# Generalization Guarantees for Neural Architecture Search with Train-Validation Split

**Samet Oymak** [1] **Mingchen Li** [2] **Mahdi Soltanolkotabi** [3]

## Abstract

Neural Architecture Search (NAS) is a popular method for automatically designing optimized deep-learning architectures. NAS methods commonly use bilevel optimization where one optimizes the weights over the training data (lower-level problem) and hyperparameters - such as the architecture - over the validation data (upper-level problem). This paper explores the statistical aspects of such problems with train-validation splits. In practice, the lower-level problem is often overparameterized and can easily achieve zero loss. Thus, a-priori, it seems impossible to distinguish the right hyperparameters based on training loss alone which motivates a better understanding of train-validation split. To this aim, we first show that refined properties of the validation loss such as risk and hyper-gradients are indicative of those of the true test loss and help prevent overfitting with a near-minimal validation sample size. Importantly, this is established for continuous search spaces which are relevant for differentiable search schemes. We then establish generalization bounds for NAS problems with an emphasis on an activation search problem and gradient-based methods. Finally, we show rigorous connections between NAS and low-rank matrix learning which leads to algorithmic insights where the solution of the upper problem can be accurately learned via spectral methods to achieve near-minimal risk.

## 1. Introduction

Hyperparameter optimization (HPO) is a critical component of modern machine learning pipelines. It is particularly important for deep learning applications where there are many possibilities for choosing a variety of hyperparameters to achieve the best test accuracy. A crucial application of HPO is Neural Architecture Search (NAS) which aims to find the most suitable architecture in an automated manner without extensive user trial and error. HPO/NAS problems are often formulated as bilevel optimization problems and critically rely on a train-validation split of the data, where the parameters of the model (e.g. network weights) and the hyperparameters are optimized over the training data (lower-level problem) and validation data (upper-level problem) respectively. With an ever growing number of configurations/architecture choices in modern learning problems, there has been a surge of interest in differentiable HPO methods that focus on continuous relaxations. For instance, differentiable architecture search schemes learn continuously parameterized architectures which are discretized only at the end of the training (Liu et al., 2018). Similar techniques have also been applied to learn data-augmentation policies (Cubuk et al., 2020) and meta-learning (Franceschi et al., 2018; Finn et al., 2017). These differentiable algorithms are often faster and seamlessly scale to millions of hyperparameters (Lorraine et al., 2020). However, the generalization capability of HPO/NAS with such large search spaces and the benefits of the train-validation split remain largely mysterious.

Addressing the above challenge is particularly important in modern overparameterized learning regimes where the training loss is often not indicative of the model's performance as large capacity networks can effortlessly fit to training data and achieve zero loss. To be concrete, let $n_{\mathcal{T}}$ and $n_{\mathcal{V}}$ denote the training and validation sample sizes and $p$ and $h$ the number parameters and hyperparmeters of the model. HPO/NAS problems typically operate in a regime where

$$\boxed{p := \#\text{ params} \geq n_{\mathcal{T}} \geq n_{\mathcal{V}} \geq h := \#\text{ hyperparams}} \quad (1)$$

Figure 1 depicts such a regime (e.g. when $p \gg \text{poly}(n_{\mathcal{T}})$) where the neural network model is in fact expressive enough to perfectly fit the dataset *for all possible combinations of hyperparameters*. Nevertheless, training with a train-validation split tends to select the right hyperparameters where the corresponding network achieves stellar test accuracy. This leads us to the main challenge of this paper[1]:

[1]Department of Electrical and Computer Eng., University of California, Riverside. [2]Department of Computer Science and Eng., University of California, Riverside. [3]Ming Hsieh Department of Electrical Eng., University of Southern California.. Correspondence to: Samet Oymak <soymak@ucr.edu>, Mingchen Li <mli176@ucr.edu>, Mahdi Soltanolkotabi <soltanol@usc.edu>.

*Proceedings of the $38^{th}$ International Conference on Machine Learning*, PMLR 139, 2021. 

[1]While we do provide guarantees for generic HPO problems

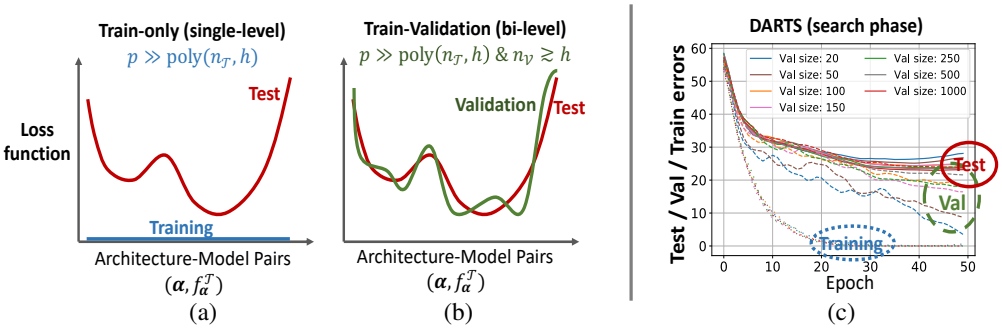



*Figure 1.* This figure depicts the typical scenario arising in modern HPO/NAS problems per Eq. 1. In Figure (a), the training loss is not indicative of the test loss as an overparameterized network can perfectly fit the training data for all choices of (continuously-parameterized) architectures. However, with train-validation split in Figure (b), the validation loss uniformly concentrates around the test loss and helps discover the best architecture. This paper rigorously establishes this phenomena (e.g., see our Theorem 3). Figure (c) shows NAS experiments with DARTS during the search phase. We evaluate train/test/validation losses of the continuously-parameterized supernet with $h = 224$ hyperparameters. We observe that training error is consistently zero after 30 epochs, whereas validation error almost perfectly tracks test error as soon as the validation size is mildly large (e.g.$\geq 250$), which is consistent with Figures (a), (b) and our theory.

*How does train-validation split for NAS/HPO over large continuous search spaces discover near-optimal hyperparameters despite overparameterization?*

To this aim, in this paper, we explore the statistical aspects of NAS with train-validation split and provide theoretical guarantees to explain its generalization capability in the practical data/parameter regime of (1). Specifically, our contributions and the basic outline of the paper are as follows:

• **Generalization with Train-Validation Split (Sec. 3):** We provide general-purpose uniform convergence arguments to show that refined properties of the validation loss (such as risk and hyper-gradients) are indicative of the test-time properties. This is shown when the lower-level training problem is optimized by an algorithm which is (approximately) Lipschitz with respect to the hyperparameters. Our result applies as soon as the validation sample size scales proportionally with the *effective dimension of the hyperparameter space* and only logarithmically in this Lipschitz constant. We then utilize this result to obtain an end-to-end generalization bound for bilevel optimization under generic conditions which are then verified for neural nets.

• **Generalization Guarantees for NAS** are established in Section 4. Specifically, we first develop results for a *neural activation search* problem that aims to determine the best activation function for shallow neural networks. We study this problem in connection to a *feature-map/kernel learning* problem involving the selection of the best feature-map among a continuously parameterized family. Furthermore, when the lower-level problem is optimized via gradient descent, we show that the *bilevel problem is guaranteed to select the activation that has the best generalization capability*. We then discuss extensions general deep architectures by similarly linking the NAS problem to the search for the optimal kernel function. With this link, we show how train-validation split achieves the best excess risk bound among all architectures using few validation samples and provide insights on the role of depth and width. Detailed results are deferred to the extended paper (Oymak et al., 2021).

• **Algorithmic Guarantees via Connection to Low-rank Learning (Section 5):** The results so far focus on generalization and are not fully algorithmic: they assume access to an approximate solution of the upper-level (validation) problem. This raises the question: Can one provably find such an approximate solution with a few validation samples and a computationally tractable algorithm? To this end, we connect the neural activation search problem to a novel low-rank matrix learning problem with an overparameterized dimension $p$. We then provide an algorithm to find the near-optimal hyperparameters via a spectral estimator that also achieves a near-optimal test risk. Interestingly, this holds as long as the matrix dimensions obey $h \times p \lesssim (n_{\mathcal{T}} + n_{\mathcal{V}})^2$ which allows for the regime (1). In essence, this shows that it is possible to tractably solve the upper problem in the regime of (1) even when the problem is potentially overfitting for all choices of hyperparameters, similar in spirit to NAS where even poor quality architectures can fit the data.

The proofs and refinements of our theorems, detailed results on deep architectures and further discussion of related works are deferred to the extended work (Oymak et al., 2021).

## 2. Preliminaries and Problem Formulation

We begin by introducing some notation used throughout the paper. We use $\boldsymbol{X}^\dagger$ to denote the Moore–Penrose inverse of a matrix $X$. $\gtrsim, \lesssim$ denote inequalities that hold up to an absolute constant. We define the norm $\|\cdot\|_{\mathcal{X}}$ over an input space $\mathcal{X}$ as $\|f\|_{\mathcal{X}} := \sup_{\boldsymbol{x}\in\mathcal{X}} |f(\boldsymbol{x})|$. $\widetilde{\mathcal{O}}(\cdot)$ implies equality up to constant/logarithmic factors. $c, C > 0$ are used to denote absolute constants. Finally, we use $\mathcal{N}_\varepsilon(\boldsymbol{\Delta})$ to denote

(cf. Sec. 3), the emphasis of this work is NAS and the search for the optimal architecture rather than broader class of hyperparameters.

an $\varepsilon$-Euclidean ball cover of a set $\mathbf{\Delta}$.

Throughout, we use $(\boldsymbol{x}, y) \sim \mathcal{D}$ with $\boldsymbol{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$, to denote the data distribution of the feature/label pair. We also use $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_\mathcal{T}}$ to denote the training dataset and $\mathcal{V} = \{(\widetilde{\boldsymbol{x}}_i, \widetilde{y}_i)\}_{i=1}^{n_\mathcal{V}}$ the validation dataset and assume $\mathcal{T}$ and $\mathcal{V}$ are drawn i.i.d. from $\mathcal{D}$. Given a loss function $\ell$ and a hypothesis $f : \mathcal{X} \to \mathcal{Y}$, we define the population risk and the empirical validation risk as follows

$$\mathcal{L}(f) = \mathrm{E}_\mathcal{D}[\ell(y, f(\boldsymbol{x}))],\ \widehat{\mathcal{L}}_\mathcal{V}(f) = \frac{1}{n_\mathcal{V}} \sum_{i=1}^{n_\mathcal{V}} \ell(\widetilde{y}_i, f(\widetilde{\boldsymbol{x}}_i)).$$

For binary classification with $y \in \{-1, 1\}$ also define the test classification error as $\mathcal{L}^{0-1}(f) = \mathrm{P}(yf(\boldsymbol{x}) \leq 0)$. We focus on a *bilevel empirical risk minimization* (ERM) problem over train/validation datasets involving a hyperparameter $\boldsymbol{\alpha} \in \mathrm{R}^h$ and a hypothesis $f$. Here, the model $f$ (which depends on the hyperparameter $\boldsymbol{\alpha}$) is typically trained over the training data $\mathcal{T}$ with the hyperparameters fixed (lower problem). Then, the best hyperparameter is selected based on the validation data (upper-level problem).

The lower problem typically solves an (possibly regularized) empirical risk of the form $\widehat{\mathcal{L}}_\mathcal{T}(f) = \frac{1}{n_\mathcal{T}} \sum_{i=1}^{n_\mathcal{T}} \ell(y_i, f(\boldsymbol{x}_i))$. In this work, we do not explicitly require a global optima of this empirical loss and assume that we have access to an algorithm $\mathcal{A}$ that returns a model based on the training data $\mathcal{T}$ given hyperparameters $\boldsymbol{\alpha}$. Specifically, this model is

$$f_{\boldsymbol{\alpha}}^\mathcal{T} = \mathcal{A}(\boldsymbol{\alpha}, \mathcal{T}).$$

We provide some example scenarios with the corresponding algorithms below.

*Scenario 1: Strongly Convex Problems.* The lower-level problem ERM is strongly convex with respect to the parameters of the model and $\mathcal{A}$ returns its unique solution. A specific example is learning the optimal kernel given a predefined set of kernels per §4.1.

*Scenario 2: Gradient Descent & NAS.* In NAS, $f$ is typically a neural network and $\boldsymbol{\alpha}$ encodes the network architecture. Given this architecture, $\mathcal{A}$ trains the weights of $f$ on dataset $\mathcal{T}$ by running fixed number of gradient descent iterations. See §4.2 for more details.

As mentioned earlier, modern NAS problems typically obey (1) where $h$ is typically less than 1000 and obeys $h = \dim(\boldsymbol{\alpha}) \leq n_\mathcal{V}$. Intuitively, this is the regime in which all lower-level problems have solutions perfectly fitting the data. However, as we will show, the under-parameterized upper problem can provably guide the algorithm towards the right model. To select the optimal model, given hyperparameter space $\mathbf{\Delta}$ and tolerance $\delta > 0$, the following Train-Validation Optimization (TVO) returns a $\delta$-approximate solution to the validation risk $\widehat{\mathcal{L}}_\mathcal{V}$ (upper problem)

$$\widehat{\boldsymbol{\alpha}} \in \{\boldsymbol{\alpha} \in \mathbf{\Delta} \;\big|\; \widehat{\mathcal{L}}_\mathcal{V}(f_{\boldsymbol{\alpha}}^\mathcal{T}) \leq \min_{\boldsymbol{\alpha} \in \mathbf{\Delta}} \widehat{\mathcal{L}}_\mathcal{V}(f_{\boldsymbol{\alpha}}^\mathcal{T}) + \delta\}. \quad \text{(TVO)}$$

# 3. Generalization with Train-Validation Split

This section provides our generic generalization bounds for train-validation split. Specifically, Section 3.1 introduce our result for generalization gap between the test and validation risk as well as the corresponding gradients. Section 3.2 provides a bound jointly capturing the role of training and validation. In Section 4, we utilize these bounds and further innovations to establish guarantees for NAS.

## 3.1. Low validation risk implies good generalization

Our first result connects the test (generalization) error to that of the validation error. A key aspect of our result is that we establish uniform convergence guarantees that hold over continuous hyperparameter spaces which is particularly insightful for differentiable HPO/NAS algorithms such as DARTS (Liu et al., 2018). Besides validation loss, we also establish the uniform convergence of the hyper-gradient $\nabla_{\boldsymbol{\alpha}} \widehat{\mathcal{L}}_\mathcal{V}(f_{\boldsymbol{\alpha}}^\mathcal{T})$ of the upper problem under similar assumptions. Such concentration of hyper-gradient is insightful for gradient-based bilevel optimization algorithms to solve (TVO). Specifically, we will answer how many validation samples are required so that upper-level problems (hyper-)gradient concentrates around its expectation. Our results rely on the following definition and assumptions.

**Definition 1 (Effective dimension)** *For a set $\mathbf{\Delta} \in \mathrm{R}^h$ of hyperparameters we define its effective dimension $h_{\mathit{eff}}$ as the smallest value of $h_{\mathit{eff}} > 0$ such that $|\mathcal{N}_\varepsilon(\mathbf{\Delta})| \leq (\bar{C}/\varepsilon)^{h_{\mathit{eff}}}$ for all $\varepsilon > 0$ and a constant $\bar{C} > 0$.*

The effective dimension captures the degrees of freedom of a set $\mathbf{\Delta}$. In particular, if $\mathbf{\Delta} \in \mathrm{R}^h$ has Euclidean radius $R$, then $h_{\text{eff}} = h$ with $\bar{C} = 3R$ so that it reduces to the number of hyperparameters. However, $h_{\text{eff}}$ is more nuanced and can also help incorporate problem structure/prior knowledge (e.g. sparse architectures have less degrees of freedom).[2]

**Assumption 1** *$\mathcal{A}(\cdot)$ is an $L$-Lipschitz function of $\boldsymbol{\alpha}$ in $\|\cdot\|_\mathcal{X}$ norm, that is, for all pairs $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in \mathbf{\Delta}$, we have $\|f_{\boldsymbol{\alpha}_1}^\mathcal{T} - f_{\boldsymbol{\alpha}_2}^\mathcal{T}\|_\mathcal{X} \leq L\|\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2\|_{\ell_2}$.*

**Assumption 2** *For all hypotheses $f_{\boldsymbol{\alpha}}^\mathcal{T}$, the loss $\ell(y, \cdot)$ is $\Gamma$-Lipschitz over the feasible set $\{f_{\boldsymbol{\alpha}}^\mathcal{T}(\boldsymbol{x}) \;\big|\; \boldsymbol{x} \in \mathcal{X}\}$. Additionally, $\ell(y, f_{\boldsymbol{\alpha}}^\mathcal{T}(\boldsymbol{x})) - \mathrm{E}[\ell(y, f_{\boldsymbol{\alpha}}^\mathcal{T}(\boldsymbol{x}))]$ has bounded subexponential norm with respect to the randomness in $(\boldsymbol{x}, y) \sim \mathcal{D}$.*

Assumption 1 is key to our NAS generalization analysis and we show it holds in a variety of scenarios. In general, we only need $\mathcal{A}$ to be approximately Lipschitz over a partitioning of the set $\mathbf{\Delta}$. Assumption 2 requires the loss or gradient on a sample $(\boldsymbol{x}, y)$ to have a sub-exponential tail. While these assumptions allow us to show that the validation error is indicative of the test error, the two additional assumptions (which parallel those above) allow us to show that the

[2] In the empirical process theory literature this is also known as the uniform entropy number e.g. see (Mendelson, 2003)[Def 2.5].

hyper-gradient is concentrated around gradient of the true loss with respect to the hyperparameters. As mentioned earlier such concentration of the hyper-gradient is insightful for gradient-based bilevel optimization algorithms.

**Assumption 1′** *For some $R \geq 1$ and all $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in \boldsymbol{\Delta}$ and $\boldsymbol{x} \in \mathcal{X}$, hyper-gradient obeys $\|\nabla_{\boldsymbol{\alpha}} f^{\mathcal{T}}_{\boldsymbol{\alpha}_1}(\boldsymbol{x})\|_{\ell_2} \leq R$ and $\|\nabla_{\boldsymbol{\alpha}} f^{\mathcal{T}}_{\boldsymbol{\alpha}_1}(\boldsymbol{x}) - \nabla_{\boldsymbol{\alpha}} f^{\mathcal{T}}_{\boldsymbol{\alpha}_2}(\boldsymbol{x})\|_{\ell_2} \leq RL\|\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2\|_{\ell_2}$.*

**Assumption 2′** *$\ell'(y, \cdot)$ is $\Gamma$-Lipschitz and the hyper-gradient noise $\nabla\ell(y, f^{\mathcal{T}}_{\boldsymbol{\alpha}}(\boldsymbol{x})) - \mathrm{E}[\nabla\ell(y, f^{\mathcal{T}}_{\boldsymbol{\alpha}}(\boldsymbol{x}))]$ over the example $(\boldsymbol{x}, y) \sim \mathcal{D}$ has bounded subexponential norm.*

Our first result establishes a generalization guarantee for (TVO) under these assumptions.

**Theorem 1** *Suppose Assumptions 1&2 hold. Let $\widehat{\boldsymbol{\alpha}}$ be an approximate minimizer of the validation risk per (TVO) and set $\bar{h}_{\mathit{eff}} := h_{\mathit{eff}} \log(\bar{C}L\Gamma n_{\mathcal{V}}/h_{\mathit{eff}})$. If $n_{\mathcal{V}} \geq \bar{h}_{\mathit{eff}} + \tau$ for some $\tau > 0$, with probability at least $1 - 2e^{-\tau}$, we have*

$$\sup_{\boldsymbol{\alpha}\in\boldsymbol{\Delta}} |\mathcal{L}(f^{\mathcal{T}}_{\boldsymbol{\alpha}}) - \widehat{\mathcal{L}}_{\mathcal{V}}(f^{\mathcal{T}}_{\boldsymbol{\alpha}})| \leq \sqrt{\frac{C(\bar{h}_{\mathit{eff}} + \tau)}{n_{\mathcal{V}}}}, \tag{2}$$

$$\mathcal{L}(f^{\mathcal{T}}_{\widehat{\boldsymbol{\alpha}}}) \leq \min_{\boldsymbol{\alpha}\in\boldsymbol{\Delta}} \mathcal{L}(f^{\mathcal{T}}_{\boldsymbol{\alpha}}) + 2\sqrt{\frac{C(\bar{h}_{\mathit{eff}} + \tau)}{n_{\mathcal{V}}}} + \delta. \tag{3}$$

*Suppose also Assumptions 1′ & 2′ hold and $n_{\mathcal{V}} \geq h + \bar{h}_{\mathit{eff}} + \tau$ for some $\tau > 0$. Then, with probability at least $1 - 2e^{-\tau}$, the hyper-gradient of the validation risk converges uniformly:*

$$\sup_{\boldsymbol{\alpha}\in\boldsymbol{\Delta}} \|\nabla\widehat{\mathcal{L}}_{\mathcal{V}}(f^{\mathcal{T}}_{\boldsymbol{\alpha}}) - \nabla\mathcal{L}(f^{\mathcal{T}}_{\boldsymbol{\alpha}})\|_{\ell_2} \leq \sqrt{\frac{C(h + \bar{h}_{\mathit{eff}} + \tau)}{n_{\mathcal{V}}}}. \tag{4}$$

This result shows that as soon as the size of the validation data exceeds the effective number of hyperparameters $n_{\mathcal{V}} \gtrsim h_{\text{eff}}$ (up to log factors) then as evident per (2) the test error is close to the validation error (i.e. validation error is indicative of the test error) and per (3) the optimization over validation is guaranteed to return a hypothesis on par with the best choice of hyperparameters in $\boldsymbol{\Delta}$. Theorem 1 has two key distinguishing features, over the prior art on cross-validation (Kearns et al., 1997; Kearns & Ron, 1999), which makes it highly relevant for modern learning problems. First, it applies to continuous hyperparameters and bounds the size of $\boldsymbol{\Delta}$ via the refined notion of effective dimension, establishing a logarithmic dependence on other problem parameters. This is particularly important for the Lipschitzness $L$ which can be rather large in practice. Second, besides the loss function, per (4) we also establish the uniform convergence of hyper-gradients. Once the validation loss satisfies favorable properties (e.g. Lojasiewicz condition), one can obtain generalization guarantees based on the stationary points of validation risk via (4) (Foster et al., 2018). Additionally, we note that (4) requires at least $h$ samples - which is the ambient dimension and greater than $h_{\text{eff}}$. This is unavoidable due to the vectorial nature of the (hyper)-gradient and is consistent with related results on uniform gradient concentration (Mei et al., 2018).

Finally, we note that the results above do not directly imply good generalization as they do not guarantee that the validation error ($\min_{\boldsymbol{\alpha}} \widehat{\mathcal{L}}_{\mathcal{V}}(f^{\mathcal{T}}_{\boldsymbol{\alpha}})$) or the generalization error ($\min_{\boldsymbol{\alpha}\in\boldsymbol{\Delta}} \mathcal{L}(f^{\mathcal{T}}_{\boldsymbol{\alpha}})$) of the model trained with the best hyperparameters is small. This is to be expected as when there are very few training data one can not hope for the model $f^{\mathcal{T}}_{\boldsymbol{\alpha}}$ to have good generalization even with optimal hyperparameters. However, whether the training phase is successful or not, the validation phase returns approximately the best hyperparameters even with a bad model! In the next section we do in fact show that with enough training data the validation/generalization of the model trained with the best hyperparameter is indeed small allowing us to establish an end-to-end generalization bound.

### 3.2. End-to-end bound with Train-Validation Split

To establish an end-to-end bound, firstly, we discuss the role of the training data by characterizing how the test risk of the algorithm $\mathcal{A}$ changes with the training data $n_{\mathcal{T}}$. Let us consider the population limit $n_{\mathcal{T}} \to +\infty$ and define the corresponding model for a given hyperparameter $\boldsymbol{\alpha}$

$$f^{\mathcal{D}}_{\boldsymbol{\alpha}} := \mathcal{A}(\boldsymbol{\alpha}, \mathcal{D}) := \lim_{n_{\mathcal{T}}\to\infty} \mathcal{A}(\boldsymbol{\alpha}, \mathcal{T}).$$

Classical learning theory results typically bound the difference between the population loss/risk of a model that is trained with finite training data ($\mathcal{L}(f^{\mathcal{T}}_{\boldsymbol{\alpha}})$) and the loss achieved by the idealized infinite data model ($\mathcal{L}(f^{\mathcal{D}}_{\boldsymbol{\alpha}})$) in terms of an appropriate complexity measure of the class and the size of the training data. In particular, for a specific choice of the hyperparameter $\boldsymbol{\alpha}$, based on classical learning theory (Bartlett & Mendelson, 2002)) a typical behavior is to have

$$\mathcal{L}(f^{\mathcal{T}}_{\boldsymbol{\alpha}}) \leq \mathcal{L}(f^{\mathcal{D}}_{\boldsymbol{\alpha}}) + \frac{\mathcal{C}^{\mathcal{T}}_{\boldsymbol{\alpha}} + C_0\sqrt{t}}{\sqrt{n_{\mathcal{T}}}}, \tag{5}$$

with probability at least $1 - e^{-t}$. Here, $\mathcal{C}^{\mathcal{T}}_{\boldsymbol{\alpha}}$ is a dataset-dependent complexity measure for the hypothesis set of the lower-level problem and $C_0$ is a positive scalar. We are now ready to state our end-to-end bound which ensures a bound of the form (5) holds simultaneously for all choices of hyperparameters $\boldsymbol{\alpha} \in \boldsymbol{\Delta}$.

**Proposition 1 (Train-validation bound)** *Consider the setting of Theorem 1 and for any fixed $\boldsymbol{\alpha} \in \boldsymbol{\Delta}$ assume (5) holds. Also assume $f^{\mathcal{D}}_{\boldsymbol{\alpha}}$ (in $\|\cdot\|_{\mathcal{X}}$ norm) and $\mathcal{C}^{\mathcal{T}}_{\boldsymbol{\alpha}}$ have bounded Lipschitz constants with respect to $\boldsymbol{\alpha}$ over $\boldsymbol{\Delta}$. Then with probability at least $1 - 3e^{-t}$ over the train $\mathcal{T}$ and validation $\mathcal{V}$ datasets*

$$\mathcal{L}(f^{\mathcal{T}}_{\widehat{\boldsymbol{\alpha}}}) \leq \min_{\boldsymbol{\alpha}\in\boldsymbol{\Delta}} \left(\mathcal{L}(f^{\mathcal{D}}_{\boldsymbol{\alpha}}) + \frac{\mathcal{C}^{\mathcal{T}}_{\boldsymbol{\alpha}}}{\sqrt{n_{\mathcal{T}}}}\right) + \sqrt{\frac{\widetilde{\mathcal{O}}(h_{\mathit{eff}} + t)}{n_{\mathcal{V}}}} + \delta.$$

In a nutshell, the above bound shows that the generalization error of a model trained with train-validation split is on par with the best train-only generalization achievable by picking the best hyperparameter $\boldsymbol{\alpha} \in \boldsymbol{\Delta}$. The only loss incurred is an extra $\sqrt{h_{\text{eff}}/n_{\mathcal{V}}}$ term which is vanishingly small as soon as the validation data is sufficiently larger than the effective dimension of the hyperparameters. We note that the Lipschitzness condition on $f_{\boldsymbol{\alpha}}^{\mathcal{D}}$ and $\mathcal{C}_{\boldsymbol{\alpha}}^{\mathcal{T}}$ can be relaxed.

## 4. Feature Maps and Shallow Networks

This section provides our main results on neural architecture/activation search which utilize the generalization bounds above. We first introduce the feature map selection problem (Khodak et al., 2019), which can be connected to NAS and kernel selection problems (Gonen & Alpaydin, 2011). Building on our findings on feature maps/kernels, Sec. 4.2 provides our results on activation search for shallow networks and discusses extensions to deep architectures.

### 4.1. Feature map selection for kernel learning

Below, the hyperparameter $\boldsymbol{\alpha} \in \mathrm{R}^{h+1}$ controls both the choice of the feature map and the regularization strength.

**Definition 2 (Feature Map Selection)** *Suppose we are given $h$ feature maps $\phi_i : \mathcal{X} \to \mathrm{R}^p$. Define the superposition $\phi_{\boldsymbol{\alpha}}(\cdot) = \sum_{i=1}^{h} \boldsymbol{\alpha}_i \phi_i(\cdot)$. Given training data $\mathcal{T}$, the algorithm $\mathcal{A}$ solves ridge regression with feature matrix $\boldsymbol{\Phi}_{\boldsymbol{\alpha}} := \boldsymbol{\Phi}_{\boldsymbol{\alpha}}^{\mathcal{T}}$ via*

$$\boldsymbol{\theta}_{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{\Phi}_{\boldsymbol{\alpha}}\boldsymbol{\theta}\|_{\ell_2}^2 + \boldsymbol{\alpha}_{h+1}\|\boldsymbol{\theta}\|_{\ell_2}^2 \tag{6}$$

$$\text{where} \quad \boldsymbol{\Phi}_{\boldsymbol{\alpha}} = [\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_1)\ \phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_2)\ \ldots\ \phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{n_{\mathcal{T}}})]^T. \tag{7}$$

*Here $\boldsymbol{\alpha}_{h+1} \in [\lambda_{\min}, \lambda_{\max}] \subset \mathrm{R}^+ \cup \{0\}$ controls the regularization strength. We then solve for optimal choice $\widehat{\boldsymbol{\alpha}}$ via* (TVO) *with hypothesis $f_{\boldsymbol{\alpha}}^{\mathcal{T}}(\boldsymbol{v}) = \boldsymbol{v}^T\boldsymbol{\theta}_{\boldsymbol{\alpha}}$.*

This problem formulation models weight-sharing which has been a key ingredient of the state-of-the-art NAS algorithms (Pham et al., 2018; Li et al., 2020). In essence, for NAS, the parameter $\boldsymbol{\theta}$ corresponds to the (super)network's weights and the feature maps $\boldsymbol{\Phi}_{\boldsymbol{\alpha}}$ will be induced by different architecture choices so that the formulation above can be viewed as the simplest of NAS problems with linear networks. Nevertheless, as we will see in the forthcoming sections this analysis serves as a stepping stone for more complex NAS problems. To apply Theorem 1 to this problem we need to verify its assumptions and characterize $\bar{h}_{\text{eff}}$.

**Lemma 1** *Suppose the feature maps and labels are bounded i.e. $\sup_{\boldsymbol{x}\in\mathcal{X}, 1\le i\le h} \|\phi_i(\boldsymbol{x})\|_{\ell_2}^2 \le B$ and $|y| \le 1$. Also assume the loss $\ell$ is bounded and 1-Lipschitz w.r.t. the model output. Set $\lambda_0 = \lambda_{\min} + \inf_{\boldsymbol{\alpha}\in\boldsymbol{\Delta}} \sigma_{\min}^2(\boldsymbol{\Phi}_{\boldsymbol{\alpha}}) > 0$. Additionally let $\boldsymbol{\Delta}$ be a convex set with $\ell_1$ radius $R \ge 1$. Then, Theorem 1 holds with $\bar{h}_{\text{eff}} = (h+1)\log(20R^3Bn_{\mathcal{T}}^2\lambda_0^{-2}(Bn_{\mathcal{T}}+1))$.*

An important component of the proof of this lemma is that we show that when $\lambda_0 > 0$, $f_{\boldsymbol{\alpha}}$ is a Lipschitz function of $\boldsymbol{\alpha}$ and Theorem 1 applies. Thus per (TVO) in this setting one can provably and jointly find the optimal feature map and the optimal regularization strength as soon as the size of the validation exceeds the number of hyperparameters.

We note that there are two different mechanisms by which we establish Lipschitzness w.r.t. $\boldsymbol{\alpha}$ in the above theorem. When $\lambda_{\min} > 0$, the lower problem is strongly-convex with respect to the model parameters. As we show in the next lemma, this is more broadly true for any training procedure which is based on minimizing a loss which is strongly convex with respect to the model parameters.

**Lemma 2** *Let $\boldsymbol{\Delta}$ be a convex set. Suppose $f_{\boldsymbol{\alpha}}$ is parameterized by $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ where $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ is obtained by minimizing a loss function $\bar{\mathcal{L}}_{\mathcal{T}}(\boldsymbol{\alpha}, \boldsymbol{\theta}) : \boldsymbol{\Delta} \times \mathrm{R}^p \to \mathrm{R}$. Suppose $\bar{\mathcal{L}}_{\mathcal{T}}(\boldsymbol{\alpha}, \boldsymbol{\theta})$ is $\mu$ strongly-convex in $\boldsymbol{\theta}$ and $\bar{L}$ smooth in $\boldsymbol{\alpha}$. Then $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ is $\sqrt{\bar{L}/\mu}$-Lipschitz in $\boldsymbol{\alpha}$.*

Importantly, Lemma 1 can also operate in the ridgeless regime ($\lambda_{\min} = 0$) even when the training loss is not strongly convex. This holds as long as the feature maps are not poorly-conditioned in the sense that

$$\inf_{\boldsymbol{\alpha}\in\boldsymbol{\Delta}} \sigma_{\min}\left(\boldsymbol{\Phi}_{\boldsymbol{\alpha}}\boldsymbol{\Phi}_{\boldsymbol{\alpha}}^T\right) = \lambda_0 > 0. \tag{8}$$

We remark that the exact value of $\lambda_0$ is not too important as our bound only depends logarithmically on this quantity. In what follows, we focus on ridgeless regression with an emphasis on neural nets which can often generalize well in an overparameterized regime without any regularization despite perfectly interpolating the training data.

Our next result utilizes Proposition 1 to provide an end-to-end generalization bound for feature map selection involving both training and validation sample sizes.

**Theorem 2** *Consider the setting of Def. 2 with $\boldsymbol{\alpha}_{h+1} = 0$. Assume that $\sup_{\boldsymbol{x}\in\mathcal{X}, 1\le i\le h} \|\phi_i(\boldsymbol{x})\|_{\ell_2}^2 \le B$ and the validation loss function $\ell$ is $\Gamma$-Lipschitz and bounded. Suppose* (8) *holds with probability at least $1 - p_0$ and $p \ge n_{\mathcal{T}} \ge n_{\mathcal{V}} \gtrsim h\log(M)$ with $M = 30R^4B^2\lambda_0^{-2}\Gamma(n_{\mathcal{T}}^2 + n_{\mathcal{V}}^2)\|\boldsymbol{y}\|_{\ell_2}$ where $R = \sup_{\boldsymbol{\alpha}\in\boldsymbol{\Delta}} \|\boldsymbol{\alpha}\|_{\ell_1}$. Define the label vector $\boldsymbol{y} = [y_1\ y_2\ \ldots\ y_{n_{\mathcal{T}}}]$. Then with probability at least $1 - 4e^{-t} - p_0$, the population risk (over $\mathcal{D}$) obeys*

$$\mathcal{L}(f_{\widehat{\boldsymbol{\alpha}}}) \le \min_{\boldsymbol{\alpha}\in\boldsymbol{\Delta}} 2\Gamma\sqrt{\frac{B\boldsymbol{y}^T\boldsymbol{K}_{\boldsymbol{\alpha}}^{-1}\boldsymbol{y}}{n_{\mathcal{T}}}} + C\sqrt{\frac{h\log(M) + \tau}{n_{\mathcal{V}}}} + \delta.$$

This theorem shows that for the feature map selection problem, bilevel optimization via a train-validation split returns a generalization guarantee on par with that of the best feature map (minimizing the excess risk) as soon as the size of the validation data exceeds the number of hyperparameters.

### 4.2. Activation search for shallow networks

In this section we focus on an *activation search* problem where the goal is to find the best activation among a parameterized family of activations for training a shallow neural networks based on a train-validation split. To this aim we consider a one-hidden layer network of the form $\boldsymbol{x} \mapsto f_{\text{nn}}(\boldsymbol{x}) = \boldsymbol{v}^T\sigma(\boldsymbol{W}\boldsymbol{x})$ and focus on a binary classification task with $y \in \{-1, +1\}$ labels. Here, $\sigma : \mathrm{R} \to \mathrm{R}$ denotes the activation, $\boldsymbol{W} \in \mathrm{R}^{k\times d}$ input-to-hidden weights, and $\boldsymbol{v} \in \mathrm{R}^d$ hidden-to-output weights. We focus on the case where the activation belongs to a family of activations of the form $\sigma_{\boldsymbol{\alpha}} = \sum_{i=1}^h \boldsymbol{\alpha}_i\sigma_i$ with $\boldsymbol{\alpha} \in \boldsymbol{\Delta}$ denoting the hyperparameters. Here, $\{\sigma_i\}_{i=1}^h$ are a list of candidate activation functions (e.g., ReLU, sigmoid, Swish). The neural net with hyperparameter $\boldsymbol{\alpha}$ is thus given by $f_{\text{nn},\boldsymbol{\alpha}}(\boldsymbol{x}) = \boldsymbol{v}^T\sigma_{\boldsymbol{\alpha}}(\boldsymbol{W}\boldsymbol{x})$. For simplicity of exposition, we will use the input layer for training thus the training weights are $\boldsymbol{W}$ with dimension $p = \dim(\boldsymbol{W}) = k \times d$ and fix $\boldsymbol{v}$ to have $\pm\sqrt{c_0/k}$ entries (half of each) for some $c_0 > 0$.

**TVO for shallow activation search:** We now explain the specific gradient-based algorithm for the lower-level optimization problem. For a fixed hyperparameter $\boldsymbol{\alpha}$, the lower-level optimization aims to minimize a quadratic loss over the training data of the form

$$\widehat{\mathcal{L}}_{\mathcal{T}}(\boldsymbol{W}) = \frac{1}{2}\sum_{i=1}^{n_{\mathcal{T}}}(y_i - f_{\text{nn},\boldsymbol{\alpha}}(\boldsymbol{x}_i, \boldsymbol{W}))^2. \tag{9}$$

To this aim, for a fixed hyperparameter $\boldsymbol{\alpha} \in \boldsymbol{\Delta}$, starting from a random initialization of the form $\boldsymbol{W}_0 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ we run gradient descent updates of the form $\boldsymbol{W}_{\tau+1} = \boldsymbol{W}_\tau - \eta\nabla\widehat{\mathcal{L}}_{\mathcal{T}}(\boldsymbol{W}_\tau)$ for $T$ iterations. Thus, the lower algorithm $\mathcal{A}$ returns the model

$$f^{\mathcal{T}}_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \boldsymbol{v}^T\sigma_{\boldsymbol{\alpha}}(\boldsymbol{W}_T\boldsymbol{x}).$$

We then solve for the $\delta$-approximate optimal activation $\widehat{\boldsymbol{\alpha}}$ via (TVO) by optimizing validation using $\ell = \textit{hinge loss}$.

To state our generalization guarantee, we need a few definitions. First, we introduce neural feature maps induced by the Neural Tangent Kernel (NTK) (Jacot et al., 2018).

**Definition 3 (Neural feature maps)** *Let $f_{nn,\boldsymbol{\alpha}}(\cdot, \boldsymbol{\theta})$ be a neural net parameterized by weights $\boldsymbol{\theta} \in \mathrm{R}^p$ and architecture $\boldsymbol{\alpha}$. Define $\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \frac{\partial f_{nn,\boldsymbol{\alpha}}(\boldsymbol{x})}{\partial \boldsymbol{\theta}_0}$ to be the* neural feature map *at the random initialization $\boldsymbol{\theta}_0 \sim \mathcal{D}_{init}$. Define the neural feature matrix $\boldsymbol{\Phi}_{\boldsymbol{\alpha}} = [\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_1) \ \dots \ \phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_{n_{\mathcal{T}}})]^T \in \mathrm{R}^{n_{\mathcal{T}}\times p}$ as in (7) i.e.*

$$\boldsymbol{\Phi}_{\boldsymbol{\alpha}} = \left[\frac{\partial f_{nn,\boldsymbol{\alpha}}(\boldsymbol{x}_1)}{\partial \boldsymbol{\theta}_0} \ \dots \ \frac{\partial f_{nn,\boldsymbol{\alpha}}(\boldsymbol{x}_n)}{\partial \boldsymbol{\theta}_0}\right]^T. \tag{10}$$

*We define the gram matrix as $\widehat{\boldsymbol{K}}_{\boldsymbol{\alpha}} = \boldsymbol{\Phi}_{\boldsymbol{\alpha}}\boldsymbol{\Phi}_{\boldsymbol{\alpha}}^T \in \mathrm{R}^{n_{\mathcal{T}}\times n_{\mathcal{T}}}$ with $(i, j)$th entry equal to $\langle\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_i), \phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_j)\rangle$ and NTK matrix is as $\boldsymbol{K}_{\boldsymbol{\alpha}} = \mathrm{E}_{\boldsymbol{\theta}_0}[\widehat{\boldsymbol{K}}_{\boldsymbol{\alpha}}]$.*

Neural feature maps are in general nonlinear functions of $\boldsymbol{\alpha}$. However, in case of shallow networks, they are nicely additive and obey $\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}_i) = \sum_{i=1}^h \boldsymbol{\alpha}_i\phi_1(\boldsymbol{x}_i)$, regardless of $\boldsymbol{\theta}_0$, establishing a link to Def. 2. The next assumption ensures the expressivity of the NTK to interpolate the data and enables us to establish regularization-free bounds.

**Assumption 3 (Expressive Neural Kernels)** *There exists $\lambda_0$>0 such that for any $\boldsymbol{\alpha} \in \boldsymbol{\Delta}$, NTK matrix $\boldsymbol{K}_{\boldsymbol{\alpha}} \succeq \lambda_0\boldsymbol{I}_{n_{\mathcal{T}}}$.*

This assumption is similar to (8) but we take expectation over random $\boldsymbol{\theta}_0$. Assumptions in a similar spirit to this are commonly used for the optimization/generalization analysis of neural nets, especially in the interpolating regime (Chizat et al., 2018; Du et al., 2018; Mei & Montanari, 2019). For fixed $\boldsymbol{\alpha}$, $\boldsymbol{K}_{\boldsymbol{\alpha}} \succ 0$ as long as no two training inputs are perfectly correlated and $\phi_{\boldsymbol{\alpha}}$ is analytic and not a polynomial (Du et al., 2019). The key aspect of our assumption is that we require the NTK matrices to be lower bounded for all $\boldsymbol{\alpha}$. In (Oymak et al., 2021), we also show this assumption can be overcome with a small ridge regularization.

With these definitions in place we are now ready to state our end-to-end generalization guarantee for Shallow activation search where the lower-level problem is optimized via gradient descent. Note that $\lambda_0$ and $\boldsymbol{K}_{\boldsymbol{\alpha}}$ scales linearly with initialization variance $c_0$. To be invariant to initialization, we will state our result in terms of the normalized eigen lower bound $\bar{\lambda}_0 = \lambda_0/c_0$ and kernel matrix $\bar{\boldsymbol{K}}_{\boldsymbol{\alpha}} = \boldsymbol{K}_{\boldsymbol{\alpha}}/c_0$.

**Theorem 3 (Neural activation search)** *Suppose input features have unit norm $\|\boldsymbol{x}\|_{\ell_2} = 1$ and labels are $\pm 1$. Pick $\boldsymbol{\Delta}$ to be a subset of the unit $\ell_1$ ball. Suppose Assumption 3 holds for $\boldsymbol{\theta}_0 \leftrightarrow \boldsymbol{W}_0$ and the candidate activations have first two derivatives ($|\sigma_i'|, |\sigma_i''|$) bounded by $B > 0$. Fix output weights $\boldsymbol{v}_i = \pm\sqrt{c_0/k}$ for a proper choice of $c_0$. Define the normalized lower bound $\bar{\lambda}_0 = \lambda_0/c_0$ and kernel matrix $\bar{\boldsymbol{K}}_{\boldsymbol{\alpha}} = \boldsymbol{K}_{\boldsymbol{\alpha}}/c_0$. Also assume the network width obeys*

$$k \gtrsim \textit{poly}(n_{\mathcal{T}}, \bar{\lambda}_0^{-1}, \varepsilon^{-1}).$$

*for a tolerance level $1 > \varepsilon > 0$ and the size of the validation data obeys $n_{\mathcal{V}} \gtrsim \widetilde{\mathcal{O}}(h)$. Consider (TVO) where lower-level (9) is optimized with a proper $\eta > 0$ choice and # of gradient iterations obeying $T \gtrsim \widetilde{\mathcal{O}}(\frac{n_{\mathcal{T}}}{\bar{\lambda}_0}\log(\varepsilon^{-1}))$. The classification error (0-1 loss) on the data distribution $\mathcal{D}$ obeys*

$$\mathcal{L}^{0-1}(f^{\mathcal{T}}_{\widehat{\boldsymbol{\alpha}}}) \le \min_{\boldsymbol{\alpha}\in\boldsymbol{\Delta}} 2B\sqrt{\frac{\boldsymbol{y}^T\bar{\boldsymbol{K}}_{\boldsymbol{\alpha}}^{-1}\boldsymbol{y}}{n_{\mathcal{T}}}} + C\sqrt{\frac{\widetilde{\mathcal{O}}(h) + t}{n_{\mathcal{V}}}} + \varepsilon + \delta,$$

*with probability at least $1 - 4(e^{-t} + n_{\mathcal{T}}^{-3} + e^{-10h})$ (over the randomness in $\boldsymbol{W}_0, \mathcal{T}, \mathcal{V}$). Here, $\boldsymbol{y} = [y_1 \ y_2 \ \dots \ y_{n_{\mathcal{T}}}]$. On the same event, for all $\boldsymbol{\alpha} \in \boldsymbol{\Delta}$, the training classification error obeys $\hat{\mathcal{L}}^{0-1}_{\mathcal{T}}(f^{\mathcal{T}}_{\boldsymbol{\alpha}}) \le \varepsilon$.*

For a fixed $\boldsymbol{\alpha}$, the norm-based excess risk term $\sqrt{\frac{\boldsymbol{y}^T\bar{\boldsymbol{K}}_{\boldsymbol{\alpha}}^{-1}\boldsymbol{y}}{n_{\mathcal{T}}}}$ quantifies the alignment between the kernel and the labeling function (which is small when $\boldsymbol{y}$ lies on the principal

eigenspace of $\boldsymbol{K}_{\boldsymbol{\alpha}}$). This generalization bound is akin to expressions that arise in norm-based NTK generalization arguments such as (Arora et al., 2019a). Critically, however, going beyond a fixed $\boldsymbol{\alpha}$, our theorem establishes this for all activations uniformly to conclude that the minimizer of the validation error also achieves minimal excess risk. The final statement of the theorem shows that the training error is arbitrarily small (essentially zero as $T \to \infty$) over all activations uniformly. Together, these results formally establish the pictorial illustration in Figures 1(a) & (b).

The proof strategy has two novelties with respect to standard NTK arguments. First, it requires a subtle uniform convergence argument on top of the NTK analysis to show that certain favorable properties that are essential to the NTK proof hold *uniformly* for all activations (i.e. choices of the hyperparameters) simultaneously with the same random initialization $\boldsymbol{W}_0$. Second, since neural nets may not obey Assumption 1, to be able to apply our generalization bounds we need to construct a uniform Lipschitz approximation via its corresponding linearized feature map ($f_{\text{lin},\boldsymbol{\alpha}}(\boldsymbol{x}) = \boldsymbol{x}^T \phi_{\boldsymbol{\alpha}}(\boldsymbol{x})$) and bound the neural net's risk over train-validation procedure in terms of this proxy. This uniform approximation is in contrast to pointwise approximation results of (Arora et al., 2019b).

**Extensions to deep architectures:** Section 5 of our extended work (Oymak et al., 2021) provides discussion and results for general multilayer architectures. Here, rather than gradient descent, we investigate training with the linearized neural features which is a good proxy for the optimization dynamics of wide nets. The main takeaway is that, a version of Theorem 3 still holds even if $\boldsymbol{\Phi}_{\boldsymbol{\alpha}}$ is a highly nonlinear function of $\boldsymbol{\alpha}$. We argue that Lipschitz constant $L$ is at most exponential in depth $D$, thus, depth costs at most a factor of $D$ validation samples. We also show the uniform concentration of the kernel matrix $\boldsymbol{K}_{\boldsymbol{\alpha}}$ over $\boldsymbol{\alpha} \in \boldsymbol{\Delta}$ once the *width* of the network is sufficiently large and discuss how Assumption 3 can be obviated via ridge regression.

## 5. Algorithmic Guarantees via Connection to Low-rank Matrix Learning

The results stated so far focus on generalization and are not fully algorithmic in nature in the sense that they assume access to an approximate solution of the upper-level problem per (TVO). In this section we wish to investigate whether it is possible to provably find such an approximate solution with a few validation samples and a computationally tractable algorithm. To this aim, we establish algorithmic connections between our activation/feature-map search problems of Section 4 and a rank-1 matrix learning problem. In Def. 2 –instead of studying $\boldsymbol{\Phi}_{\boldsymbol{\alpha}}$ for fixed $\boldsymbol{\alpha}$– let us consider the matrix of feature maps

$$\boldsymbol{X} = [\phi_1(\boldsymbol{x})\ \phi_2(\boldsymbol{x})\ \ldots\ \phi_h(\boldsymbol{x})]^T \in \mathrm{R}^{h \times p}$$

for a fixed input $\boldsymbol{x}$. Then, the population squared-loss risk of a $(\boldsymbol{\alpha}, \boldsymbol{\theta})$ pair predicting $\boldsymbol{\theta}^T \phi_{\boldsymbol{\alpha}}(\boldsymbol{x})$ is given by

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\theta}) := \mathrm{E}[(y - \boldsymbol{\alpha}^T \boldsymbol{X} \boldsymbol{\theta})^2] = \mathrm{E}[(y - \langle \boldsymbol{X}, \boldsymbol{\alpha}\boldsymbol{\theta}^T \rangle)^2].$$

Thus, the search for the optimal model pair $(\boldsymbol{\alpha}_\star, \boldsymbol{\theta}_\star)$ is simply a rank-1 matrix learning task with $\boldsymbol{M}_\star = \boldsymbol{\alpha}_\star \boldsymbol{\theta}_\star{}^T$. We ask: Can we learn a useful matrix in the regime (1)?

This is a rather subtle question as in the regime (1) there is not enough samples to reconstruct $\boldsymbol{M}_\star$ as anything algorithm regardless of computational tractability requires $n_{\mathcal{T}} + n_{\mathcal{V}} \gtrsim p + h$! But this of course does not rule out the possibility of finding an approximately optimal hyperparameter close to $\boldsymbol{\alpha}_\star$. To answer this –rather tricky question– we study a discriminative data model commonly used for modeling low-rank learning. Consider a realizable setup $y = \boldsymbol{\alpha}_\star{}^T \boldsymbol{X} \boldsymbol{\theta}_\star$ where we ignore noise for ease of exposition. We also assume that the feature matrix $\boldsymbol{X}$ has i.i.d. $\mathcal{N}(0,1)$ entries. Suppose we have $\mathcal{T} = (y_i, \boldsymbol{X}_i)_{i=1}^{n_{\mathcal{T}}}, \mathcal{V} = (\widetilde{y}_i, \tilde{\boldsymbol{X}}_i)_{i=1}^{n_{\mathcal{V}}}$ datasets with equal sample split $n = n_{\mathcal{T}} = n_{\mathcal{V}}$. If we combine these datasets into $\mathcal{T}$ and solve ERM, when $2n \le p$, for *any choice of* $\boldsymbol{\alpha}$, weights $\boldsymbol{\theta} \in \mathrm{R}^p$ can perfectly fit the labels. Instead, we propose a two-stage algorithm to achieve a near-optimal learning guarantee. Set $\hat{\boldsymbol{M}} = \sum_{i=1}^n \widetilde{y}_i \tilde{\boldsymbol{X}}_i$.

1. **Spectral estimation** : Set $\widehat{\boldsymbol{\alpha}} = \text{top\_eigen\_vec}(\hat{\boldsymbol{M}} \hat{\boldsymbol{M}}^T)$.

2. **Solve ERM on** $\mathcal{T}$ : Set $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^n (y_i - \widehat{\boldsymbol{\alpha}}^T \boldsymbol{X}_i \boldsymbol{\theta})^2$

We have the following guarantee for this procedure.

**Theorem 4 (Low-rank learning with $p > n$)** *Let $(\boldsymbol{X}_i, \tilde{\boldsymbol{X}}_i)_{i=1}^n$ be i.i.d. matrices with i.i.d. $\mathcal{N}(0,1)$ entries. Let $y_i = \boldsymbol{\alpha}_\star{}^T \boldsymbol{X}_i \boldsymbol{\theta}_\star$ for unit norm $\boldsymbol{\alpha} \in \mathrm{R}^h, \boldsymbol{\theta} \in \mathrm{R}^p$. Consider an asymptotic setting where $p, n, h$ grow to infinity with fixed ratios given by $\bar{p} = p/n > 1$, $\bar{h} = h/n < 1$ and consider the asymptotic performance of $(\widehat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}})$.*

*Let $1 \ge \rho_{\boldsymbol{\alpha}_\star, \widehat{\boldsymbol{\alpha}}} \ge 0$ be the correlation between $\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}$ i.e. $\rho_{\boldsymbol{\alpha}_\star, \widehat{\boldsymbol{\alpha}}} = |\boldsymbol{\alpha}^T \widehat{\boldsymbol{\alpha}}|$. Suppose $\bar{p}\bar{h} \le 1/6$. We have that*

$$\lim_{n \to \infty} \rho^2_{\boldsymbol{\alpha}_\star, \widehat{\boldsymbol{\alpha}}} \ge 1 - 64 \bar{p} \bar{h} \tag{11}$$

*Additionally, if $\bar{p}\bar{h} \le c$ for sufficiently small constant $c > 0$,*

$$\lim_{n \to \infty} \mathcal{L}(\widehat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}) \le \underbrace{1 - \frac{1}{\bar{p}}}_{\textit{risk}(\boldsymbol{\alpha}_\star)} + \underbrace{\frac{200\bar{h}}{1 - 1/\bar{p}}}_{\textit{estimating } \boldsymbol{\alpha}_\star}. \tag{12}$$

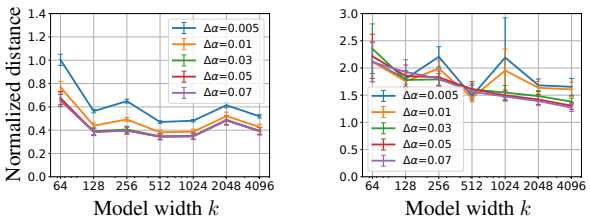
A few remarks are in order. First, the result applies in the regime $p \gg n$ as long as –the rather surprising condition– $ph \lesssim n^2$ holds (see (11)). Numerical experiments in appendix verify that this condition is indeed necessary. Here, $\text{risk}(\boldsymbol{\alpha}_\star) = 1 - n/p$ is the *exact asymptotic risk* one would achieve by solving ERM with the knowledge of optimal $\boldsymbol{\alpha}_\star$. Our result shows that one can approximately recover

this optimal $\boldsymbol{\alpha}_\star$ up to an error that scales with $ph/n^2$. Our second result achieves a near-optimal risk via $\widehat{\boldsymbol{\alpha}}$ without knowing $\boldsymbol{\alpha}_\star$. Since $1-1/\bar{p}$ is essentially constant, the risk due to $\boldsymbol{\alpha}_\star$-search is proportional to $\bar{h}=h/n$. This rate is consistent with Theorem 1 which would achieve a risk of $1-n/p+\widetilde{\mathcal{O}}(\sqrt{h/n})$. Remarkably, we obtain a slightly more refined rate ($h/n \leq \sqrt{h/n}$) using a spectral estimator with a completely different mathematical machinery based on high-dimensional learning. To the best of our knowledge, our spectral estimation result (11) in the $p>n$ regime is first of its kind (with a surprising $ph \lesssim n^2$ condition) and might be of independent interest. Finally, while this result already provides valuable algorithmic insights, it would be desirable to extend this result to general feature distributions to establish algorithmic guarantees for the original activation/feature map search problems.

## 6. Related Works

Our work establishes generalization guarantees for architecture search and is closely connected to the literature on deep learning theory, statistical learning, and NAS.

**Statistical learning:** The statistical learning theory provide rich tools for analyzing test performance of algorithms (Bartlett & Mendelson, 2002; Vapnik, 2006). Our discussion on learning with bilevel optimization and train-validation split connects to the model selection literature (Kearns, 1996; Kearns et al., 1997; Vuong, 1989) as well as the more recent works on architecture search (Khodak et al., 2020; 2019). The model selection literature is mostly concerned with controlling the model complexity (e.g. via nested hypothesis), which is not directly applicable to high-capacity deep nets. The latter two works are closer to us and also establish connections between feature maps and NAS. However, there are key distinctions. First, we operate on continuous hyperparameter spaces whereas these consider discrete hyperparameters which are easier to analyze. Second, their approaches do not directly apply to neural nets as they have to control the space of all networks with zero training loss which is large. In contrast, we analyze tractable lower-level algorithms such as gradient-descent and study the properties of the specific model returned by the algorithm. (Guyon et al., 1997) discuss methods for determining train-validation split ratios. Favorable learning theoretic properties of (cross-)validation are studied by (Kearns & Ron, 1999; Xu et al., 2020). These works either apply to specific scenarios such as tuning lasso penalty or do not consider hyperparameters. We also note that algorithmic stability of (Bousquet & Elisseeff, 2001) utilizes stability of an algorithm to changes in the training set. In contrast, we consider the stability with respect to hyperparameters. Finally, (Wang et al., 2020b) explores tuning the learning rate for improved generalization. They focus on a simple quadratic objective using hyper-gradient methods and characterize when train-validation split provably helps.




(a) Input layer stability for a one-hidden layer network (b) Stability of weights of a deeper four layer network

*Figure 2.* We visualize the Lipschitzness of the algorithm when $\mathcal{A}(\cdot)$ is stochastic gradient descent. We train networks with activation parameters $\alpha$ and $\alpha+\Delta\alpha$ and display the normalized distances $\|\boldsymbol{\theta}_\alpha - \boldsymbol{\theta}_{\alpha+\Delta\alpha}\|_{\ell_2}/\Delta\alpha$ for different perturbation strengths $\Delta\alpha$.

**Generalization in deep learning:** The statistical study of neural networks can be traced back to 1990's (Anthony & Bartlett, 2009; Bartlett et al., 1998; Bartlett, 1998). With the success of deep learning, the generalization properties of deep networks received a renewed interest in recent years (Dziugaite & Roy, 2017; Arora et al., 2018; Neyshabur et al., 2017a; Golowich et al., 2018). (Bartlett et al., 2017; Neyshabur et al., 2017b) establish spectrally normalized risk bounds for deep networks and (Nagarajan et al., 2018) provides refined bounds by exploiting inter-layer Jacobian. (Arora et al., 2018) proposes tighter bounds using compression techniques. More recently, (Jacot et al., 2018) has introduced the neural tangent kernel which relates training dynamics of wide deep nets to kernel regression. NTK received significant attention for analyzing the optimization and learning dynamics of wide networks (Du et al., 2018; Zhang et al., 2019; Nitanda & Suzuki, 2019; Zou et al., 2018; Wang et al., 2020a; Oymak & Soltanolkotabi, 2019; Chizat & Bach, 2018). Closer to us, (Arora et al., 2019a; Ma et al., 2019; Oymak et al., 2019; Allen-Zhu et al., 2018; Arora et al., 2019a) provide generalization bounds for gradient descent training. A line of research implements neural kernels for convolutional networks and ResNets (Yang, 2019; Li et al., 2019; Huang et al., 2020). Related to us (Arora et al., 2019b) mention the possibility of using NTK for NAS and recent work by (Park et al., 2020) shows that such an approach can indeed produce good results and speed up NAS. In connection to these, Section 4 establishes the first provable guarantees for NAS and also provide a rigorous justification of the NTK-based NAS by establishing data-dependent bounds under verifiable assumptions.

## 7. Numerical Experiments

We provide two sets of experiments to verify our theory. First, to test Theorem 3, we verify the (approximate) Lipschitzness of trained neural nets to perturbations in the activation function. Second, to test Theorem 1, we will study the test-validation gap for DARTS search space.

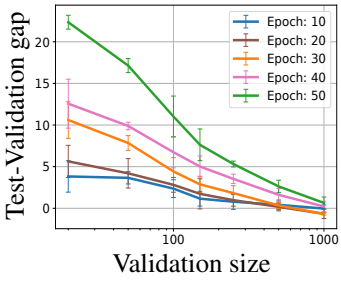



*Figure 3.* The test-validation gap for the continuously parameterized architecture during the search phase of DARTS.

**a. Lipschitzness of Trained Networks.** To verify Assumption 1, we consider a single hyperparameter $\alpha \in \mathrm{R}$ to control the activation via a combination of ReLU and Sigmoid i.e. $\sigma_\alpha(x) = (1-\alpha)\mathrm{ReLU}(x) + \alpha \cdot \mathrm{Sigmoid}(x)$. Training the network weights $\boldsymbol{\theta}$ with this activation from the same random initialization leads to the weights $\boldsymbol{\theta}_\alpha$. We are interested in testing the stability of these weights to slight $\alpha$ perturbations by studying the normalized distance $\|\boldsymbol{\theta}_\alpha - \boldsymbol{\theta}_{\alpha+\Delta\alpha}\|_{\ell_2}/\Delta\alpha$. This in turn ensures the Lipschitzness of the model output via a standard bound. Fig. 2 presents our results on both shallow and deeper networks on a binary MNIST task which uses the first two classes with squared loss. This setup is in a similar spirit to our theory. In Fig. 2(a) we train input layer of a shallow network $f_\alpha(x) = \boldsymbol{v}^T\sigma_\alpha(\boldsymbol{W}\boldsymbol{X})$ where $\boldsymbol{W} \in \mathrm{R}^{k\times 784}$. In Fig. 2(b), a deeper fully connected network with 4 layers is trained. Here, the number of neurons from input to output are $k$, $k/2$, $k/4$ and 1 and the same activation $\sigma_\alpha(\boldsymbol{X})$ is used for all layers. Finally, we initialize the network with He initialization and train the model for 60 epochs with batch size 128 with SGD optimizer and learning rate 0.003. For each curve and width level, we average 20 experiments where we first pick 20 random $\alpha \in [0,1]$ and their perturbation $\alpha + \Delta\alpha$. We then compute the average of normalized distances $\|\boldsymbol{\theta}_\alpha - \boldsymbol{\theta}_{\alpha+\Delta\alpha}\|_{\ell_2}/\Delta\alpha$.

All figures support our theory and show that, the normalized distance is indeed stable to the perturbation level $\Delta\alpha$ across different widths and only mildly changes. Note that $\Delta\alpha \in \{0.01, 0.005\}$ result in a slightly larger normalized distance compared to larger perturbations. Such behavior for small $\Delta\alpha$ is not surprising and is likely due to the imperfect Lipschitzness of the network (especially with ReLU activation). Fortunately, our theory allows for this as it only requires an approximate Lipschitz property (recall the discussion below Theorem 1).

**b. Test-Validation Gap for DARTS.** In this experiment, we study a realistic architecture search space via DARTS algorithm (Liu et al., 2018) over CIFAR-10 dataset using 10k training samples. We only consider the search phase of DARTS and train for 50 epochs using SGD. This phase outputs a *continuously-parameterized architecture*, which can be computed on DARTS' supernet. Each operation on the edges of the final architecture is a linear superposition of eight predefined operations (e.g. conv3x3, zero, skip). The curves are obtained by averaging five independent runs. In Figures 3 and 1(c), we assess the gap between the test and validation errors while varying validation sizes from 20 to 1000. Our experiments reveal two key findings via Figure 3. First, consistent with Theorem 1, the train-validation gap decreases rapidly as soon as the validation size is only mildly large, e.g. around $n_{\mathcal{V}} = 250$ –much smaller than the typical validation size used in practice. On the other hand, there is indeed a potential of overfitting to validation for $n_{\mathcal{V}} \leq 100$. We also observe that the gap noticeably increases with more epochs. The small gaps at initial epochs may be due to insufficient training For later epochs, since early-stopping has a ridge regularization effect, we suspect that widening gap may be due to the growing Lipschitz constant with respect to the architecture choice. Such behavior would be consistent with Thm 1 as well as Lemma 1 (smaller ridge penalty leads to more excess validation risk). Figure 1(c) displays the train/validation/test errors by epoch for different validation sample sizes. The training loss/error quickly goes down to zero. Validation contains much fewer samples but it is difficult to overfit (despite continuous architecture parameterization). However, as discussed above, below a certain threshold ($n_{\mathcal{V}} \leq 100$), differentiable search indeed overfits to the validation leading to deteriorating test risk.

## 8. Conclusions

In this paper, we explored theoretical aspects of the NAS problem. We first provided statistical guarantees when solving bilevel optimization with train-validation split. We showed that even if the lower-level problem overfits –which is common in deep learning– the upper-level problem can guarantee generalization with a few validation data. We applied these results to establish guarantees for the optimal activation search problem and extended our theory to generic neural architectures. These formally established the high-level intuition in Figure 1. We also showed interesting connections between the activation search and a novel low-rank matrix learning problem and provided sharp algorithmic guarantees for the latter.

## Acknowledgements

S.O. and M.L. are supported by NSF-CNS award #1932254 and by an NSF-CAREER award #2046816. M.S. is supported by the Packard Fellowship in Science and Engineering, a Sloan Research Fellowship in Mathematics, an NSF-CAREER award #1846369, the AFOSR Young Investigator Award #FA9550-18-1-0078, DARPA LwLL and FastNICS programs, and NSF-CIF awards #1813877 and #2008443.

## References


Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. *arXiv preprint arXiv:1811.04918*, 2018.

Anthony, M. and Bartlett, P. L. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pp. 254–263. PMLR, 2018.

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332. PMLR, 2019a.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019b.

Bartlett, P. L. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Bartlett, P. L., Maiorov, V., and Meir, R. Almost linear vc-dimension bounds for piecewise polynomial networks. *Neural computation*, 10(8):2159–2173, 1998.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pp. 6241–6250, 2017.

Bousquet, O. and Elisseeff, A. Algorithmic stability and generalization performance. *Advances in Neural Information Processing Systems*, pp. 196–202, 2001.

Chizat, L. and Bach, F. A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pp. 1675–1685. PMLR, 2019.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Foster, D. J., Sekhari, A., and Sridharan, K. Uniform convergence of gradients for non-convex learning and optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8759–8770, 2018.

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pp. 1568–1577. PMLR, 2018.

Golowich, N., Rakhlin, A., and Shamir, O. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pp. 297–299. PMLR, 2018.

Gonen, M. and Alpaydin, E. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12: 2211–2268, 2011.

Guyon, I. et al. A scaling law for the validation-set training-set size ratio. *AT&T Bell Laboratories*, 1(11), 1997.

Huang, K., Wang, Y., Tao, M., and Zhao, T. Why do deep residual networks generalize better than deep feedforward networks?—a neural tangent kernel perspective. *Advances in Neural Information Processing Systems*, 33, 2020.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

Kearns, M. A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. *Advances in Neural Information Processing Systems*, pp. 183–189, 1996.

Kearns, M. and Ron, D. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural computation*, 11(6):1427–1453, 1999.

Kearns, M., Mansour, Y., Ng, A. Y., and Ron, D. An experimental and theoretical comparison of model selection methods. *Machine Learning*, 27(1):7–50, 1997.

Khodak, M., Li, L., Balcan, M.-F., and Talwalkar, A. On weight-sharing and bilevel optimization in architecture search. *preprint available at https://openreview.net/forum?id=HJgRCyHFDr*, 2019.

Khodak, M., Li, L., Roberts, N., Balcan, M.-F., and Talwalkar, A. A simple setting for understanding neural architecture search with weight-sharing. *ICML AutoML Workshop*, 2020.

Li, L., Khodak, M., Balcan, M.-F., and Talwalkar, A. Geometry-aware gradient algorithms for neural architecture search. *arXiv preprint arXiv:2004.07802*, 2020.

Li, Z., Wang, R., Yu, D., Du, S. S., Hu, W., Salakhutdinov, R., and Arora, S. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*, 2019.

Liu, H., Simonyan, K., and Yang, Y. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Lorraine, J., Vicol, P., and Duvenaud, D. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1540–1552. PMLR, 2020.

Ma, C., Wu, L., et al. A comparative analysis of the optimization and generalization property of two-layer neural network and random feature models under gradient descent dynamics. *arXiv preprint arXiv:1904.04326*, 2019.

Mei, S. and Montanari, A. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

Mei, S., Bai, Y., Montanari, A., et al. The landscape of empirical risk for nonconvex losses. *Annals of Statistics*, 46(6A):2747–2774, 2018.

Mendelson, S. A few notes on statistical learning theory. In *Advanced lectures on machine learning*, pp. 1–40. Springer, 2003.

Nagarajan, P., Warnell, G., and Stone, P. Deterministic implementations for reproducibility in deep reinforcement learning. *arXiv preprint arXiv:1809.05676*, 2018.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017a.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017b.

Nitanda, A. and Suzuki, T. Refined generalization analysis of gradient descent for over-parameterized two-layer neural networks with smooth activations on classification problems. *arXiv preprint arXiv:1905.09870*, 2019.

Oymak, S. and Soltanolkotabi, M. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In *International Conference on Machine Learning*, pp. 4951–4960. PMLR, 2019.

Oymak, S., Fabian, Z., Li, M., and Soltanolkotabi, M. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *arXiv preprint arXiv:1906.05392*, 2019.

Oymak, S., Li, M., and Soltanolkotabi, M. Generalization guarantees for neural architecture search with train-validation split. *arXiv preprint arXiv:2104.14132*, 2021.

Park, D. S., Lee, J., Peng, D., Cao, Y., and Sohl-Dickstein, J. Towards nngp-guided neural architecture search. *arXiv preprint arXiv:2011.06006*, 2020.

Pham, H., Guan, M., Zoph, B., Le, Q., and Dean, J. Efficient neural architecture search via parameters sharing. In *International Conference on Machine Learning*, pp. 4095–4104. PMLR, 2018.

Vapnik, V. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

Vuong, Q. H. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica: Journal of the Econometric Society*, pp. 307–333, 1989.

Wang, H., Sun, R., and Li, B. Global convergence and induced kernels of gradient-based meta-learning with neural nets. *arXiv preprint arXiv:2006.14606*, 2020a.

Wang, X., Yuan, S., Wu, C., and Ge, R. Guarantees for tuning the step size using a learning-to-learn approach. *arXiv preprint arXiv:2006.16495*, 2020b.

Xu, N., Fisher, T. C., and Hong, J. Rademacher upper bounds for cross-validation errors with an application to the lasso. *arXiv preprint arXiv:2007.15598*, 2020.

Yang, G. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.

Zhang, H., Yu, D., Chen, W., and Liu, T.-Y. Training over-parameterized deep resnet is almost as easy as training a two-layer network. *arXiv preprint arXiv:1903.07120*, 2019.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018.