PROVABLE REGRET BOUNDS FOR DEEP ONLINE LEARNING AND CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

The use of deep neural networks has been highly successful in reinforcement learning and control, although few theoretical guarantees for deep learning exist for these problems. There are two main challenges for deriving performance guarantees: a) control has state information and thus is inherently online and b) deep networks are non-convex predictors for which online learning cannot provide provable guarantees in general.

Building on the linearization technique for overparameterized neural networks, we derive provable regret bounds for efficient online learning with deep neural networks. Specifically, we show that over any sequence of convex loss functions, any low-regret algorithm can be adapted to optimize the parameters of a neural network such that it competes with the best net in hindsight. As an application of these results in the online setting, we obtain provable bounds for online episodic control with deep neural network controllers.

1 INTRODUCTION

In many machine learning problems, the environment cannot be represented by a distribution. One reason for the inadequacy of this modeling assumption is the nonstochastic nature of the environment. For example, in control for dynamical systems and reinforcement learning, the environment has a temporal state that is affected by feedback. Likewise in the setting of spam filtering, spam emails are generated adversarially to bypass email filters. Another example is the problem of portfolio selection, where the stock market behavior is governed by multiple players and is thus nonstochastic.

The accepted framework to study learning in nonstochastic environments is online learning in games. Since the environment can change arbitrarily, there is no fixed, a priori optimal decision. Instead, the notion of generalization is replaced by the game-theoretic concept of regret: the difference between the overall performance of an algorithm and that of the best fixed decision in hindsight. Efficient algorithms for online learning are based on Online Convex Optimization (OCO), which is restricted to convex predictors and losses. As a result, the framework cannot be readily applied when the learners are deep neural networks – the driving force behind many breakthroughs in modern machine learning. It is therefore desirable to extend OCO to bridge this gap.

In this paper, we bring recent ideas from deep learning theory to enable online learning with neural networks, and apply this method to control. We derive efficient algorithms, by reduction from *any OCO algorithm*, that attain provable regret bounds using deep neural networks. These bounds apply to the full online learning setting with any convex decision set and loss functions. Moreover, they are *agnostic*, which means that they show competitive performance to the best neural network in hindsight without assuming it achieves zero loss.

An interesting conclusion from this reduction is that provable bounds for training deep neural networks can be derived from any OCO method, beyond online (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 (Wu et al., 2019; Cai et al., 2019; Wu et al., 2019; 2021).

We apply this general online deep learning method to obtain provable regret guarantees for control of dynamical systems with deep neural networks. Provable regret bounds in this domain have thus

far been limited to linear dynamics and/or linear controllers. However, most dynamical systems in the physical world are nonlinear.

In order to go beyond linear control, we consider 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 a convex reparametrization of policies. Therefore, our extension of OCO to deep neural networks naturally leads to regret bounds for deep neural network controllers in this setting.

Our contributions can be summarized as follows:

- We give a general reduction from any online convex optimization algorithm to online deep learning. The regret bounds obtain depend on the original regret for online convex optimization, the width of the network, and the diameter of neural network parameters over which we optimize. The precise statements are given in Theorems 3.1, 3.2.
- These regret bounds imply generalization in the statistical setting, and go beyond SGD: any online convex optimization algorithm can be shown to generalize over deep neural networks according to its regret bound in the OCO framework. This includes commonly used algorithms such as Adagrad/Adam, as well as more recent regularization functions for mirror descent (Ghai et al., 2020).
- We apply this reduction to the framework of online nonstochastic control, and obtain provable regret bounds for deep controllers in the episodic setting. These can be used to derive iterative linear control algorithms, as well as regret for online single trajectory control.

1.1 Related work

Online learning and online convex optimization. The framework of learning in games has been extensively studied as a model for learning in adversarial and nonstochastic environments (Cesa-Bianchi & 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.

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 or linearization approach has emerged as the most general and pervasive. This technique shows that neural networks behave similar to their linearization and prove that gradient descent converges to a global minimizer of the training loss (Soltanolkotabi et al., 2018; Du et al., 2018a;b; Allen-Zhu et al., 2019; Zou et al., 2018; Jacot et al., 2018; Bai & Lee, 2019). Techniques related to this have been expanded to provide generalization error bounds in the i.i.d. statistical setting (Arora et al., 2019; Wei et al., 2019), and generalization bounds for SGD (Cao & Gu, 2019; Ji & Telgarsky, 2020).

Online-to-batch linearization. The linearization technique has been combined with online learning and online-to-batch conversion to yield generalization bounds for SGD Cao & Gu (2019). In the adversarial training setting, the online gradient proof technique was used in Gao et al. (2019); Zhang et al. (2020) to handle non i.i.d. functions. In relationship to these works, our reduction takes in a general OCO algorithm, rather than only OGD/SGD. In addition, we generalize this reduction to the full OCO model, including high dimensional output predictors and general convex costs to enable the application to control.

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 & Szepesvári (2011) initiated the study of online control under the regret benchmark for Linear Time-invariant (LTI) dynamical systems. Our work instead adopts the *nonstochastic control setting* (Agarwal et al., 2019), that allows for adversarially chosen (i.e. non-Gaussian) noise and costs that may vary with time. The nonstochastic control model was recently extended to consider nonlinear and time-varying dynamics in (Gradu et al., 2020). See Hazan & Singh (2021) for a comprehensive survey of results and advances in online and nonstochastic control. Similar to our setting, online

episodic control is also studied in Kakade et al. (2020), but the regret definition differs from ours, and the system is not linear.

Nonlinear systems and deep neural network based controls. Nonlinear control is computationally intractable in general (Blondel & Tsitsiklis, 2000). One approach to deal with the computational difficulty 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 & Li, 2005) fall into this category. Neural networks were also used to directly control the dynamical system since the 90s, for example in Lewis et al. (1997). More recently, deep neural networks were used in applications of a variety of control problems, including Lillicrap et al. (2016) and Levine et al. (2016). A critical study of neural-network based controllers vs. linear controllers appears in Rajeswaran et al. (2017)

2 PRELIMINARIES

Notation. Let $\langle \cdot, \cdot \rangle$ denote the element-wise inner product between two vectors, matrices, or tensors of the same dimension: $\langle x, y \rangle = \text{vec}(x)^\top \text{vec}(y)$. Let $\mathbb{S}_p = \{x \in \mathbb{R}^p : \|x\|_2 = 1\}$ denote the unit sphere, and for a convex set $\mathcal{K}$, let $\prod_{\mathcal{K}}$ denote projection onto $\mathcal{K}$.

2.1 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 $l_t(\theta_t)$ according to a convex loss function $l_t : \mathcal{K} \mapsto \mathbb{R}$. The goal of the learner is to minimize her regret, defined as

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

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

2.2 DEEP NEURAL NETWORKS AND THE NEURAL TANGENT KERNEL (NTK)

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:

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

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 $\mathcal{N}(0, 1)$.

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: with data points $\{(x_t, y_t)\}_t$ the losses $\ell_t(\cdot) = \ell(\cdot, y_t)$ reduce the setting to supervised learning. We make the following standard assumptions on the inputs and the loss functions:

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 $f(\theta; x)$.*

Two-layer Neural Networks. For analysis simplicity we also consider the special case of two-layer network architecture. For inputs $x \in \mathbb{R}^p$, define the coordinate-wise 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 $\theta[i] \in \mathbb{R}^{m \times p}$, $f(\theta; x) = (f_1(\theta[1]; x), \ldots, f_d(\theta[d]; x))^\top \in \mathbb{R}^d$, where

$$f_i(\theta[i]; x) = \frac{1}{b}\Big(\sum_{r=1}^{m/2} a_{i,r}\sigma(\theta[i,r]^\top x) + \sum_{r=1}^{m/2} \bar{a}_{i,r}\sigma(\bar{\theta}[i,r]^\top x)\Big). \tag{2.2}$$

In the above expression, $\theta[i,r]^\top$ denotes the r-th row of $\theta[i]$, and $\bar{\theta}[i,r]$ denotes the $r + \frac{m}{2}$-th row of $\theta[i]$, such that $\theta[i] = (\theta[i,1], \ldots, \theta[i, \frac{m}{2}], \bar{\theta}[i,1], \ldots, \bar{\theta}[i, \frac{m}{2}])$. We include a scaling factor $b \in \mathbb{R}$ to demonstrate how its value affects the convergence and generalization properties of the network, and in Section 3 we study the tradeoff between the two properties and choose an optimal b. 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 $i \in [d]$, $\theta_1[i,r] \sim N(0, I_d)$ for $r = 1, \ldots, \frac{m}{2}$, 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 $x \in \mathbb{S}_p$. We make the following assumption on the general activation function:

Assumption 3. *The activation function σ is $C-$smooth and $C-$Lipschitz: $|\sigma'(z) - \sigma'(z')| \le C|z - z'|$, $|\sigma'(z)| \le C$.*

The above architectures, initialization schemes, and assumptions are consistent with Gao et al. (2019).

Neural Tangent Kernel. The Neural Tangent Kernel (NTK) was first introduced in Jacot et al. (2018), who showed that in the infinite width limit, a randomly initialized deep neural network trained with gradient descent is equivalent to a kernel method with the NTK kernel:

$$K(x, x') = \mathbb{E}_{\theta \sim \mathcal{D}}\big[\langle \frac{\partial f(\theta; x)}{\partial \theta}, \frac{\partial f(\theta; x')}{\partial \theta} \rangle\big],$$

where $\mathcal{D}$ denotes the initialization distribution of θ. Since a deep neural network contains a two-layer neural network, for quantitative results on the relationship between network overparameterization and its expressivity, we focus on the two-layer network (2.2) for simplicity. We present the explicit form of the NTK for our two-layer neural network in the following definition.

Definition 2.1. *The NTK for the scalar two-layer neural network with activation σ and initialization distribution $w \sim \mathcal{N}(0, I_p)$ is defined as $K_\sigma(x, y) = \mathbb{E}_{w \sim \mathcal{N}(0, I_p)} \langle x\sigma'(w^\top x), y\sigma'(w^\top y) \rangle$.*

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. Depending on the choice of σ, $\mathcal{H}(K_\sigma)$ can contain meaningful classes of functions; for example, $\mathcal{H}(K_{\sigma_{\text{relu}}})$ contains all even functions with bounded derivatives (Bietti & Mairal, 2019). Since our goal is to obtain nonasymptotic approximation guarantee, we focus on RKHS functions of bounded norm. Following Gao et al. (2019), we define the closely related Random Feature (RF) space of functions, its norm and restrict to functions of bounded RF-norm.

Definition 2.2 ((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

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

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

By Lemma C.1 in Gao et al. (2019), $\mathcal{F}_{RF}(\infty)$ 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.

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

$$\forall k \in [1, K], \quad x_{k+1} = A_k x_k + B_k u_k + w_k, \tag{2.3}$$

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 cost functions $c_k : \mathbb{R}^{d_x} \times \mathbb{R}^{d_u} \to \mathbb{R}$ of state and action, i.e. for a policy π

$$J(\pi; x_1, c_{1:K}) = \sum_{k=1}^{K} c_k(x_k^\pi, u_k^\pi).$$

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 for different episodes. The goal of the learner is to minimize *episodic* regret by adapting its output policies π_t for $t \in [1, T]$,

$$\text{Regret}_T = \sum_{t=1}^{T} J_t(\pi_t; x_1^t, c_{1:K}^t) - \min_{\pi\in\Pi} \sum_{t=1}^{T} J_t(\pi; x_1^t, c_{1:K}^t), \tag{2.4}$$

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

We make the following basic assumptions about the dynamical system *in each episode* that are common in the nonstochastic control literature : the disturbances are bounded, the system is sequentially stable[1], and the cost functions are well-behaved for efficient optimization.

Assumption 4. *All disturbances have a uniform bound on their norms:* $\max_{k\in[K]} \|w_k\|_2 \le W$.

Assumption 5. *There exist* $C_1, C_2 \ge 1$ *and* $0 < \rho_1 < 1$ *such that the system matrices satisfy:*

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

Assumption 6. *Each cost function* $c_k : \mathbb{R}^{d_x} \times \mathbb{R}^{d_u} \to \mathbb{R}$ *is jointly convex and satisfies* $\|\nabla c_k(x, y)\| \le L_c \max\{1, \|x\| + \|u\|\}$ *for some* $L_c > 0$.

Policy Class. The performance of the learner given by (2.4) 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 $u_k = f(w_{1:k-1})$. These policies are parameterized w.r.t. *policy-independent* inputs, in this case the sequence $w_{1:K}$. This is in contrast to the commonly used state feedback policy $u_k = f(x_k)$.

In particular, the DAC policy class (Agarwal et al., 2019) (as well as DRC (Simchowitz et al., 2020)) that outputs controls linear in past finite disturbances has a convex parameterization (Agarwal et al., 2019). Recent works have devised efficient online methods with provable guarantees for these policy classes both for LTI and LTV systems in the single trajectory setting. We expand the comparator class by considering policies with controls that are *nonlinear* in the past disturbances, represented by a neural network.

Definition 2.3. *(Disturbance Neural Feedback Control) A disturbance neural feedback policy* π^θ_{dnn} *chooses control* u_k *output by a neural network over the past disturbances,*

$$u_k = f_\theta(w_{k-1}, w_{k-2}, \ldots, w_1),$$

where $f_\theta(\cdot)$ *is a neural network with parameters* θ.

[1] Extension to the stabilizable case can be found in appendix.

The reasoning behind this policy class expansion is twofold. First, for general LTI systems, the best in hindsight DAC policy is not close to the optimal open-loop control sequence given adversarial disturbances w_k and general convex costs c_k. Furthermore, our episodic setting can be used for trajectory-based first-order policy optimization over *nonlinear* dynamics (Ahn et al., 2007). Hence, competing against the rich policy class of neural network controllers is highly desirable. For a given neural network architecture let $f_\theta(\cdot) = f(\theta; \cdot)$, and let Θ be the set of permissible parameters θ. The class of deep controller policies is then given by $\Pi_{\text{dnn}}(f; \Theta) = \{\pi^\theta_{\text{dnn}} : \theta \in \Theta\}$.

3 Online Learning with Neural Networks

In this section, we describe the general framework of online learning with neural networks and derive accompanying regret guarantees. Our framework can use any OCO algorithm as a black-box, and we focus on Online Gradient Descent (OGD) in our main algorithm below, since it is widely used in practice. Observe that the update is equivalent to OGD on the original losses $\{\ell_t\}$.

Algorithm 1 OGD for Neural Networks

1: Input: step size $\eta_t > 0$, initial parameters θ_1, decision set $B(R)$.
2: **for** $t = 1 \dots T$ **do**
3: Play θ_t, receive loss $\ell_t(\theta) = \ell_t(f(\theta; x_t))$. Construct $h_t(\theta) = \ell_t(\theta_t) + \nabla\ell_t(\theta_t)^\top(\theta - \theta_t)$.
4: Update $\theta_{t+1} = \prod_{B(R)}(\theta_t - \eta_t \nabla h_t(\theta_t)) = \prod_{B(R)}(\theta_t - \eta_t \nabla \ell_t(\theta_t))$.
5: **end for**

In the following theorem, we give an end-to-end bound on the performance of our algorithm compared to the best-in-hindsight function in $\mathcal{F}^d_{RF}(D)$ for two-layer neural networks. The regret bound consists of two parts: the regret for learning the optimal neural network parameters in a ball around initialization (Section 3.2), and the approximation error of neural networks to the target function in $\mathcal{F}^d_{RF}(D)$ (Section 3.3).

Theorem 3.1. *Let f be a two-layer neural network as in* (2.2) *with scaling factor $b = \sqrt{m}$ and decision set $B(R) = \{\theta \in \mathbb{R}^{d\times m\times p} : \|\theta - \theta_1\|_F \le R\}$, and suppose Assumptions 1, 2, 3 are satisfied. For any $\delta > 0, D > 1$, let $R = D\sqrt{d}$, then with probability at least $1 - \delta$ over the random initialization, Algorithm 1 with $\eta_t = \frac{2Rb}{CL\sqrt{m}} \cdot t^{-1/2}$ satisfies*

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

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

Note that the radius R of the permissible set for the parameters $B(R)$ is constant with respect to the network width m when the number of parameters initialized as $\mathcal{N}(0, I_d)$ is linear in m. This indicates very little movement allowed in the parameter space, consistent with the recent insights in deep learning theory. The above bound is more conveniently stated in terms of average regret.

Corollary 3.1. *Under the conditions of Theorem 3.1, the average regret is bounded by any $\varepsilon > 0$*

$$\frac{1}{T}\textit{Regret}_T = \frac{1}{T}\left[\sum_{t=1}^{T} \ell_t(f(\theta_t; x_t)) - \min_{g\in\mathcal{F}^d_{RF}(D)} \sum_{t=1}^{T} \ell_t(g(x_t))\right] \le \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{T}}\right) \le \varepsilon$$

for large values of network width $m = \Omega(\varepsilon^{-2})$ and large number of iterations $T = \Omega(\varepsilon^{-2})$.

A similar analysis applies even beyond the simple two-layer networks. In particular, we also derive regret bounds for learning with deep neural networks.

Theorem 3.2. *Let f be a deep neural network with ReLU activation defined as in* (2.1)*, and suppose Assumptions 1 and 2 are satisfied. Let $B(R) = \{\theta : \|\theta[i] - \theta_1[i]\|_F \le R \ \forall\ i \in [d]\}$. Take $R = \tilde{O}(H^{-3/2}m^{-3/2})$, then for $m \ge \max\{d, H^3\}$, with probability at least $1 - O(H)e^{-\Omega(p\log m)}$*

over the random initialization θ_1, Algorithm 1 with $\eta_t = \frac{2R}{LH\sqrt{m}} t^{-1/2}$ has regret bound

$$\sum_{t=1}^{T} \ell_t(f(\theta_t; x_t)) \leq \min_{\theta \in \bar{B}(R)} \sum_{t=1}^{T} \ell_t(f(\theta; x_t)) + \tilde{O}\left(\frac{L\sqrt{dT}}{\sqrt{H}m} + \frac{L\sqrt{HdT}}{m\sqrt{m}}\right)$$

where $\tilde{O}(\cdot)$ hides terms polylogarithmic in m.

Similar to the case of two-layer neural networks, the radius R of the decision set is small, as it scales inversely with m. Moreover, if T and m are large, we can achieve small average regret. In particular, with $m = \Omega(\varepsilon^{-2})$ and $T = \Omega(-\varepsilon^2)$, the average regret is bounded by ε.

3.1 Online Nearly Convex Optimization

We first prove regret bounds for nearly convex functions, a slight extension to the OCO framework. As we show in the next sections, these regret bounds naturally carry over to the setting of online learning over neural networks.

Definition 3.1. *A function $\ell : \mathbb{R}^n \to \mathbb{R}$ is ε-nearly convex over the convex, compact set $\mathcal{K} \subset \mathbb{R}^n$ if and only if*

$$\forall x, y \in \mathcal{K},\ \ell(x) \geq \ell(y) + \nabla \ell(y)^\top (x - y) - \varepsilon\ . \tag{3.1}$$

The analysis of any algorithm for OCO, including the most fundamental method OGD, extends to this case in a straightforward manner. Let $\mathcal{A}$ be any algorithm for OCO that accepts a sequence of convex losses $\{h_t\}$ by an *adaptive* adversary, and returns a decision sequence $\{\theta_t\}_{t\in[T]} \subseteq \mathcal{K}$ with the following regret guarantee,

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

Then the given OCO algorithm $\mathcal{A}$ can be applied to online nearly-convex optimization as per algorithm 2 to obtain a regret bound.

Algorithm 2 Online Nearly-Convex Optimization

1: Input: OCO algorithm $\mathcal{A}$ for convex decision set $\mathcal{K}$.
2: **for** $t = 1 \dots T$ **do**
3: Play θ_t, observe ℓ_t. Construct $h_t(\theta) = \ell_t(\theta_t) + \nabla \ell_t(\theta_t)^\top (\theta - \theta_t)$.
4: Update $\theta_{t+1} = \mathcal{A}(h_1, ..., h_t) \in \mathcal{K}$.
5: **end for**

Lemma 3.1. *Suppose $\ell_1, \dots, \ell_T$ are ε-nearly-convex, then Algorithm 2 attains the following regret:*

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

Much of the literature in the theory of deep learning has focused on the analysis of stochastic gradient descent (SGD) and gradient descent (GD). However, this general reduction allows for results over any OCO algorithm with sublinear regret, such as mirror descent and adaptive gradient methods (AdaGrad and further enhancements) eliminating the need to devise isolate analyses for separate algorithms. For simplicity, we consider OGD for the rest of the paper and state the corresponding regret bound below.

Corollary 3.2. *Suppose $\{\ell_t\}_{t=1}^T$ are ε-nearly convex and let $\mathcal{A}$ be OGD, then Algorithm 2 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 $\mathcal{K}$.

3.2 Online Gradient Descent for Two-layer Neural Networks

We proceed to show in Lemma 3.3 that two-layer neural networks as defined by (2.2) satisfy the nearly convex property. Consequently, we can obtain regret guarantees in Lemma 3.2 for learning the class of two-layer neural networks near initialization in an online fashion using OGD. We define the convex decision set as $B(R) = \{\theta : \|\theta - \theta_1\|_F \leq R\}$ where θ_1 denotes the value of θ at initialization. Throughout this section, denote $\ell_t(\theta) = \ell_t(f(\theta; x_t))$.

Lemma 3.2. *Under Assumptions 1, 2, 3, Algorithm 1 with $\eta_t = \frac{2Rb}{CL\sqrt{m}} \cdot t^{-1/2}$ attains regret bound*

$$\sum_{t=1}^{T} \ell_t(\theta_t) \leq \min_{\theta \in B(R)} \sum_{t=1}^{T} \ell_t(\theta) + \frac{3CLR\sqrt{mT}}{b} + \frac{2CLR^2}{b}T. \tag{3.2}$$

The following lemma quantifies the margin of near-convexity for two-layer neural networks.

Lemma 3.3. *Under Assumptions 1, 2, 3, the functions $\ell_t(\theta)$ satisfy the near-convexity property* (3.1) *with $\varepsilon_{nc} = \frac{2CLR^2}{b}$ for $\theta \in B(R)$.*

3.3 Expressivity of Two-layer Neural Networks

Lemma 3.2 shows that we can efficiently (online) learn over the class of two-layer neural networks near initialization, but how well does the best neural network in that class perform? Recall the function space $\mathcal{F}_{RF}^d(D)$ from Definition 2.2. In the results below, we show that the class of two-layer neural networks considered $\{f(\theta; \cdot) : \theta \in B(R)\}$ can approximate any function in $\mathcal{F}_{RF}^d(D)$, and derive nonasymptotic rates on the approximation error in terms of the network width m. The analysis follows the outline from Gao et al. (2019).

Lemma 3.4. *For any $\delta, D > 0$, let $g : \mathbb{R}^p \to \mathbb{R}^d \in \mathcal{F}_{RF}^d(D)$, and let $R = \frac{bD\sqrt{d}}{\sqrt{m}}$, then with probability at least $1 - \delta$ over the random initialization θ_1, there exists $\theta^* \in B(R)$ such that for all $x \in \mathbb{S}_p$,*

$$\ell_t(f(\theta^*; x)) \leq \ell_t(g(x)) + \frac{Lb\sqrt{d}CD^2}{2m} + \frac{L\sqrt{d}CD}{\sqrt{m}}(2\sqrt{2p} + 2\sqrt{\log d/\delta}).$$

3.4 Extending to Deep Neural Networks

Beyond two-layer neural networks, the nearly-convex property holds for deep neural networks as well. We show this result for networks with ReLU activation, where the gradient itself may be large but changes very slowly. This phenomenon is studied in Allen-Zhu et al. (2019) on finitely many points, and Gao et al. (2019) further extends the result to hold for inputs over the unit sphere. In the lemma below, we adapt the result in Gao et al. (2019) to hold for our deep neural network architecture as in (2.1). Throughout this section, we use $\bar{B}(R) = \{\theta : \|\theta[i] - \theta_1[i]\|_F \leq R \, \forall \, i \in [d]\}$.

Lemma 3.5. *For $m = \Omega(\frac{p\log(1/R) + \log d}{R^{2/3}H})$, and $R = O(\frac{1}{H^6 \log^3 m})$, with probability at least $1 - O(H)e^{-\Omega(mR^{2/3}H)}$ over the random initialization θ_1, for any $\theta, \theta' \in \bar{B}(R)$ and any $x \in \mathbb{S}$, under Assumption 2,*

$$\ell_t(f(\theta'; x)) - \ell_t(f(\theta; x)) \geq \langle \nabla_\theta \ell_t(f(\theta; x)), \theta' - \theta \rangle - O\big(R^{4/3}H^{5/2}\sqrt{m \log m}\big)L\sqrt{d}, \tag{3.3}$$

$$\|\nabla_{\theta[i]} f_i(\theta[i]; x)\|_F \leq O(H\sqrt{m}) \, \forall i \in [d]. \tag{3.4}$$

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 (Section 3) where f is defined as in (2.1) with a permissible parameter set Θ. For simplicity, we temporarily drop the index $t \in [T]$ of a single episode and denote the 0-padded network input $z_k = \text{vec}([w_{k-1}, \ldots, w_1, 0, \ldots, 0]) \in \mathbb{R}^{K \cdot d_x}$. To

satisfy Assumption 1, normalize the input $\bar{z}_k = \frac{z_k}{\|z_k\|_2} \in \mathbb{S}_{K \cdot d_x}$. The controls of a policy $\pi_{\text{dnn}}(\theta)$ parameterized by $\theta \in \Theta$ are given by $u_k = f(\theta; \bar{z}_k)$ for all $k \in [K]$. The episode loss equals

$$\mathcal{L}(\theta) = J(\pi_{\text{dnn}}(\theta); x_1, c_k) = \sum_{k=1}^{K} c_k(x_k^\theta, f(\theta; \bar{z}_k)) \, .$$

Note that the episode loss $\mathcal{L}$ depends on the parameter θ through all the K controls $f(\theta, \bar{z}_k), k \in [K]$. Denote $\bar{f}(\theta) = [f(\theta, \bar{z}_1), \ldots, f(\theta, \bar{z}_K)]^\top \in \mathbb{R}^{K \times d_u}$ and let $\mathcal{L}(\theta) = \mathcal{L}(\bar{f}(\theta))$ by overload 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 B.1) and Lipschitz (Lemma B.3) conditions. This means that for each episode $t \in [T]$, the episode loss $\mathcal{L}_t(\theta) = J_t(\pi_{\text{dnn}^\theta}; x_1^t, c_{1:K}^t)$ satsifies Assumption 2 over the argument $\bar{f}(\theta)$. The rest of the derivation is analogous to that in Section 3 and yields the theorem below (see Appendix B for the proof). The algorithm itself for online episodic control is simply OGD over the losses $\mathcal{L}_t(\theta)$ given in detail in Algorithm 4.

Algorithm 3 Deep Neural Episodic Control with OGD

1: Input: $\eta_t > 0$, initial parameter θ_1, permissible set Θ.
2: **for** $t = 1 \ldots T$ **do**
3: **for** $k = 1 \ldots K$ **do**
4: Observe x_k^t and play $u_k^t = f(\theta_t, \bar{z}_k^t)$.
5: **end for**
6: Construct loss function

$$\mathcal{L}_t(\theta) = \sum_{k=1}^{K} c_k^t(x_k^{t,\theta}, f(\theta, \bar{z}_k^t).$$

7: Perform gradient update $\theta_{t+1} = \Pi_\Theta(\theta_t - \eta_t \nabla_\theta \mathcal{L}_t(\theta_t))$.
8: **end for**

Let f denote the neural network as in (2.1) and π_{dnn}^θ the policy class with $u_k^\theta = f(\theta; \bar{z}_k)$. Define $\Pi_{\text{dnn}}(f; \Theta) = \{\pi_{\text{dnn}}^\theta : \theta \in \Theta\}$ with $\Theta = B(R)$ as in Theorem 3.2

Theorem 4.1. *Given the setting in Section 2.3, suppose the Assumptions 4, 5, 6 hold. Let R, m, H satisfy the conditions in Theorem 3.2. Then, Algorithm 4 with $\eta_t = O(\frac{2R}{L_c H \sqrt{m}} t^{-1/2})$, attains episodic regret bound given by*

$$\textit{Regret}_T = \sum_{t=1}^{T} J_t(\pi_t; x_1^t, c_{1:K}^t) - \min_{\pi \in \Pi} \sum_{t=1}^{T} J_t(\pi; x_1^t, c_{1:K}^t) \leq \tilde{O}\left(\frac{K L_c \sqrt{d_u T}}{\sqrt{H} m} + \frac{K L_c \sqrt{H d_u} T}{m \sqrt{m}} \right),$$

where $\tilde{O}(\cdot)$ hides terms polylogarithmic in m.

This theorem statement, analogous to Section 3, can be interpreted as arbitrarily small average regret ε when the network width m and number of episodes T are both large, i.e. $\Omega(\varepsilon^{-2})$.

5 Conclusion and Future Work

In this work, we study online learning with the class of deep neural networks, and apply this general framework to online episodic control over LTV systems with deep neural network controllers. This leads to the first provable performance guarantees for neural network based controllers.

Interestingly, our derivation of provable regret bounds for online learning with deep neural networks can be applied to *any* OCO algorithm, creating a unifying framework for studying optimization methods in deep learning. For example, generalization bounds can be obtained by an online-to-batch reduction.

In terms of control, our results can open a new line of work showing guarantees in different control settings with the policy class of neural network controllers. Particularly, one can derive provable bounds for single-trajectory online control or nonlinear control with regret competing against these policies.

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*, pp. 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*, pp. 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.

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.

Alberto Bietti and Julien Mairal. *On the Inductive Bias of Neural Tangent Kernels*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Vincent D Blondel and John N Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

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.

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.

Ruiqi Gao, Tianle Cai, Haochuan Li, Liwei Wang, Cho-Jui Hsieh, and Jason D. Lee. Convergence of adversarial training in overparametrized neural networks, 2019.

Udaya Ghai, Elad Hazan, and Yoram Singer. Exponentiated gradient meets gradient descent. In *Algorithmic Learning Theory*, pp. 386–407. PMLR, 2020.

Paula Gradu, Elad Hazan, and Edgar Minasyan. Adaptive regret for control of time-varying dynamics. *arXiv preprint arXiv:2007.04393*, 2020.

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 (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15312–15325. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/aee5620fa0432e528275b8668581d9a8-Paper.pdf.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

F.L. Lewis, S. Jagannathan, and A. Yeşildirek. Chapter 7 - neural network control of robot arms and nonlinear systems. In Omid Omidvar and David L. Elliott (eds.), *Neural Systems for Control*, pp. 161–211. Academic Press, San Diego, 1997. ISBN 978-0-12-526430-3. doi: https://doi.org/10.1016/B978-012526430-3/50008-8.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *ICLR (Poster)*, 2016.

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

Aravind Rajeswaran, Kendall Lowrey, Emanuel Todorov, and Sham Kakade. Towards generalization and simplicity in continuous control. *arXiv preprint arXiv:1703.02660*, 2017.

Max Simchowitz, Karan Singh, and Elad Hazan. Improper learning for non-stochastic control, 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*, pp. 4906–4913, 2012.

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.*, pp. 300–306. IEEE, 2005.

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

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.

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.

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

A PROOFS FOR SECTION 3

Proof of Theorem 3.1. Let $g \in \mathcal{F}_{RF}^d(D)$. By Lemma 3.4, with probability at least $1-\delta$ over the random initialization θ_1, there exists $\theta^* \in B(R)$ such that for all $x \in \mathbb{S}$,

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

By the regret guarantee in Lemma 3.2, Algorithm 1 has regret

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

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

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

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

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

Proof of Theorem 3.2. By Lemma 3.5, with our choice of m and R, 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 $\|\nabla_{\theta[i]} f_i(\theta[i];x)\|_F \le O(H\sqrt{m})$ for all $i \in [d], x \in \mathbb{S}, \theta \in \bar{B}(R)$. Since the decision set is $\bar{B}(R)$, its radius in Forbenius norm is at most $R\sqrt{d}$. We can bound the gradient norm as follows, for all $x \in \mathbb{S}$,

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

By Corollary 3.2, the regret is bounded by

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

Note that the conditions of Lemma 3.5 are satisfied with the choice of $R = \tilde{O}(H^{-3/2}m^{-3/2})$.

$$\begin{aligned}R^{2/3}H = \tilde{O}(m^{-1}) \text{implies} m &= \Omega(\frac{p\log(1/R) + \log d}{R^{2/3}H}), \\ m \ge H^3 \text{implies} R \;\le\; \tilde{O}(H^{-3/2} * H^{-9/2}) &= O(H^{-6}\log^{-3} m) .\end{aligned}$$

Finally, $R^{2/3}Hm = \Omega(p\log m)$ and $m \ge H^3$ implies the probability of the bound is high. □

Proof of Lemma 3.1. Observe that by the 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 $\theta^* \in \mathcal{K}$,

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

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

Proof of Lemma 3.2. The theorem statement is shown by using Corollary 3.2 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 3.3 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,

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

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 (A.3). This means that $G = \frac{CL\sqrt{m}}{b}$ and we can use the Corollary 3.2 to conclude the final statement in (3.2). □

Proof of Lemma 3.3. We extend the original proof in Gao et al. (2019). Let $\text{diag}(a_i)$ be a diagonal matrix with
$(a_{1,i}, \ldots, a_{m/2,i}, -a_{1,i}, \ldots, -a_{m/2,i})$ on the diagonal. Note that the gradient of the network at the i-th coordinate is

$$\nabla_{\theta[i]} f_i(\theta[i]; x) = \frac{1}{b} \text{diag}(a_i) \sigma'(\theta[i] x) x^\top. \tag{A.3}$$

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

$$\begin{aligned} \|\nabla_{\theta[i]} f_i(\theta[i]; x) - \nabla_{\theta[i]} f_i(\theta'[i]; x)\|_F &\leq \frac{1}{b} \|\text{diag}(a_i)\|_2 \|\sigma'(\theta[i] x) - \sigma'(\theta'[i] x)\|_2 \|x\|_2 && \text{(A.4)} \\ &\leq \frac{C}{b} \|\theta[i] - \theta'[i]\|_F. && (|a_{r,i}| = 1, \|x\|_2 = 1) \end{aligned}$$

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

$$f_i(\theta'[i]; x_t) - f_i(\theta[i]; x_t) = \langle \nabla_{\theta[i]} f_i(\theta[i]; x_t), \theta'[i] - \theta[i] \rangle + \mathcal{R}(f_i, \theta[i], \theta'[i]) \tag{A.5}$$

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

Note that since the gradient of f_i is Lipschitz given by (A.4), 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 of Lemma 3.4. Let $g = (g_1, \ldots, g_d) \in \mathcal{F}_{RF}^d(D)$. By Lemma A.1, if $R' = \frac{bD}{\sqrt{m}}$, with probability at least $1 - \delta/d$, for each i there exists $\theta^*[i]$ such that $\|\theta^*[i] - \theta_1[i]\|_F \leq R'$, and

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

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

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

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

Lemma A.1. *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 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 $x \in \mathbb{S}_p$ and any $i \in [d]$,*

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

Proof. Since the neural network architectures are the same for all $i \in [d]$, we fix an arbitrary i and drop the index i 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 (A.5) 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

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

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

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

$$\begin{aligned}
|f_i(\theta^*; x) - g_1(x)| = \frac{1}{b}\Big|&\sum_{r=1}^{m/2} a_r \int_0^1 x^\top(\theta^*[r] - \theta_1[r])(\sigma'((t\theta^*[r] + (1-t)\theta_1[r])^\top x) - \sigma'((\theta_1[r])^\top x))dt \\
&- \sum_{r=1}^{m/2} a_r \int_0^1 x^\top(\bar{\theta}^*[r] - \bar{\theta}_1[r])(\sigma'((t\bar{\theta}^*[r] + (1-t)\bar{\theta}_1[r])^\top x) - \sigma'((\bar{\theta}_1[r])^\top x))dt\Big| \\
&\frac{mC}{b}\|x\|_2^2\frac{b^2}{4}\max_r\|c_r\|_2^2 \cdot \frac{1}{2} \\
&\le \frac{mC}{b}\frac{b^2}{4}\frac{4\|g\|_{RF}^2}{2m^2} = \frac{bC\|g\|_{RF}^2}{2m}.
\end{aligned}$$

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:

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

□

Proof of Lemma 3.5. 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 $\mathbb{S}$ is defined as a collection of points $\{x_r\} \in \mathbb{S}$ such that for all $x \in \mathbb{S}$, 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 $x' \in \mathbb{S}$ 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 (A.5), 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,

$$f_i(\theta'[i]; x') - f_i(\theta[i]; x') = \langle \nabla_{\theta[i]} f_i(\theta[i]; x'), \theta'[i] - \theta[i] \rangle + \mathcal{R}(f_i, \theta[i], \theta'[i], x') \tag{A.6}$$

$$\mathcal{R}(f_i, \theta[i], \theta'[i], x') = \int_0^1 \big\langle \nabla_{\theta[i]} f_i(s\theta'[i] + (1-s)\theta[i]; x') - \nabla_{\theta[i]} f_i(\theta[i]; x'), \theta'[i] - \theta[i] \big\rangle ds \tag{A.7}$$

Since the gradient changes slowly, we can bound the residual term as follows

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

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 $\|x' - x_r\|_2 \leq \delta$,

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

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

B PROOFS FOR SECTION 4

Proof of Theorem 4.1. This is identical to that of Theorem 3.2 given the result in Lemma B.4 and the fact that the norm bound given by Lemma 3.5 is invariant. The conclusion is the same as in Theorem 3.2 with $L = O(L_c), p = K \cdot d_x, d = d_u$. □

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 $\|x_1\| \le W$.

Sequential stabilizability. Furthermore, note that Assumption 5 can be relaxed to assuming there exists a sequence of linear operators $F_{1:K}$ such that for $C_1 \ge 1$ and $\rho_1 \in (0, 1)$

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

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 $(A_k + F_k B_k, B_k)_{1:K}$.

Lemma B.1. *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 $f(\theta, \bar{z}_i), i = 1, \dots, K$. 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 B.2. *For any $\theta \in \Theta$, the states and actions over an episode are bounded, $\max_k \|u_k^\theta\| \le D_u$ and $\max_k \|x_k^\theta\| \le D_x$ for $D_u = \sqrt{d_u} N$, $D_x = \frac{C_1}{1-\rho_1} \cdot (W + D_u C_2)$.*

Proof. First, note that $u_k^\theta = f(\theta; \bar{z}_k)$ and $\bar{z}_k \in \mathbb{S}_{K \cdot d_x}$. Given the output magnitude bound $|u_k^\theta[i]| \le N$ for the network for all $i \in [d_u]$, which means that $\|u_k^\theta\| \le \sqrt{d_u} N = D_u$. By definition of x_k^{nat}, we have that

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

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

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

□

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

Lemma B.3. *The function $\mathcal{L}(\bar{f}(\theta))$ is L-Lipschitz w.r.t. each $f(\theta; \bar{z}_k)$ for $k \in [K]$ with $L = L'_c \cdot \frac{C_2 \cdot C_1}{1-\rho_1}$.*

Proof. We use Corollary B.1 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

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

Therefore, we conclude that

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

□

Lemma B.4. *Suppose the conditions on R, m, H from Lemma 3.5 and Assumptions 1, 3, 6 hold. Let $\mathcal{L}(\theta)$ denote $\mathcal{L}(\bar{f}(\theta))$. Then, with probability at least $1 - O(H)e^{-\Omega(mR^{2/3}H)}$ over the random initialization θ_1, for any $\theta, \theta' \in \bar{B}(R)$ and any $\bar{z} \in \mathbb{S}$,*

$$\mathcal{L}(\theta') \geq \mathcal{L}(\theta) + \nabla\mathcal{L}(\theta)^\top(\theta' - \theta) - O\big(L_c K R^{4/3} H^{5/2}\sqrt{m\log m}\sqrt{d}\big)$$

Proof. Since $\mathcal{L}$ is convex in $\bar{f}$ by Lemma B.1, we have that

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

Using the linearization trick as in (A.6) and the L-Lipschitz property of $\mathcal{L}(\bar{f}(\theta))$ w.r.t each $f(\theta, \bar{z}_k)$, and continuing exactly as in proof of Lemma 3.5 we obtain that by Assumption 5

$$\mathcal{L}(\bar{f}(\theta')) - \mathcal{L}(\bar{f}(\theta)) \geq \langle\nabla_\theta\mathcal{L}(\bar{f}(\theta)), \theta' - K\cdot\theta\rangle - O\big((R+\delta)^{1/3}H^{5/2}\sqrt{m\log m}\big)O(L_c)\sqrt{d}R.$$

where $L = O(L_c)$ according to Lemma B.3. □