Proceedings of Machine Learning Research vol XX:1–34, 2023

Regret Guarantees for Online Deep Control

Xinyi Chen XINYIC@PRINCETON.EDU
Princeton University, Google AI Princeton

Edgar Minasyan MINASYAN@PRINCETON.EDU
Princeton University, Google AI Princeton

Jason D. Lee JASONLEE@PRINCETON.EDU
Princeton University, Google AI Princeton

Elad Hazan EHAZAN@PRINCETON.EDU
Princeton University, Google AI Princeton

Abstract

Despite the immense success of deep learning in reinforcement learning and control, few theoretical guarantees for neural networks exist for these problems. Deriving performance guarantees is challenging because control is an online problem with no distributional assumptions and an agnostic learning objective, while the theory of deep learning so far focuses on supervised learning with a fixed known training set.

In this work, we begin to resolve these challenges and derive the first regret guarantees in online control over a *neural network*-based policy class. In particular, we show sublinear *episodic* regret guarantees against a policy class parameterized by deep neural networks, a much richer class than previously considered linear policy parameterizations. Our results center on a reduction from online learning of neural networks to online convex optimization (OCO), and can use any OCO algorithm as a blackbox. Since online learning guarantees are inherently agnostic, we need to quantify the performance of the best policy in our policy class. To this end, we introduce the interpolation dimension, an expressivity metric, which we use to accompany our regret bounds. The results and findings in online deep learning are of independent interest and may have applications beyond online control.

1. Introduction

The use of deep neural networks has been highly successful in reinforcement learning (RL) and continuous control problems. However, a theory for deep control and RL remains challenging. The main difficulty in applying the theory developed for supervised learning to the RL domain is the distributional assumptions and realizability goal made in the literature thus far. In control and RL, the environment is inherently online and often nonstochastic, and the goal is usually agnostic learning with respect to a policy class.

In this work, we consider the problem of online episodic control with neural network-based policies. We begin to resolve the aforementioned challenges and derive the first regret bound guarantees in this setting. Provable regret bounds in this domain have thus far been limited to linear controllers. However, most dynamical systems in the physical world are nonlinear and/or require nonlinear controls. An important tool that allows us to go beyond linear controllers is the emerging paradigm of online nonstochastic control: a methodology for control that is robust to adversarial noise in the dynamics. The important aspect of this paradigm to our study is that it uses policy classes that admit a convex parameterization.

It is natural to consider the online *episodic* control setting: although it is less challenging from a technical perspective than single-trajectory control, the policy learning procedure in empirical deep control is often done episodically as detailed in the related work section. The main technical challenge to this goal is formalizing online learning over deep neural networks and proving accompanying regret bounds. Given this result, an extension to the single-trajectory setting is possible.

As the major technical component of this work, we propose a black-box reduction from online deep learning to online convex optimization (OCO) that attains provable regret bounds. These bounds apply to the general online learning setting with vector output predictors and arbitrary convex loss functions. Moreover, the regret guarantees are naturally *agnostic*, i.e. they show performance competitive to the best neural network in our policy class in hindsight without assuming it achieves zero loss. To capture agnostic learning and derive meaningful guarantees for online learning and control, we also introduce a new metric of expressivity, namely the “interpolation dimension”, that accompanies our regret bounds.

An interesting conclusion from this reduction is the unifying view that provable convergence and/or generalization bounds for training deep neural networks can be derived for any OCO method, beyond online and stochastic gradient descent. This includes mirror descent, adaptive gradient methods, follow-the-perturbed leader and other algorithms. Previously, convergence and generalization analyses for neural networks were done in isolation for different optimization algorithms as detailed in the related work section.

Our contributions in this work can be summarized as follows:

- **Online episodic deep control:** We derive the first provable regret guarantees in online episodic control with policies based on deep neural networks. Furthermore, we demonstrate the richness of the considered policy class by showing that it can output the optimal *open-loop* control sequence of any single episode.
- **Online learning over neural networks:** We give a general reduction from online learning of neural networks to OCO that can use any OCO algorithm as a blackbox.
- **Interpolation dimension:** To state meaningful guarantees in online agnostic learning, we introduce the interpolation dimension as an expressivity metric. It is a fundamental notion and applies to any hypothesis class.
- **Unifying analysis:** Our proposed method applies to any OCO algorithm, including mirror descent and adaptive gradient methods widely used in deep learning. This leads to a unifying framework for optimization in deep learning: the online learning framework implies both convergence and generalization bounds in the supervised learning setting.

1.1. Related work

Online and nonstochastic control. Our study focuses on algorithms which enjoy sublinear regret for online control of dynamical systems; that is, whose performance tracks a given benchmark of policies up to a term which is vanishing relative to the problem horizon. Abbasi-Yadkori and Szepesvári (2011) initiated the study of online control under the regret benchmark for linear time-invariant (LTI) dynamical systems. Bounds for this setting have since been improved and refined in Dean et al. (2018); Mania et al. (2019); Cohen et al. (2019); Simchowitz and Foster (2020). Our work instead adopts the online *nonstochastic* control setting (Agarwal et al., 2019), that allows for adversarially chosen (e.g. non-Gaussian) noise and general convex costs that may vary with time. This model has been studied for many extended settings, see Hazan and Singh (2021) for a comprehensive survey. Similar to our control framework, online episodic control is also studied in Kakade et al. (2020), but the regret definition differs from ours, the results are only information-theoretic and the system is linear in a kernel space. In terms of nonlinear systems, one common approach in control is iterative linearization which takes the local linear approximation via the gradient of the nonlinear dynamics. One can apply techniques from optimal control to solve the resulting changing linear system. Iterative planning methods such as iLQR (Tassa et al., 2012), iLC (Moore, 2012) and iLQG (Todorov and Li, 2005) fall into this category. Recent works (Roulet et al., 2022; Westenbroek et al., 2021) provide theoretical results and insights to this approach but many theoretical questions about the approach remain open.

The emerging theory of deep learning. For detailed background on the developing theory for deep learning, see the book draft (Arora et al., 2021). Among the various studies on the theory of deep learning, the neural tangent kernel (NTK or linearization) approach has emerged as the most complete and pervasive: it is not currently believed to fully explain the practical success but there is no alternative substantial theory yet. This technique shows that neural networks behave similar to their local linearization and proves that gradient descent converges to a global minimizer of the training loss (Soltanolkotabi et al., 2018; Du et al., 2018a,b; Jacot et al., 2018; Bai and Lee, 2019; Lee et al., 2019). The NTK approach/regime has been expanded to provide various generalization error bounds (Arora et al., 2019; Wei et al., 2019; Cao and Gu, 2019; Ji and Telgarsky, 2020), and adversarial training guarantees (Gao et al., 2019; Zhang et al., 2020). As opposed to our generic approach, a number of different optimization algorithms have been considered in isolation for analyzing deep learning theory in the NTK regime including (Wu et al., 2019; Cai et al., 2019; Wu et al., 2021; Zhang et al., 2019).

The results in this work extend upon the described deep learning theory literature; in particular, we use the same deep learning setup, and follow techniques and results from Gao et al. (2019); Allen-Zhu et al. (2019). Furthermore, several works in the literature (Cao and Gu, 2019; Gao et al., 2019; Zhang et al., 2020) have observed and used online components in their derivations of generalization and adversarial training guarantees. We note that all these works, unlike our contributions, operate in the supervised learning setting.

Online convex optimization and dimensionality notions in learning. The framework of learning in games has been extensively studied as a model for learning in adversarial and nonstochastic environments (Cesa-Bianchi and Lugosi, 2006). Online learning was infused with algorithmic techniques from mathematical optimization into the setting of online convex optimization, see (Hazan, 2019) for a comprehensive introduction. Learnability in the statistical and online learning settings was characterized using various notions of dimensionality, starting from the VC-dimension, fat-shattering dimension, Rademacher complexity, Littlestone dimension and more. For an extensive treatment see (Mohri et al., 2018; Shalev-Shwartz and Ben-David, 2014; Vapnik, 1999). Regarding interpolation, Bubeck and Sellke (2021) establish an inverse relationship between the interpolation ability and robustness of a function class. The notion of interpolation dimension that we introduce here has found applications in the theory of boosting (Alon et al., 2021).

Deep control. Deep neural networks have advanced the state of the art for continuous control, not only in simulated environments Tassa et al. (2018); Zhang et al. (2016); Duan et al. (2016), but also in real-world tasks such as robotic manipulation OpenAI et al. (2018, 2019) and temperature control in office buildings and data centers Wang et al. (2017); Lazic et al. (2018). In many of these applications, the policy learning procedure is episodic, where the environment resets at the beginning of an episode. For example, OpenAI et al. (2018, 2019) train an LSTM policy for manipulating a rubik's cube with a robotic hand in the following manner: an environment is generated at the beginning of the episode, which interacts with the current policy for a fixed number of time steps; then, after collecting the episodic trajectory, the policy is updated according to a chosen optimization scheme. This setting is closely related to online episodic control, which we formally describe in Section 2, and motivates our theoretical analysis of neural network-based policies in this framework.

2. Problem Setting and Preliminaries

Notation. Let $\|\cdot\|$ denote the Euclidean norm and $\langle\cdot,\cdot\rangle$ the corresponding inner product between two vectors, matrices, or tensors of the same dimension: $\langle x, y\rangle = \text{vec}(x)^\top \text{vec}(y)$. Let $\mathbb{S}_p = \{x \in \mathbb{R}^p : \|x\| = 1\}$ denote the unit p-dimensional sphere, and for a convex set $\mathcal{K}$, let $\prod_{\mathcal{K}}$ denote projection onto $\mathcal{K}$.

2.1. Deep neural networks and the interpolation dimension

Deep neural networks. Let $x \in \mathbb{R}^p$ be the p-dimensional input. We define the depth H network with ReLU activation and scalar output as follows:

$$x^0 = Ax, \qquad x^h = \sigma_{\text{relu}}(\theta^h x^{h-1}),\ h \in [H], \qquad f(\theta, x) = a^\top x^H,$$

where $\sigma_{\text{relu}}(\cdot)$ is the ReLU function $\sigma_{\text{relu}}(z) = \max(0, z)$, $A \in \mathbb{R}^{m\times p}$, $\theta^h \in \mathbb{R}^{m\times m}$, and $a \in \mathbb{R}^m$. Let $\theta = (\theta^1, \ldots, \theta^H)^\top \in \mathbb{R}^{H\times m\times m}$ denote the trainable parameters of the network and the parameters A, a are fixed after initialization. The initialization scheme is as follows: each entry in A and θ^h is drawn i.i.d. from the Gaussian distribution $\mathcal{N}(0, \frac{2}{m})$, and each entry in a is drawn i.i.d. from $\mathcal{N}(0, 1)$. This setup is common in recent literature and follows that of Gao et al. (2019).

For vector-valued outputs, we consider a scalar output network for each coordinate. Suppose for $i \in [d]$, f_i is a deep neural network with a scalar output; with a slight abuse of notation, for input $x \in \mathbb{R}^p$, denote

$$f(\theta; x) = (f_1(\theta[1]; x), \ldots, f_d(\theta[d]; x))^\top \in \mathbb{R}^d, \tag{2.1}$$

where $\theta[i] \in \mathbb{R}^{H\times m\times m}$ denotes the trainable parameters for the network f_i for coordinate i. Let $\theta = (\theta[1], \theta[2], \ldots, \theta[d]) \in \mathbb{R}^{d\times H\times m\times m}$ denote all the parameters for f.

In the online setting, the neural net receives an input $x_t \in \mathbb{R}^p$ at each round $t \in [T]$, and with parameter θ suffers loss $\ell_t(f(\theta; x_t))$. Note that this framework generalizes the supervised learning paradigm. We make the following standard assumptions:

Assumption 1 *The input x has unit norm, i.e. $x \in \mathbb{S}_p$, $\|x\|_2 = 1$.*

Assumption 2 *The loss functions $\ell_t(f(\theta; x))$ are L-Lipschitz and convex in $f(\theta; x)$.*

Interpolation dimension. Since we aim to prove regret bounds for online learning with families of deep neural networks as the comparator class, we need to ensure these families have non-trivial representation power. To this end, we introduce interpolation dimension, an expressivity metric that can be naturally applied to our setting. In real-valued learning, we say that a hypothesis class has interpolation dimension of at least k if one can assign arbitrary real labels to *any* k different inputs using a hypothesis from that class.

Definition 1 *The* **interpolation dimension** *of a hypothesis class $\mathcal{H} = \{h : \mathcal{X} \subseteq \mathbb{R}^p \to \mathbb{R}^d\}$ over input domain $\mathcal{X}$ at non-degeneracy $\gamma > 0$, denoted $\mathcal{I}_{\mathcal{X},\gamma}(\mathcal{H})$, is the largest cardinality k such that for* **any** *set of data points $\{(x_j, y_j)\}_{j=1}^k$ satisfying $\min_{j \neq l} \|x_j - x_l\|_2 \geq \gamma$, $y_j \in [-1, 1]^d$, $\forall j \in [k]$, $\inf_{h \in \mathcal{H}} \left[\sum_{j=1}^k \|y_j - h(x_j)\|^2\right] = 0$.*

The label bound above is 1 for simplicity, but can be extended to any $B > 0$. Henceforth, we show that over input domain $\mathcal{X} = \mathbb{S}_p$, neural networks that have poly$(k, \frac{1}{\gamma})$ width have $\mathcal{I}_{\mathcal{X},\gamma}(\mathcal{H}) \geq k$. This enables us to derive regret bounds for online agnostic learning over a class of neural networks that has interpolation dimension at least k.

In the case of binary classification, interpolation dimension can be seen as the "dual" of the VC dimension. More details on the interpolation dimension in binary classification, connection to VC dimension, and additional examples can be found in Appendix A.1 of the full manuscript: https://xinyi.github.io/submission_1.pdf.

2.2. Online convex optimization

In Online Convex Optimization (OCO), a decision maker sequentially chooses a point in a convex set $\theta_t \in \mathcal{K} \subseteq \mathbb{R}^d$, and suffers loss $\ell_t(\theta_t)$ according to a convex loss function $\ell_t : \mathcal{K} \mapsto \mathbb{R}$. The goal of the learner is to minimize her regret, defined as

$$\text{Regret}_T = \sum_{t=1}^{T} \ell_t(\theta_t) - \min_{\theta^* \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(\theta^*) \ .$$

A host of techniques from classical optimization are applicable to this setting and give rise to efficient low-regret algorithms. To name a few methods, mirror descent, Newton's method, Frank-Wolfe and follow-the-perturbed leader all have online analogues, see e.g. Hazan (2019) for a comprehensive treatment.

As an extension to the OCO framework, we show that regret bounds hold analogously for the online optimization of *nearly* convex functions. As we show in later sections, these regret bounds naturally carry over to the setting of online learning over neural networks.

Definition 2 *A function $\ell : \mathbb{R}^n \to \mathbb{R}$ is ε-nearly convex over the convex, compact set $\mathcal{K} \subseteq \mathbb{R}^n$ iff $\forall x, y \in \mathcal{K}$, $\ell(x) \geq \ell(y) + \nabla \ell(y)^\top (x - y) - \varepsilon$.*

The analysis of any algorithm for OCO, including the most fundamental method of online gradient descent (OGD), extends to this case in a straightforward manner. Let $\mathcal{A}$ be any regret minimization algorithm for OCO with a regret bound given by $\text{Regret}_T(\mathcal{A})$. This algorithm $\mathcal{A}$ can be applied on the surrogate loss functions $h_t(\theta) = \ell_t(\theta_t) + \nabla \ell_t(\theta_t)^\top (\theta - \theta_t)$ to obtain regret bounds on the nearly convex losses ℓ_t as given below. The described method is presented in Algorithm 3 which along with more details can be found in Appendix A.2 of the aforementioned full manuscript.

Lemma 3 *Suppose $\ell_1, \ldots, \ell_T$ are ε-nearly convex, then Algorithm 3 has regret bounded by*

$$\sum_{t=1}^{T} \ell_t(\theta_t) - \min_{\theta^* \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(\theta^*) \leq \textit{Regret}_T(\mathcal{A}) + \varepsilon T \ .$$

2.3. Online episodic control

Consider the following online episodic learning problem for nonstochastic control over linear time-varying (LTV) dynamics: there is a sequence of T control problems each with a horizon K and an initial state $x_1 \in \mathbb{R}^{d_x}$. In each episode, the state transition is given by

$$\forall k \in [1, K], \quad x_{k+1} = A_k x_k + B_k u_k + w_k, \tag{2.2}$$

where $x_k \in \mathbb{R}^{d_x}, u_k \in \mathbb{R}^{d_u}$. The system matrices $A_k \in \mathbb{R}^{d_x \times d_x}, B_k \in \mathbb{R}^{d_x \times d_u}$ along with the next state x_{k+1} are revealed to the learner *after* taking the action u_k. The disturbances $w_k \in \mathbb{R}^{d_x}$ are unknown and adversarial but can be a posteriori computed by the learner $w_k = x_{k+1} - A_k x_k - B_k u_k$. An episode loss is defined cumulatively over the rounds $k \in [1, K]$ according to the convex cost functions $c_k : \mathbb{R}^{d_x} \times \mathbb{R}^{d_u} \to \mathbb{R}$ of state and action: for a policy π, the loss is $J(\pi; x_1, c_{1:K}) = \sum_{k=1}^{K} c_k(x_k^\pi, u_k^\pi)$. Like the system matrices, the cost function c_k is also revealed after taking action u_k. The transition matrices $(A_k, B_k)_{1:K}$, initial state x_1, disturbances $w_{1:K}$ and costs $c_{1:K}$ can *change arbitrarily* over different episodes. The goal of the learner is to minimize *episodic* regret by adapting its output policies π_t for $t \in [1, T]$,

$$\text{Regret}_T(\Pi) = \sum_{t=1}^{T} J_t(\pi_t; x_1^t, c_{1:K}^t) - \min_{\pi \in \Pi} \sum_{t=1}^{T} J_t(\pi; x_1^t, c_{1:K}^t), \tag{2.3}$$

where Π denotes the class of policies the learner competes against.

The model above is presented in its utmost generality: the system in an episode is LTV and these LTVs are allowed to change arbitrarily throughout episodes. Results for this model can be applied to derive guarantees for: (1) a simpler setting, learning to control a single LTV episodically; (2) a more complex setting, first-order guarantees in control or planning over *nonlinear* dynamics by taking the Jacobian linearization of the dynamics (Ahn et al., 2007; Westenbroek et al., 2021; Roulet et al., 2022). We make the following basic assumptions about the dynamical system *in each episode* that are common in the nonstochastic control literature (Agarwal et al., 2019).

Assumption 3 *The disturbances satisfy $\forall k \in [K]$, $\|w_k\|_2 \leq W$.*

Assumption 4 (Sequential stability) [1] *There exist $C_1, C_2 \geq 1$, $0 < \rho_1 < 1$ such that the system matrices satisfy:*

$$\forall k \in [K], \forall n \in [1, k), \qquad \left\| \prod_{i=k}^{k-n+1} A_i \right\|_{op} \leq C_1 \cdot \rho_1^n, \quad \|B_k\|_{op} \leq C_2 \ .$$

Assumption 5 *Each cost function $c_k : \mathbb{R}^{d_x} \times \mathbb{R}^{d_u} \to \mathbb{R}$ is jointly convex and satisfies a generalized Lipschitz condition $\|\nabla c_k(x,u)\| \leq L_c \max\{1, \|x\| + \|u\|\}$ for some $L_c > 0$.*

The performance of the learner given by (2.3) directly depends on the policy class Π. In this work, we focus on disturbance based policies, i.e. policies that take past perturbations as input $u_k = f(w_{1:k-1})$, which are parameterized w.r.t. *policy-independent* inputs. This is in contrast to the commonly used state feedback policy $u_k = f(x_k)$. For example, the Disturbance Action Control (DAC) policy class, shown to be more general than linear state feedback policies (Agarwal et al., 2019), outputs controls linear in past finite disturbances, resulting in a *convex* parameterization of the state/control and enabling the design of efficient provable online methods. Our work expands the comparator class by considering policies that are *nonlinear* in the past disturbances, represented by neural networks.

Definition 4 (Disturbance Neural Feedback Control) *Let π^θ_{dnn} denote the policy with control outputs u_k given by*

$$\forall k \in [K], \quad u_k = f_\theta(w_{k-1}, w_{k-2}, \ldots, w_1) \in \mathbb{R}^{d_u},$$

where $f_\theta(\cdot) = f(\theta; \cdot)$ is a neural network defined in (2.1). The policy class is defined as $\Pi_{dnn}(f; \Theta) = \{\pi^\theta_{dnn} : \theta \in \Theta\}$ with Θ being the set of permissible parameters.

3. Online learning of deep neural networks

We present our technical results in the following two sections; due to space constraints, all proofs are included in the full manuscript `https://xinyi.github.io/submission_1.pdf`. We first present the general framework of online learning with deep neural networks and state the accompanying regret guarantees. Our framework can use any OCO algorithm as a black-box as in Algorithm 1, but for our main result, we use projected Online Gradient Descent (OGD). Projected OGD has explicit regret bounds and variants of GD are widely used in practice. Observe that, in this case, the parameter update is equivalent to OGD on the original losses.

The main technical result, provided in Theorem 5, gives a regret bound on the online agnostic learning of deep neural networks. The benchmark hypothesis class is a class of deep neural networks with interpolation dimension of at least k where k is decided a priori and used in the construction of the network.

Theorem 5 *Suppose Assumptions 1 and 2 hold, and let $\mathcal{H}_{\rm NN}(R; \theta_1) = \{f(\theta; \cdot) : \theta \in \Theta\}$ denote the class of neural networks $f(\theta; \cdot)$ as in (2.1) with parameter set $\Theta = B(R; \theta_1) = \{\theta : \|\theta[i] - \theta_1[i]\|_F \leq R, \forall\ i \in [d]\}$ and $\mathcal{X} = \mathbb{S}_p$. Suppose $\gamma \in (0, O\left(\frac{1}{H}\right)]$, take $R = O\left(\frac{k^3 \log m}{\gamma\sqrt{m}}\right)$,*

[1] This condition is relaxed to sequential stabilizability in Appendix E

Algorithm 1 Online Learning over Neural Networks

```
Input: OCO algorithm $\mathcal{A}$, neural network $f(\cdot;\cdot)$, initial $\theta_1$, parameter set $\Theta = B(R;\theta_1)$.
for $t = 1 \dots T$ do
    Play $\theta_t$, receive loss $\ell_t(\theta) = \ell_t(f(\theta; x_t))$.
    Construct $h_t(\theta) = \ell_t(\theta_t) + \nabla \ell_t(\theta_t)^\top (\theta - \theta_t)$.
    Update $\theta_{t+1} = \mathcal{A}(h_1, \dots, h_t) \in \Theta$.
end
```

then for $m \geq O(\frac{p^{3/2}(k^{24}H^{12}\log^8 m + d)^{3/2}}{\gamma^8})$, *with probability* $1 - O(H+d)e^{-\Omega(\log^2 m)}$ *over the random initialization,*

- *The function class* $\mathcal{H}_{\mathrm{NN}}(R;\theta_1)$ *has interpolation dimension* $\mathcal{I}_{\mathcal{X},\gamma}(\mathcal{H}_{\mathrm{NN}}(R;\theta_1)) \;\geq\; k$.
- *Algorithm 1 using OGD with* $\eta_t = \frac{2R\sqrt{d}}{LH\sqrt{m}} \cdot t^{-1/2}$ *for* $\mathcal{A}$ *attains regret bound*
$$\sum_{t=1}^T \ell_t(f(\theta_t; x_t)) \leq \min_{g \in \mathcal{H}_{\mathrm{NN}}(R;\theta_1)} \sum_{t=1}^T \ell_t(g(x_t)) + \tilde{O}\left(\frac{k^3 LH\sqrt{dT}}{\gamma} + \frac{k^4 LH^{5/2}\sqrt{d}T}{\gamma^{4/3}m^{1/6}}\right),$$
where $\tilde{O}(\cdot)$ *hides terms polylogarithmic in* m.

The above theorem indicates that the average regret can be minimized up to arbitrary precision: for any $\varepsilon > 0$, if one chooses sufficiently large network width $m = \Omega(\varepsilon^{-6})$ and sufficiently large number of iterations $T = \Omega(\varepsilon^{-2})$, the average regret is bounded by ε. The interpolation dimension bound is established due to the seminal work Allen-Zhu et al. (2019), spelled out in the following lemma and proven in Appendix A.1.

Lemma 6 *Let* $\mathcal{H}_{\mathrm{NN}}(R;\theta_1) = \{f(\theta;\cdot) : \theta \in \Theta\}$ *denote the class of neural networks as in* (2.1) *where* $\Theta = B(R;\theta_1)$ *and* $\mathcal{X} = \mathbb{S}_p$. *Suppose* $\gamma \in \big(0, O(\frac{1}{H})\big]$, $m \;\geq\; \Omega\left(\frac{k^{24}H^{12}\log^5 m}{\gamma^8}\right)$ *and* $R = O\left(\frac{k^3 \log m}{\gamma\sqrt{m}}\right)$, *then with probability* $1 - d \cdot e^{-\Omega(\log^2 m)}$ *over random initialization of* θ_1,

$$\mathcal{I}_{\mathcal{X},\gamma}\left(\mathcal{H}_{\mathrm{NN}}(R;\theta_1)\right) \;\geq\; k \; . \tag{3.1}$$

3.1. Proof Sketch

Due to space constraints, we give a proof sketch here; for a more detailed analysis outline, see Appendix C, and for the full proof see Appendix D. There are 3 steps to the proof of Theorem 5. First, we show that the considered loss functions $\ell_t : \Theta \to \mathbb{R}$, $\ell_t(\theta) = \ell_t(f(\theta; x_t))$ are *nearly convex* with respect to the parameter θ. This is due to the observation that in the overparameterized regime, neural networks behave similarly to their local linearization.

Second, we can use the near convexity of the loss functions $\ell_t(\theta)$ for all $\theta \in B(R;\theta_1)$, and Lemma 3 to show a regret bound over the parameter set $\Theta = B(R;\theta_1)$. The bound is comprised of the sublinear regret of the OCO algorithm used for parameter update, and the worst-case linear penalty of near convexity $\varepsilon_{\mathrm{nc}} \cdot T$, where $\varepsilon_{\mathrm{nc}}$ is in terms of R and m.

Finally, we use Lemma 6 to ensure that our choice of R and m give the desired interpolation dimension, and derive the final regret guarantee in terms of k, γ and m, among other parameters.

4. Online episodic control with neural network controllers

The online episodic control problem described in Section 2.3 with the policy class $\Pi = \Pi_{\text{dnn}}(f;\Theta)$ can be reduced to online learning for neural networks. This reduction is done by following the current policy each episode, constructing the episode loss, and updating the policy via an OCO algorithm. Algorithm 2 below uses projected OGD but as in the previous section, any OCO algorithm can be used instead.

Algorithm 2 Deep Neural Network Episodic Control with OGD

```
Input: stepsize $\eta_t > 0$, initial parameter $\theta_1$, parameter set $\Theta = B(R;\theta_1)$.
for $t = 1 \dots T$ do
    for $k = 1 \dots K$ do
        Observe $x_k^t$ and construct $z_k^t = \text{vec}([w_{k-1}^t, \dots, w_1^t, \mathbf{0}, \dots, \mathbf{0}, k]) \in \mathbb{R}^{K \cdot d_x + 1}$.
        Normalize $\bar{z}_k^t = \frac{z_k^t}{\|z_k^t\|}$, and play $u_k^t = f(\theta_t, \bar{z}_k^t)$.
    end
    Construct loss function $\mathcal{L}_t(\theta) = \sum_{k=1}^K c_k^t(x_k^{t,\theta}, f(\theta, \bar{z}_k^t))$.
    Perform gradient update $\theta_{t+1} = \prod_\Theta[\theta_t - \eta_t \nabla_\theta \mathcal{L}_t(\theta_t)]$.
end
```

Theorem 7 *Suppose Assumptions 3, 4, 5 hold and let $\Pi_{dnn}(f;\Theta)$ denote the policy class given by Definition 4 with $\Theta = B(R;\theta_1)$. Take $R = O\left(\frac{K^3(2KW+H)\log m}{\sqrt{m}}\right)$, then for $m \geq \Omega(K^{46}H^{20}W^8(d_xd_u)^{3/2}\log^{12} m)$ with probability at least $1 - O(H+d_u)e^{-\Omega(\log^2 m)}$ over the randomness of initialization θ_1, Algorithm 2 with $\eta_t = O(\frac{R\sqrt{d_u}}{LH\sqrt{m}}t^{-1/2})$ satisfies*

$$\text{Regret}_T(\Pi_{dnn}(f;\Theta)) \leq \tilde{O}\left(K^{10}L_cH^4W^2d_ud_x^{1/2} \cdot \sqrt{T} + \frac{K^{12}L_cH^6W^3d_ud_x^{1/2}}{m^{1/6}} \cdot T\right),$$

where $\Pi_{dnn}(f;\Theta)$ can output the optimal open-loop control sequence $u_{1:K}^\star \in [-1,1]^{K\times d_u}$ of any episode and $\tilde{O}(\cdot)$ hides terms polylogarithmic in m.

This theorem statement, analogous to Theorem 5, implies that arbitrarily small $\varepsilon > 0$ average episodic regret is attained with a large network width $m = \Omega(\varepsilon^{-6})$ and large number of iterations $T = \Omega(\varepsilon^{-2})$. The regret bound is against the benchmark policy class $\Pi_{\text{dnn}}(f;\Theta)$ which is chosen such that the neural network class has interpolation dimension $k = K$. This implies open-loop control optimality over a single episode in the following way. For simplicity, drop the episode index $t \in [T]$ and define the optimal *open-loop* control sequence of an episode.

Definition 8 *Define the optimal open-loop control sequence $u_{1:K}^\star \in [-1,1]^{K\times d_u}$ to be*

$$u_{1:K}^\star = \underset{\forall k, u_k \in [-1,1]^{d_u}}{\arg\min} \left\{ J(u_{1:K}; x_1, c_{1:K}) = \sum_{k=1}^K c_k(x_k, u_k) \right\}.$$

To demonstrate the capacity of the benchmark policy class $\Pi_{\text{dnn}}(f;\Theta)$ with $\Theta = B(R;\theta_1)$ we show that it can output the *optimal* open-loop control sequence of any *single* episode as detailed below.

Lemma 9 *Take* $R = O\left(\frac{K^3 \log m(2KW+H)}{\sqrt{m}}\right)$*, suppose* $m \geq \Omega\left(K^{24}H^{12}\log^5 m(2KW+H)^8\right)$*, then with probability* $1 - d_u \cdot e^{-\Omega(\log^2 m)}$ *over the random initialization of* θ_1*,* $\Pi_{dnn}(f;\Theta)$ *can output any open-loop control sequence* $u^\star_{1:K} \in [-1,1]^{K\times d_u}$*:*

$$\inf_{\pi^\theta_{dnn}\in\Pi_{dnn}(f;\Theta)} \left[\sum_{k=1}^{K} \|u^\theta_k - u^\star_k\|^2\right] = 0 \ .$$

4.1. Proof Sketch

To extend the online learning results of Theorem 5 to the online episodic control setting, we ensure the control setting satisfies the corresponding assumptions. For each $k \in [K]$, denote the padded input $z_k = \text{vec}([w_{k-1},\ldots,w_1,\mathbf{0},\ldots,\mathbf{0},k]) \in \mathbb{R}^{K\cdot d_x+1}$ where the index is padded to ensure inputs are separable (Definition 1). To satisfy Assumption 1, normalize the network inputs $\bar{z}_k = \frac{z_k}{\|z_k\|_2} \in \mathbb{S}_{K\cdot d_x+1}$.

For a policy π^θ_{dnn} the episode loss $\mathcal{L}(\theta) = J(\pi^\theta_{\text{dnn}}; x_1, c_{1:K})$ depends on the parameter θ through all the K controls $u^\theta_k = f(\theta;\bar{z}_k)$. Denote $\bar{f}(\theta) = [u^\theta_1,\ldots,u^\theta_K]^\top \in \mathbb{R}^{K\times d_u}$ and let $\mathcal{L}(\theta) = \mathcal{L}(\bar{f}(\theta))$ by abuse of notation. We demonstrate that the reduction to the online learning setting is achieved by showing that $\mathcal{L}(\bar{f}(\theta))$ satisfies the convexity (Lemma 23) and Lipschitz (Lemma 26) conditions. Hence, for each episode $t \in [T]$, the episode loss $\mathcal{L}_t(\theta) = J_t(\pi^\theta_{\text{dnn}}; x^t_1, c^t_{1:K})$ satisfies Assumption 2 and the rest of the derivation is analogous to that of Theorem 5. Finally, Lemma 9 uses the interpolation dimension property of the neural network class to conclude the open-loop optimality stated in the theorem. See Appendix E for full details.

5. Conclusions and Future Work

In this work, we derive the first regret guarantees for neural network based controllers in online control. Our results are in the online episodic control setting, which is motivated by empirical research in control and deep reinforcement learning. We propose algorithms that obtain sublinear episodic regret against the optimal open-loop control sequence of any episode, which relies on a general reduction from online deep learning to regret minimization.

We also introduce a new metric for the expressive power of a hypothesis class and use it for characterizing the expressivity of the benchmark neural network class. The definition of interpolation dimension enables this characterization to be isolated to neural networks but is in no way specific to them. Many intriguing questions about this expressivity notion remain, such as its broader connection to statistical learning theory given its close relationship to the VC dimension.

We use the NTK paradigm to derive the control and online learning results in this work. However, there still are open questions to understand the empirical success of neural networks. As deep learning theory advances in this direction, the question of extending these results to reinforcement learning and control problems remains open too.

References

Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 1–26, 2011.

Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. Online control with adversarial disturbances. In *International Conference on Machine Learning*, pages 111–119, 2019.

Hyo-Sung Ahn, YangQuan Chen, and Kevin L. Moore. Iterative learning control: Brief survey and categorization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1099–1121, 2007. doi: 10.1109/TSMCC.2007.905759.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization, 2019.

Noga Alon, Alon Gonen, Elad Hazan, and Shay Moran. Boosting simple learners. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 481–489, 2021.

Raman Arora, Sanjeev Arora, Joan Bruna, Nadav Cohen, Simon Du, Rong Ge, Suriya Gunasekar, Chi Jin, Jason Lee, Tengyu Ma, and Behnam Neyshabur. *Theory of Deep Learning*. 2021.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019.

Yu Bai and Jason D Lee. Beyond linearization: On quadratic and higher-order approximation of wide neural networks. *arXiv preprint arXiv:1910.01619*, 2019.

Sebastien Bubeck and Mark Sellke. A universal law of robustness via isoperimetry. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=z71OSKqTFh7.

Tianle Cai, Ruiqi Gao, Jikai Hou, Siyu Chen, Dong Wang, Di He, Zhihua Zhang, and Liwei Wang. Gram-gauss-newton method: Learning overparameterized neural networks for regression problems. *arXiv preprint arXiv:1905.11675*, 2019.

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in Neural Information Processing Systems*, 32: 10836–10846, 2019.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Alon Cohen, Tomer Koren, and Yishay Mansour. Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret. In *International Conference on Machine Learning*, pages 1300–1309, 2019.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. Regret bounds for robust adaptive control of the linear quadratic regulator. In *Advances in Neural Information Processing Systems*, pages 4188–4197, 2018.

Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018a.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018b.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1329–1338. JMLR.org, 2016.

Ruiqi Gao, Tianle Cai, Haochuan Li, Cho-Jui Hsieh, Liwei Wang, and Jason D Lee. Convergence of adversarial training in overparametrized neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/file/348a38cd25abeab0e440f37510e9b1fa-Paper.pdf.

Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Elad Hazan and Karan Singh. Tutorial: online and non-stochastic control, July 2021.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks, 2020.

Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15312–15325. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/aee5620fa0432e528275b8668581d9a8-Paper.pdf.

Nevena Lazic, Tyler Lu, Craig Boutilier, MK Ryu, Eehern Jay Wong, Binz Roy, and Greg Imwalle. Data center cooling using model-predictive control. In *Proceedings of the Thirty-second Conference on Neural Information Processing Systems (NeurIPS-18)*, pages 3818–3827, Montreal, QC, 2018. URL https://papers.nips.cc/paper/7638-data-center-cooling-using-model-predictive-control.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. The connection between approximation, depth separation and learnability in neural networks. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 3265–3295. PMLR, 15–19 Aug 2021. URL https://proceedings.mlr.press/v134/malach21a.html.

Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalence is efficient for linear quadratic control. In *Advances in Neural Information Processing Systems*, pages 10154–10164, 2019.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Kevin L Moore. *Iterative learning control for deterministic systems*. Springer Science & Business Media, 2012.

OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation, 2018. URL https://arxiv.org/abs/1808.00177.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand, 2019. URL https://arxiv.org/abs/1910.07113.

Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561, 2008. doi: 10.1109/ALLERTON.2008.4797607.

Vincent Roulet, Siddhartha Srinivasa, Maryam Fazel, and Zaid Harchaoui. Complexity bounds of iterative linear quadratic optimization algorithms for discrete time nonlinear control. *arXiv preprint arXiv:2204.02322*, 2022.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Max Simchowitz and Dylan Foster. Naive exploration is optimal for online lqr. In *International Conference on Machine Learning*, pages 8937–8948. PMLR, 2020.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 2018.

Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913, 2012.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018. URL https://arxiv.org/abs/1801.00690.

Emanuel Todorov and Weiwei Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 300–306. IEEE, 2005.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

Yuan Wang, Kirubakaran Velswamy, and Biao Huang. A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems. *Processes*, 5(3), 2017. ISSN 2227-9717. doi: 10.3390/pr5030046. URL https://www.mdpi.com/2227-9717/5/3/46.

Colin Wei, Jason Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. 2019.

Tyler Westenbroek, Max Simchowitz, Michael I Jordan, and S Shankar Sastry. On the stability of nonlinear receding horizon control: a geometric perspective. *arXiv preprint arXiv:2103.15010*, 2021.

Xiaoxia Wu, Simon S Du, and Rachel Ward. Global convergence of adaptive gradient methods for an over-parameterized neural network. *arXiv preprint arXiv:1902.07111*, 2019.

Xiaoxia Wu, Yuege Xie, Simon Du, and Rachel Ward. Adaloss: A computationally-efficient and provably convergent adaptive gradient method. *arXiv preprint arXiv:2109.08282*, 2021.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5481b2f34a74e427a2818014b8e103b0-Paper.pdf.

Guodong Zhang, James Martens, and Roger B Grosse. Fast convergence of natural gradient descent for over-parameterized neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/file/1da546f25222c1ee710cf7e2f7a3ff0c-Paper.pdf.

Tianhao Zhang, Gregory Kahn, Sergey Levine, and P. Abbeel. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 528–535, 2016.

Yi Zhang, Orestis Plevrakis, Simon S. Du, Xingguo Li, Zhao Song, and Sanjeev Arora. Over-parameterized adversarial training: An analysis overcoming the curse of dimensionality, 2020.

Contents

Appendix A. Details for Section 2

A.1. Interpolation dimension

Characterizing the expressive power of the hypothesis class of deep neural networks is an active area of investigation e.g. (Malach et al., 2021; Yehudai and Shamir, 2019; Rahimi and Recht, 2008). The literature mostly focuses on differentiating the expressivity of networks according to their depth, or on proving lower bounds for sample complexity. Our focus, however, is different. We prove regret bounds for online learning with families of deep neural networks as the comparator classe, thus we need to ensure that these families have non-trivial representation power.

It is useful to recall the theory of supervised learning for binary classification. Vapnik's theorem asserts that the VC dimension characterizes the learnability of a hypothesis class. For many common examples of hypothesis classes, the VC dimension also characterizes their expressive power. For example, linear classifiers of dimension k are capable of shattering *any* training set of size $k+1$, as long as it is in general linear position (non-degenerate). This is, however, not a requirement of the VC dimension, which only requires the *existence* of a set that can be shattered by the hypothesis class. It is thus useful to consider the ”dual” of the VC dimension, which we formally define as follows:

Definition 10 *The* **interpolation dimension** *of a hypothesis class $\mathcal{H} = \{h : \mathcal{X} \mapsto \{0,1\}\}$ over input domain $\mathcal{X}$, denoted $\mathcal{I}_{\mathcal{X}}(\mathcal{H})$, is the largest cardinality k such that for* **any non-degenerate** *set of examples $x_1, ..., x_k \in \mathcal{X}$, and any labels $y_1, ..., y_k \in \{0,1\}$, the mapping $h(x_i) = y_i$ for all $i \in [1,k]$ can be expressed by a hypothesis $h \in \mathcal{H}$.*

By definition, the inequality $\mathcal{I}_{\mathcal{X}}(\mathcal{H}) \leq \text{VC}(\mathcal{H})$ clearly holds, and it is equal for many common examples, including linear classifiers. Notice that the non-degeneracy assumption is necessary to avoid non-separability due to “trivial” reasons, such as having two different labels assigned to the same example.

Observe that while interpolation dimension of k requires all sets of points (under certain conditions) to be shattered by the hypothesis class, VC dimension of k requires all sets of points to **not** be shattered by the hypothesis class. That is, while VC dimension upper bounds the complexity of a hypothesis class the interpolation dimension does the exact opposite, i.e. it lower bounds the complexity of the given hypothesis class. The intuitive conjecture then is that since VC dimension acts as an upper bound on learnability and generalization for the class, interpolation dimension should provide the dual characterization: a hypothesis class is difficult to generalize over (need more samples) if it has a large interpolation dimension. Formalizing these ideas and developing a theory behind it is left for future work.

To illustrate this definition of interpolation dimensions as a measure for expressivity, its relation to the VC dimension and its applicability, consider the following examples:

1. The hypothesis class of linear hyperplanes of dimension k (with bias) has interpolation dimension $k+1$ for any $\gamma > 0$ non-degeneracy *if* the input domain $\mathcal{X}$ is restricted to linearly independent points. In this case, the VC dimension of this class is also equal to $k+1$. However, if the input domain $\mathcal{X}$ is *not* restricted to linearly independent

points, the interpolation dimension of this hypothesis class equals 2 for any $\gamma > 0$ non-degeneracy while the VC dimension remains unchanged.

Observe that the choice of input domain $\mathcal{X}$ is crucial and can determine whether there is equality or a huge gap between VC dimension and interpolation dimension.

2. Consider online learning of a Boolean function $\{0,1\}^{\log k} \mapsto \{0,1\}$. There are $2^{2^{\log k}} = 2^k$ such functions, and running an experts algorithm such as Hedge on all possible such functions results in $O(\sqrt{Tk})$ regret, but requires maintaining 2^k weights on the experts. On the other hand, online learning of a deep network (or any hypothesis class that can be learned efficiently) that has interpolation dimension k can learn the same class of functions, with regret that is $\mathrm{poly}(k)\sqrt{T}$ and $\mathrm{poly}(k)$ running time.

   Note that it is possible to learn this problem efficiently using an experts algorithms on the possible entries of the truth table of these functions.

Proof [Proof of Lemma 6] Let $\{(x_j, y_j)\}_{j=1}^k$ be a set of examples where $x_j \in \mathbb{S}_p$, $y_j \in [-1,1]^d$, and the x_j's are at least γ apart, i.e. $\min_{j \neq l} \|x_j - x_l\|_2 \geq \gamma$ with $\gamma \in (0, O\left(\frac{1}{H}\right)]$. Let $y_{j,i}$ denote the i-th coordinate of the label y_j, and recall that $f_i(\theta[i]; x)$ is the scalar output of the vector-valued deep neural network at coordinate i, with parameters $\theta[i]$ and input x. Fix any arbitrary $\varepsilon > 0$. By Theorem 1 in Allen-Zhu et al. (2019), for $m \geq \Omega(\frac{k^{24} H^{12} \log^5 m}{\gamma^8})$, $R = O(\frac{k^3 \log m}{\gamma \sqrt{m}})$, for any fixed $i \in [d]$, with probability at least $1 - e^{-\Omega(\log^2 m)}$, there exists $\theta^*[i]$ such that $\|\theta^*[i] - \theta_1[i]\|_F \leq R$, and

$$\sum_{j=1}^k (f_i(\theta^*[i]; x_j) - y_{j,i})^2 \leq \frac{\varepsilon}{d}.$$

The existence of such θ^* follows from the statement of the aforementioned theorem, i.e. gradient descent finds such θ^* in a finite number of iterations (convergence rate is irrelevant). Note that our choice of m and R satisfy the above conditions. Taking a union bound, we conclude that with probability at least $1 - d \cdot e^{-\Omega(\log^2 m)}$, there exists $\theta^* = (\theta^*[1], \ldots, \theta^*[d])$ such that for all i, $\|\theta^*[i] - \theta_1[i]\|_F \leq R$, and

$$\sum_{j=1}^k \|f(\theta^*; x_j) - y_j\|_2^2 = \sum_{i=1}^d \sum_{j=1}^k (f_i(\theta^*[i]; x_j) - y_{j,i})^2 \leq \varepsilon.$$

This conclusion is true for any $\varepsilon > 0$ and training set $\{(x_j, y_j)\}_{j=1}^k$ satisfying the stated conditions. In other words, we have that for $\mathcal{H} = \mathcal{H}_{\mathrm{NN}}(R; \theta_1)$

$$\forall \varepsilon > 0, \, \exists h \in \mathcal{H}, \quad \sum_{j=1}^k \|h(x_j) - y_j\|^2 \ \leq \ \varepsilon \implies \inf_{h \in \mathcal{H}} \left[\sum_{j=1}^k \|h(x_j) - y_j\|^2 \right] = 0 \, .$$

Thus, by the definition of interpolation dimension, the hypothesis class $\mathcal{H}_{\mathrm{NN}}(R; \theta_1)$, under input domain $\mathcal{X}$ and at non-degeneracy γ, has interpolation dimension at least k, which concludes the proof of this lemma. ■

A.2. Online nearly convex optimization

The full algorithm for extending OCO to nearly convex loss functions ℓ_t is presented in Algorithm 3. In addition to the proof of Lemma 3, we provide a corollary with OGD as the OCO algorithm $\mathcal{A}$ to use the explicit regret bound in further derivations. The proof of the corollary simply follows by plugging in the appropriate regret (and stepsize) value for OGD.

```
Algorithm 3 Online Nearly-Convex Optimization
Input: OCO algorithm $\mathcal{A}$ for convex decision set $\mathcal{K}$.
for $t = 1 \dots T$ do
    Play $\theta_t$, observe $\ell_t$.
    Construct $h_t(\theta) = \ell_t(\theta_t) + \nabla \ell_t(\theta_t)^\top (\theta - \theta_t)$.
    Update $\theta_{t+1} = \mathcal{A}(h_1, ..., h_t) \in \mathcal{K}$.
end
```

Proof [Proof of Lemma 3] Observe that by the ε-nearly convex property, for all $\theta \in \mathcal{K}$,

$$h_t(\theta) - \ell_t(\theta) = \ell_t(\theta_t) + \nabla \ell_t(\theta_t)^\top (\theta - \theta_t) - \ell_t(\theta) \leq \varepsilon.$$

Moreover, by construction the functions $h_t(\cdot)$ are convex and $h_t(\theta_t) = \ell_t(\theta_t)$ for all $t \in [T]$. The regret can be decomposed as follows, for any fixed $\theta^* \in \mathcal{K}$,

$$\sum_{t=1}^{T} \big(\ell_t(\theta_t) - \ell_t(\theta^*)\big) \leq \sum_{t=1}^{T} \big(h_t(\theta_t) - h_t(\theta^*)\big) + \varepsilon T \leq \mathrm{Regret}_T(\mathcal{A}) + \varepsilon T.$$

Taking $\theta^* \in \mathcal{K}$ to be the best decision in hindsight concludes the lemma proof. ■

Corollary 11 *Suppose $\{\ell_t\}_{t=1}^T$ are ε-nearly convex and let $\mathcal{A}$ be OGD with stepsizes $\eta_t = \frac{2R}{G} \cdot t^{-1/2}$, then Algorithm 3 has regret*

$$\sum_{t=1}^{T} \ell_t(\theta_t) - \min_{\theta^* \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(\theta^*) \leq 3RG\sqrt{T} + \varepsilon T,$$

where G is the gradient norm upper bound for all $\ell_t, t \in [T]$, and R is the radius of $\mathcal{K}$.

Appendix B. Online learning with two-layer neural networks

To showcase the key ideas behind the main result in this work and a connection to the NTK line fo works, we consider a simpler setting as a warmup: online learning of two-layer neural networks. The setup in this section along with many of the derivations follow that of Gao et al. (2019).

Two-layer Neural Networks. For inputs $x \in \mathbb{R}^p$, define the vector-valued two-layer neural network $f : \mathbb{R}^p \to \mathbb{R}^d$ with a smooth activation function $\sigma : \mathbb{R} \to \mathbb{R}$, even hidden layer width m and weights $\theta \in \mathbb{R}^{d \times m \times p}$ expressed as follows: for all $i \in [d]$ with parameter $\theta[i] \in \mathbb{R}^{m \times p}$, $f(\theta; x) = (f_1(\theta[1]; x), \dots, f_d(\theta[d]; x))^\top \in \mathbb{R}^d$, where

$$f_i(\theta[i]; x) = \frac{1}{\sqrt{m}} \big(\sum_{r=1}^{m/2} a_{i,r} \sigma(\theta[i,r]^\top x) + \sum_{r=1}^{m/2} \bar{a}_{i,r} \sigma(\bar{\theta}[i,r]^\top x) \big). \tag{B.1}$$

The parameter for each $i \in [d]$ is given by $\theta[i] = (\theta[i,1], \dots, \theta[i, \frac{m}{2}], \bar{\theta}[i,1], \dots, \bar{\theta}[i, \frac{m}{2}])$. The scaling factor $\frac{1}{\sqrt{m}}$ is chosen optimally in retrospect of the analysis. We initialize $a_{i,r}$ to be randomly drawn from $\{\pm 1\}$, choose $\bar{a}_{i,r} = -a_{i,r}$, and fix them throughout training. The initialization scheme for θ is as follows: for all $i \in [d]$, $\theta_1[i,r] \sim N(0, I_p)$ for $r = 1, \dots, \frac{m}{2}$, and $\bar{\theta}_1[i,r] = \theta_1[i,r]$. This symmetric initialization scheme is chosen so that $f_i(\theta_1[i]; x) = 0$ for all $x \in \mathbb{S}_p$ to avoid some technical nuisance. We make the following assumption on the general activation function:

Assumption 6 *The activation function σ is $C-$Lipschitz and $C-$smooth: $|\sigma'(z)| \leq C$, $|\sigma'(z) - \sigma'(z')| \leq C|z - z'|$.*

This warmup setting serves two purposes: (1) the key analysis structure is analogous to that of Section 3 with simpler details so this section can be a stepping stone to the main analysis; (2) to contrast with our approach of measuring the expressivity of the neural networks with interpolation dimension, in this setting we quantify the expressivity of the hypothesis class via the Neural Tangent Kernel (NTK) (Jacot et al., 2018) approach by relating the considered class of neural networks to a different function class.

Consider the neural network given in (B.1). Let K_σ denote the NTK of the two-layer network and $\mathcal{H}_{\text{RKHS}}(K_\sigma)$ denote the RKHS of the K_σ kernel. To obtain non-asymptotic guarantees, we restrict to RKHS functions of bounded norm. In this pursuit, we define the class of Random Feature functions $\mathcal{H}_{\text{RF}}(\infty)$, which is *dense* in $\mathcal{H}_{\text{RKHS}}(K_\sigma)$, construct its multidimensional analog $\mathcal{H}^d_{\text{RF}}(\infty)$, and restrict it to the functions $\mathcal{H}^d_{\text{RF}}(D)$ of bounded RF-norm D. See Appendix B.1 for formal treatment. The regret bound given below consists of two parts: the regret for learning the optimal neural network parameters in the parameter set Θ, and the approximation error of neural networks to the target function in $\mathcal{H}^d_{\text{RF}}(D)$.

Theorem 12 *Let f be a two-layer neural network as in (B.1) with the parameter set $\Theta = B(R; \theta_1) = \{\theta \in \mathbb{R}^{d \times m \times p} : \|\theta - \theta_1\|_F \leq R\}$, and suppose Assumptions 1, 2, 6 hold. For any $\delta > 0, D > 1$, take $R = D\sqrt{d}$, then with probability at least $1 - \delta$ over the random initialization, Algorithm 1 with OGD as the base algorithm and stepsizes $\eta_t = \frac{2R}{CL} \cdot t^{-1/2}$ satisfies*

$$\sum_{t=1}^T \ell_t(f(\theta_t; x_t)) \leq \min_{g \in \mathcal{H}^d_{\text{RF}}(D)} \sum_{t=1}^T \ell_t(g(x_t)) + \tilde{O}\left(\frac{L\sqrt{dp}CD^2T}{\sqrt{m}} \right) + O\left(CLD\sqrt{dT} + \frac{CLD^2dT}{\sqrt{m}} \right)$$

where $\tilde{O}(\cdot)$ hides factors that are polylogarithmic in δ, d.

The radius R being a constant w.r.t. m indicates small movement of the parameters (number of parameters is linear in m). The regret bound is increasing in terms of D which characterizes the expressivity of the benchmark function class $\mathcal{H}_{\mathrm{RF}}^d(D)$. Finally, to achieve ε average regret, it suffices to take large enough width $m = \Omega(\varepsilon^{-2})$ and large number of iterations $T = \Omega(\varepsilon^{-2})$.

Proof structure. The analysis of the above theorem goes through 3 main components similar to the proof of the deep net case:

1. **near convexity:** the losses $\ell_t(\theta) = \ell_t(f(\theta; x_t))$ are nearly convex (Lemma 13).
2. **regret guarantee:** regret is bounded against the parameter set $\Theta = B(R; \theta_1)$ (Lemma 14).
3. **expressivity:** any function $g \in \mathcal{H}_{\mathrm{RF}}^d(D)$ is approximated by a network (Lemma 15).

Lemma 13 *For any $\theta \in B(R; \theta_1)$ and any $t \in [T]$, the loss function $\ell_t(\theta) = \ell_t(f(\theta; x_t))$ is ε_{nc}-nearly convex as in (2) with $\varepsilon_{nc} = \frac{2CLR^2}{\sqrt{m}}$.*

Lemma 14 *Algorithm 1 with $\eta_t = \frac{2R}{CL} \cdot t^{-1/2}$ attains regret bound*

$$\sum_{t=1}^{T} \ell_t(\theta_t) \leq \min_{\theta \in B(R;\theta_1)} \sum_{t=1}^{T} \ell_t(\theta) + 3CLR \cdot \sqrt{T} + \frac{2CLR^2}{\sqrt{m}} \cdot T \ . \tag{B.2}$$

Lemma 15 *For any $\delta, D > 0$, let $g : \mathbb{R}^p \to \mathbb{R}^d \in \mathcal{H}_{\mathrm{RF}}^d(D)$, and let $R = D\sqrt{d}$, then with probability at least $1 - \delta$ over the random initialization of θ_1, there exists $\theta^* \in B(R; \theta_1)$ such that*

$$\forall x \in \mathbb{S}_p, \quad \ell_t(f(\theta^*; x)) \leq \ell_t(g(x)) + \frac{L\sqrt{d}CD^2}{2\sqrt{m}} + \frac{L\sqrt{d}CD}{\sqrt{m}}(2\sqrt{2p} + 2\sqrt{\log d/\delta}) \ .$$

B.1. Further details for Section B

Neural Tangent Kernel. The Neural Tangent Kernel (NTK) was first introduced in Jacot et al. (2018), who showed a connection between overparameterized neural networks and kernel methods. We characterize the net's expressivity by the capacity of learning functions in the RKHS of the NTK, which for our two-layer neural network has the following form:

Definition 16 *The NTK for the scalar two-layer neural network with activation σ and intialization distribution $\theta \sim \mathcal{N}(0, I_p)$ is defined as $K_\sigma(x, y) = \mathbb{E}_{\theta \sim \mathcal{N}(0,I_p)} \langle x\sigma'(\theta^\top x), y\sigma'(\theta^\top y) \rangle$.*

Let $\mathcal{H}(K_\sigma)$ denote the RKHS of the NTK. Intuitively, $\mathcal{H}(K_\sigma)$ represents the space of functions that can be approximated by a neural network with kernel K_σ. To obtain non-asymptotic approximation guarantees, we focus on RKHS functions of bounded norm, specifically the RF-norm as defined below.

Definition 17 **((Gao et al., 2019))** *Consider functions of the form*

$$h(x) = \int_{\mathbb{R}^d} c(w)^\top x \sigma'(w^\top x) dw.$$

Define the RF-norm of h as $\|h\|_{RF} = \sup_w \frac{\|c(w)\|_2}{p_0(w)}$, where $p_0(w)$ is the probability density function of $\mathcal{N}(0, I_p)$. Let

$$\mathcal{F}_{RF}(D) = \{h(x) = \int_{\mathbb{R}^d} c(w)^\top x \sigma'(w^\top x) dw \ : \ \|h\|_{RF} \le D\}, \tag{B.3}$$

and extend to the multi-dimensional case, $\mathcal{F}^d_{RF}(D) = \{h = (h_1, h_2, \dots, h_d) : h_i \in \mathcal{F}_{RF}(D)\}$.

By Lemma C.1 in Gao et al. (2019), the class of Random Feature functions, $\mathcal{F}_{RF}(\infty)$, is dense in $\mathcal{H}(K_\sigma)$ with respect to the $\|\cdot\|_{\infty,\mathbb{S}}$ norm, where $\|h\|_{\infty,\mathbb{S}} = \sup_{x\in\mathbb{S}_p} |h(x)|$. Since we are concerned with the approximation of the function value over the unit sphere, it is sufficient to consider $\mathcal{F}^d_{RF}(\infty)$, and further restrict to $\mathcal{F}^d_{RF}(D)$ for explicit nonasymptotic guarantees. The remaining of this section covers the proofs of the claims in Section B. We remark that the scaling factor in (B.1) is optimally chosen to be $b = \sqrt{m}$ in the proof of Theorem 12.

Proof [Proof of Theorem 12] Let $g \in \mathcal{F}^d_{RF}(D)$. By Lemma 15, with probability at least $1-\delta$ over the random initialization θ_1, there exists $\theta^* \in B(R)$ such that for all $x \in \mathbb{S}_p$,

$$\begin{aligned}\ell_t(f(\theta^*; x)) &\le \ell_t(g(x)) + \frac{L\sqrt{d}bCD^2}{2m} + \frac{L\sqrt{d}CD}{\sqrt{m}}(2\sqrt{2p} + 2\sqrt{\log d/\delta}) \\ &\le \ell_t(g(x)) + \tilde{O}\left(\frac{L\sqrt{dp}CD^2}{\sqrt{m}}\right),\end{aligned}$$

using the optimal scaling factor choice $b = \sqrt{m}$. By the regret guarantee in Lemma 14, Algorithm 1 has regret

$$\sum_{t=1}^T \ell_t(\theta_t) \le \min_{\theta\in B(R)} \sum_{t=1}^T \ell_t(\theta) + \frac{3CLR\sqrt{mT}}{b} + \frac{2CLR^2}{b}T \tag{B.4}$$

$$= \min_{\theta\in B(R)} \sum_{t=1}^T \ell_t(\theta) + O(CLR\sqrt{T} + \frac{CLR^2}{\sqrt{m}}T). \tag{B.5}$$

Combining them and using $R = D\sqrt{d}$, we conclude

$$\begin{aligned}\sum_{t=1}^T \ell_t(\theta_t) &\le \min_{\theta\in B(R)} \sum_{t=1}^T \ell_t(\theta) + O(CLR\sqrt{T} + \frac{CLR^2}{\sqrt{m}}T) \\ &\le \sum_{t=1}^T \ell_t(\theta^*) + O(CLR\sqrt{T} + \frac{CLR^2}{\sqrt{m}}T) \\ &\le \sum_{t=1}^T \ell_t(g(x_t)) + O(CLR\sqrt{T} + \frac{CLR^2}{\sqrt{m}}T) + \tilde{O}(\frac{L\sqrt{dp}CD^2T}{\sqrt{m}})\end{aligned}$$

The theorem follows by noticing that the inequality holds for any arbitrary $g \in \mathcal{F}^d_{RF}(D)$. ■

Proof [Proof of Lemma 13] We extend the original proof in Gao et al. (2019). Let $\operatorname{diag}(a_i)$ be a diagonal matrix with $(a_{1,i}, \dots, a_{m/2,i}, -a_{1,i}, \dots, -a_{m/2,i})$ on the diagonal. Note that the gradient of the network at the i-th coordinate is

$$\nabla_{\theta[i]} f_i(\theta[i]; x) = \frac{1}{b} \operatorname{diag}(a_i) \sigma'(\theta[i]x) x^\top. \tag{B.6}$$

We can show that the gradient is Lipschitz as follows, for all $x \in \mathbb{S}_p$,

$$\begin{aligned} \|\nabla_{\theta[i]} f_i(\theta[i]; x) - \nabla_{\theta[i]} f_i(\theta'[i]; x)\|_F &\leq \frac{1}{b} \|\operatorname{diag}(a_i)\|_2 \|\sigma'(\theta[i]x) - \sigma'(\theta'[i]x)\|_2 \|x\|_2 && \text{(B.7)} \\ &\leq \frac{C}{b} \|\theta[i] - \theta'[i]\|_F. && (|a_{r,i}| = 1, \|x\|_2 = 1) \end{aligned}$$

For each $\ell_t(f(\theta; x_t))$ according to the convexity property we have

$$\begin{aligned} \ell_t(\theta') - \ell_t(\theta) &\geq \nabla_f \ell_t(\theta)^\top (f(\theta'; x_t) - f(\theta; x_t)) \\ &= \sum_{i=1}^d \frac{\partial \ell_t(\theta)}{\partial f_i(\theta[i]; x_t)} (f_i(\theta'[i]; x_t) - f_i(\theta[i]; x_t)) \end{aligned}$$

For each $i \in [d]$, we use the fundamental theorem of calculus to rewrite function value difference as

$$\begin{aligned} f_i(\theta'[i]; x_t) - f_i(\theta[i]; x_t) &= \langle \nabla_{\theta[i]} f_i(\theta[i]; x_t), \theta'[i] - \theta[i] \rangle + \mathcal{R}(f_i, \theta[i], \theta'[i]) && \text{(B.8)} \\ \mathcal{R}(f_i, \theta[i], \theta'[i]) &= \int_0^1 \langle \nabla_{\theta[i]} f_i(s\theta'[i] + (1-s)\theta[i]; x_t) - \nabla_{\theta[i]} f_i(\theta[i]; x_t), \theta'[i] - \theta[i] \rangle ds. \end{aligned}$$

Note that since the gradient of f_i is Lipschitz given by (B.7), the residual term is bounded in magnitude as follows,

$$|\mathcal{R}(f_i, \theta[i], \theta'[i])| \ \leq \ \int_0^1 \frac{C}{b} \|s(\theta'[i] - \theta[i])\|_F \cdot \|\theta'[i] - \theta[i]\|_F ds = \frac{C}{2b} \|\theta'[i] - \theta[i]\|_F^2.$$

Hence we can show that the loss is nearly convex with respect to θ,

$$\begin{aligned} \ell_t(\theta') - \ell_t(\theta) &\geq \sum_{i=1}^d \frac{\partial \ell_t(\theta)}{\partial f_i(\theta[i]; x_t)} (f_i(\theta'[i]; x_t) - f_i(\theta[i]; x_t)) \\ &= \sum_{i=1}^d \frac{\partial \ell_t(\theta)}{\partial f_i(\theta[i]; x_t)} \left(\langle \nabla_{\theta[i]} f_i(\theta[i]; x_t), \theta'[i] - \theta[i] \rangle + \mathcal{R}(f_i, \theta[i], \theta'[i]) \right) \\ &\geq \ \sum_{i=1}^d \langle \frac{\partial \ell_t(\theta)}{\partial f_i(\theta[i]; x_t)} \nabla_{\theta[i]} f_i(\theta[i]; x_t), \theta'[i] - \theta[i] \rangle - \frac{C}{2b} \sum_{i=1}^d \left| \frac{\partial \ell_t(\theta)}{\partial f_i(\theta[i]; x_t)} \right| \cdot \|\theta'[i] - \theta[i]\|_F^2 \\ &\geq \ \langle \nabla_\theta \ell_t(\theta), \theta' - \theta \rangle - \frac{CL}{2b} \|\theta' - \theta\|_F^2, \end{aligned}$$

where the last inequality uses the L-Lipschitz property of the loss $\ell_t(\cdot)$ with respect to f. Using a diameter bound for $\theta, \theta' \in B(R)$ we get that $\|\theta - \theta'\|_F \leq 2R$ which results in near convexity of $\ell_t(\cdot)$ with $\varepsilon_{\text{nc}} = \frac{2CLR^2}{b}$ with respect to θ. ■

Proof [Proof of Lemma 14] The theorem statement is shown by using Corollary 11 and showing that the loss functions $\ell_t : \mathbb{R}^{d\times m\times p} \to \mathbb{R}^d$ satisfy near-convexity with respect to θ. First, the decision set in this case is $\mathcal{K} = B(R)$ so its radius is R. Lemma 13 shows that the loss functions $\ell_t(\theta)$ are ε_{nc}-nearly convex with $\varepsilon_{\text{nc}} = \frac{2CLR^2}{b}$. Finally, we can show that the gradient norm is bounded as follows,

$$\|\nabla_\theta \ell_t(\theta)\|_F^2 = \sum_{i=1}^d \|\nabla_{\theta[i]} \ell_t(\theta)\|_F^2 = \sum_{i=1}^d \left|\frac{\partial \ell_t(\theta)}{\partial f_i(\theta[i]; x_t)}\right|^2 \cdot \|\nabla_{\theta[i]} f_i(\theta[i]; x_t)\|_F^2 \;\leq\; \frac{C^2L^2m}{b^2},$$

where we use the L-Lipschitz property of $\ell_t(f(\theta; x))$ and the fact that the f_i gradient is bounded $\|\nabla_{\theta[i]} f_i(\theta[i]; x_t)\|_F \leq \sqrt{m}C/b$ given (B.6). This means that $G = \frac{CL\sqrt{m}}{b}$ and we can use the Corollary 11 to conclude the final statement in (B.2). ■

Lemma 18 *For any $\delta, D > 0$, let $g : \mathbb{R}^p \to \mathbb{R} \in \mathcal{F}_{RF}(D)$ and let $R' = \frac{bD}{\sqrt{m}}$, then for a fixed $i \in [d]$, with probability at least $1 - \delta$ over the random initialization θ_1, there exists $\theta^* \in \mathbb{R}^{m\times p}$ such that $\|\theta^* - \theta_1\|_F \leq R'$, and for all $x \in \mathbb{S}_p$,*

$$|f_i(\theta^*; x) - g(x)| \leq \frac{bCD^2}{2m} + \frac{CD}{\sqrt{m/2}}(2\sqrt{p} + \sqrt{2\log 1/\delta}).$$

Proof Since the neural network architectures are the same for all $i \in [d]$, we fix an arbitrary i and drop the index i for $\theta[i]$ throughout the proof. By Proposition C.1 in Gao et al. (2019), for any $\delta > 0$, with probability at least $1 - \delta$ over the randomness of θ_1, there exist $c_1, \cdots, c_{m/2} \in \mathbb{R}^p$ with $\|c_r\|_2 \leq \frac{2\|g\|_{RF}}{m}$ $\forall\, r \in [\frac{m}{2}]$, such that $g_1(x) = \sum_{r=1}^{m/2} c_r^\top x\sigma'((\theta_1[r])^\top x)$ satisfies

$$\forall\, x \in \mathbb{S},\ |g_1(x) - g(x)| \leq \frac{C\|g\|_{RF}}{\sqrt{m/2}}(2\sqrt{p} + \sqrt{2\log 1/\delta}),$$

where $\theta_1[r]$ represents the r-th row of θ_1. Now, we proceed to construct a θ^* such that $f_i(\theta^*; x)$ is close to $g_1(x)$. We note that by symmetric initialization $f_i(\theta_1; x) = 0$ for all $x \in \mathbb{S}_p$. Then, use the fundamental theorem of calculus similarly to (B.8) to decompose f_i as follows:

$$\begin{aligned}
f_i(\theta; x) &= f_i(\theta; x) - f_i(\theta_1; x) \\
&= \frac{1}{b}\Big(\sum_{r=1}^{m/2} a_r(\theta[r] - \theta_1[r])^\top x\sigma'((\theta_1[r])^\top x) - \sum_{r=1}^{m/2} a_r(\bar{\theta}[r] - \bar{\theta}_1[r])^\top x\sigma'((\bar{\theta}_1[r])^\top x)\Big) \\
&+ \frac{1}{b}\Big(\sum_{r=1}^{m/2} a_r \int_0^1 x^\top(\theta[r] - \theta_1[r])(\sigma'((t\theta[r] + (1-t)\theta_1[r])^\top x) - \sigma'((\theta_1[r])^\top x))dt \\
&- \sum_{r=1}^{m/2} a_r \int_0^1 x^\top(\bar{\theta}[r] - \bar{\theta}_1[r])(\sigma'((t\bar{\theta}[r] + (1-t)\bar{\theta}_1[r])^\top x) - \sigma'((\bar{\theta}_1[r])^\top x))dt\Big).
\end{aligned}$$

Consider $\theta^* \in \mathbb{R}^{m\times p}$ such that $\theta^*[r] = \theta_1[r] + \frac{b}{2}c_r a_r$, $\bar{\theta}^*[r] = \bar{\theta}_1[r] - \frac{b}{2}c_r a_r$, where $\bar{\theta}^*[r]^\top$ represents the $\frac{m}{2}+r$-th row of θ^*. Then

$$\|\theta^*[r]-\theta_1[r]\|_2,\ \|\bar{\theta}^*[r]-\bar{\theta}_1[r]\|_2 \le \frac{b\|g\|_{RF}}{m}, \qquad \text{and the linear part of } f_i \text{ satisfies}$$

$$\begin{aligned}
&\frac{1}{b}\big(\sum_{r=1}^{m/2} a_r(\theta^*[r]-\theta_1[r])^\top x\sigma'((\theta_1[r])^\top x) - \sum_{r=1}^{m/2} a_r(\bar{\theta}^*[r]-\bar{\theta}_1[r])^\top x\sigma'((\bar{\theta}_1[r])^\top x)\big)\\
&= \frac{1}{b}\big(\sum_{r=1}^{m/2} a_r^2\frac{b}{2}c_r^\top x\sigma'((\theta_1[r])^\top x) + \sum_{r=1}^{m/2} a_r^2\frac{b}{2}c_r^\top x\sigma'((\bar{\theta}_1[r])^\top x)\big)\\
&= \frac{1}{b}\big(\sum_{r=1}^{m/2} \frac{b}{2}c_r^\top x\sigma'((\theta_1[r])^\top x) + \sum_{r=1}^{m/2} \frac{b}{2}c_r^\top x\sigma'((\theta_1[r])^\top x)\big)\\
&= \sum_{r=1}^{m/2} c_r^\top x\sigma'((\theta_1[r])^\top x) = g_1(x).
\end{aligned}$$

Now we bound the residual part of f_i, by using the triangle inequality, and the smoothness of $\sigma(\cdot)$, as follows

$$\begin{aligned}
|f_i(\theta^*;x)-g_1(x)| = \frac{1}{b}\big|&\sum_{r=1}^{m/2} a_r\int_0^1 x^\top(\theta^*[r]-\theta_1[r])(\sigma'((t\theta^*[r]+(1-t)\theta_1[r])^\top x)-\sigma'((\theta_1[r])^\top x))dt\\
&-\sum_{r=1}^{m/2} a_r\int_0^1 x^\top(\bar{\theta}^*[r]-\bar{\theta}_1[r])(\sigma'((t\bar{\theta}^*[r]+(1-t)\bar{\theta}_1[r])^\top x)-\sigma'((\bar{\theta}_1[r])^\top x))dt\big|\\
&\le \frac{mC}{b}\frac{b^2}{4}\frac{4\|g\|_{RF}^2}{2m^2} = \frac{bC\|g\|_{RF}^2}{2m}.
\end{aligned}$$

Using the triangle inequality, we can bound the approximation error as follows,

$$\begin{aligned}
|f_i(\theta^*;x)-g(x)| &\le |f_i(\theta^*;x)-g_1(x)| + |g_1(x)-g(x)|\\
&\le \frac{bC\|g\|_{RF}^2}{2m} + \frac{C\|g\|_{RF}}{\sqrt{m/2}}(2\sqrt{p}+\sqrt{2\log 1/\delta}).
\end{aligned}$$

Finally, observe that θ^* is close to θ_1:

$$\|\theta^*-\theta_1\|_F^2 \le \sum_{r=1}^{m}\|\theta^*[r]-\theta_1[r]\|_2^2 \le \frac{b^2\|g\|_{RF}^2}{m} \le \frac{b^2D^2}{m} = (R')^2.$$

$\blacksquare$

Proof [Proof of Lemma 15] Let $g=(g_1,\ldots,g_d)\in\mathcal{F}_{RF}^d(D)$. By Lemma 18, if $R'=\frac{bD}{\sqrt{m}}$, with probability at least $1-\delta/d$, for each i there exists $\theta^*[i]$ such that $\|\theta^*[i]-\theta_1[i]\|_F\le R'$, and

$$|f_i(\theta^*[i];x)-g_i(x)| \le \frac{bCD^2}{2m} + \frac{CD}{\sqrt{m/2}}(2\sqrt{p}+\sqrt{2\log d/\delta}).$$

Let $\theta^* = (\theta^*[1], \ldots, \theta^*[d])$. Taking a union bound, with probability at least $1-\delta$,

$$\begin{aligned}\ell_t(f(\theta^*; x)) &= \ell_t(f_1(\theta^*[1]; x), \ldots, f_d(\theta^*[d]; x)) \\ &\le \ell_t(g_1(x), \ldots, g_d(x)) + L\sqrt{\sum_{i=1}^{d} \left(f_i(\theta^*[i]; x) - g_i(x)\right)^2} \\ &\le \ell_t(g(x)) + \frac{Lb\sqrt{d}CD^2}{2m} + \frac{L\sqrt{d}CD}{\sqrt{m/2}}(2\sqrt{d} + \sqrt{2\log d/\delta}).\end{aligned}$$

Finally, observe that $\|\theta^* - \theta_1\|_F \le \sqrt{d}R' = R$. ■

Appendix C. Analysis Outline

In this section, we showcase the key ideas behind the analysis of the main results in this work (see Appendix D for full details). The regret guarantee in online learning of deep neural networks, Theorem 5, is the main technical component of our work and is of potential independent interest. The theorem applies to the general online learning setting, i.e. high-dimensional outputs, adversarial data and adversarial convex losses, as well as agnostic bounds, which enables us to derive the final result in Theorem 7: episodic regret bounds in online episodic learning with neural network based policies.

Proof of Theorem 5. The structure of the theorem proof can be broken down into 3 main parts. First, one has to show that the considered loss functions $\ell_t : \Theta \to \mathbb{R}$, $\ell_t(\theta) = \ell_t(f(\theta; x_t))$ are *nearly convex* with respect to the parameter θ. This property holds due to the common observation in the community that in the described overparameterized regime the neural networks behave similar to their local linearization, and $\ell_t(\cdot)$ simply applies a convex loss over this local linearization.

Lemma 19 *Suppose $m \ge \Omega\left(\frac{p\log(1/R)+\log d}{R^{2/3}H}\right)$, and $R \le O\left(\frac{1}{H^6\log^3 m}\right)$, then with probability $1 - O(H)e^{-\Omega(mR^{2/3}H)}$ over the random initialization, for any $\theta \in B(R;\theta_1)$, $x \in \mathbb{S}_p$, and $t \in [T]$, the loss function $\ell_t(\theta) = \ell_t(f(\theta; x))$ is ε_{nc}-nearly convex with $\varepsilon_{nc} = O\left(R^{4/3}LH^{5/2}\sqrt{dm\log m}\right)$.*

The near convexity of the loss functions $\ell_t(\theta)$ for all $\theta \in B(R;\theta_1)$ along with the observations from Section 2.2, in particular Lemma 3, result in a regret bound over the parameter set $\Theta = B(R;\theta_1)$. The bound itself is comprised of the regret of the OCO algorithm used for parameter update, sublinear in learning horizon T, and the worst-case linear penalty of near convexity $\varepsilon_{\mathrm{nc}} \cdot T$.

Lemma 20 *Under conditions of Lemma 19, Algorithm 1 with $\mathcal{A}$ as the projected OGD algorithm over stepsizes $\eta_t = \frac{2R\sqrt{d}}{LH\sqrt{m}} \cdot t^{-1/2}$ suffers regret*

$$\sum_{t=1}^{T} \ell_t(\theta_t) \le \min_{\theta \in B(R;\theta_1)} \sum_{t=1}^{T} \ell_t(\theta) + O\left(RLH\sqrt{dmT}\right) + O\left(R^{4/3}LH^{5/2}\sqrt{dm\log m}T\right) . \quad \text{(C.1)}$$

The obtained regret bound is with respect to the best-in-hindsight parameter $\theta^\star$ in the parameter set $\Theta = B(R;\theta_1)$ which is a ball around the initialization. The bound above can be interpreted in terms of the radius of the parameter set: an increasing radius R means a stronger comparator $\theta^\star$ but a larger regret bound at the same time. This tradeoff is natural but the neural network class given by parameters $\theta \in B(R;\theta_1)$ has its capacity characterized by R, which is a formulation that is highly specific to the considered setup. Instead, we use the notion of interpolation dimension to characterize function class expressivity and use Lemma 6 to derive the final guarantee of the theorem where the described tradeoff is in terms of the interpolation dimension itself.

Appendix D. Details and Proofs for Section 3

This and the subsequent sections detail the analysis and proofs of the main theorems in this work following the outline of C. A simpler warmup setting is introduced and analyzed in isolation in Appendix B to provide more intuition behind the deep learning derivations.
Proof [Proof of Theorem 5] To prove this theorem, we will use both Lemmas 20 and 6. First, let us verify that the conditions of Lemma 20, i.e. conditions of Lemma 19, are satisfied given the choice of m, R in the theorem statement. Indeed, under our choice of m, as long as $m \geq \frac{c_1 k^6 \log^8 m H^{12}}{\gamma^2}$ for some sufficiently large $c_1 > 0$, we have $\frac{k^3 \log m}{\gamma\sqrt{m}} \leq \frac{1}{\sqrt{c_1} H^6 \log^3 m}$. Suppose for some constant c_2, taking

$$R = \frac{c_2 k^3 \log m}{\gamma\sqrt{m}}$$

satisfies the condition required for Lemma 6. Then we can set c_1 to be large enough such that $\frac{c_2}{\sqrt{c_1}} \leq c'$ for c' specified in Lemma 19, and choosing

$$m \geq \frac{c_1 p^{3/2}(k^{24}H^{12}\log^8 m + d)^{3/2}}{\gamma^8} \geq \Omega\left(\frac{k^{24}H^{12}\log^5 m}{\gamma^8}\right)$$

gives us an R that satisfies the Lemma 19's condition.

For the condition on m, we simply have

$$\begin{aligned} mR^{2/3}H = \frac{c_2^{2/3}k^2 m^{2/3} H \log^{2/3} m}{\gamma^{2/3}} &\geq \frac{(c_1 c_2)^{2/3} p (k^{24}H^{12}\log^8 m + d) k^2 H \log^{2/3} m}{\gamma^6} \\ &\geq (c_1 c_2)^{2/3} p (k^{24}H^{12}\log^8 m + d) \\ &\geq \Omega(p \log O(1/R) + \log d). \end{aligned}$$

Observe that under these choices of m, R the conditions from Lemma 6 are trivially satisfied. Hence, we plug in the value of R into the regret bound (C.1) in Lemma 20 and use Lemma 6 to conclude the final regret bound in Theorem 5. Finally, note that $mR^{2/3}H = \Omega(\log^2 m)$, and by taking a union bound over the events of Lemma 6 and Lemma 20, the failure probability for the regret bound is

$$d \cdot e^{-\Omega(\log^2 m)} + O(H) \cdot e^{-\Omega(mR^{2/3}H)} = O(H + d) \cdot e^{-\Omega(\log^2 m)} \ .$$

This concludes the theorem, verifying that the failure probability is low, since $m \gg \max(d, H)$.
■

Proof [Proof of Lemma 19] Our proof extends Lemma A.6 in Gao et al. (2019) to our setting, where the loss is defined over a vector whose coordinates are outputs of different deep neural networks. A δ-net over $\mathbb{S}_p$ is defined as a collection of points $\{x_r\} \in \mathbb{S}_p$ such that for all $x \in \mathbb{S}_p$, there exists an x_j in the δ-net such that $\|x_j - x\|_2 \leq \delta$. Consider a δ-net of the unit sphere consisting of $\{x_r\}_{r=1}^{N}$, and standard results show that such a δ-net exists with $N = (O(1/\delta))^p$. Let $i \in [d]$ and $r \in [N]$. By Lemma A.5 in Gao et al. (2019), if $m \geq \max\{d, \Omega(H \log H)\}$, $R + \delta \leq \frac{c}{H^6 \log^3 m}$ for some sufficiently small constant c, then with probability at least $1 - O(H)e^{-\Omega(m(R+\delta)^{2/3}H)}$ over the random initialization, for any $\theta'[i], \theta[i] \in B(R)$ and any $x' \in \mathbb{S}_p$ with $\|x' - x_r\|_2 \leq \delta$,

$$\|\nabla_{\theta^h[i]} f_i(\theta'[i]; x') - \nabla_{\theta^h[i]} f_i(\theta[i]; x')\|_F = O((R+\delta)^{1/3} H^2 \sqrt{m \log m}),$$

$$\|\nabla_{\theta^h[i]} f_i(\theta'[i]; x')\|_F = O(\sqrt{mH}),$$

where $\theta^h[i]$ denotes the parameter for layer h in the network for the i-th coordinate of the output. Summing over the layers, we have

$$\|\nabla_{\theta[i]} f_i(\theta'[i]; x') - \nabla_{\theta[i]} f_i(\theta[i]; x')\|_F = O((R+\delta)^{1/3} H^{5/2} \sqrt{m \log m}),$$

$$\|\nabla_{\theta[i]} f_i(\theta'[i]; x')\|_F = O(H\sqrt{m}).$$

Similar to (B.8), we can write the difference of f_i evaluated on $\theta'[i]$ and $\theta[i]$ as a sum of a linear term and a residual term $\mathcal{R}(f_i, \theta[i], \theta'[i], x')$ using the Fundamental Theorem of Calculus,

$$f_i(\theta'[i]; x') - f_i(\theta[i]; x') = \langle \nabla_{\theta[i]} f_i(\theta[i]; x'), \theta'[i] - \theta[i] \rangle + \mathcal{R}(f_i, \theta[i], \theta'[i], x') \tag{D.1}$$

$$\mathcal{R}(f_i, \theta[i], \theta'[i], x') = \int_0^1 \left\langle \nabla_{\theta[i]} f_i(s\theta'[i] + (1-s)\theta[i]; x') - \nabla_{\theta[i]} f_i(\theta[i]; x'), \theta'[i] - \theta[i] \right\rangle ds \tag{D.2}$$

Since we can bound the change of the gradient, we can bound the residual term as follows

$$\begin{aligned} |\mathcal{R}(f_i, \theta[i], \theta'[i], x')| &\leq \int_0^1 \|\nabla_{\theta[i]} f_i(s\theta'[i] + (1-s)\theta[i]; x') - \nabla_{\theta[i]} f_i(\theta[i]; x')\|_F \|\theta'[i] - \theta[i]\|_F ds \\ &\leq O\big((R+\delta)^{1/3} H^{5/2} \sqrt{m \log m}\big) \|\theta'[i] - \theta[i]\|_F. \end{aligned}$$

Taking a union bound over the i's, with probability at least $1 - O(H)de^{-\Omega(m(R+\delta)^{2/3}H)}$, for all x' such that $\|x' - x_r\|_2 \le \delta$,

$$\begin{aligned}
\ell_t(f(\theta';x')) - \ell_t(f(\theta;x')) &\ge \sum_{i=1}^{d} \frac{\partial \ell_t(f(\theta;x'))}{\partial f_i(\theta[i];x')}(f_i(\theta'[i];x') - f_i(\theta[i];x')) \\
&= \sum_{i=1}^{d} \frac{\partial \ell_t(f(\theta;x'))}{\partial f_i(\theta[i];x')}\left(\langle \nabla_{\theta[i]} f_i(\theta[i];x'), \theta'[i] - \theta[i]\rangle + \mathcal{R}(f_i, \theta[i], \theta'[i], x')\right) \\
&\ge \ \sum_{i=1}^{d} \langle \frac{\partial \ell_t(f(\theta;x'))}{\partial f_i(\theta[i];x')} \nabla_{\theta[i]} f_i(\theta[i];x'), \theta'[i] - \theta[i]\rangle \\
&\quad - O\big((R+\delta)^{1/3} H^{5/2}\sqrt{m\log m}\big) \sum_{i=1}^{d} \left| \frac{\partial \ell_t(f(\theta;x'))}{\partial f_i(\theta[i];x')} \right| \cdot \|\theta'[i] - \theta[i]\|_F \\
&\ge \ \langle \nabla_\theta \ell_t(f(\theta;x')), \theta' - \theta\rangle - O\big((R+\delta)^{1/3} H^{5/2}\sqrt{m\log m}\big) L\sqrt{d}R.
\end{aligned}$$

We take $\delta = R$, and by our choice of R, the condition $R + \delta \le \frac{c}{H^6 \log^3 m}$ is satisfied. Taking a union over bound all points in the δ-net, the above inequality holds for all $x \in \mathbb{S}_p$ with probability at least

$$\begin{aligned}
1 - dO(H)O(1/R)^p e^{-\Omega(mR^{2/3}H)} &= 1 - O(H)e^{-\Omega(mR^{2/3}H) + p\log(O(1/R)) + \log d} \\
&= 1 - O(H)e^{-\Omega(mR^{2/3}H)},
\end{aligned}$$

where the last inequality is due to our choice of m. This applies to the gradient bound too, i.e.

$$\|\nabla_{\theta[i]} f_i(\theta[i];x)\|_F = O(H\sqrt{m}), \ \forall i \in [d], \tag{D.3}$$

holds for any $\theta \in B(R)$ and any $x \in \mathbb{S}_p$ with the same failure probability. ■

Proof [Proof of Lemma 20] Given that the identical conditions of Lemma 19 hold, then with probability at least $1 - O(H)e^{-\Omega(mR^{2/3}H)}$ over the randomness of θ_1, ℓ_t is ε_{nc}-nearly convex with $\varepsilon_{\text{nc}} = O(R^{4/3}H^{5/2}\sqrt{m\log m}L\sqrt{d})$, and $\|\nabla_{\theta[i]} f_i(\theta[i];x)\|_F \le O(H\sqrt{m})$ according to (D.3) for all $i \in [d], x \in \mathbb{S}_p, \theta \in B(R)$. Since the decision set is $B(R)$, its radius in Frobenius norm is at most $R\sqrt{d}$. We can bound the gradient norm as follows, for all $x \in \mathbb{S}_p$,

$$\begin{aligned}
\|\nabla_\theta \ell_t(f(\theta;x))\|_F^2 &= \sum_{i=1}^{d} \|\nabla_{\theta[i]} \ell_t(f_i(\theta[i];x))\|_F^2 \\
&= \sum_{i=1}^{d} \left| \frac{\partial \ell_t(f(\theta;x))}{\partial f_i(\theta[i];x)} \right|^2 \cdot \|\nabla_{\theta[i]} f_i(\theta[i];x)\|_F^2 \\
&\le L^2 \max_i \|\nabla_{\theta[i]} f_i(\theta[i];x)\|_F^2 \le O(L^2H^2m).
\end{aligned}$$

By Corollary 11, the regret is bounded by

$$3R\sqrt{d}G\sqrt{T} + \varepsilon T \le O(RLH\sqrt{dmT}) + O(R^{4/3}H^{5/2}TL\sqrt{dm\log m}) \ .$$

which concludes the proof. ■

D.1. Auxiliary Lemmas

Lemma 21 *For* $m \geq \Omega(\frac{p\log(1/R)+\log(d/\delta)}{R^{2/3}H})$, *and* $R = O(\frac{1}{H^6\log^3 m})$, *with probability at least* $1-\delta$ *over the randomness of initialization, for all* $x \in \mathbb{S}_p$ *and all* $i \in [d]$, $|f_i(\theta_1[i];x)| \leq O\left(\sqrt{\log\frac{d}{\delta}} + \sqrt{p\log\frac{1}{R}}\right)$.

Proof As in the proof of Lemma 19, we consider an ε-net consisting of $O(1/\varepsilon)^p$ points over the unit sphere in dimension p, and fix x_r in the ε-net. Let $i \in [d]$, and define $B_i(R) = \{\theta[i] : \|\theta[i]-\theta_1[i]\|_2 \leq R\}$. Let $f_i^h(\theta[i];x)$ denote output at the h-th layer of the network after activation, with weights $\theta[i]$ and input x.

By Lemma A.4 in Gao et al. (2019), if $R = O(1)$, with probability $1-O(H)e^{-\Omega(m/H)}$ over random initialization, for any $x' \in \mathbb{S}_p$ such that $\|x_r - x'\|_2 \leq \varepsilon$, and any $\theta[i] \in B_i(R)$, in particular $\theta_1[i]$, there exists $\tilde{\theta}[i] \in B_i(R+O(\varepsilon))$ such that

$$f_i^H(\tilde{\theta}[i];x_r) = f_i^H(\theta_1[i];x').$$

We first decompose the output of the neural net as follows,

$$\begin{aligned}|f_i(\theta_1[i];x')| = |a^\top f_i^H(\theta_1[i];x')| &= |a^\top f_i^H(\tilde{\theta}[i];x_r)| \\ &\leq |a^\top (f_i^H(\tilde{\theta}[i];x_r) - f_i^H(\theta_1[i];x_r))| + |a^\top f_i^H(\theta_1[i];x_r))|.\end{aligned}$$

Note that since $a \sim \mathcal{N}(0, I_m)$, for any fixed vector v, we have $a^\top v \sim \mathcal{N}(0, \|v\|_2^2)$. By Hoeffding's inequality, for all $c \geq 0$

$$\mathbb{P}[|a^\top v| \geq c\|v\|] \leq 2e^{-\frac{c^2}{2}}.$$

Now we bound the first term. According to Lemma 8.2 in Allen-Zhu et al. (2019), for $R + O(\varepsilon) \leq \frac{c'}{H^6\log^3 m}$ for some sufficiently small c', with probability at least $1 - e^{-\Omega(m(R+O(\varepsilon))^{2/3}H)}$, $\|f_i^H(\tilde{\theta}[i];x_r) - f_i^H(\theta_1[i];x_r)\|_2 \leq c_1 \cdot (R+O(\varepsilon))H^{5/2}\sqrt{\log m}$ for some constant c_1. Under this event, with probability at least $1-\delta'$,

$$\begin{aligned}|a^\top (f_i^H(\tilde{\theta}[i];x_r) - f_i^H(\theta_1[i];x_r))| &\leq \sqrt{2\ln\left(\frac{2}{\delta'}\right)}c_1(R+O(\varepsilon))H^{5/2}\sqrt{\log m} \\ &= O\left(\sqrt{\ln\frac{1}{\delta'}}(R+O(\varepsilon))H^{5/2}\sqrt{\log m}\right).\end{aligned}$$

For the second term, by Lemma A.2 in Gao et al. (2019), with probability at least $1-O(H)e^{-\Omega(m/H)}$ over the randomness of $\theta_1[i]$, $\|f_i^H(\theta_1[i];x_r)\|_2 \leq c_2$ for some constant c_2. Under this event, with probability at least $1-\delta'$,

$$|a^\top f_i^H(\theta_1[i];x_r))| \leq O\left(\sqrt{\ln\frac{1}{\delta'}}\right).$$

We take $\varepsilon = R$, and $R = O(\frac{1}{H^6\log^3 m})$, then the conditions on R and ε are satisfied.

We set $\delta' = \frac{\delta O(R)^p}{d}$, with our choice of m and R, $O(H)e^{-\Omega(m/H)} = e^{-\Omega(m/H)}$, and $e^{-\Omega(mR^{2/3}H)} \leq \delta'$. Taking a union bound on the mentioned events, with probability at least $1-\delta'$,

$$|f_i(\theta_1[i];x')| \leq O\left(\sqrt{\ln\frac{1}{\delta'}}RH^{5/2}\sqrt{\log m}\right) + O\left(\sqrt{\ln\frac{1}{\delta'}}\right) = O\left(\sqrt{\ln\frac{d}{\delta}} + \sqrt{p\ln\frac{1}{R}}\right).$$

Now take a union bound over the ε-net and over the d coordinates, we conclude that for all $x \in \mathbb{S}_p$, for all $i \in [d]$ $|f_i(\theta_1[i];x)| \leq O\left(\sqrt{\ln\frac{d}{\delta}} + \sqrt{p\ln\frac{1}{R}}\right)$ with probability at least

$$1 - dO(1/R)^p\delta' = 1-\delta.$$

■

Lemma 22 *For $m \geq \Omega(\frac{p^{3/2}(k^{24}H^{12}\log^8 m+d)^{3/2}}{\gamma^8})$, and $R = O\left(\frac{k^3\log m}{\gamma\sqrt{m}}\right)$, with probability at least $1-O(H+d)e^{-\Omega(\log^2 m)}$ over the randomness of initialization, for all $x \in \mathbb{S}_p$ and all $\theta \in B(R)$, for all $i \in [d]$, $|f_i(\theta[i];x)| \leq O\left(\frac{k^3(H+\sqrt{p})\log m}{\gamma}\right)$.*

Proof Observe that for each $i \in [d]$ and any $x \in \mathbb{S}_p$, the inequality

$$|f_i(\theta[i];x)| \ \leq \ |f_i(\theta_1[i];x)| + |f_i(\theta[i];x) - f_i(\theta_1[i];x)|$$

holds. The choice of m, R satisfies the conditions in Lemma 21, take $\delta = de^{-\Omega(\log^2 m)}$ and note that $mR^{2/3}H = \Omega(\log^2 m)$. We can use the decomposition in (D.1) to bound the difference between the neural network output at $\theta[i]$ and that at $\theta_1[i]$.

$$f_i(\theta[i];x) - f_i(\theta_1[i];x) = \int_0^1 \left\langle \nabla_{\theta[i]} f_i(s\theta[i] + (1-s)\theta_1[i];x), \theta[i]-\theta_1[i]\right\rangle ds$$

By Lemma 19, with our choice of m and R, with probability at least $1-O(H)e^{-\Omega(\log^2 m)}$,

$$\|\nabla_{\theta[i]} f_i(s\theta[i] + (1-s)\theta_1[i];x)\|_F = O(H\sqrt{m}), \ \forall \ s \in [0,1].$$

Therefore the integral can be bounded as

$$\begin{aligned}&\left|\int_0^1 \left\langle \nabla_{\theta[i]} f_i(s\theta[i] + (1-s)\theta_1[i];x), \theta[i]-\theta_1[i]\right\rangle ds\right| \\ &\leq \int_0^1 \|\nabla_{\theta[i]} f_i(s\theta[i] + (1-s)\theta_1[i];x)\|_F \|\theta[i]-\theta_1[i]\|_F ds \\ &\leq O(RH\sqrt{m}).\end{aligned}$$

Combining with Lemma 21, we conclude that

$$|f_i(\theta[i];x)| \leq |f_i(\theta[i];x) - f_i(\theta_1[i];x)| + |f_i(\theta_1[i];x)| \leq O\left(\frac{k^3(H+\sqrt{p})\log m}{\gamma}\right), \quad \forall i \in [d]$$

with probability at least $1-O(H+d)e^{-\Omega(\log^2 m)}$. ■

Appendix E. Details and Proofs for Section 4

We provide some additional notations and terms before going ahead with the proofs in this section.

Dynamics rollout. Before proving the lemmas necessary for the theorem proof, we rewrite the state x_k^θ by rolling out the dynamics from $i = k$ to $i = 1$ as follows

$$x_k^\theta = x_k^{\text{nat}} + \sum_{i=1}^{k-1} M_i^k f(\theta; \bar{z}_i),\; x_k^{\text{nat}} = \prod_{j=k-1}^{1} A_j x_1 + \sum_{i=1}^{k-1} \prod_{j=k-2}^{i} A_j w_i,\; M_i^k = \prod_{j=k-1}^{i+1} A_j \cdot B_i,$$

and for simplicity $\|x_1\|_2 \leq W$.

Sequential stabilizability. Furthermore, note that Assumption 4 can be relaxed to assuming there exists a sequence of linear operators $F_{1:K}$ such that for $C_1 \geq 1$ and $\rho_1 \in (0, 1)$

$$\forall k \in [K], n \in [1, k), \qquad \left\| \prod_{i=k}^{k-n+1} (A_i + B_i F_i) \right\|_{\text{op}} \;\leq\; C_1 \cdot \rho_1^n \;.$$

This condition is called *sequential stabilizability* and it reduces to the stable case by taking the actions $u_k' = F_k x_k + u_k$, yielding the stable dynamics of $(A_k + B_k F_k, B_k)_{1:K}$.

Proof [Proof of Lemma 9] This lemma is shown by reducing it to the interpolation dimension lemma for deep neural networks, Lemma 6. The class of policies $\Pi_{\text{dnn}}(f; \Theta)$ is at the same time a hypothesis class of functions of type $\mathbb{R}^{K \cdot d_x + 1} \to \mathbb{R}^{d_u}$, i.e. $p = K \cdot d_x + 1$, $d = d_u$. Observe that the domain is still the unit sphere $\mathcal{X} = \mathbb{S}_{K \cdot d_x + 1}$ given the normalization of inputs $\bar{z}_k$. Furthermore, the inputs are separated in ℓ_2 norm by $\gamma > 0$ for $\gamma = \frac{1}{2KW+H}$:

$$\forall k \in [K], \|z_k\|_2^2 \leq K \cdot W^2 + K^2 \;\leq\; K^2(W^2 + 1) \;\leq\; 4K^2W^2,$$

assuming $W = \max(1, W)$ since $\max_{k \in [K]} \|w_k\|_2 \leq W$ according to Assumption 3. This means that

$$\forall j, l \in [K],\; \|\bar{z}_j - \bar{z}_l\|_2^2 \geq \left(\frac{k}{\|z_k\|_2} - \frac{l}{\|z_l\|_2} \right) \geq \frac{1}{4K^2W^2},$$

so taking $\gamma = \frac{1}{2KW+H}$ satisfies separability and also the condition in Lemma 6. Finally, the conditions on m, R coincide with those in Lemma 6 for $\gamma = \frac{1}{2KW+H}$ and interpolation dimension K. Hence, according to Lemma 6, the function class $\Pi_{\text{dnn}}(f; \Theta)$, with probability $1 - d_u \cdot e^{-\Omega(\log^2 m)}$, has interpolation dimension $\mathcal{I}_{\mathcal{X},\gamma}(\Pi_{\text{dnn}}(f; \Theta)) \geq K$. Therefore, by definition of interpolation dimension, the function class $\Pi_{\text{dnn}}(f; \Theta)$ can *interpolate* any dataset of size K, e.g. inputs $\{\bar{z}_k\} + k = 1^K$ and labels $\{u^\star\}_{k=1}^K$ for any arbitrary fixed $u_{1:K}^\star \in [-1, 1]^{K \times d_u}$. This directly implies that it can output any open-loop control sequence $u_{1:K}^*$ of length K up to arbitrary precision, including the optimal one. ∎

Proof [Proof of Theorem 7] The proof is very similar to that of Theorem 5. The theorem conditions are at least as strong as those in Lemma 27, hence we can use Lemma 27 to claim that $\mathcal{L}_t(\theta)$ is ε_{nc}-nearly convex with $\varepsilon_{nc} = O(L_c R^{4/3} H^{9/2} K^6 W d_u \sqrt{d_x m} \log^{3/2} m)$, and $\|\nabla_{\theta[i]} f_i(\theta[i]; \bar{z}_k^t)\|_F \leq O(H\sqrt{m})$ for all $i \in [d], \bar{z}_k^t \in \mathbb{S}_{K \cdot d_x + 1}, \theta \in B(R; \theta_1)$. We first bound the gradient norm of $\mathcal{L}_t(\theta)$:

$$\begin{aligned}
\|\nabla_\theta \mathcal{L}_t(\bar{f}(\theta))\|_F^2 &= \|\sum_{k=1}^{K}\sum_{i=1}^{d_u} \frac{\partial \mathcal{L}(\theta)}{\partial f_i(\theta[i]; \bar{z}_k^t)} \nabla_{\theta[i]} f_i(\theta[i]; \bar{z}_k^t)\|_F^2 \\
&\leq \sum_{k=1}^{K}\sum_{i=1}^{d_u} \left| \frac{\partial \mathcal{L}_t(\theta)}{\partial f_i(\theta[i]; \bar{z}_k^t)} \right|^2 \cdot \|\nabla_{\theta[i]} f_i(\theta[i]; \bar{z}_k^t)\|_F^2 \\
&\leq O(K L_c'^2) \max_{i,k} \|\nabla_{\theta[i]} f_i(\theta[i]; \bar{z}_k^t)\|_F^2 \\
&\leq O(K^{11} L_c^2 H^6 W^2 d_u d_x m \log^2 m),
\end{aligned}$$

where the second to last inequality is due to Lemma 26 and the last inequality holds because $L_c' = O(K^5 L_c H^2 W \sqrt{d_x d_u} \log m)$. We can proceed to bound the regret as follows

$$\begin{aligned}
3R\sqrt{d_u}G\sqrt{T} + \varepsilon_{nc}T &\leq O(RL_cK^{11/2}H^3Wd_u\sqrt{d_x m}\log m\sqrt{T}) + \\
&\quad + O(R^{4/3}L_cK^6H^{9/2}Wd_u\sqrt{d_x m}\log^{3/2} mT) \\
&= \tilde{O}(K^{19/2}L_cH^4W^2d_u\sqrt{d_x}\cdot\sqrt{T}) + \tilde{O}(\frac{K^{34/3}L_cH^{35/6}W^{7/3}d_u\sqrt{d_x}}{m^{1/6}}\cdot T) \\
&= \tilde{O}(K^{10}L_cH^4W^2d_ud_x^{1/2}\cdot\sqrt{T}) + \tilde{O}\left(\frac{K^{12}L_cH^6W^3d_ud_x^{1/2}}{m^{1/6}}\cdot T\right).
\end{aligned}$$

■

Lemma 23 *The function $\mathcal{L}(\bar{f}(\theta))$ is convex in $\bar{f}(\theta)$.*

Proof The function $\mathcal{L}(\bar{f}(\theta))$ is a sum of K functions. For an arbitrary $k \in [K]$, note that x_k^θ is a affine function of $\bar{f}(\theta)$ w.r.t. the components $f(\theta, \bar{z}_i), i = 1, \ldots, K$. The other argument is $f(\theta; \bar{z}_k)$ which is also an affine function of $\bar{f}(\theta)$. Hence, both arguments in $c_k(\cdot, \cdot)$, which is jointly convex in its arguments, are affine in $\bar{f}(\theta)$, which means that $c_k(x_k^\theta, f(\theta; \bar{z}_k))$ is convex in $\bar{f}(\theta)$. Since $\mathcal{L}(\bar{f}(\theta))$ is defined as the sum over $c_k(x_k^\theta, f(\theta; \bar{z}_k))$, it is also convex in the argument $\bar{f}(\theta)$. ■

Lemma 24 *Under the identical conditions of Lemma 22, the states and actions over an episode are bounded,* $\max_k \|u_k^\theta\|_2 \leq D_u$ *and* $\max_k \|x_k^\theta\|_2 \leq D_x$ *for* $D_u = O(K^5H^2W\sqrt{d_ud_x}\log m)$, $D_x = \frac{C_1}{1-\rho_1} \cdot (W + D_uC_2)$.

Proof First, note that $u_k^\theta = f(\theta; \bar{z}_k)$ and $\bar{z}_k \in \mathbb{S}_{K \cdot d_x + 1}$. Given the output magnitude bound for the network in Lemma 22, i.e. $\|u_k^\theta[i]\| \leq O(K^3(H + \sqrt{Kd_x + 1})(2KW + H)\log m)$ we have $\|u_k^\theta\|_2 \leq O(\sqrt{d_u}K^3(H + \sqrt{Kd_x + 1})(2KW + H)\log m) = O(K^5H^2W\sqrt{d_ud_x}\log m) = D_u$. By definition of x_k^{nat}, we have that

$$\|x_k^{\text{nat}}\|_2 \leq W \cdot \frac{C_1}{1-\rho_1}$$

Plugging this bound in the expression for x_k^θ, we get

$$\|x_k^\theta\|_2 \le W \cdot \frac{C_1}{1-\rho_1} + D_u \cdot \sum_{i=1}^{k-1} C_2 \cdot C_1 \cdot \rho_1^{k-i-1} \;\le\; \frac{C_1}{1-\rho_1} \cdot (W + D_u C_2).$$

■

Corollary 25 *The cost function c_k is L_c'-Lipschitz with $L_c' = L_c \cdot \max\{1, D_x + D_u\}$.*

Lemma 26 *The function $\mathcal{L}(\bar{f}(\theta))$ is L-Lipschitz w.r.t. each $f(\theta;\bar{z}_k)$ for $k \in [K]$ with $L = L_c' \cdot \frac{C_2 \cdot C_1}{1-\rho_1}$, i.e. $L = O(K^5 L_c H^2 W \sqrt{d_x d_u} \log m)$ under the identical conditions of Lemma 22.*

Proof We use Corollary 25 with L_c' to conclude this lemma statement. For any arbitrary $k \in [K]$, denote $f_k = f(\theta;\bar{z}_k)$ and note that in the expression of $\mathcal{L}(\bar{f}(\theta))$ we have

$$\begin{aligned}
&\forall i < k, \quad \|\nabla_{f_k} c_i(x_k^\theta, u_k^\theta)\|_2 = 0,\\
&\text{for } i = k, \quad \|\nabla_{f_k} c_i(x_k^\theta, u_k^\theta)\|_2 = \|\nabla_u c_i(x_k^\theta, u_k^\theta)\|_2 \;\le\; L_c',\\
&\forall i > k, \quad \|\nabla_{f_k} c_i(x_k^\theta, u_k^\theta)\|_2 = \|(M_k^i)^\top \nabla_x c_i(x_k^\theta)\|_2 \le \|M_k^i\|_{\mathrm{op}} \cdot L_c'
\end{aligned}$$

Therefore, we conclude that

$$\|\nabla_{f_k} \mathcal{L}\|_2 \le \sum_{i=1}^{K} \|\nabla_{f_k} c_i\|_2 \le L_c' \cdot \sum_{i\,\ge\,k} \|M_k^i\|_{\mathrm{op}} \;\le\; L_c' \cdot \frac{C_2 \cdot C_1}{1-\rho_1}\;.$$

■

Lemma 27 *For $m \ge \Omega((K^{25} H^{12} d_x d_u \log^8 m)^{3/2} (2KW+H)^8)$, and $R = O\left(\frac{K^3(2KW+H)\log m}{\sqrt{m}}\right)$, with probability at least $1 - O(H + d_u)e^{-\Omega(\log^2 m)}$ over the randomness of initialization θ_1, the loss $\mathcal{L}(\theta) = \mathcal{L}(\bar{f}(\theta))$, for any $\theta \in B(R;\theta_1)$ and any $\bar{z} \in \mathbb{S}_{Kd_x+1}$, is ε_{nc}-nearly convex with $\varepsilon_{nc} = O(L_c R^{4/3} H^{9/2} K^6 W d_u \sqrt{d_x m} \log^{3/2} m)$.*

Proof Since $\mathcal{L}$ is convex in $\bar{f}$ by Lemma 23, we have that

$$\begin{aligned}
\mathcal{L}(\bar{f}(\theta')) - \mathcal{L}(\bar{f}(\theta)) &\ge \nabla_{\bar{f}} \mathcal{L}(\bar{f}(\theta))^\top (\bar{f}(\theta') - \bar{f}(\theta))\\
&= \sum_{k=1}^{K} \sum_{j=1}^{d_u} \frac{\partial \mathcal{L}}{\partial f_j(\theta;\bar{z}_k)} (f_j(\theta';\bar{z}_k) - f_j(\theta;\bar{z}_k))
\end{aligned}$$

Using the linearization trick as in (D.1), we can write

$$\mathcal{L}(\bar{f}(\theta')) - \mathcal{L}(\bar{f}(\theta)) \ge \sum_{k=1}^{K} \sum_{j=1}^{d_u} \frac{\partial \mathcal{L}}{\partial f_j(\theta[j];\bar{z}_k)} (\langle \nabla_{\theta[j]} f_j(\theta[j];\bar{z}_k), \theta'[j] - \theta[j] \rangle + \mathcal{R}(f_j, \theta[j], \theta'[j], \bar{z}_k)).$$

Pulling out the first term in the sum, we have

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{j=1}^{d_u}\frac{\partial\mathcal{L}}{\partial f_j(\theta[j];\bar{z}_k)}\langle\nabla_{\theta[j]}f_j(\theta[j];\bar{z}_k),\theta'[j]-\theta[j]\rangle\\
&=\sum_{j=1}^{d_u}\langle\sum_{k=1}^{K}\frac{\partial\mathcal{L}}{\partial f_j(\theta[j];\bar{z}_k)}\nabla_{\theta[j]}f_j(\theta[j];\bar{z}_k),\theta'[j]-\theta[j]\rangle\\
&=\sum_{i=1}^{d_u}\langle\nabla_{\theta[j]}\mathcal{L}(\theta),\theta'[j]-\theta[j]\rangle=\langle\nabla_\theta\mathcal{L}(\theta),\theta'-\theta\rangle.
\end{aligned}$$

We can use the proof of Lemma 19 to bound the other term as follows,

$$\begin{aligned}
&\left|\sum_{k=1}^{K}\sum_{j=1}^{d_u}\frac{\partial\mathcal{L}}{\partial f_j(\theta[j];\bar{z}_k)}\mathcal{R}(f_j,\theta[j],\theta'[j],\bar{z}_k)\right|\\
&\leq O(R^{1/3}H^{5/2}\sqrt{m\log m})\sum_{k=1}^{K}\sum_{j=1}^{d_u}\left|\frac{\partial\mathcal{L}}{\partial f_j(\theta[j];\bar{z}_k)}\right|\|\theta'[j]-\theta[j]\|_F\\
&\leq O(R^{4/3}H^{5/2}\sqrt{m\log m})\sum_{k=1}^{K}\sum_{j=1}^{d_u}\left|\frac{\partial\mathcal{L}}{\partial f_j(\theta[j];\bar{z}_k)}\right|\\
&\leq O(R^{4/3}H^{5/2}KL_c'\sqrt{d_u m\log m})
\end{aligned}$$

We obtain that by Assumption 4

$$\mathcal{L}(\bar{f}(\theta'))-\mathcal{L}(\bar{f}(\theta))\geq\langle\nabla_\theta\mathcal{L}(\bar{f}(\theta)),\theta'-\theta\rangle-O(L_cR^{4/3}H^{9/2}K^6Wd_u\sqrt{d_x m}\log^{3/2}m),$$

where $L_c'=O(K^5L_cH^2W\sqrt{d_xd_u}\log m)$ by Lemma 26 and Corollary 25. ■