpha_{l-2}\boldsymbol{A}^{(l-1)} \,.
\end{aligned}$$

According to Huang et al. [2020, Proposition 4], written as matrix form,we have:

$$\boldsymbol{K}^{(L)} = \sum_{l=1}^{L} \boldsymbol{G}^{(l)} \circ \dot{\boldsymbol{G}}^{(l+1)} \circ (\dot{\boldsymbol{G}}^{(l+2)} + \alpha_l \mathbf{1}_{N\times N}) \circ \cdots \circ (\dot{\boldsymbol{G}}^{(L)} + \alpha_{L-2}\mathbf{1}_{N\times N}),$$

where the $\dot{\boldsymbol{G}}^{(s)}$ satisfy that $\dot{\boldsymbol{G}}^{(s)} = 2\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\mathbf{0},\mathbb{I}_N)}[\sigma'_{s-1}(\sqrt{\boldsymbol{A}^{(s-1)}}\boldsymbol{w})\sigma'_{s-1}(\sqrt{\boldsymbol{A}^{(s-1)}}\boldsymbol{w})^\top]$. Combining the above results, we finish the proof.

□

B.2 Proof of Theorem 1

In this part, we present the proof of Theorem 1. Differently from Oymak and Soltanolkotabi [2020], our result allows for activation functions search in each layer.

Before we prove Theorem 1, we provide some propositions that are helpful to our proof. To facilitate the writing of the proof, let $\alpha_0 := 0$.

Proposition 1. *When σ_1 is Tanh, the remaining layers are with LeakyReLU and for $l \in [L-2]$, $\alpha_l = 1$, the quantity $G_{ii}^{(l)}$ has the largest upper bound:*

$$G_{ii}^{(l)} \leq \begin{cases} 1 & \text{if } l = 1 \\ 2(2+\eta^2)^{l-2} & \text{if } l \geq 2\,. \end{cases} \tag{3}$$

We set $G_{\max} = 2(2+\eta^2)^{L-2}$ as the upper bound of $G_{ii}^{(L)}$.

Proof. To prove our result, we need bound $\boldsymbol{G}^{(l)}$ under different activation functions. We summarize them as below.

When σ_{l-1} is ReLU:

$$\begin{aligned} G_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}(w)^2] = \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \max(0,x)^2 \mathrm{d}x \\ &= \int_0^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} x^2 \mathrm{d}x \\ &= A_{ii}^{(l-1)}\,. \end{aligned} \tag{4}$$

When σ_{l-1} is LeakyReLU:

$$\begin{aligned} G_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}(w)^2] = \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \max(\eta x,x)^2 \mathrm{d}x \\ &= \int_0^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} x^2 \mathrm{d}x + \int_{-\infty}^{0} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \eta^2 x^2 \mathrm{d}x \\ &= (1+\eta^2) A_{ii}^{(l-1)}\,. \end{aligned} \tag{5}$$

When σ_{l-1} is Sigmoid:

$$\begin{aligned} G_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}(w)^2] = \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} f_{\text{Sigmoid}}(x)^2 \mathrm{d}x \\ &\leq \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} (\frac{1}{2})^2 \mathrm{d}x \\ &= \frac{1}{2}\,. \end{aligned} \tag{6}$$

σ_{l-1}	ReLU	LeakyReLU	Sigmoid	Tanh	Swish
Upper bound	1	$1+\eta^2$	$\frac{1}{8}$	2	1
Lower bound	1	$1+\eta^2$	$(\frac{1}{2}-\frac{1}{2\sqrt{1+\frac{G_{\max}}{4}}})\frac{1}{G_{\max}}$	$(2-\frac{2}{\sqrt{1+G_{\max}}})\frac{1}{G_{\max}}$	$\frac{1}{2}$

Table 4: Upper and lower bound for $A_{ii}^{(l)}/A_{ii}^{(l-1)}-\alpha_{l-2}$ for different activation functions σ_{l-1} and the binary variable $\alpha_{l-2}\in\{0,1\}$ indicates whether $(l-1)$-th layer has a skip connection or not.

When σ_{l-1} is Tanh:

$$\begin{aligned} G_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}(w)^2] = \int_{-\infty}^{\infty}\frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}}e^{-\frac{x^2}{2A_{ii}^{(l-1)}}}f_{\text{Tanh}}(x)^2\mathrm{d}x \\ &\leq \int_{-\infty}^{\infty}\frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}}e^{-\frac{x^2}{2A_{ii}^{(l-1)}}}\mathrm{d}x \\ &= 2\,. \end{aligned} \tag{7}$$

When σ_{l-1} is Swish:

$$\begin{aligned} G_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}(w)^2] = \int_{-\infty}^{\infty}\frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}}e^{-\frac{x^2}{2A_{ii}^{(l-1)}}}f_{\text{Swish}}(x)^2\mathrm{d}x \\ &= \int_{-\infty}^{\infty}\frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}}e^{-\frac{x^2}{2A_{ii}^{(l-1)}}}\frac{x^2}{(1+e^{-x})^2}\mathrm{d}x \\ &= \int_{0}^{\infty}\frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}}e^{-\frac{x^2}{2A_{ii}^{(l-1)}}}x^2\times\left(\frac{1}{(1+e^{-x})^2}+\frac{1}{(1+e^{x})^2}\right)\mathrm{d}x \\ &\leq \int_{0}^{\infty}\frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}}e^{-\frac{x^2}{2A_{ii}^{(l-1)}}}x^2\mathrm{d}x \\ &= A_{ii}^{(l-1)}\,. \end{aligned} \tag{8}$$

Combining Eqs. (4), (5), (6), (7) and (8) with Lemma 1 we draw the conclusion and finish the proof. □

Proposition 2. *The relationship between $A_{ii}^{(l)}$ and $A_{ii}^{(l-1)}$ for different activation functions can be summarized as Table 4 according to the difference of σ_{l-1}.*

Proof. To prove our result, we need to bound the ratio $A_{ii}^{(l)}/A_{ii}^{(l-1)}$ for different activation functions. We illustrate how this is achieved in different cases below:

For $l\geq 2$:

When σ_{l-1} is ReLU by Eq. (4) we have:

$$A_{ii}^{(l)} = G_{ii}^{(l)}+\alpha_{l-2}A_{ii}^{(l-1)} = (1+\alpha_{l-2})A_{ii}^{(l-1)}\,. \tag{9}$$

When σ_{l-1} is LeakyReLU by Eq. (5) we have:

$$A_{ii}^{(l)} = G_{ii}^{(l)}+\alpha_{l-2}A_{ii}^{(l-1)} = (1+\alpha_{l-2}+\eta^2)A_{ii}^{(l-1)}\,. \tag{10}$$

When σ_{l-1} is Swish, $G_{ii}^{(l)}$ can be upper by Eq. (8) and lower bounded by:

$$
\begin{aligned}
G_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}(w)^2] = \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} f_{\text{Swish}}(x)^2 \mathrm{d}x \\
&= \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \frac{x^2}{(1+e^{-x})^2} \mathrm{d}x \\
&= \int_{0}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} x^2 \times \left(\frac{1}{(1+e^{-x})^2} + \frac{1}{(1+e^{x})^2} \right) \mathrm{d}x \\
&\geq \int_{0}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} x^2 \times \frac{1}{2} \mathrm{d}x \\
&= \frac{1}{2} A_{ii}^{(l-1)} ,
\end{aligned}
\tag{11}
$$

which implies:

$$
\left(\frac{1}{2} + \alpha_{l-2} \right) A_{ii}^{(l-1)} \leq A_{ii}^{(l)} \leq (1+\alpha_{l-2}) A_{ii}^{(l-1)} . \tag{12}
$$

When σ_{l-1} is Sigmoid, $G_{ii}^{(l)}$ can be upper by:

$$
\begin{aligned}
G_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}(w)^2] = \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} f_{\text{Sigmoid}}(x)^2 \mathrm{d}x \\
&\leq \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \left(\frac{1}{4} - e^{-\frac{x^2}{4}} \right) \mathrm{d}x = \frac{1}{2} - \frac{1}{2\sqrt{1+\frac{A_{ii}^{(l-1)}}{2}}} \\
&\leq \frac{A_{ii}^{(l-1)}}{8} , \qquad \text{holds for } x \geq 0 .
\end{aligned}
\tag{13}
$$

Then $G_{ii}^{(l)}$ can be lower bounded by:

$$
\begin{aligned}
G_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}(w)^2] = \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} f_{\text{Sigmoid}}(x)^2 \mathrm{d}x \\
&\geq \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \left(\frac{1}{4} - e^{-\frac{x^2}{8}} \right) \mathrm{d}x \\
&= \frac{1}{2} - \frac{1}{2\sqrt{1+\frac{A_{ii}^{(l-1)}}{4}}} \\
&\geq \left(\frac{1}{2} - \frac{1}{2\sqrt{1+\frac{G_{\max}}{4}}} \right) \frac{A_{ii}^{(l-1)}}{G_{\max}} ,
\end{aligned}
\tag{14}
$$

where we use the fact that the penultimate line is a concave function with respect to $A_{ii}^{(l-1)}$. When $A_{ii}^{(l-1)} = 0$, the function value is 0. That means $G_{ii}^{(l)}/A_{ii}^{(l-1)}$ obtains the minimum value at $G_{ii}^{(l)} = G_{\max}$. Combined with Eq. (3), we get the last inequality.

Then, we have:

$$
\left(\left[\frac{1}{2} - \frac{1}{2\sqrt{1+\frac{G_{\max}}{4}}} \right] \frac{1}{G_{\max}} + \alpha_{l-2} \right) A_{ii}^{(l-1)} \leq A_{ii}^{(l)} \leq \left(\frac{1}{8} + \alpha_{l-2} \right) A_{ii}^{(l-1)} . \tag{15}
$$

σ_{l-1}	ReLU	LeakyReLU	Sigmoid	Tanh	Swish
Upper bound for $\dot{G}_{ii}^{(l)}$	1	$1+\eta^2$	$1/8$	2	1.22
Lower bound for $\dot{G}_{ii}^{(l)}$	1	$1+\eta^2$	$f_{\mathrm{S}}(G_{\max})$	$f_{\mathrm{T}}(G_{\max})$	$1/2$

Table 5: Upper and lower bound for $\dot{G}_{ii}^{(l)}$ for different activation function σ_{l-1}.

When σ_{l-1} is Tanh, $G_{ii}^{(l)}$ can be upper bounded by:

$$\begin{aligned} G_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}(w)^2] = \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} f_{\mathrm{Tanh}}(x)^2 \mathrm{d}x \\ &\leq \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} (1-e^{-x^2}) \mathrm{d}x = 2 - \frac{2}{\sqrt{1+2A_{ii}^{(l-1)}}} \\ &\leq 2A_{ii}^{(l-1)}\,, \qquad \text{holds for } x \geq 0\,. \end{aligned} \tag{16}$$

The lower bound is:

$$\begin{aligned} G_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}(w)^2] = \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} f_{\mathrm{Tanh}}(x)^2 \mathrm{d}x \\ &\geq \int_{-\infty}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} (1-e^{-\frac{x^2}{2}}) \mathrm{d}x \\ &= 2 - \frac{2}{\sqrt{1+A_{ii}^{(l-1)}}} \\ &\geq \left(2 - \frac{2}{\sqrt{1+G_{\max}}}\right) \frac{A_{ii}^{(l-1)}}{G_{\max}}\,. \end{aligned} \tag{17}$$

Similar to the Sigmoid, the penultimate line is an concave function with respect to $A_{ii}^{(l-1)}$. When $A_{ii}^{(l-1)} = 0$, the function value is 0. That means $G_{ii}^{(l)}/A_{ii}^{(l-1)}$ obtains the minimum value at $G_{ii}^{(l)} = G_{\max}$. Combined with Eq. (3), we get the last inequality.

Then, we have:

$$\left(\left[2 - \frac{2}{\sqrt{1+G_{\max}}}\right] \frac{1}{G_{\max}} + \alpha_{l-2}\right) A_{ii}^{(l-1)} \leq A_{ii}^{(l)} \leq (2+\alpha_{l-2}) A_{ii}^{(l-1)}\,. \tag{18}$$

According to Eqs. (9), (10), (12), (15) and (18), we can summarized the results about bound of $A_{ii}^{(l)}/A_{ii}^{(l-1)}$ in Table 4.

□

Proposition 3. *The bound of $\dot{G}_{ii}^{(l)}$ with respect to different activation function σ_{l-1} can be summarized in Table 5.*

Proof. To prove our result, we need to bound $\dot{G}_{ii}^{(l)}$ with respect to different activation function σ_{l-1} as follows.

When σ_{l-1} is ReLU:

$$\begin{aligned} \dot{G}_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma'_{l-1}(w)^2] = \int_{0}^{\infty} \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \mathrm{d}x \\ &= 1\,. \end{aligned} \tag{19}$$

When σ_{l-1} is LeakyReLU:

$$\begin{aligned}\dot{G}_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}'(w)^2] \\ &= \int_0^\infty \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \mathrm{d}x + \int_{-\infty}^0 \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \eta^2 \mathrm{d}x \\ &= 1+\eta^2 \,.\end{aligned} \tag{20}$$

When σ_{l-1} is Sigmoid, according to the monotonicity of the f_{S}, we have:

$$f_{\mathrm{S}}(G_{\max}) \le \dot{G}_{ii}^{(l)} = 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}'(w)^2] = f_{\mathrm{S}}(A_{ii}^{(l-1)}) \le f_{\mathrm{S}}(0) \le \frac{1}{8} \,. \tag{21}$$

When σ_{l-1} is Tanh, according to the monotonicity of the f_{T}, we have:

$$f_{\mathrm{T}}(G_{\max}) \le \dot{G}_{ii}^{(l)} = 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}'(w)^2] = f_{\mathrm{T}}(A_{ii}^{(l-1)}) \le f_{\mathrm{T}}(0) \le 2 \,. \tag{22}$$

When σ_{l-1} is Swish, The quantity $\dot{G}_{ii}^{(l)}$ can be upper bounded by:

$$\begin{aligned}\dot{G}_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}'(w)^2] = \int_{-\infty}^\infty \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} f_{\mathrm{Swish}}'(x)^2 \mathrm{d}x \\ &\le \int_0^\infty \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \left(\sup_x f_{\mathrm{Swish}}'(x)\right)^2 \mathrm{d}x \\ &\quad + \int_{-\infty}^0 \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \left(\inf_x f_{\mathrm{Swish}}'(x)\right)^2 \mathrm{d}x \\ &= \left(\inf_x f_{\mathrm{Swish}}'(x)\right)^2 + \left(\sup_x f_{\mathrm{Swish}}'(x)\right)^2 \\ &\le 1.22 \,,\end{aligned} \tag{23}$$

where the last inequality holds by $1.099 < \sup_x f_{\mathrm{Swish}}'(x) < 1.1$ and $-0.1 < \inf_x f_{\mathrm{Swish}}'(x) < -0.099$.

Then the quantity $\dot{G}_{ii}^{(l)}$ can be lower bounded by:

$$\begin{aligned}\dot{G}_{ii}^{(l)} &= 2\mathbb{E}_{w\sim\mathcal{N}(0,A_{ii}^{(l-1)})}[\sigma_{l-1}'(w)^2] = \int_{-\infty}^\infty \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} f_{\mathrm{Swish}}'(x)^2 \mathrm{d}x \\ &= \int_0^\infty \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \left(f_{\mathrm{Swish}}'(x)^2 + f_{\mathrm{Swish}}'(-x)^2\right) \mathrm{d}x \\ &\ge \int_0^\infty \frac{2}{\sqrt{2\pi A_{ii}^{(l-1)}}} e^{-\frac{x^2}{2A_{ii}^{(l-1)}}} \frac{1}{2} \mathrm{d}x \\ &= \frac{1}{2} \,.\end{aligned} \tag{24}$$

Combining Eqs. (19), (20), (21), (22), (23) to (24), we can summarize the results about bound of $\dot{G}_{ii}^{(l)}$ in Table 5.

□

Proposition 4. *Let $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ be an matrix that every row $\boldsymbol{x}_i$ sampled from data distribution $\mathcal{D}_X$. when $N \ge \Omega(d^4)$, with probability at least $1-e^{-d}$, we have*

$$\lambda_{\min}(\boldsymbol{X}^\top \boldsymbol{X}) \ge \frac{N}{d} - 9N^{2/3}d^{1/3} = \Theta(N/d) \,.$$

Proof. Firstly, we can compute that, for any $\alpha \in (0, 2]$,

$$\sup_{\|\boldsymbol{v}\|_2=1} \mathbb{E}\left|\left\langle \sqrt{d}\boldsymbol{x}_i, \boldsymbol{v}\right\rangle\right|^{2+\alpha} = \left\|\sqrt{d}\boldsymbol{x}_i\right\|_2^{2+\alpha} = d^{(2+\alpha)/2} < \infty\,.$$

Then from Assumption 3, we have $\sqrt{d}\boldsymbol{x}_i$ is an isotropic random vector in $\mathbb{R}^d$.

According to Yaskov [2014, Corollary 3.1], if we choose $\alpha = 1$, then with probability at least $1 - e^{-d}$ we have:

$$\begin{aligned}
\lambda_{\min}(\frac{d}{N}\boldsymbol{X}^\top\boldsymbol{X}) &\geq 1 - C_\alpha(\frac{d}{N})^{1/3} \\
&= 1 - 9(L_p(\alpha))^{2/(2+\alpha)}(\frac{d}{N})^{1/3} \\
&= 1 - 9(d^{(2+\alpha)/2})^{2/(2+\alpha)}(\frac{d}{N})^{1/3} \\
&= 1 - \frac{9d^{4/3}}{N^{1/3}}\,.
\end{aligned}$$

That is, with probability at least $1 - e^{-d}$ we have:

$$\lambda_{\min}(\boldsymbol{X}^\top\boldsymbol{X}) \geq \frac{N}{d} - 9N^{2/3}d^{1/3} = \Theta(N/d)\,.$$

□

Now we are ready to prove Theorem 1.

Proof of Theorem 1. Now we are ready to present the estimation on $\lambda_{\min}(\boldsymbol{K}^{(L)})$ as below.

From Lemma 1, we have the NTK formulation for residual neural networks:

$$\boldsymbol{K}^{(L)} = \boldsymbol{G}^{(L)} + \sum_{l=1}^{L-1} \boldsymbol{G}^{(l)} \circ (\dot{\boldsymbol{G}}^{(l+1)} + \alpha_l \mathbf{1}_{N\times N}) \circ \cdots \circ (\dot{\boldsymbol{G}}^{(L)} + \alpha_{L-1}\mathbf{1}_{N\times N})\,.$$

$$\boldsymbol{K}^{(L)} = \boldsymbol{G}^{(L)} + \sum_{l=1}^{L-1} \boldsymbol{G}^{(l)} \circ \dot{\boldsymbol{G}}^{(l+1)} \circ (\dot{\boldsymbol{G}}^{(l+2)} + \alpha_l \mathbf{1}_{N\times N}) \circ \cdots \circ (\dot{\boldsymbol{G}}^{(L)} + \alpha_{L-2}\mathbf{1}_{N\times N})\,.$$

It is clear that all the matrices $\boldsymbol{G}^{(l)}$, $\dot{\boldsymbol{G}}^{(l)}$ are positive semi-definite (PSD), then $(\dot{\boldsymbol{G}}^{(l+1)} + \alpha_l \mathbf{1}_{N\times N})$ are also PSD. For two PSD matrices $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{R}^{N\times N}$, it holds $\lambda_{\min}(\boldsymbol{P} \circ \boldsymbol{Q}) \geq \lambda_{\min}(\boldsymbol{P}) \min_{i\in[N]} Q_{ii}$ [Schur, 1911]. Accordingly, we have:

$$\lambda_{\min}(\boldsymbol{K}^{(L)}) \geq \sum_{l=1}^{L} \lambda_{\min}(\boldsymbol{G}^{(l)}) \min_{i\in[N]} \prod_{p=l+1}^{L} \left(\dot{G}_{ii}^{(p)} + \alpha_{p-2}\right).$$

Then we bound $\lambda_{\min}(\boldsymbol{G}^{(2)})$:

$$\begin{aligned}
\lambda_{\min}(\boldsymbol{G}^{(2)}) &= \lambda_{\min}\left(2\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(0,\mathbb{I}_d)}[\sigma_1(\boldsymbol{X}\boldsymbol{w})\sigma_1(\boldsymbol{X}\boldsymbol{w})^\top]\right) \\
&= 2\lambda_{\min}\left(\sum_{s=0}^{\infty} \mu_s(\sigma_1)^2 \bigcirc_{i=1}^{s} (\boldsymbol{X}\boldsymbol{X}^\top)\right) \quad \text{[Nguyen et al., 2021, Lemma D.3]} \\
&\geq 2\mu_1(\sigma_1)^2\lambda_{\min}(\boldsymbol{X}\boldsymbol{X}^\top)\,.
\end{aligned}$$

As a reminder, the symbol $\bigcirc$ denotes the Hadamard product, which is defined in Appendix A.1.2.

We know that when $N > d$, $\boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{X}^\top\boldsymbol{X}$ have the same non-zero eigenvalues. Then according to Proposition 4, with probability at least $1 - e^{-d}$ we have:

$$\lambda_{\min}(\boldsymbol{G}^{(2)}) \geq 2\mu_1(\sigma_1)^2(\frac{N}{d} - 9N^{2/3}d^{1/3}) = 2\mu_1(\sigma_1)^2\Theta(N/d)\,.$$

That means, with probability at least $1 - e^{-d}$ we have the lower bound of $\lambda_{\min}(\boldsymbol{K}^{(L)})$:

$$\begin{aligned}\lambda_{\min}(\boldsymbol{K}^{(L)}) &\geq \sum_{l=1}^{L}\lambda_{\min}(\boldsymbol{G}^{(l)})\min_{i\in[N]}\prod_{p=l+1}^{L}(\dot{G}_{ii}^{(p)} + \alpha_{p-2}) \geq \lambda_{\min}(\boldsymbol{G}^{(2)})\min_{i\in[N]}\prod_{p=3}^{L}(\dot{G}_{ii}^{(p)} + \alpha_{p-2}) \\ &\geq 2\mu_1(\sigma_1)^2\Theta(N/d)\min_{i\in[N]}\prod_{p=3}^{L}(\dot{G}_{ii}^{(p)} + \alpha_{p-2})\,.\end{aligned} \tag{25}$$

According to Table 1 and Table 5, we have:

$$\prod_{p=l+1}^{L}(\dot{G}_{ii}^{(p)} + \alpha_{p-2}) \leq \prod_{p=l+1}^{L}\Big(\beta_2(\sigma_{p-1}) + \alpha_{p-2}\Big)\,, \tag{26}$$

$$\prod_{p=3}^{L}(\dot{G}_{ii}^{(p)} + \alpha_{p-2}) \geq \prod_{p=3}^{L}\Big(\beta_3(\sigma_{p-1}) + \alpha_{p-2}\Big)\,. \tag{27}$$

According to the previous analysis, we know that $\boldsymbol{K}^{(L)}$ has d non-zero eigenvalues. Their sum is equal to the trace of $\boldsymbol{K}^{(L)}$. The upper bound of $\lambda_{\min}(\boldsymbol{K}^{(L)})$ is directly given by:

$$\lambda_{\min}(\boldsymbol{K}^{(L)}) \leq \frac{1}{d}\sum_{i=1}^{N}\sum_{l=1}^{L}(G_{ii}^{(l)})\prod_{p=l+1}^{L}(\dot{G}_{ii}^{(p)} + \alpha_{p-2})\,. \tag{28}$$

Final result - upper bound

For $G_{ii}^{(l)}$ we have the following bound:

$$G_{ii}^{(l)} = \beta_1(\sigma_{l-1})A_{ii}^{(l-1)} \leq \beta_1(\sigma_{l-1})\prod_{p=3}^{l-1}\Big(\beta_1(\sigma_{p-1}) + \alpha_{p-2}\Big)A_{ii}^{(2)} \leq \beta_1(\sigma_{l-1})\prod_{p=2}^{l-1}\Big(\beta_1(\sigma_{p-1}) + \alpha_{p-2}\Big)\,. \tag{29}$$

By Eqs. (26), (28) and (29), we have:

$$\lambda_{\min}(\boldsymbol{K}^{(L)}) \leq \frac{N}{d}\sum_{l=1}^{L}\Bigg(\beta_1(\sigma_{l-1})\prod_{p=2}^{l-1}\Big[\beta_1(\sigma_{p-1}) + \alpha_{p-2}\Big]\prod_{p=l+1}^{L}\Big[\beta_2(\sigma_{p-1}) + \alpha_{p-2}\Big]\Bigg)\,. \tag{30}$$

Final result - lower bound

By Eq. (25) and Eq. (27), with probability at least $1 - e^{-d}$, we have:

$$\lambda_{\min}(\boldsymbol{K}^{(L)}) \geq 2\mu_1(\sigma_1)^2\Theta(N/d)\prod_{p=3}^{L}\Big(\beta_3(\sigma_{p-1}) + \alpha_{p-2}\Big)\,.$$

□

B.3 Special cases

To provide further insights into our proofs of mixed activation functions as provided in the previous sections, we now consider the special case of a single activation function in each layer.

Corollary 4. *Under Assumption 1, 3, for a deep fully-connected ResNet with the same activation functions in every layer and for a not very large L, let $\boldsymbol{K}^{(L)}$ be the limiting NTK recursively defined in Lemma 1. Then, with probability at least $1 - e^{-d}$, we have:*

For ReLU:

$$2\mu_1(\sigma_1)^2\Theta(N/d)\prod_{p=3}^{L}(1+\alpha_{p-2}) \leq \lambda_{\min}(\boldsymbol{K}^{(L)}) \leq \frac{N}{d}\sum_{l=1}^{L}\left(\frac{\prod_{p=2}^{L}(1+\alpha_{p-2})}{1+\alpha_{l-2}}\right).$$

For LeakyReLU:

$$2\mu_1(\sigma_1)^2\Theta(N/d)\prod_{p=3}^{L}(1+\eta^2+\alpha_{p-2}) \leq \lambda_{\min}(\boldsymbol{K}^{(L)}) \leq \frac{N}{d}(1+\eta^2)\sum_{l=1}^{L}\left(\frac{\prod_{p=2}^{L}(1+\eta^2+\alpha_{p-2})}{1+\eta^2+\alpha_{l-2}}\right).$$

For Sigmoid:

$$2\mu_1(\sigma_1)^2\Theta(N/d)\prod_{p=3}^{L}(f_{\mathrm{S}}(\frac{1}{2})+\alpha_{p-2}) \leq \lambda_{\min}(\boldsymbol{K}^{(L)}) \leq \frac{N}{8d}\sum_{l=1}^{L}\left(\frac{\prod_{p=2}^{L}(\frac{1}{8}+\alpha_{p-2})}{\frac{1}{8}+\alpha_{l-2}}\right). \quad (31)$$

For Tanh:

$$2\mu_1(\sigma_1)^2\Theta(N/d)\prod_{p=3}^{L}(f_{\mathrm{T}}(2)+\alpha_{p-2}) \leq \lambda_{\min}(\boldsymbol{K}^{(L)}) \leq \frac{2N}{d}\sum_{l=1}^{L}\left(\frac{\prod_{p=2}^{L}(2+\alpha_{p-2})}{2+\alpha_{l-2}}\right). \quad (32)$$

For Swish:

$$2\mu_1(\sigma_1)^2\Theta(N/d)\prod_{p=3}^{L}(\frac{1}{2}+\alpha_{p-2}) \leq \lambda_{\min}(\boldsymbol{K}^{(L)}) \leq \frac{N}{d}\sum_{l=1}^{L}\left(\prod_{p=2}^{l-1}(1+\alpha_{p-2})\prod_{p=l+1}^{L}(1.22+\alpha_{p-2})\right).$$

The $\mu_1(\sigma_1)$ is 1-st Hermite coefficient of the activation function.

Proof. By, Table 1, Eqs. (25) and (30). we can have this result.

It should be noted that for Sigmoid network (all of activation functions are Sigmoid) and Tanh (all of activation functions are Tanh) network , the upper bound of $G_{\max}$ will change. By Eq. (6) and Eq. (7) we have for Sigmoid $G_{\max} = \frac{1}{2}$, For Tanh $G_{\max} = 2$. That means $f_{\mathrm{S}}(G_{\max})$ in the Theorem 1 is replaced by $f_{\mathrm{S}}(\frac{1}{2})$ in Eq. (31) and $f_{\mathrm{T}}(G_{\max})$ in the Theorem 1 is replaced by $f_{\mathrm{T}}(2)$ in Eq. (32).

□

C The bound of the minimum eigenvalues of NTK for finite width

We present the details of our results on sec. 4.4 in this section. Firstly, we introduce the specific expression form for NTK of finite width network in Appendix C.1. Then, we introduce some lemmas in Appendix C.2 to facilitate the proof of theorems, after that we provide the results of multiple activation functions are mixed in one network in Appendix C.3 directly, finally we discuss the results.

C.1 Neural Tangent Kernel for finite width

$$\bar{\boldsymbol{K}}^{(L)} = \boldsymbol{J}\boldsymbol{J}^{\top} = \sum_{l=1}^{L}\left[\frac{\partial \boldsymbol{F}}{\partial \mathrm{vec}(\boldsymbol{W}_l)}\right]\left[\frac{\partial \boldsymbol{F}}{\partial \mathrm{vec}(\boldsymbol{W}_l)}\right]^{\top}.$$

Let $\boldsymbol{F}_k = [\boldsymbol{f}_k(\boldsymbol{x}_1), \ldots, \boldsymbol{f}_k(\boldsymbol{x}_N)]^T$, by chain rule and some standard calculation, we have,

$$\boldsymbol{J}\boldsymbol{J}^\top = \sum_{k=0}^{L-1} \boldsymbol{F}_k \boldsymbol{F}_k^\top \circ \boldsymbol{B}_{k+1}\boldsymbol{B}_{k+1}^\top \,,$$

where $\boldsymbol{B}_k \in \mathbb{R}^{N\times m}$ is a matrix of which the i-th row is given by

$$(\boldsymbol{B}_k)_{i:} = \begin{cases} \boldsymbol{D}_{i,k}\prod_{l=k+1}^{L-1}(\boldsymbol{W}_l\boldsymbol{D}_{i,l} + \alpha_{l-1}\boldsymbol{I}_{m\times m})\boldsymbol{W}_L, & k \in [L-2]\,, \\ \boldsymbol{D}_{i,L-1}\boldsymbol{W}_L, & k = L-1\,, \\ 1, & k = L\,. \end{cases}$$

C.2 Relevant Lemmas

Lemma 2. *Fix any $k \in [0, L-1]$ and $\boldsymbol{x} \sim P_X$, then for ReLU, LeakyReLU, Sigmoid, Tanh and Swish we have*

$$\|\boldsymbol{f}_k(\boldsymbol{x})\|_2^2 = \Theta(1)\,,$$

with probability at least $1 - \sum_{l=1}^{k}\exp(-\Omega(m))$ over $(\boldsymbol{W}_l)_{l=1}^k$ and $\boldsymbol{x}$. Moreover,

$$\mathbb{E}_{\boldsymbol{x}}\,\|\boldsymbol{f}_k(\boldsymbol{x})\|_2^2 = \Theta(1)\,,$$

with probability at least $1 - \sum_{l=1}^{k-1}\exp(-\Omega(m))$ over $(\boldsymbol{W}_l)_{l=1}^k$.

Proof. We prove this by induction.

The result holds for $k = 0$ due to Assumption 1 and Assumption 3.

Assume that the lemma holds for some $k-1$, i.e.

$$\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2 = \Theta(1)\,,$$

with probability at least $1 - \sum_{l=1}^{k-1}\exp(-\Omega(m))$ over $(\boldsymbol{W}_l)_{l=1}^k$ and $\boldsymbol{x}$.

Let us condition on this event of $(\boldsymbol{W}_l)_{l=1}^{k-1}$ and study probability bounds over $\boldsymbol{W}_k$: Let $\boldsymbol{W}_k = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m]^\top$ where $\boldsymbol{w}_j \sim \mathcal{N}(0, \mathbb{I}_m/m)$ and $f_k^{[j]}$ represents the j-th element of $\boldsymbol{f}_k$. Note that:

$$\|\boldsymbol{f}_k(\boldsymbol{x})\|_2^2 = \sum_{j=1}^{m} f_k^{[j]}(\boldsymbol{x})^2\,. \tag{33}$$

Then we have:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{W}_k} \left\| \boldsymbol{f}_k(\boldsymbol{x}) \right\|_2^2 &= \sum_{j=1}^{m} \mathbb{E}_{\boldsymbol{w}_j}[f_k^{[j]}(\boldsymbol{x})^2] \\
&= \sum_{j=1}^{m} \mathbb{E}_{\boldsymbol{w}_j}\left(\left[\sigma_k\left(\ \langle \boldsymbol{w}_j, \boldsymbol{f}_{k-1}(\boldsymbol{x})\rangle\ \right) + \alpha_{k-1} f_{k-1}^{[j]}(\boldsymbol{x})\right]^2\right) \qquad Eq.\ (1) \\
&= \sum_{j=1}^{m} \left(\mathbb{E}_{\boldsymbol{w}_j}\left[\left(\sigma_k(\langle \boldsymbol{w}_j, \boldsymbol{f}_{k-1}(\boldsymbol{x})\rangle\right)^2\right] + \mathbb{E}_{\boldsymbol{w}_j}\left(\alpha_{k-1}^2 f_{k-1}^{[j]}(\boldsymbol{x})^2\right)\right. \\
&\left. + \mathbb{E}_{\boldsymbol{w}_j}\left[2\sigma_k\left(\ \langle \boldsymbol{w}_j, \boldsymbol{f}_{k-1}(\boldsymbol{x})\rangle\ \right)\alpha_{k-1} f_{k-1}^{[j]}(\boldsymbol{x})\right]\right) \\
&= m\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)^2) + \sum_{j=1}^{m} \alpha_{k-1}^2 \mathbb{E}_{\boldsymbol{w}_j}\left(f_{k-1}^{[j]}(\boldsymbol{x})^2\right) \\
&+ 2\sum_{j=1}^{m} \alpha_{k-1} \mathbb{E}_{\boldsymbol{w}_j}\left(\sigma_k\left[\ \langle \boldsymbol{w}_j, \boldsymbol{f}_{k-1}(\boldsymbol{x})\rangle\ \right]\right)\mathbb{E}_{\boldsymbol{w}_j}\left(f_{k-1}^{[j]}(\boldsymbol{x})\right) \\
&= m\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)^2) + \alpha_{k-1}^2 \left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2^2 \\
&+ 2\alpha_{k-1}\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)) \sum_{j=1}^{m} f_{k-1}^{[j]}(\boldsymbol{x})\,.
\end{aligned} \tag{34}
$$

According to Eqs. (4), (5), (8), (11), (13), (14), (16) and (17), we know that when σ_{k-1} are in ReLU, LeakyReLU, Sigmoid, Tanh and Swish we have:

$$
m\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)^2) = m\Theta\left(\frac{\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2^2}{m}\right) = \Theta(\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2^2)\,. \tag{35}
$$

When σ_{k-1} is ReLU, LeakyReLU or Swish, Eq. (35) can be written as:

$$
\frac{1}{2}\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2^2 \leq m\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)^2) \leq (1+\eta^2)\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2^2\ ,
$$

$$
0 < \mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)) \leq \mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(f_{\mathrm{ReLU}}(w)) = \frac{2\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2}{5\sqrt{m}}\,.
$$

According to the relationship between the vectors 1-norm and 2-norm, we have:

$$
-\sqrt{m}\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2 \leq \sum_{j=1}^{m} f_{k-1}^{[j]}(\boldsymbol{x}) \leq \sqrt{m}\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2\ .
$$

Then:

$$
-\frac{2\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2^2}{5} \leq 2\alpha_{k-1}\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)) \sum_{j=1}^{m} f_{k-1}^{[j]}(\boldsymbol{x}) \leq \frac{2\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2^2}{5}\,.
$$

If we substitute into Eq. (34), we have upper bound and lower bound for $\mathbb{E}_{\boldsymbol{W}_k}\left\|\boldsymbol{f}_k(\boldsymbol{x})\right\|_2^2$:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{W}_k}\left\|\boldsymbol{f}_k(\boldsymbol{x})\right\|_2^2 &= m\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)^2) + \alpha_{k-1}^2\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2^2 \\
&+ 2\alpha_{k-1}\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)) \sum_{j=1}^{m} f_{k-1}^{[j]}(\boldsymbol{x}) \\
&\leq \left(1+\eta^2+\alpha_{k-1}+\frac{2}{5}\right)\left\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\right\|_2^2 \\
&\leq \left(\eta^2+\frac{12}{5}\right)\Theta(1)\,,
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{W}_k} \left\| \boldsymbol{f}_k(\boldsymbol{x}) \right\|_2^2 &= m\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)^2) + \alpha_{k-1}^2 \left\| \boldsymbol{f}_{k-1}(\boldsymbol{x}) \right\|_2^2 \\
&\quad + 2\alpha_{k-1}\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}(\sigma_k(w)) \sum_{j=1}^{m} f_{k-1}^{[j]}(\boldsymbol{x}) \\
&\geq \left(\frac{1}{2} + \alpha_{k-1} - \frac{2}{5} \right) \left\| \boldsymbol{f}_{k-1}(\boldsymbol{x}) \right\|_2^2 \\
&\geq \frac{1}{10}\Theta(1)\,.
\end{aligned}
$$

That means, when σ_{k-1} is ReLU, LeakyReLU or Swish we have:

$$
\mathbb{E}_{\boldsymbol{W}_k} \left\| \boldsymbol{f}_k(\boldsymbol{x}) \right\|_2^2 = \Theta(1)\,. \tag{36}
$$

When σ_{k-1} is Sigmoid or Tanh, according to symmetry we have:

$$
\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}[\sigma_k(w)] = 0\,.
$$

Then:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{W}_k} \left\| f_k(\boldsymbol{x}) \right\|_2^2 &= \Theta(\left\| \boldsymbol{f}_{k-1}(\boldsymbol{x}) \right\|_2^2) + \alpha_{k-1}^2 \left\| \boldsymbol{f}_{k-1}(\boldsymbol{x}) \right\|_2^2 \\
&= \Theta(\left\| \boldsymbol{f}_{k-1}(\boldsymbol{x}) \right\|_2^2)\,.
\end{aligned} \tag{37}
$$

By Eq. (36) and Eq. (37) when σ_{k-1} is ReLU, LeakyReLU, Sigmoid, Tanh or Swish we have:

$$
\mathbb{E}_{\boldsymbol{W}_k} \left\| \boldsymbol{f}_k(\boldsymbol{x}) \right\|_2^2 = \Theta(1).
$$

Thus, by applying Bernstein's inequality to the sum of i.i.d. random variables in Eq. (33), we have:

$$
\frac{1}{2}\mathbb{E}_{\boldsymbol{W}_k} \left\| \boldsymbol{f}_k(\boldsymbol{x}) \right\|_2^2 \leq \left\| \boldsymbol{f}_k(\boldsymbol{x}) \right\|_2^2 \leq \frac{3}{2}\mathbb{E}_{\boldsymbol{W}_k} \left\| \boldsymbol{f}_k(\boldsymbol{x}) \right\|_2^2\,,
$$

with probability at least $1 - \exp(-\Omega(m))$. i.e.:

$$
\left\| \boldsymbol{f}_k(\boldsymbol{x}) \right\|_2^2 = \Theta(1)\,,
$$

with probability at least $1 - \sum_{l=1}^{k} \exp(-\Omega(m))$.

The proof for $\mathbb{E}_x \left\| \boldsymbol{f}_k(\boldsymbol{x}) \right\|_2^2$ can be done by following similar passages and using that $\left\| \mathbb{E}_{\boldsymbol{x}}[f_k^{[j]}(\boldsymbol{x})^2] \right\|_{\psi_1} \leq \mathbb{E}_{\boldsymbol{x}} \left\| f_k^{[j]}(\boldsymbol{x})^2 \right\|_{\psi_1}$. □

Lemma 3. *Fix any layer $k \in [L-1]$, and $\boldsymbol{x} \sim P_X$. Then, we have that $\left\| \boldsymbol{D}_k \right\|_{\mathrm{F}}^2 = \Theta(m)$ with probability at least $1 - \sum_{l=1}^{k} \exp(-\Omega(m))$ over $(\boldsymbol{W}_l)_{l=1}^{k}$ and $\boldsymbol{x}$.*

Proof. By Lemma 2, we have $f_{k-1}(\boldsymbol{x}) \neq 0$ with probability at least $1 - \sum_{l=1}^{k} \exp(-\Omega(m))$ over $(\boldsymbol{W}_l)_{l=1}^{k}$ and $\boldsymbol{x}$. Let us condition on this event and derive probability bounds over $\boldsymbol{W}_k$. Let $\boldsymbol{W}_k = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_{n_k}]$. Then, $\left\| \boldsymbol{D}_k \right\|_{\mathrm{F}}^2 = \sum_{j=1}^{m} {\sigma_k'}^2(\langle \boldsymbol{f}_{k-1}(\boldsymbol{x}), \boldsymbol{w}_j \rangle)$. Thus:

$$
\mathbb{E}_{\boldsymbol{W}_k} \left\| \boldsymbol{D}_k \right\|_{\mathrm{F}}^2 = m\mathbb{E}_{\boldsymbol{w}_1}[{\sigma_k'}^2(\langle \boldsymbol{f}_{k-1}(\boldsymbol{x}), \boldsymbol{w}_1 \rangle)] = m\mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{k-1}(\boldsymbol{x})\|_2^2/m)}[{\sigma_k'}^2(w)]\,.
$$

By Eqs. (19), (20), (21), (22), (23) and (24), we know that when σ_k are in ReLU, LeakyReLU, Sigmoid, Tanh and Swish we have:

$$\mathbb{E}_{\boldsymbol{W}_k} \|\boldsymbol{D}_k\|_{\mathrm{F}}^2 = m\Theta(1) = \Theta(m)\,.$$

By Hoeffding's inequality on bounded random variables, we have:

$$\mathbb{P}\left(\left|\|\boldsymbol{D}_k\|_{\mathrm{F}}^2 - \mathbb{E}_{\boldsymbol{W}_k} \|\boldsymbol{D}_k\|_{\mathrm{F}}^2\right| > t\right) \leq 2\exp\left(-\frac{2t^2}{m}\right)\,.$$

Picking $t := 0.01m$ concludes the proof.

□

Lemma 4. *For any $k \in [L-1]$, $k \leq p \leq L-1$ and $\boldsymbol{x} \sim P_X$, we have that:*

$$\Theta\Big(m \prod_{i=k+1}^{p} (\beta_3(\sigma_i)+\alpha_{i-1})\Big) \leq \left\|\boldsymbol{D}_k \prod_{l=k+1}^{p} \Big(\boldsymbol{W}_l\boldsymbol{D}_l + \alpha_{l-1}\boldsymbol{I}_{m\times m}\Big)\right\|_F^2 \leq \Theta\Big(m \prod_{i=k+1}^{p} (\beta_2(\sigma_i)+\alpha_{i-1})\Big)\,,$$

with probability at least $1 - \sum_{l=k+1}^{p} \exp(-\Omega(m))$ over $(\boldsymbol{W}_l)_{l=k+1}^{p}$ and $\boldsymbol{x}$.

Proof. We prove this by induction on p.

Lemma 3 implies that the statement holds for $p = k$.

Suppose it holds for some $p-1$. Let $\boldsymbol{S}_p = \boldsymbol{D}_k \prod_{l=k+1}^{p}(\boldsymbol{W}_l\boldsymbol{D}_l + \alpha_{l-1}\boldsymbol{I}_{m\times m})$. Then,$\boldsymbol{S}_p = \boldsymbol{S}_{p-1}(\boldsymbol{W}_p\boldsymbol{D}_p + \alpha_{p-1}\boldsymbol{I}_{m\times m}) = \boldsymbol{S}_{p-1}\boldsymbol{W}_p\boldsymbol{D}_p + \alpha_{p-1}\boldsymbol{S}_{p-1}$. Let $\boldsymbol{W}_p = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{n_p}]$. Then:

$$\|\boldsymbol{S}_p\|_{\mathrm{F}}^2 = \sum_{j=1}^{m} \|\boldsymbol{S}_{p-1}\boldsymbol{w}_j\|_2^2\, \sigma_p'(\langle \boldsymbol{f}_{p-1}(\boldsymbol{x}), \boldsymbol{w}_j\rangle)^2 + \alpha_{p-1}\|\boldsymbol{S}_{p-1}\|_{\mathrm{F}}^2\,.$$

Then we have:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{W}_p} \|\boldsymbol{S}_p\|_{\mathrm{F}}^2 &= m\mathbb{E}_{w\sim\mathcal{N}(0,\mathbb{I}_m/m)} \|\boldsymbol{S}_{p-1}\boldsymbol{w}_j\|_2^2\, \sigma_p'(\langle \boldsymbol{f}_{p-1}(\boldsymbol{x}), \boldsymbol{w}_j\rangle)^2 + \alpha_{p-1}\|\boldsymbol{S}_{p-1}\|_{\mathrm{F}}^2 \\
&= m\mathbb{E}_{w\sim\mathcal{N}(0,\mathbb{I}_m/m)} \|\boldsymbol{S}_{p-1}\boldsymbol{w}_j\|_2^2\, \mathbb{E}_{w\sim\mathcal{N}(0,\mathbb{I}_m/m)}\sigma_p'(\langle \boldsymbol{f}_{p-1}(\boldsymbol{x}), \boldsymbol{w}_j\rangle)^2 + \alpha_{p-1}\|\boldsymbol{S}_{p-1}\|_{\mathrm{F}}^2 \\
&= \|\boldsymbol{S}_{p-1}\|_{\mathrm{F}}^2\, \mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{p-1}(\boldsymbol{x})\|_2^2/m)}\sigma_p'(w)^2 + \alpha_{p-1}\|\boldsymbol{S}_{p-1}\|_{\mathrm{F}}^2\,.
\end{aligned}$$

From the previous result Eqs. (19), (20), (21), (22), (23) and (24) we have:

$$\beta_3(\sigma_p) \leq \mathbb{E}_{w\sim\mathcal{N}(0,\|\boldsymbol{f}_{p-1}(\boldsymbol{x})\|_2^2/m)}\sigma_p'(w)^2 \leq \beta_2(\sigma_p)\,.$$

That is:

$$(\beta_3(\sigma_p) + \alpha_{p-1})\|\boldsymbol{S}_{p-1}\|_{\mathrm{F}}^2 \leq \mathbb{E}_{\boldsymbol{W}_p}\|\boldsymbol{S}_p\|_{\mathrm{F}}^2 \leq (\beta_2(\sigma_p) + \alpha_{p-1})\|\boldsymbol{S}_{p-1}\|_{\mathrm{F}}^2\,.$$

Moreover:

$$\left\|\|\boldsymbol{S}_{p-1}\boldsymbol{w}_j\|_2^2\, \sigma_p'(\langle \boldsymbol{f}_{p-1}(\boldsymbol{x}), \boldsymbol{w}_j\rangle)^2\right\|_{\psi_1} \leq \left\|\|\boldsymbol{S}_{p-1}\boldsymbol{w}_j\|_2\right\|_{\psi_2}^2 \leq \frac{c}{m}\|\boldsymbol{S}_{p-1}\|_{\mathrm{F}}^2\,.$$

By Bernstein's inequality [Vershynin, 2018], we have:

$$\frac{1}{2}\mathbb{E}_{\boldsymbol{W}_p}\|\boldsymbol{S}_p\|_{\mathrm{F}}^2 \leq \|\boldsymbol{S}_p\|_{\mathrm{F}}^2 \leq \frac{3}{2}\mathbb{E}_{\boldsymbol{W}_p}\|\boldsymbol{S}_p\|_{\mathrm{F}}^2\,,$$

with probability at least $1 - \exp(-\Omega(m))$. Finally, taking the intersection of all the events finishes the proof.

□

Lemma 5. *For any layer $k \in [L-2]$ and $\boldsymbol{x} \sim P_X$, we have:*

$$\Theta\Big(\prod_{i=k+1}^{L-1}(\beta_3(\sigma_i)+\alpha_{i-1})\Big) \leq \left\|\boldsymbol{D}_k \prod_{l=k+1}^{L-1}\Big(\boldsymbol{W}_l\boldsymbol{D}_l + \alpha_{l-1}\boldsymbol{I}_{m\times m}\Big)\boldsymbol{W}_L\right\|_2^2 \leq \Theta\Big(\prod_{i=k+1}^{L-1}(\beta_2(\sigma_i)+\alpha_{i-1})\Big),$$

with probability at least $1-\sum_{l=k+1}^{L-1}\exp(-\Omega(m)) - \exp(-\Omega(1))$.

Proof. Let $\boldsymbol{B} = \boldsymbol{D}_k \prod_{l=k+1}^{L-1}(\boldsymbol{W}_l\boldsymbol{D}_l + \alpha_{l-1}\boldsymbol{I}_{m\times m})$.
By Lemma 4, we have:

$$\Theta\Bigg(m\prod_{i=k+1}^{L-1}(\beta_3(\sigma_i)+\alpha_{i-1})\Bigg) \leq \|\boldsymbol{B}\|_{\mathrm{F}}^2 \leq \Theta\Bigg(m\prod_{i=k+1}^{L-1}(\beta_2(\sigma_i)+\alpha_{i-1})\Bigg), \tag{38}$$

with probability at least $1-\sum_{l=k+1}^{L-1}\exp(-\Omega(m))$.
Then, by Hanson-Wright inequality [Vershynin, 2018], we have:

$$\frac{1}{2m}\|\boldsymbol{B}\|_{\mathrm{F}}^2 = \frac{1}{2}\mathbb{E}_{\boldsymbol{W}_L}\|\boldsymbol{B}\boldsymbol{W}_L\|_2^2 \leq \|\boldsymbol{B}\boldsymbol{W}_L\|_2^2 \leq \frac{3}{2}\mathbb{E}_{\boldsymbol{W}_L}\|\boldsymbol{B}\boldsymbol{W}_L\|_2^2 = \frac{3}{2m}\|\boldsymbol{B}\|_{\mathrm{F}}^2, \tag{39}$$

with probability at least $1-\exp(-\Omega(\|\boldsymbol{B}\|_{\mathrm{F}}^2 / \|\boldsymbol{B}\|_2^2)) \geq 1-\exp(-\Omega(1))$ over $\boldsymbol{W}_L$.
According to Eq. (38) and Eq. (39) we can get the result.

□

C.3 Results for mixed activation functions under the finite-width setting (Proof of Theorem 2)

Proof. We firstly present the lower bound of the minimal eigenvalue of $\boldsymbol{J}\boldsymbol{J}^\top$ and then derive its upper bound.

Lower bound
For PSD matrices $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{R}^{N\times N}$, it holds $\lambda_{\min}(\boldsymbol{P}\circ\boldsymbol{Q}) \geq \lambda_{\min}(\boldsymbol{P})\min_{i\in[N]} Q_{ii}$. Then, by Theorem 1 and Theorem 5.1 of Nguyen et al. [2021]:

$$\begin{aligned}
\lambda_{\min}(\boldsymbol{J}\boldsymbol{J}^\top) &\geq \sum_{k=0}^{L-1}\lambda_{\min}(\boldsymbol{F}_k\boldsymbol{F}_k^\top)\min_{i\in[N]}\|(\boldsymbol{B}_{k+1})_{i:}\|_2^2 \\
&\geq \lambda_{\min}(\boldsymbol{F}_0\boldsymbol{F}_0^\top)\min_{i\in[N]}\|(\boldsymbol{B}_1)_{i:}\|_2^2 \\
&\geq \Theta\Bigg(\frac{N}{d}\prod_{i=2}^{L-1}(\beta_3(\sigma_i)+\alpha_{i-1})\Bigg),
\end{aligned}$$

with probability at least $1-e^{-d}-\sum_{l=1}^{L-1}\exp(-\Omega(m)) - \exp(-\Omega(1))$. where the last inequality hold by Lemma 5 and Proposition 4.

Upper bound
For ReLU, LeakyReLU and Swish we have:

$$\begin{aligned}
\lambda_{min}(\boldsymbol{J}\boldsymbol{J}^\top) \leq \sum_{i=0}^{N}(\boldsymbol{J}\boldsymbol{J}^\top)_{ii}/d &= \frac{1}{d}\sum_{i=0}^{N}\sum_{k=0}^{L-1}\|(\boldsymbol{F}_k)_{1:}\|_2^2\,\|(\boldsymbol{B}_{k+1})_{1:}\|_2^2 \\
&= \frac{1}{d}\sum_{i=0}^{N}\sum_{k=0}^{L-1}\|\boldsymbol{f}_k(\boldsymbol{x_1})\|_2^2\,\|(\boldsymbol{B}_{k+1})_{1:}\|_2^2 .
\end{aligned}$$

By Lemma 2 and Lemma 5 we have:

$$\lambda_{min}(\boldsymbol{J}\boldsymbol{J}^\top) \leq \frac{N}{d}\sum_{k=0}^{L-1}\Theta\Bigg(\prod_{i=k+2}^{L-1}(\beta_2(\sigma_i)+\alpha_{i-1})\Bigg),$$

with probability at least $1-\sum_{l=1}^{L-1}\exp(-\Omega(m))-\exp(-\Omega(1))$.

□

D Generalization error via the minimum eigenvalue of NTK

In this section, firstly, we provide some useful lemmas in Appendix D.1,then present the proof of Theorem 3 in Appendix D.2.

D.1 Relevant Lemmas

Lemma 6. *(Vershynin [2018, Theorem 4.4.5]) Let $\boldsymbol{A}$ be an $N\times n$ matrix whose entries are independent standard normal random variables. Then for every $t\geq 0$, with probability at least $1-2\exp(-t^2/2)$, one has:*

$$s(\boldsymbol{A})_{\max}\leq\sqrt{N}+\sqrt{n}+t\,.$$

We need the following lemma to show that the output of each neuron with any activation function does not change too much if the input weights are close.

Lemma 7. *Let $\boldsymbol{W}\in\mathbb{R}^{m\times m}$ be the random Gaussian matrix with $W_{i,j}\sim\mathcal{N}(0,1/m)$, $\mathrm{Lip}_{\max}$ be the maximum value of the Lipschitz constants of the all activation functions, with $\omega=\mathcal{O}((3\mathrm{Lip}_{\max}+1)^{-(L-1)})$, assuming $\widetilde{\boldsymbol{W}}\in\mathcal{B}(\boldsymbol{W},\omega)$, for any $l\in[L]$, it holds that $\left\|\hat{\boldsymbol{f}}_{i,l}\right\|_2=\mathcal{O}(1)$ with probability at least $1-2l\exp(-m/2)-l\exp(-\Omega(m))$.*

Proof. We provide the estimation on $\hat{\boldsymbol{f}}_{i,1}$ and $\hat{\boldsymbol{f}}_{i,l}$ ($l=2,3,\cdots,L$) in Definition 2, respectively. Firstly, $\hat{\boldsymbol{f}}_{i,1}$ admits:

$$\begin{aligned}\left\|\hat{\boldsymbol{f}}_{i,1}\right\|_2&=\left\|\widetilde{\boldsymbol{f}}_{i,1}-\boldsymbol{f}_{i,1}\right\|_2=\left\|\sigma_1(\widetilde{\boldsymbol{W}}_1\boldsymbol{x}_i)-\sigma_1(\boldsymbol{W}_1\boldsymbol{x}_i)\right\|_2\\&\leq\mathrm{Lip}_{\sigma_1}\left\|\widetilde{\boldsymbol{W}}_1-\boldsymbol{W}_1\right\|_2\|\boldsymbol{x}_i\|_2\leq\omega\mathrm{Lip}_{\sigma_1}\\&=\mathcal{O}(1)\,.\end{aligned}$$

For $\hat{\boldsymbol{f}}_{i,l}$ with $l=2,3,\ldots,L$, we have:

$$\begin{aligned}\left\|\hat{\boldsymbol{f}}_{i,l}\right\|_2&=\left\|\widetilde{\boldsymbol{f}}_{i,l}-\boldsymbol{f}_{i,l}\right\|_2\\&=\left\|\sigma_l(\widetilde{\boldsymbol{W}}_l\widetilde{\boldsymbol{f}}_{i,l-1})+\alpha_{l-1}\widetilde{\boldsymbol{f}}_{i,l-1}-\sigma_l(\boldsymbol{W}_l\boldsymbol{f}_{i,l-1})-\alpha_{l-1}\boldsymbol{f}_{i,l-1}\right\|_2\\&\leq\left\|\sigma_l(\widetilde{\boldsymbol{W}}_l\widetilde{\boldsymbol{f}}_{i,l-1})-\sigma_l(\boldsymbol{W}_l\boldsymbol{f}_{i,l-1})\right\|_2+\alpha_{l-1}\left\|\hat{\boldsymbol{f}}_{i,l-1}\right\|_2\\&\leq\mathrm{Lip}_{\sigma_l}\left\|\widetilde{\boldsymbol{W}}_l\widetilde{\boldsymbol{f}}_{i,l-1}-\boldsymbol{W}_l\boldsymbol{f}_{i,l-1}\right\|_2+\left\|\hat{\boldsymbol{f}}_{i,l-1}\right\|_2\qquad\text{[Lipschitz continuity of }\sigma_l]\\&=\mathrm{Lip}_{\sigma_l}\left\|\boldsymbol{W}_l(\widetilde{\boldsymbol{f}}_{i,l-1}-\boldsymbol{f}_{i,l-1})+(\widetilde{\boldsymbol{W}}_l-\boldsymbol{W}_l)\widetilde{\boldsymbol{f}}_{i,l-1}\right\|_2+\left\|\hat{\boldsymbol{f}}_{i,l-1}\right\|_2\\&\leq\mathrm{Lip}_{\sigma_l}\left\{\left\|\boldsymbol{W}_l(\widetilde{\boldsymbol{f}}_{i,l-1}-\boldsymbol{f}_{i,l-1})\right\|_2+\left\|(\widetilde{\boldsymbol{W}}_l-\boldsymbol{W}_l)\widetilde{\boldsymbol{f}}_{i,l-1}\right\|_2\right\}+\left\|\hat{\boldsymbol{f}}_{i,l-1}\right\|_2\\&\leq\mathrm{Lip}_{\sigma_l}\left\{\|\boldsymbol{W}_l\|_2\left\|\widetilde{\boldsymbol{f}}_{i,l-1}-\boldsymbol{f}_{i,l-1}\right\|_2+\left\|\widetilde{\boldsymbol{W}}_l-\boldsymbol{W}_l\right\|_2\left\|\widetilde{\boldsymbol{f}}_{i,l-1}\right\|_2\right\}+\left\|\hat{\boldsymbol{f}}_{i,l-1}\right\|_2\\&\leq(\mathrm{Lip}_{\sigma_l}\|\boldsymbol{W}_l\|_2+1)\left\|\hat{\boldsymbol{f}}_{i,l-1}\right\|_2+\mathrm{Lip}_{\sigma_l}\omega\Bigg(\left\|\widetilde{\boldsymbol{f}}_{i,l-1}-\boldsymbol{f}_{i,l-1}\right\|_2+\|\boldsymbol{f}_{i,l-1}\|_2\Bigg)\\&=\left\{\mathrm{Lip}_{\sigma_l}(\|\boldsymbol{W}_l\|_2+\omega)+1\right\}\left\|\hat{\boldsymbol{f}}_{i,l-1}\right\|_2+\mathrm{Lip}_{\sigma_l}\omega\|\boldsymbol{f}_{i,l-1}\|_2\,.\end{aligned}\tag{40}$$

By Lemma 6, choosing $t = \sqrt{m}$, with probability at least $1 - 2\exp(-m/2)$, we have:

$$\|\boldsymbol{W}_l\|_2 = s(\boldsymbol{W}_l)_{\max} \leq \frac{\sqrt{m} + \sqrt{m} + \sqrt{m}}{\sqrt{m}} = 3\,.$$

Then, $\|\hat{\boldsymbol{f}}_{i,l}\|_2$ in Eq. (40) can be further upper bounded with probability at least $1 - 2l\exp(-m/2) - l\exp(-\Omega(m))$:

$$\begin{aligned}
\left\|\hat{\boldsymbol{f}}_{i,l}\right\|_2 &\leq \Big((3+\omega)\mathrm{Lip}_{\max} + 1\Big)\left\|\hat{\boldsymbol{f}}_{i,l-1}\right\|_2 + \mathrm{Lip}_{\max}\omega\,\|\boldsymbol{f}_{i,l-1}\|_2 \\
&\leq \Big([(3+\omega)\mathrm{Lip}_{\max} + 1]^{l-1} - 1\Big)\left(\mathrm{Lip}_{\sigma_1}\omega + \frac{\mathrm{Lip}_{\max}\omega\,\|\boldsymbol{f}_{i,l-1}\|_2}{(3+\omega)\mathrm{Lip}_{\max}}\right) + \mathrm{Lip}_{\sigma_1}\omega \\
&\leq (3\mathrm{Lip}_{\max} + 1)^{L-1}\Theta(1)\omega + \mathrm{Lip}_{\sigma_1}\omega \\
&= \mathcal{O}(1)\Theta(1) + \mathcal{O}(1) \\
&= \mathcal{O}(1)\,,
\end{aligned}$$

where the second inequality holds by the recursion which conclude the proof.

□

We also need the following lemma, demonstrating that the neural network function is almost linear in terms of its weights if the initializations are close to each other.

Lemma 8. *Let $\boldsymbol{W}, \boldsymbol{W}' \in \mathcal{B}(\boldsymbol{W}^{(0)}, \omega)$ with $\omega = \mathcal{O}((3\mathrm{Lip}_{\max} + 1)^{-(L-1)})$, for any $i \in [N]$, with probability at least $1 - 2(L-1)\exp(-m/2) - L\exp(-\Omega(m)) - 2/m$, we have:*

$$|f(\boldsymbol{x}_i; \boldsymbol{W}') - f(\boldsymbol{x}_i; \boldsymbol{W}) - \langle \nabla f(\boldsymbol{x}_i; \boldsymbol{W}), \boldsymbol{W}' - \boldsymbol{W}\rangle| = \mathcal{O}(1)\,.$$

Proof. We have the following expression:

$$\begin{aligned}
&\left|f(\boldsymbol{x}_i; \boldsymbol{W}') - f(\boldsymbol{x}_i; \boldsymbol{W}) - \big\langle \nabla f(\boldsymbol{x}_i; \boldsymbol{W}), \boldsymbol{W}' - \boldsymbol{W}\big\rangle\right| \\
=&\left|\sum_{l=1}^{L-1} \boldsymbol{W}_L \prod_{r=l+1}^{L-1} (\boldsymbol{D}_{i,r}\boldsymbol{W}_r + \alpha_{r-1}\boldsymbol{I}_{m\times m})\boldsymbol{D}_{i,l}(\boldsymbol{W}_l' - \boldsymbol{W}_l)\boldsymbol{f}_{i,l-1} + \boldsymbol{W}_L'(\boldsymbol{f}_{i,L-1}' - \boldsymbol{f}_{i,L-1})\right| \\
\leq&\sum_{l=1}^{L-1}\left|\boldsymbol{W}_L \prod_{r=l+1}^{L-1} (\boldsymbol{D}_{i,r}\boldsymbol{W}_r + \alpha_{r-1}\boldsymbol{I}_{m\times m})\boldsymbol{D}_{i,l}(\boldsymbol{W}_l' - \boldsymbol{W}_l)\boldsymbol{f}_{i,l-1}\right| + \left|\boldsymbol{W}_L'(\boldsymbol{f}_{i,L-1}' - \boldsymbol{f}_{i,L-1})\right| \\
\leq&\sum_{l=1}^{L-1}\|\boldsymbol{W}_L\|_2\left\|\prod_{r=l+1}^{L-1} (\boldsymbol{D}_{i,r}\boldsymbol{W}_r + \alpha_{r-1}\boldsymbol{I}_{m\times m})\boldsymbol{D}_{i,l}(\boldsymbol{W}_l' - \boldsymbol{W}_l)\boldsymbol{f}_{i,l-1}\right\|_2 \\
&+ \left\|\boldsymbol{W}_L'\right\|_2\left\|\boldsymbol{f}_{i,L-1}' - \boldsymbol{f}_{i,L-1}\right\|_2 \\
\leq&\sum_{l=1}^{L-1}\|\boldsymbol{W}_L\|_2 \prod_{r=l+1}^{L-1}\left(\|\boldsymbol{D}_{i,r}\|_2\|\boldsymbol{W}_r\|_2 + \alpha_{r-1}\right)\|\boldsymbol{D}_{i,l}\|_2\left\|\boldsymbol{W}_l' - \boldsymbol{W}_l\right\|_2\|\boldsymbol{f}_{i,l-1}\|_2 + \left\|\boldsymbol{W}_L'\right\|_2\left\|\boldsymbol{f}_{i,L-1}' - \boldsymbol{f}_{i,L-1}\right\|_2\,.
\end{aligned} \tag{41}$$

Here we require the derivative of the activation function σ' is bound, i.e., $\|\boldsymbol{D}\|_2 \leq \mathrm{Lip}_{\max}$. The considered activation functions in this paper satisfy this condition.

By Lemma 6, Lemma 7 and Lemma 2 with probability at least $1 - 2(L-1)\exp(-m/2) - L\exp(-\Omega(m))$, we have $\left\|\boldsymbol{f}_{i,L-1}' - \boldsymbol{f}_{i,L-1}\right\|_2 \leq \mathcal{O}(1)$, $\|\boldsymbol{f}_{i,l-1}\|_2 = \Theta(1)$ and $\|\boldsymbol{W}_r\|_2 \leq 3 \quad \forall r \in [L-1]$.

Moreover, $m\|\boldsymbol{W}_L\|_2^2$ is a random Variables obey chi-square distribution with m degrees of freedom. That means $\mathbb{E}(m\|\boldsymbol{W}_L\|_2^2) = m$ and $\mathbb{V}(m\|\boldsymbol{W}_L\|_2^2) = 2m$. By Chebyshev's Inequality we have $P(|m\|\boldsymbol{W}_L\|_2^2 - m| \geq m) \leq 2m/m^2$. i.e.:

$$\|\boldsymbol{W}_L\|_2 \leq \sqrt{2}\,,$$

with probability at least $1 - 2/m$.

Accordingly, Equation (41) can be further upper bounded by:

$$
\begin{aligned}
&|f(\boldsymbol{x}_i; \boldsymbol{W}') - f(\boldsymbol{x}_i; \boldsymbol{W}) - \langle \nabla f(\boldsymbol{x}_i; \boldsymbol{W}), \boldsymbol{W}' - \boldsymbol{W} \rangle| \\
&\leq \sum_{l=1}^{L-1} (3\mathrm{Lip}_{\max} + 1)^{L-l-1} \omega \sqrt{2} \mathrm{Lip}_{\max} \Theta(1) + (\sqrt{2} + \omega) \mathcal{O}(1) \\
&= \frac{(3\mathrm{Lip}_{\max} + 1)^{L-1} - 1}{3\mathrm{Lip}_{\max}} \omega \sqrt{2} \mathrm{Lip}_{\max} \Theta(1) + (\sqrt{2} + \omega) \mathcal{O}(1) \\
&= \mathcal{O}(1) \,.
\end{aligned}
$$

□

We define $L_i(\boldsymbol{W}) = \ell[y_i f(\boldsymbol{x}_i; \boldsymbol{W})]$, then the following lemma shows that, $L_i(\boldsymbol{W})$ is almost a convex function of $\boldsymbol{W}$ for any $i \in [N]$ if the initilizations are close.

Lemma 9. *Let $\boldsymbol{W}, \boldsymbol{W}' \in \mathcal{B}(\boldsymbol{W}^{(0)}, \omega)$ with $\omega = \mathcal{O}((3\mathrm{Lip}_{\max} + 1)^{-(L-1)})$, for any $i \in [N]$, it holds that:*

$$
L_i(\boldsymbol{W}') \geq L_i(\boldsymbol{W}) + \langle \nabla_{\boldsymbol{W}} L_i(\boldsymbol{W}), \boldsymbol{W}' - \boldsymbol{W} \rangle - \mathcal{O}(1) \,,
$$

with probability at least $1 - 2(L-1)\exp(-m/2) - L\exp(-\Omega(m)) - 2/m$.

Proof. By the convexity of $\ell(z)$, we have:

$$
L_i(\boldsymbol{W}') - L_i(\boldsymbol{W}) = \ell[y_i f(\boldsymbol{x}_i; \boldsymbol{W}')] - \ell[y_i f(\boldsymbol{x}_i; \boldsymbol{W})] \geq \ell'[y_i f(\boldsymbol{x}_i; \boldsymbol{W})] \cdot y_i \cdot [f(\boldsymbol{x}_i; \boldsymbol{W}') - f(\boldsymbol{x}_i; \boldsymbol{W})] \,.
$$

Using the chain rule leads to:

$$
\sum_{l=1}^{L} \langle \nabla_{\boldsymbol{W}_l} L_i(\boldsymbol{W}), \boldsymbol{W}_l' - \boldsymbol{W}_l \rangle = \ell'[y_i f(\boldsymbol{x}_i; \boldsymbol{W})] \cdot y_i \cdot \langle \nabla f(\boldsymbol{x}_i; \boldsymbol{W}), \boldsymbol{W}' - \boldsymbol{W} \rangle \,.
$$

Combining the above two equations, by triangle inequality, we have:

$$
\begin{aligned}
\ell'[y_i f(\boldsymbol{x}_i; \boldsymbol{W})] \cdot y_i \cdot [f(\boldsymbol{x}_i; \boldsymbol{W}') - f(\boldsymbol{x}_i; \boldsymbol{W})] &\geq \ell'[y_i f(\boldsymbol{x}_i; \boldsymbol{W})] \cdot y_i \cdot \left\langle \nabla f(\boldsymbol{x}_i; \boldsymbol{W}), \boldsymbol{W}' - \boldsymbol{W} \right\rangle - \varepsilon \\
&= \textstyle\sum_{l=1}^{L} \langle \nabla_{\boldsymbol{W}_l} L_i(\boldsymbol{W}), \boldsymbol{W}_l' - \boldsymbol{W}_l \rangle - \varepsilon \,,
\end{aligned}
$$

where $\varepsilon := |\ell'[y_i f(\boldsymbol{x}_i; \boldsymbol{W})] \cdot y_i \cdot [f(\boldsymbol{x}_i; \boldsymbol{W}') - f(\boldsymbol{x}_i; \boldsymbol{W}) - \langle \nabla f(\boldsymbol{x}_i; \boldsymbol{W}), \boldsymbol{W}' - \boldsymbol{W} \rangle]|$. Then by upper-bounding ε with Lemma 8 and the fact that $|\ell'[y_i f(\boldsymbol{x}_i; \boldsymbol{W})] \cdot y_i| \leq 1$, we have:

$$
\begin{aligned}
L_i(\boldsymbol{W}') - L_i(\boldsymbol{W}) &\geq \sum_{l=1}^{L} \langle \nabla_{\boldsymbol{W}_l} L_i(\boldsymbol{W}), \boldsymbol{W}_l' - \boldsymbol{W}_l \rangle - \varepsilon \\
&= \sum_{l=1}^{L} \langle \nabla_{\boldsymbol{W}_l} L_i(\boldsymbol{W}), \boldsymbol{W}_l' - \boldsymbol{W}_l \rangle - \mathcal{O}(1) \,.
\end{aligned}
$$

□

We need the following lemma to show that, the gradient of the neural network function can be upper bounded under near initialization.

Lemma 10. *Let $\boldsymbol{W} \in \mathcal{B}(\boldsymbol{W}^{(0)}, \omega)$ with $\omega = \mathcal{O}((3\mathrm{Lip}_{\max} + 1)^{-(L-1)})$, for any $i \in [N]$, with probability at least $1 - 2(L-l)\exp(-m/2) - l\exp(-\Omega(m)) - 2/m$, it holds that:*

$$
\|\nabla_{\boldsymbol{W}_l} f(\boldsymbol{x}_i; \boldsymbol{W})\|_2 \,, \|\nabla_{\boldsymbol{W}_l} L_i(\boldsymbol{W})\|_2 \leq \Theta(3\mathrm{Lip}_{\max} + 1)^{L-l} \,.
$$

Proof. According to the triangle inequality and definition of operator norm, we have:

$$\|\nabla_{\boldsymbol{W}_l} f(\boldsymbol{x}_i;\boldsymbol{W})\|_2 = \left\| \boldsymbol{f}_{i,l-1}\boldsymbol{W}_L \prod_{r=l+1}^{L-1} (\boldsymbol{D}_{i,r}\boldsymbol{W}_r + \alpha_{r-1}\boldsymbol{I}_{m\times m})\boldsymbol{D}_{i,l} \right\|_2$$
$$\leq \|\boldsymbol{f}_{i,l-1}\|_2 \left\| \boldsymbol{W}_L \prod_{r=l+1}^{L-1} (\boldsymbol{D}_{i,r}\boldsymbol{W}_r + \alpha_{r-1}\boldsymbol{I}_{m\times m})\boldsymbol{D}_{i,l} \right\|_2$$
$$\leq \|\boldsymbol{f}_{i,l-1}\|_2 \|\boldsymbol{W}_L\|_2 \prod_{r=l+1}^{L-1} (\|\boldsymbol{D}_{i,r}\|_2 \|\boldsymbol{W}_r\|_2 + \alpha_{r-1}) \|\boldsymbol{D}_{i,l}\|_2 \,.$$

By Lemma 2 and Lemma 6, with probability at least $1-2(L-l-1)\exp(-m/2)-l\exp(-\Omega(m))-2/m$ we have $\|\boldsymbol{f}_{i,l-1}\|_2 = \Theta(1)$, $\left\|\boldsymbol{W}_i^{(0)}\right\|_2 \leq 3$ for $i = l+1,\cdots,L-1$, $\left\|\boldsymbol{W}_L^{(0)}\right\|_2 \leq \sqrt{2}$ and $\|\boldsymbol{D}\|_2 \leq \mathrm{Lip}_{\max}$ due to σ' is bounded, then we have:

$$\|\nabla_{\boldsymbol{W}_l} f(\boldsymbol{x}_i;\boldsymbol{W})\|_2 \leq \Theta(1)(3\mathrm{Lip}_{\max}+1)^{L-l-1}\sqrt{2}\mathrm{Lip}_{\max} = \Theta(3\mathrm{Lip}_{\max}+1)^{L-l-1}\,,$$

which implies:

$$\|\nabla_{\boldsymbol{W}_l} L_i(\boldsymbol{W})\|_2 \leq |\ell'[y_i\cdot f(\boldsymbol{x}_i;\boldsymbol{W})]\cdot y_i|\cdot\|\nabla_{\boldsymbol{W}_l} f(\boldsymbol{x}_i;\boldsymbol{W})\|_2 \leq \|\nabla_{\boldsymbol{W}_l} f(\boldsymbol{x}_i;\boldsymbol{W})\|_2 \leq \Theta(3\mathrm{Lip}_{\max}+1)^{L-l-1}\,,$$

where we use the fact that $|\ell'[y_i f(\boldsymbol{x}_i;\boldsymbol{W})]\cdot y_i| \leq 1$. □

We need the following lemma to show that, the cumulative loss can be upper bounded under small changes on the parameters (i.e., weights).

Lemma 11. *For any $\epsilon,\delta,R>0$, there exists:*

$$m^\star = \frac{(3\mathrm{Lip}_{\max}+1)^{4L-4}L^2R^4}{4\varepsilon^2}\,,$$

such that if $m \geq m^*(\epsilon,\delta,R,L)$, then with probability at least $1-\delta$ over the randomness of $\boldsymbol{W}^{(1)}$, for any $\boldsymbol{W}^* \in \mathcal{B}(\boldsymbol{W}^{(1)}, Rm^{-1/2})$, Algorithm 1 with $\gamma = \varepsilon/[m(3\mathrm{Lip}_{\max}+1)^{2L-2}]$, $N = LR^2(3\mathrm{Lip}_{\max}+1)^{2L-2}/(2\varepsilon^2)$, the cumulative loss can be upper bounded by:

$$\sum_{i=1}^N L_i(\boldsymbol{W}^{(i)}) \leq \sum_{i=1}^N L_i(\boldsymbol{W}^*) + 3N\epsilon\,.$$

Remark: Discussion on the required width m refer to Appendix E.

Proof. Set $\omega = 1/(3\mathrm{Lip}_{\max}+1)^{L-1}$ such that the conditions on ω given in Lemmas 7, 8, 9 and 10 hold. It is easy to see that as long as $m \geq R^2(3\mathrm{Lip}_{\max}+1)^{2L-2}$, we have $\boldsymbol{W}^* \in \mathcal{B}(\boldsymbol{W}^{(1)},\omega)$. We now show that under our parameter choice, $\boldsymbol{W}^{(1)},\ldots,\boldsymbol{W}^{(N)}$ are inside $\mathcal{B}(\boldsymbol{W}^{(1)},\omega)$ as well.

This result follows by simple induction. Clearly we have $\boldsymbol{W}^{(1)} \in \mathcal{B}(\boldsymbol{W}^{(1)},\omega)$. Suppose that $\boldsymbol{W}^{(1)},\ldots,\boldsymbol{W}^{(i)} \in \mathcal{B}(\boldsymbol{W}^{(1)},\omega)$. Then by Lemma 10, for $l \in [L]$, we have $\|\nabla_{\boldsymbol{W}_l} L_i(\boldsymbol{W}^{(i)})\|_2 \leq \Theta(3\mathrm{Lip}_{\max}+1)^{L-l-1}$.

Therefore:

$$\left\|\boldsymbol{W}_l^{(i+1)} - \boldsymbol{W}_l^{(1)}\right\|_2 \leq \sum_{j=1}^i \left\|\boldsymbol{W}_l^{(j+1)} - \boldsymbol{W}_l^{(j)}\right\|_2 \leq \Theta((3\mathrm{Lip}_{\max}+1)^{L-l-1}\gamma N)\,.$$

Plugging in our parameter choice $\gamma = \varepsilon/[m(3\mathrm{Lip}_{\max}+1)^{2L-2}]$, $N = LR^2(3\mathrm{Lip}_{\max}+1)^{2L-2}/(2\varepsilon^2)$ for some small enough absolute constant ν provides:

$$\left\|\boldsymbol{W}_l^{(i+1)} - \boldsymbol{W}_l^{(1)}\right\|_{\mathrm{F}} \leq \Theta\left(\sqrt{m}(3\mathrm{Lip}_{\max}+1)^{L-l-1}\frac{LR^2}{2m\varepsilon}\right) \leq \omega\,,$$

where the last inequality holds as long as $m \geq (3\text{Lip}_{\max}+1)^{4L-4}L^2R^4/(4\varepsilon^2)$. Therefore by induction we see that $\boldsymbol{W}^{(1)},\ldots,\boldsymbol{W}^{(N)} \in \mathcal{B}(\boldsymbol{W}^{(1)},\omega)$. As a result, the conditions of Lemmas. 7, 8, 9 and 10 are satisfied for $\boldsymbol{W}^*$ and $\boldsymbol{W}^{(1)},\ldots,\boldsymbol{W}^{(N)}$.

In the following, we utilize the results of Lemmas 7, 8, 9 and 10 to prove the bound of cumulative loss. First of all, by Lemma 9, we have:

$$\begin{aligned} L_i(\boldsymbol{W}^{(i)}) - L_i(\boldsymbol{W}^*) &\leq \left\langle \nabla_{\boldsymbol{W}} L_i(\boldsymbol{W}^{(i)}), \boldsymbol{W}^{(i)} - \boldsymbol{W}^* \right\rangle + \epsilon \\ &= \sum_{l=1}^{L} \frac{\left\langle \boldsymbol{W}_l^{(i)} - \boldsymbol{W}_l^{(i+1)}, \boldsymbol{W}_l^{(i)} - \boldsymbol{W}_l^* \right\rangle}{\gamma} + \epsilon\,. \end{aligned}$$

Note that for the matrix inner product we have the equality $2\langle \boldsymbol{A},\boldsymbol{B}\rangle = \|\boldsymbol{A}\|_{\mathrm{F}}^2 + \|\boldsymbol{B}\|_{\mathrm{F}}^2 - \|\boldsymbol{A}-\boldsymbol{B}\|_{\mathrm{F}}^2$. Applying this equality to the right hand side above provides:

$$L_i(\boldsymbol{W}^{(i)}) - L_i(\boldsymbol{W}^*) \leq \sum_{l=1}^{L} \frac{\|\boldsymbol{W}_l^{(i)} - \boldsymbol{W}_l^{(i+1)}\|_{\mathrm{F}}^2 + \|\boldsymbol{W}_l^{(i)} - \boldsymbol{W}_l^*\|_{\mathrm{F}}^2 - \|\boldsymbol{W}_l^{(i+1)} - \boldsymbol{W}_l^*\|_{\mathrm{F}}^2}{2\gamma} + \epsilon\,.$$

By Lemma 10, for $l \in [L]$ we have $\|\boldsymbol{W}_l^{(i)} - \boldsymbol{W}_l^{(i+1)}\|_{\mathrm{F}} \leq \gamma\sqrt{m}\|\nabla_{\boldsymbol{W}_l} L_i(\boldsymbol{W}^{(i)})\|_2 \leq \Theta(\gamma\sqrt{m}(3\text{Lip}_{\max}+1)^{L-l-1})$.

Therefore:

$$L_i(\boldsymbol{W}^{(i)}) - L_i(\boldsymbol{W}^*) \leq \sum_{l=1}^{L} \frac{\|\boldsymbol{W}_l^{(i)} - \boldsymbol{W}_l^*\|_{\mathrm{F}}^2 - \|\boldsymbol{W}_l^{(i+1)} - \boldsymbol{W}_l^*\|_{\mathrm{F}}^2}{2\gamma} + \Theta((3\text{Lip}_{\max}+1)^{2L-2}\gamma m) + \epsilon\,.$$

Telescoping over $i = 1,\ldots,N$, we obtain:

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^{N} L_i(\boldsymbol{W}^{(i)}) &\leq \frac{1}{N}\sum_{i=1}^{N} L_i(\boldsymbol{W}^*) + \sum_{l=1}^{L} \frac{\|\boldsymbol{W}_l^{(1)} - \boldsymbol{W}_l^*\|_{\mathrm{F}}^2}{2N\gamma} + \Theta((3\text{Lip}_{\max}+1)^{2L-2}\gamma m) + \epsilon \\ &\leq \frac{1}{N}\sum_{i=1}^{N} L_i(\boldsymbol{W}^*) + \frac{LR^2}{2\gamma m N} + \Theta((3\text{Lip}_{\max}+1)^{2L-2}\gamma m) + \epsilon\,, \end{aligned}$$

where in the first inequality we simply remove the term $-\|\boldsymbol{W}_l^{(N+1)} - \boldsymbol{W}_l^*\|_{\mathrm{F}}^2/(2\gamma)$ to obtain an upper bound, the second inequality follows by the assumption that $\boldsymbol{W}^* \in \mathcal{B}(\boldsymbol{W}^{(1)}, Rm^{-1/2})$. Plugging in the parameter choice $\gamma = \varepsilon/[m(3\text{Lip}_{\max}+1)^{2L-2}]$, $N = LR^2(3\text{Lip}_{\max}+1)^{2L-2}/(2\varepsilon^2)$, then:

$$\frac{1}{N}\sum_{i=1}^{N} L_i(\boldsymbol{W}^{(i)}) \leq \frac{1}{N}\sum_{i=1}^{N} L_i(\boldsymbol{W}^*) + 3\epsilon\,,$$

which finishes the proof. □

D.2 Proof of Theorem 3

Proof. By Lemmas 8, Lemma 11 and Theorem 3.3, Lemma 4.4, Corollary 3.10 in [Cao and Gu, 2019], let $C_1(L) = \sqrt{L}/(3\text{Lip}_{\max}+1)^{L-1}$ and $C_2(L) = \sqrt{L}(3\text{Lip}_{\max}+1)^{L-1}$, bring in our γ and N with a not very large L, we have:

$$\mathbb{E}[\ell_{\mathcal{D}}^{0-1}(\hat{\boldsymbol{W}})] \leq \tilde{\mathcal{O}}\left(C_2\sqrt{\frac{\boldsymbol{y}^\top(\boldsymbol{K}^{(L)})^{-1}\boldsymbol{y}}{N}}\right) + \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{N}}\right).$$

According to the courant minimax principle [Golub and Van Loan, 1996]: $\frac{1}{\lambda_{\min}(\boldsymbol{K}^{(L)})} = \lambda_{\max}((\boldsymbol{K}^{(L)})^{-1}) = \max \frac{\boldsymbol{y}^\top(\boldsymbol{K}^{(L)})^{-1})\boldsymbol{y}}{\boldsymbol{y}^\top\boldsymbol{y}}$, that means $\boldsymbol{y}^\top((\boldsymbol{K}^{(L)})^{-1})\boldsymbol{y} \leq \frac{\boldsymbol{y}^\top\boldsymbol{y}}{\lambda_{\min}(\boldsymbol{K}^{(L)})}$, then we have the final bound:

$$\mathbb{E}[\ell_{\mathcal{D}}^{0-1}(\hat{\boldsymbol{W}})] \leq \tilde{\mathcal{O}}\left(C_2\sqrt{\frac{\boldsymbol{y}^\top \boldsymbol{y}}{\lambda_{\min}(\boldsymbol{K}^{(L)})N}}\right) + \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{N}}\right).$$

□

E Discussion on the key points and the motivation of the NTK analysis

In this section, we discuss the motivation and few key points in the proof of this paper and we also explain how the proof differs from previous results.

The motivation for studying the minimum eigenvalue of NTK:

To make this clearer, we provide an illustrative example on the significance of the minimum eigenvalue. Let us consider the square loss $\Phi(\theta) = \frac{1}{2}\sum_{i=1}^{n} \|f(x_i) - y_i\|^2$. A simple calculation shows that $\Phi(\theta) \leq \frac{[\nabla\Phi(\theta)]^2}{2\lambda_{\min}(K)}$. Thus if the minimum eigenvalue of NTK is strictly greater than 0, then minimizing the gradient on the LHS will drive the loss to zero. The larger the minimum eigenvalue, the smaller the loss.

Therefore, in this work, we are using the minimum eigenvalue to derive the generalization bound of NAS.

Key points in the proof:

- *Minimum eigenvalue*: Our proof framework is motivated by Nguyen et al. [2021] on minimal eigenvalue of NTK of ReLU neural networks. However, our proofs differ from them in two aspects. Firstly, as we discussed in sec. 1, extension to mixed activation functions is non-trivial due to the special properties of ReLU. More importantly, we remark that the lower bound of the minimal eigenvalue of NTK in [Nguyen et al., 2021, Theorem 3.2] holds with probability at least $1 - Ne^{-\Omega(d)} - N^2 e^{-\Omega\left(dN^{-\frac{2}{r-0.5}}\right)}$, where $r \geq 2$ is some constant. It can be found that, this concentration probability decreases as the number of training data increases. Thus, it could be negative for a large N. This is due to the use of Gershgorin circle theorem leading to a loose probability estimation. Instead, in this paper, we do not use this theorem, and we develop a tighter estimation based on Yaskov [2014] under the assumption of isotropic data distribution. Accordingly, we achieve the reasonable $1 - e^{-d}$ probability, *c.f.* Theorem 1.
- *Generalization*: Our proof framework is based on Cao and Gu [2019] for generalization guarantees of deep ReLU neural networks requiring $m = \Omega(L^{56})$. Their results cannot be directly extended to other activation functions as the nice homogeneity and the derivative property of ReLU are used in their proof. To make our result feasible to various activation functions, we employ Lipschitz continuous properties of all activation functions, and achieve the generalization guarantees with $m = \Omega(4^{4L})$, *c.f.* Theorem 3 and Lemma 11. Admittedly, our result is in an exponential increasing order of the depth. However, in practice, the depth of neural networks in NAS is usually smaller than 20, or even 10 [Liu et al., 2018, Dong et al., 2021], which leads to $4^{4L} \ll L^{56}$ in this case when compared to their result. This result makes our theory reasonable and fair for NAS.

F Auxiliary numerical validations

F.1 Dataset details and algorithm

We describe here the datasets that we have used for the numerical validation of our theory. Those are the following five datasets:

1. *Fashion-MNIST* [Xiao et al., 2017] includes grayscale images of clothing. The training set consists of $60,000$ examples and the test set of $10,000$ examples. The resolution of each image is 28×28, with each image belonging to one of the 10 classes.

2. *MNIST* [Lecun et al., 1998] includes handwritten digits images. MNIST has a training set of $60,000$ examples and a test set of $10,000$ examples. The resolution of each image is 28×28.
3. *CIFAR-10 and CIFAR-100 [Krizhevsky et al., 2014]* depicts images of natural scenes. CIFAR-100 has a training set of $50,000$ examples and a test set of $10,000$ examples. The resolution of each RGB image is 32×32.
4. *ImageNet-16* [Chrabaszcz et al., 2017] is the down-sampled version of ImageNet [Deng et al., 2009] with image size 16×16 on 120 classes.

Our Eigen-NAS algorithm used in sec. 5.2 is summarized as below.

Algorithm 2: Eigen-NAS Algorithm

Require: Search space $\mathcal{S}$, training data $\mathcal{D}_{tr} = \{(\boldsymbol{x}_i, y_i)_{i=1}^{N}\}$, validation data $\mathcal{D}_{val} = \{(\boldsymbol{x}_j, y_j)_{j=1}^{N_v}\}$.
  Initialize max_iteration $= M$
  Initialize candidate set $\mathcal{C} = []$
  for search_iteration in $1, 2, \ldots,$max_iteration **do**
    Randomly sample architecture s from search space $\mathcal{S}$.
    Compute $Eigen :=$ minimum eigenvalue of NTK.
    $\mathcal{C}$.append$(s, Eigen)$
    update $\mathcal{C}$ to kept top-K best architectures
  end for
  $s^\star = best_s(\mathcal{C}, \mathcal{D}_{tr}, \mathcal{D}_{val})$ # Choose the best architecture based on validation error after training 20 epochs.
  Output $s^\star$

F.2 Compared algorithms

We provide a thorough comparison with the following baselines:

1. *Classical network:* ResNet [He et al., 2016], which is the default baseline used widely in image-related tasks.
2. *Reinforcement learning based algorithm:* NAS-RL [Zoph and Le, 2017] with the validation accuracy as a reward, which is an classical and representative NAS Algorithm.
3. *Differentiable algorithm:* DARTS [Liu et al., 2019b][4], which is the earliest and basic gradient-based NAS algorithm.
4. *Train-free algorithms using metrics to guide NAS:* A new type of NAS algorithm, they use some special metrics to pick models directly from candidate models. Common Train-free algorithms are: NASWOT [Mellor et al., 2021] using the output of ReLU; TE-NAS [Chen et al., 2021] leveraging the spectrum of NTK and linear partition of the input space; KNAS [Xu et al., 2021] employing the Frobenius norm of NTK. Our Eigen-NAS algorithm also belongs to this type.

F.3 Training/test accuracy of DNNs by NAS

Here we evaluate the classification results with 5 runs of the obtained architecture by DARTS under varying widths $m \in \{64, 128, 256, 512, 1024\}$ and depths $L \in \{5, 10\}$ on Fashion-MNIST. Figure 2 shows that nearly 90% accuracy is achieved on the test set under different depth and width settings. The result is competitive on FC/residual networks within 10 layers and without training tricks, e.g., data augmentation, batch norm and drop out. We find that when compared to the depth, the network width also contributes on test accuracy. As suggested by Eq. (2), the amount of parameters in the neural network is approximately proportional to the depth, but squared to the width.

F.4 Simulation of minimum eigenvalues of NTK

We calculate the minimum eigenvalue of each NTK matrix under different architectures with activation functions, skip connections and depths, according to Lemma 1. We consider four special cases on skip connections: a) no skip connections with $\boldsymbol{\alpha} = \mathbf{0}$, i.e., fully connected neural network in Figure 3(a);

[4] We directly use the results from Xu et al. [2021].

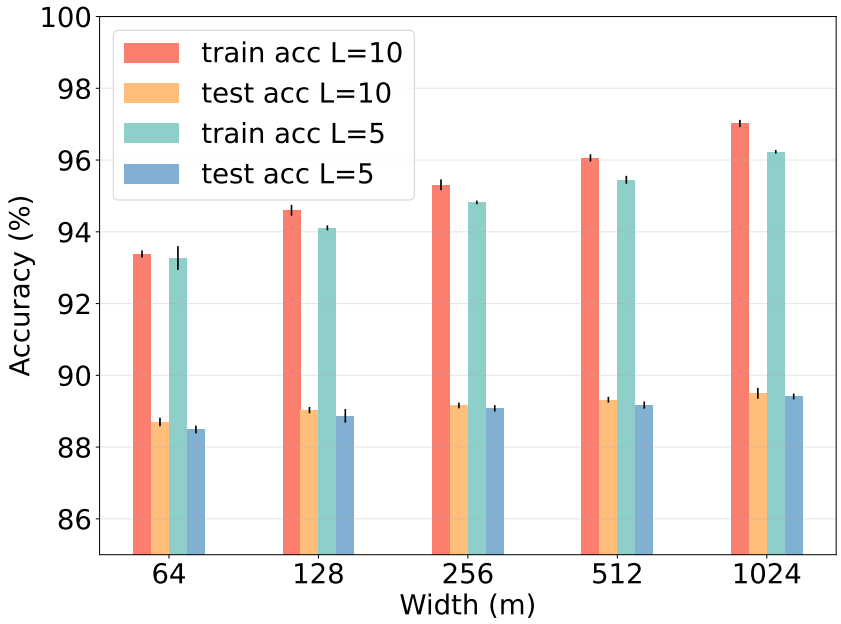


Figure 2: The accuracy of neural networks by NAS under different widths and depths.

b) skip connections between all consecutive layers with $\boldsymbol{\alpha} = \mathbf{1}$, in Figure 3(b); c) the alpha of the first half (of the network) is 1 and the alpha of the second half is 0 shown in Figure 4(a); d) The alpha of the first half is 0 and the alpha of the second half is 1. The results are shown in Figure 4(b).

Figure 3(a) indicates that as the network depth increases, the minimum eigenvalue of NTK will become larger when LeakyReLU, ReLU, Swish and Tanh employed, but Sigmoid leads to a decreasing minimum eigenvalue, which is consistent with the upper bound shown in Theorem 1. The LeakyReLU, ReLU and Swish generate the fastest increasing rate of depth, while Tanh and Sigmoid are slow, which coincides with the derived lower bound in Theorem 1 and previous work Bietti and Bach [2021]. Figure 3(b) shows that, under the skip connection, the tendency of the minimum eigenvalue of NTK is similar to that of FC neural networks when various activation functions are employed. However, the specific values and the growth rate are significantly larger than FC neural networks. This result is consistent with the conclusion we state in Theorem 1 about skip layers leading to the increase of minimum eigenvalue of NTK with respect to the depth. Moreover, Figure 4 show similar growth speed.

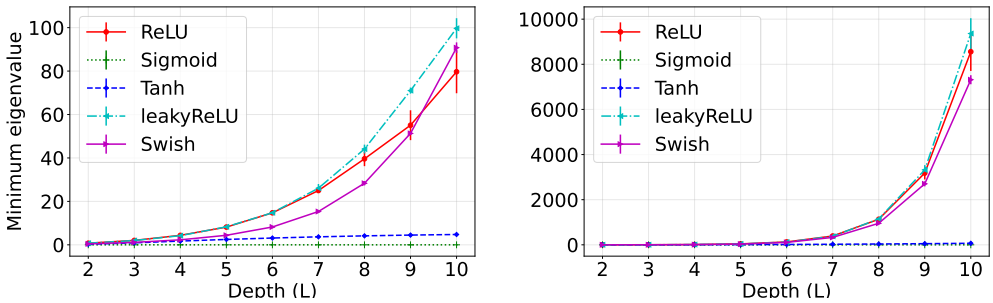
Then, we plot the comparison figure of NTK under above two settings and two settings in main paper for the same activation function in Figure 5. In addition to reconfirming the order between different activation functions, we can also see that the effect of adding an activation layer in the second half of the 10-layer neural network is better than the first half of the neural network. This verifies the experimental results in Figure 1(b).

F.5 Additional experiments on NAS-Bench-101 and ranking correlations

In this section, we conduct more experiments on two new benchmarks NAS-Bench-101 [Ying et al., 2019] and Network Design Spaces (NDS) [Radosavovic et al., 2019] using the same setting as sec. 5.2.

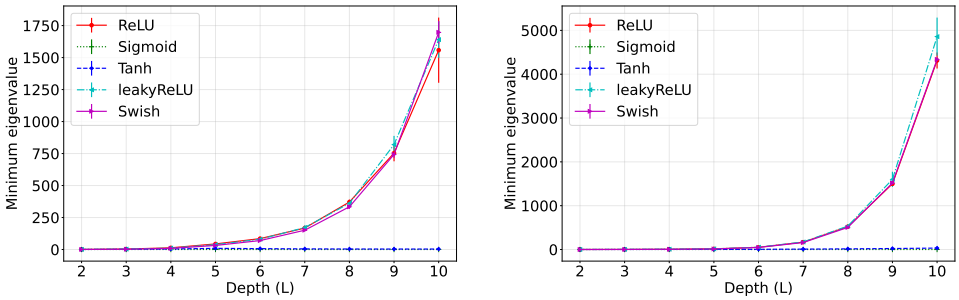
Table 6 provides a comparison of the accuracy of Eigen-NAS, KNAS and NASWOT on four new search spaces. For all of four search spaces, our method achieves the best results with $1\% - 2\%$ accuracy improvement.

Moreover, we conduct more detailed experiments using the CIFAR-10 dataset on NAS-Bench-101. Table 7 provides the running time and Kendall rank correlation coefficient between minimum eigenvalues and accuracy for the above three train-free NAS algorithm. We can see that our Eigen-NAS method can get the best rank correlation coefficient with the fastest speed among three methods.



(a) no skip connections (b) skip connections

Figure 3: Minimum eigenvalue of NTK *vs.* depth (L) under various activation functions with/without skip connections in each layer.



(a) the first half has a skip layer (b) the second half has a skip layer

Figure 4: Minimum eigenvalue of NTK *vs.* depth (L) under various activation functions with/without skip connections in each layer.

The scatter plot of the relationship between the minimum eigenvalue and the accuracy is shown in Figure 6.

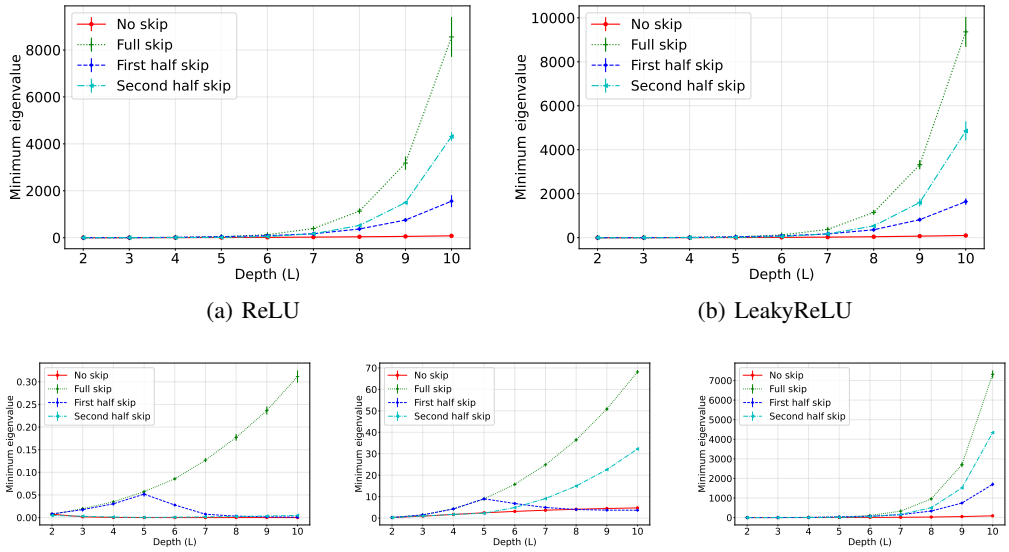
F.6 Transfer learning experiment

Here we evaluate the proposed NAS framework on transfer learning. The algorithm from sec. 5.1 is employed for this experiment, e.g., the same search space and search strategy. The experiment setting is the following: we train the model on FashionMNIST for 20 epochs, then we use the pretrained weights and fine-tune them for 5 epochs on MNIST, with repeated three times.

Table 8 show that, after the fine-tuning for just 2 epochs, the method obtains up to 95% accuracy and after fine-tuning for 5 epochs it obtains up to 97% accuracy. This verifies our intuition that the proposed NAS framework can obtain architectures that generalize well beyond the dataset they were optimized on.

Table 6: New results on NAS-Benchmark-101, NDS-DARTS and NDS-PNAS using CIFAR-10 and ImageNette2, a subset of ImageNet.

Benchmark	NAS-Bench-101	NDS-DARTS	NDS-PNAS	NDS-PNAS
Dataset	CIFAR-10	CIFAR-10	CIFAR-10	ImageNette2
Eigen-NAS ($k = 20$)	**92.7**%	**92.6**%	**93.8**%	**69.2**%
KNAS ($k = 20$)	91.7%	90.1%	91.7%	67.3%
NASWOT	91.3%	90.6%	93.3%	68.4%



(a) ReLU (b) LeakyReLU

(c) Sigmoid (d) Tanh (e) Swish

Figure 5: Minimum eigenvalue of NTK for different activation function. The red line have skip connections in each layer, green line does not contain any skip connections, the blue line represents the skip connections in the first half and the cyan line represents the skip connections in the second half.

Table 7: Running time (in Second) and the Kendall rank correlation coefficient on NAS-Bench-101, CIFAR-10 (the larger the absolute value of Rank correlation, the stronger the correlation between the guide used by the algorithm and the network accuracy).

Method	Eigen-NAS ($k = 20$)	KNAS ($k = 20$)	NASWOT
Running time	**1136**	1967	1468
Rank correlation	$\mathbf{-0.355}$	0.309	-0.313

F.7 DARTS experiment on CNN

Our theory relies on fully-connected matrices and we have indeed verified experimentally the validity of our theoretical findings. To scrutinize our method even further, we attempt to extend our results to the popular convolutional neural networks. We believe this will provide some further insights on future extensions of our theory. In particular, we use DARTS (similarly with the experiment in sec. 5.1) with convolutional layers. The standard dataset of CIFAR-10 is selected; the details of the dataset are shared in Appendix F.1. The search space and search strategy follow sec. 5.1 with one differentiating point: we use convolutional layers instead of fully connected layers in Eq. (2).

We select DARTS on a Convolutional Neural Network with $L = 10$ and $m = 1024$, while we repeat the experiment for 5 times. After training, the probability of these activation functions and skip connections in each layer are reported in Figure 7(a) and 7(b), respectively. Compared with the Figure 1, the activation function search exhibits similar characteristics with the results of the fully connected network. Namely: (1) ReLU and LeakyReLU have the highest probability to be selected, (2) the difference of probability between different activation functions in the first layer is the largest.

Table 8: Transfer learning result of our network for different width (m) which training in FashionMNIST (domain dataset) for 20 epochs and then training in MNIST (target dataset) for 2 or 5 epochs. (the accuracy in the table are displayed in percentages)

Epochs	$m = 64$	$m = 128$	$m = 256$	$m = 512$	$m = 1024$
$20 + 2$	94.13 ± 0.64	95.18 ± 0.25	94.73 ± 0.22	94.40 ± 0.80	$\mathbf{95.41} \pm 0.03$
$20 + 5$	95.73 ± 0.28	96.12 ± 0.32	96.73 ± 0.29	96.73 ± 0.11	$\mathbf{96.96} \pm 0.22$

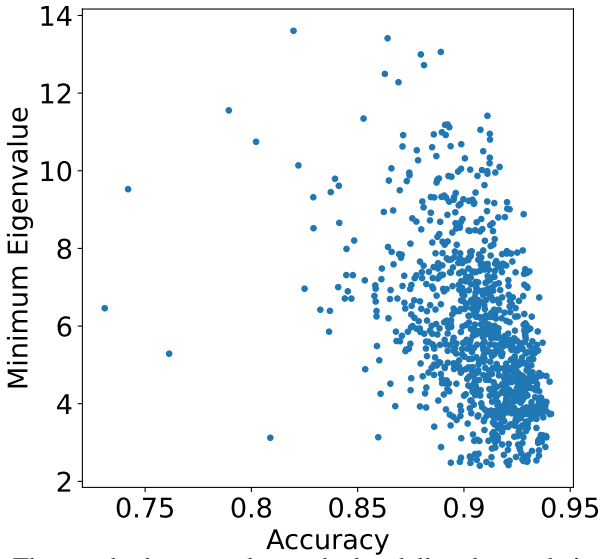
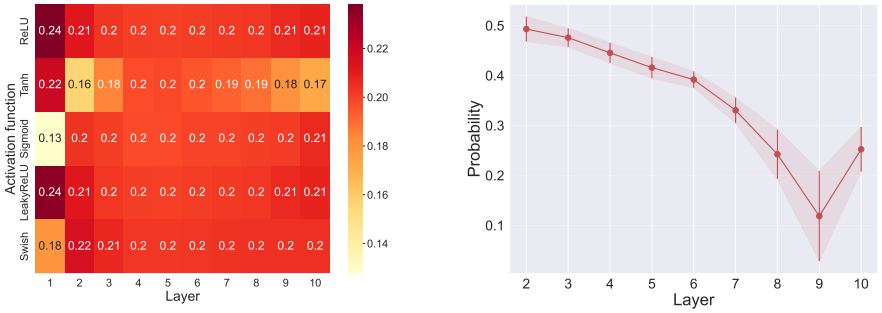


Figure 6: The standard scatter plot on the kendall rank correlation coefficient.



(a) activation functions $\boldsymbol{\sigma}$

(b) skip connections $\boldsymbol{\alpha}$

Figure 7: Architecture search results on activation functions indicated by the probability of $\boldsymbol{\sigma}$ in (a) and skip connections indicated by $\boldsymbol{\alpha}$ in (b). We notice that for each layer, ReLU and LeakyReLU are selected with the higher probability.

But for skip layer search, CNN exhibits the opposite results with fully connected network, that is, almost all of the skip connections have a probability of being selected less than 50%.

Based on the above results, our theory can still explain some of the phenomena observed in CNNs, e.g., activation functions search. Nevertheless, our theory on skip connections search on CNNs mismatches with experimental demonstration in practice to some extent, which motivates us to conduct a refined analysis on CNNs for NAS.

F.8 β-DARTS experiment on MLP

In this section, we use an improved DARTS-based algorithm, β-DARTS [Ye et al., 2022], for doing the activation function search. Our experiments are performed on a 5-layers MLP and the experimental results are presented in Figure 8. Compared with the results of DARTS in Figure 1, the experimental results of β-DARTS indicate that the probability difference between different activation functions is smaller, which may verify that DART is more easily to overfit . This is also the advantage mentioned in the β-DARTS paper.

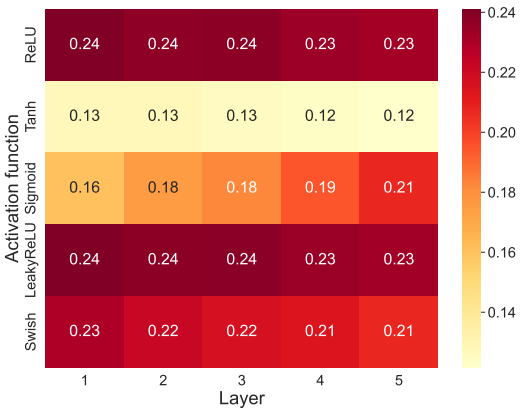


Figure 8: Architecture search results using β-DARTS on activation functions indicated by the probability of $\boldsymbol{\sigma}$. We notice that for each layer, ReLU and LeakyReLU are selected with the higher probability.

G Societal impact

This is a theoretical work that derived generalization bounds for the architectures obtained by NAS. As such, we do not expect our work to have negative societal bias, as we do not focus on obtaining state-of-the-art results in a particular task. On the contrary, our work can have various benefits for the community:

- We provide the first generalization bounds for the class of NAS architectures, which is expected to have a positive impact on the understanding and the application of such architectures.
- As we illustrate in sec. 5, we can use the minimum eigenvalue as a promising metric to guide NAS. This can lead to further investigation on techniques for efficient evaluation of NAS by avoiding solving the intensive bi-level optimization of NAS explicitly.

Nevertheless, we encourage researchers to further investigate the impact of different architectures and their inductive biases on the society.