Learning Single-Index Models with Shallow Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Single-index models are a class of functions given by an unknown univariate "link" function applied to an unknown one-dimensional projection of the input. These models are particularly relevant in high dimension, when the data might present low-dimensional structure that learning algorithms should adapt to. While several statistical aspects of this model, such as the sample complexity of recovering the relevant (one-dimensional) subspace, are well-understood, they rely on tailored algorithms that exploit the specific structure of the target function. In this work, we introduce a natural class of shallow neural networks and study its ability to learn single-index models via *gradient descent*. More precisely, we consider shallow networks in which the first layer weights are tied, to mirror the single-index model structure, and biases of the neurons are frozen at random initialization. We show that the corresponding optimization landscape is benign, which in turn leads to generalization guarantees that match the optimal sample complexity of dedicated semi-parametric methods.

1 Introduction

High-dimensional learning with both computational and statistical guarantees remains an outstanding challenge, with even more relevance in the face of current scaling trends. One important question which has seen considerable attention recently is to understand the advantages brought by non-linear learning models, such as neural networks, over more mature (from a theoretical standpoint) counterparts, such as kernel methods [33, 34, 53, 13], even in their simplest shallow instantiations.

While approximation benefits of shallow neural networks over non-adaptive kernels have been known for decades [8, 41], an important piece of this theoretical puzzle was provided by the *statistical* analysis of [5], which hinted at an inherent advantage of neural networks for extracting information from high-dimensional data with a "hidden" low-dimensional structure. Alas, the remaining piece of this puzzle, namely providing *computational* guarantees, is still mostly unresolved.

Several computational *hardness* results for learning functions that can be efficiently approximated by shallow neural networks [18, 26, 48, 17] have been established in the literature, which rule out positive results in the general setting. On the other hand, progress has been made on the positive side [1, 2, 45] by focusing on function classes with strong structural properties, thereby showcasing the adaptive representation learning capabilities of neural networks.

This work aligns with the latter effort, and focuses on the class of *single-index models*. Single-index models are high-dimensional functions $F : \mathbb{R}^d \to \mathbb{R}$ of the form $F_*(x) = f_*(\langle \theta^*, x \rangle)$, where *both* the univariate "link" function $f_* : \mathbb{R} \to \mathbb{R}$ and relevant (one-dimensional) subspace $\theta^* \in \mathbb{S}^{d-1}$ are unknown. These models have been extensively studied in the statistics literature [16, 28, 19], leading to dedicated algorithmic procedures, and can be provably approximated with shallow neural networks without incurring a curse-of-dimensionality [5]. In contrast, the analysis of neural network

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

learning using gradient-descent techniques has been more heavily focused on the so-called "Teacher-Student" setup [27, 24, 54, 55, 3], where the link function f is assumed to be known.

The mix of a high-dimensional parametric component (the hidden direction) with a low-dimensional non-parametric one (the link function) in single-index models naturally suggests a shallow neural network architecture where the inner weights are "active", while the biases are "lazy" [14]. We instantiate such an architecture by *freezing* the biases at their random initialization, and analyze gradient descent dynamics on the remaining free parameters in the continuous-time limit.

Our main results establish that as soon as the width N of the network is bigger than a quantity, which depends solely on the smoothness properties of the link function f_*[1], gradient descent recovers the unknown direction θ^* and approximates the univariate link function f_* with optimal sample and time complexity $O(d^s)$ (Theorem 5.1), where s is the so-called *information-exponent* of the link function [3]. The information exponent roughly captures the signal strength, which here refers to the alignment between the network direction θ and the hidden direction θ^*, at typical initializations.

The success of gradient descent relies on the benign optimization landscape of the empirical loss, though the presence of degenerate saddles requires a careful convergence analysis leveraging uniform convergence results for the empirical landscape [37, 20]. We show that gradient descent over our proposed neural network architecture solves two distinct problems–univariate non-parametric kernel ridge regression and non-convex optimization in high dimension–*simultaneously* and *efficiently*, cementing its role as a versatile algorithm for high-dimensional learning. We illustrate our theoretical results with experiments in Appendix D.

2 Related Work

Our work builds upon methods from different areas in statistics, optimization, and machine learning.

Single–index and multi-index models. A useful modeling assumption in high-dimensional regression is that the regression function $F(x) = \mathbb{E}[y|x]$ only depends on one or a few directions. This leads to the single-index model $F(x) = f_*(\langle\theta^*, x\rangle)$, and multi-index model $F(x) = f_*(\langle\theta_1^*, x\rangle, \ldots, \theta_k^*, x\rangle)$, with k typically much smaller than the dimension. Such models have a long history in the statistics literature and different methods exist for various estimation problems, including projection pursuit [21], slicing [31], gradient-based estimators [32], and moment-based estimators [16]. When the function f_* is also to be estimated, we face a semi-parametric problem involving parameter recovery of θ^* and non-parametric estimation of f_*. Our work is closely related to [19], which also characterizes the population landscape of certain objectives by leveraging Gaussian data. Single and multi-index models are also studied in [5] in the context of shallow neural networks, where it is shown that certain models of infinite-width shallow networks can adapt to such low-dimensional structure, though no tractable algorithms are introduced. Finally, [1] studies the learnability of functions on the hypercube by shallow neural networks and stochastic gradient descent and introduces the *merged staircase property*, which provides necessary and sufficient conditions for learnability with linear sample complexity $n = O(d)$. While they learn a broader class of functions (including multi-index model) for a more efficient sample complexity regime ($O(d)$ vs $O(d^s)$), our work captures the regime of semi-parametric estimation by considering Gaussian data without the sparsity constraints restrictions on F implied by their merged staircase property.

Teacher–student models. In the context of neural networks, several works have considered the teacher–student setting, where the target function F takes the form of a neural network with the same activation as network used for learning [27, 24, 54, 55, 3]. In this case the problem does not involve non-parametric estimation as in our setup, but this line of work often involves studying optimization landscapes similar to ours for estimating hidden directions.

Kernels and random features. Our work builds on the kernel methods literature for approximation and non-parametric regression [47, 7], their links with neural networks [15, 5], and in particular on random feature approximation [42, 6, 44, 38].

[1] Note that this quantity is independent of the input dimension d as the link function f_* is univariate.

Non-convex optimization landscapes. There is a vast literature studying tractable non-convex optimization landscapes, arising from high-dimensional statistics and statistical physics [36, 35, 22, 4, 43, 49, 10, 24]. A particular aspect of our setup is that the optimization landscape does *not* have the strict saddle property, which is often leveraged to establish global convergence [29].

3 Preliminaries

We focus on regression problems under a single-index model with Gaussian input data. Specifically, we assume d-dimensional inputs $x \sim \gamma_d := \mathcal{N}(0, I_d)$, and labels

$$y = F^*(x) + \epsilon = f_*(\langle \theta^*, x \rangle) + \epsilon,$$

where $\theta^* \in \mathbb{S}^{d-1}$ and ϵ is random noise such that $\mathbb{E}[\epsilon|x] = 0$, $\mathbb{E}[\epsilon^2|x] \leq \sigma^2$.

Shallow networks and random features. We consider learning algorithms based on shallow neural networks of the form

$$G(x; c, \theta) = c^\top \Phi(\langle \theta, x \rangle) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} c_i \phi(\langle \theta, x \rangle - b_i),$$

with $\theta \in \mathbb{S}^{d-1}$, where $\phi(u) = \max\{0, u\}$ is the ReLU activation, $b_i \sim \mathcal{N}(0, \tau^2)$ (we assume $\tau > 1$) are random "bias" scalars that are frozen throughout training, and $\Phi(u) = \frac{1}{\sqrt{N}}(\phi(u - b_i))_{i \in [N]}$ is the vector of random features. If we also fix the direction θ, learning c alone may be seen as a random feature model [42] that approximates a kernel method with the following kernel:

$$\kappa(u, v) = \mathbb{E}_{b \sim \gamma_\tau}[\phi_b(u)\phi_b(v)] \ , \tag{1}$$

where $\phi_b(u) := \phi(u - b)$ and $\gamma_\tau = \mathcal{N}(0, \tau^2)$ is the Gaussian measure on $\mathbb{R}$ with mean zero and variance τ^2. The kernel κ corresponds to a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$, which we precisely define as

$$\mathcal{H} := \left\{ f : \mathbb{R} \to \mathbb{R} \mid \ f(u) = \int c(b)\phi(u - b) d\gamma_\tau(b), \ c \in L_2(\gamma_\tau) \right\} \ .$$

The following lemma characterizes the corresponding RKHS norm $\| \cdot \|_{\mathcal{H}}$.

Lemma 3.1 (RKHS Sobolev Representation, [52, Theorem 5])**.** *The RKHS norm in $\mathcal{H}$ is given by*

$$\|f\|_{\mathcal{H}}^2 = \begin{cases} \int \frac{|f''(u)|^2}{\gamma_\tau(u)} du & \text{if } \lim_{u \to -\infty} f(u) = \lim_{u \to -\infty} f'(u) = 0, \\ \infty & \text{otherwise} \ . \end{cases} \tag{2}$$

where f'' is defined in the generalised (Sobolev) sense.

Remark 3.2. *The boundary conditions for this RKHS can be relaxed by symmetrizing the kernel, ie $\kappa_s(u, v) = \frac{1}{2}(\kappa(u, v) + \kappa(-u, -v))$, but we will keep the original boundary conditions and assume them on f_* for simplicity and w.l.o.g.*

Hermite decompositions. Given that our data is normally distributed, we consider the family of (normalized) Hermite polynomials $\{h_j\}_{j \in \mathbb{N}}$, which form an orthonormal basis of $L^2(\gamma)$, the space of squared-integrable function under the Gaussian measure $\gamma := \mathcal{N}(0, 1)$. We will denote by $f_* = \sum_j \alpha_j h_j$ the Hermite decomposition of the target link function.

We apply the following useful properties of Hermite polynomials [40]:

$$h_j' = \sqrt{j} h_{j-1} \quad \text{and} \quad \langle h_j(\langle \theta, \cdot \rangle), h_{j'}(\langle \theta', \cdot \rangle) \rangle_{\gamma_d} = \delta_{j,j'} \langle \theta, \theta' \rangle^j,$$

where $\langle \cdot, \cdot \rangle_{\gamma_d}$ is the inner product in $L^2(\gamma_d)$ and δ the Kronecker delta. We will assume throughout that $\|f_*\|_\gamma^2 = \sum_j \alpha_j^2$, $\|f_*'\|_\gamma^2 = \sum_j j\alpha_j^2$, and $\|f_*''\|_\gamma^2 = \sum_j j(j-1)\alpha_j^2$ are all finite (see Assumption 4.1).

Definition 3.3 (Population loss)**.** *We define the L_2-regularized population loss by*

$$L(c, \theta) = \mathop{\mathbb{E}}_{x \sim \gamma_d}[(F^*(x) - G(x; c, \theta))^2] + \lambda \|c\|^2 \ .$$

From features to Hermite coefficients. To precisely characterize the landscape of $L(c, \theta)$, we introduce notation to represent the relationship between features ϕ_b in terms of Hermite polynomials h_j. We define the linear integral operator $\mathcal{T} : L_2(\gamma) \to \mathbb{R}^N$ by

$$(\mathcal{T} f)_i := \frac{1}{\sqrt{N}} \langle f, \phi_{b_i} \rangle_\gamma \ . \tag{3}$$

Note that the operator $\mathcal{T}$ has rank N almost surely. We abuse notation to equivalently let $\mathcal{T}(j, b_i) := \frac{1}{\sqrt{N}} \langle h_j, \phi_{b_i} \rangle_\gamma = (\mathcal{T} h_j)_i$ and observe that $(\mathcal{T} f)_i = \sum_j \mathcal{T}(j, b_i)$. Observe that $\mathcal{T}$ has an adjoint $\mathcal{T}^* : \mathbb{R}^N \to L_2(\gamma)$ defined as $(\mathcal{T}^* c)(u) = \frac{1}{\sqrt{N}} \sum_{i=1}^N c_i \phi(u - b_i) = c^\top \Phi(u)$. Let $\mathcal{T}_j := \mathcal{T} h_j = (\mathcal{T}(j, b_i))_{i=1 \dots N} \in \mathbb{R}^N$. We can then write down the Hermite expansion of the network:

$$G(x; c, \theta) = c^\top \Phi(\langle \theta, x \rangle) = \sum_{j \geq 0} \langle c, \mathcal{T}_j \rangle h_j(\langle \theta, x \rangle).$$

Denoting $m := \langle \theta, \theta^* \rangle$, this allows us to rewrite the regularized objective as

$$L(c, \theta) = \sum_j \alpha_j^2 + \sum_j \langle c, \mathcal{T}_j \rangle^2 - 2 \sum_j \alpha_j \langle c, \mathcal{T}_j \rangle m^j + \lambda \|c\|^2,$$

where the term $\sum_j \alpha_j^2$ is a constant that can be ignored. Let $Q := \mathcal{T}\mathcal{T}^* \in \mathbb{R}^{N \times N}$ be a feature covariance matrix and $Q_\lambda = Q + \lambda I$. Note that Q_λ is positive semi-definite for $\lambda > 0$. We define the regularized projection $\hat{P}_\lambda = \hat{\Sigma}(\hat{\Sigma} + \lambda I)^{-1}$ onto the random feature space $\hat{\mathcal{H}}$ for $\hat{\Sigma} = \mathcal{T}^* \mathcal{T}$. Observe that $(\hat{P}_\lambda f)(u) = c^{*\top} \Phi(u)$, where $c^* \in \mathbb{R}^N$ is the solution to the following objective:

$$\min_{c \in \mathbb{R}^N} \left\| f - c^\top \Phi \right\|_\gamma^2 + \lambda \|c\|_2^2 \ . \tag{4}$$

Geometry on the sphere. Because the direction θ is constrained to lie on the sphere, our optimization algorithms rely on spherical (Riemannian) gradients, which are defined as follows:

$$\nabla_\theta^{\mathbb{S}^{d-1}} L(c, \theta) = \mathrm{Proj}_{\theta^\perp} \nabla_\theta L(c, \theta) \ ,$$

where $\mathrm{Proj}_{\theta^\perp} v = v - \langle \theta, v \rangle \theta$. We say that (c, θ) is a *critical point* of L if $\nabla_\theta^{\mathbb{S}^{d-1}} L(c, \theta) = 0$ and $\nabla_c L(c, \theta) = 0$.

4 Population Landscape under Frozen Random Biases

In order to understand the optimization and generalization properties of applying gradient descent to the empirical loss $L_n(c, \theta)$, we first study the optimization landscape of the population loss $L(c, \theta)$. We precisely characterize the critical points of the population loss in order to show that a gradient-based optimizer of a neural network of sufficient width N converges only when its direction vector θ is either parallel or orthogonal to the target direction θ^*. In the subsequent section, we show that having sufficiently many training samples n provides a similar optimization landscape and has favorable generalization properties for most initializations.

We will make the following regularity assumptions on the target link function f_*, in order to ensure small approximation error from random features, and a benign empirical landscape.

Assumption 4.1 (Regularity of f_*)**.** *Let $f_*^{(p)}$ be the pth derivative of f_*. We assume that $f_*^{(p)} \in L^2(\gamma)$ (i.e. $\|f_*^{(p)}\|_\gamma < \infty$) for all $p \leq 6$, that f_* is Lipschitz and that f_* is polynomially bounded.*

The index of the first non-zero Hermite coefficient α_j of f_* is referred to as the information exponent *and is given by $s := \min\{j : \alpha_j \neq 0\}$.*

To consider the critical points of $L(c, \theta)$, we introduce the *projected population loss* at a given direction θ:

$$\bar{L}(\theta) := \min_c L(c, \theta) \ .$$

Crucially, the sufficient number of features N depends on the regularization parameter $\lambda \in (0, 1)$, the variance $\tau^2 > 1$ of the random biases, and the target link function f_*, but *not* on the input dimension d.

Theorem 4.2 (Critical Points of Population Loss)**.** *For $\tau > 1$, and $\delta \in (0,1)$, there exist constants C_τ and C_{f_*} depending only on τ and the target link function f_* respectively such that if*

$$\lambda < C_{f_*}^{-(2+1/\tau^2)} \qquad \text{and} \qquad N \geq \frac{C_\tau}{\lambda} \ln\left(\frac{1}{\lambda\delta}\right) \tag{5}$$

then with probability $1-\delta$ over the random biases $b_1, \ldots, b_N$, the set of first-order critical points $\mathcal{C}_L := \{(c,\theta) : \nabla_\theta^{\mathbb{S}^{d-1}} L(c,\theta) = 0,\ \nabla_c L(C,\theta) = 0\}$ satisfies:

1. *(orientation relative to θ^*) if $(c,\theta) \in \mathcal{C}_L$, then either $\theta \in \{\pm\theta^*\}$ or $\langle \theta, \theta^* \rangle = 0$; and*
2. *(existence and uniqueness of c) if $\nabla_\theta^{\mathbb{S}^{d-1}} \bar{L}(\theta) = 0$, then there exists a unique c such that $(c,\theta) \in \mathcal{C}_L$.*

Theorem 4.2 thus establishes a benign optimization landscape in the population limit, rejoining several known non-convex objectives with similar behavior, such as tensor decomposition [25] or matrix completion [23]. It implies that all critical points are either at the equator or the poles and thus a gradient flow has to either avoid or escape the "spurious" critical points in the equator, i.e., θ's satisfying $\langle \theta^*, \theta \rangle = 0$ to succeed. Importantly, this optimization landscape has the same topology as if one were using Hermite basis functions rather than random scalar features [19, Theorem 5], the tailored choice that exploits the knowledge that the data is being generated by a single-index model on Gaussian samples. We view this as an interesting robustness property of shallow neural networks – at least in the regime where biases are randomly frozen.

We now outline the analysis that leads to the proof of the theorem before proving it formally in Section 4.3.

4.1 Exact expression of first-order critical points

The first step is to characterize the critical points of the regularized population loss. This is described in the following result.

Claim 4.3 (Critical point equation)**.** *Let $g_m, \bar{g}_m \in L_2(\gamma)$ be defined by*

$$g_m = \sum_j \alpha_j m^j h_j \ , \ \bar{g}_m = \sum_j \alpha_j j m^{j-1} h_j \ , \tag{6}$$

where $m = \langle \theta, \theta^* \rangle$. Then, the critical points of $L(c,\theta)$ satisfy the following equations:

$$c = Q_\lambda^{-1} \sum_j \alpha_j m^j \mathcal{T}_j \ ,$$

$$\sum_j \alpha_j^2 j m^{2j-1} - \langle (I - \hat{P}_\lambda) g_m, \bar{g}_m \rangle_\gamma = 0 \ .$$

Remark 4.4 (Infinite-width networks)**.** *If the kernel is universal and $N \to \infty$, then $\hat{P}_\lambda$ is the identity operator when $\lambda = 0$. Hence, $\langle h_j, P_\lambda h_{j'} \rangle = \delta_{j,j'}$ in this limit and the target can be exactly represented when $m = 1$.*

We prove Claim 4.3 in Appendix B.1 by exactly computing the gradients and relating the critical points of $\bar{L}(\theta)$ and $L(c,\theta)$.

4.2 Controlling random feature approximation

We now consider (finite) random feature approximations to functions in the RKHS. We show that the best possible loss of a linear combination of sufficiently many finite features is bounded above by the best approximation with infinitely many features with high probability.

Let $A(f,\lambda)$ be the (regularized) L_2 approximation error for functions in the space $\mathcal{H}$ with respect to the target function f and measure γ. Formally,

$$A(f,\lambda) := \min_{g \in \mathcal{H}} \|f - g\|_\gamma^2 + \lambda \|g\|_{\mathcal{H}}^2 \ .$$

Lemma 4.5 (Random features approximation, adapted from [6]). *Let $\delta \in (0,1)$, $\tau > 1$, and $f \in L^2(\gamma)$. There exists a constant $C_\tau > 0$ such that if $N \geq \frac{C_\tau}{\lambda} \ln \frac{1}{\lambda\delta}$, then with probability at least $1-\delta$,*

$$\|(I - \hat{P}_\lambda)f\|_\gamma^2 \leq 4A(f,\lambda). \tag{7}$$

Proof. The claim is immediate from Lemmas B.5 and B.6 in Appendix B.2. □

Next, we show that the approximation error of the RKHS corresponding to an infinite number of random features can be bounded in terms of the regularization λ and the γ-norm of the second derivative of the target function. The bound is independent of d, as we consider τ and λ fixed as d grows.

Lemma 4.6 (RKHS approximation error). *Assume $f \in L^2(\gamma)$ is in the weighted Sobolev space $H^2(\gamma)$ and has polynomial growth (i.e. there exist $C > 0$ and $M > 0$ such that $|f(u)| \leq C(u^M + u^{-M})$). Then,*

$$A(f,\lambda) = O\left(\lambda^\beta \|f''\|_\gamma^2\right), \tag{8}$$

with $\beta := \frac{\tau^2}{\tau^2+2}$.

The proof appears in Appendix B.3.

4.3 Proof of Theorem 4.2

Proof. For (c,θ) to be a critical point of L, it must satisfy $\nabla_\theta^{\mathbb{S}^{d-1}} L(c,\theta) = 0$ and $\nabla_c L(\theta,c) = 0$. According to Claim 4.3, every fixed θ has a unique complementary c such that $\nabla_c L(\theta,c) = 0$, and $\nabla_\theta^{\mathbb{S}^{d-1}} L(c,\theta) = 0$ if and only if $\theta \in \{-\theta^*, \theta^*\}$ or

$$\sum_j \alpha_j^2 j m^{2j-1} = \langle (I - \hat{P}_\lambda) g_m, \bar{g}_m \rangle_\gamma. \tag{9}$$

The remainder of the proof involves showing that the latter condition is true only if $m = 0$. We derive a contradiction whenever $m \neq 0$. Let $\beta := \frac{\tau^2}{\tau^2+2}$ and take $N \geq \frac{C_\tau}{\lambda} \ln \frac{1}{\lambda\delta}$ and

$$\lambda < \left(\frac{C' \|f_*''\|_\gamma \sqrt{\|f_*''\|_\gamma^2 + \|f_*'\|_\gamma^2}}{\alpha_s^2 s} \right)^{-2/\beta}$$

for sufficiently large C' and C_τ. We contradict the equality with probability at least $1-\delta$ over the randomly sampled biases $b_1, \ldots, b_N$:

$$\begin{aligned}
\left|\langle (I - \hat{P}_\lambda) g_m, \bar{g}_m \rangle_\gamma\right| &\leq \|(I - \hat{P}_\lambda) g_m\|_\gamma \|\bar{g}_m\|_\gamma \\
&\leq 2\sqrt{A(g_m,\lambda)} \|\bar{g}_m\|_\gamma && \text{[Lemma 4.5]} \\
&\leq C' \lambda^{\beta/2} \|g_m''\|_\gamma \|\bar{g}_m\|_\gamma && \text{[Lemma 4.6]} \\
&\leq C' \lambda^{\beta/2} |m|^{2s-1} \|f_*''\|_\gamma \sqrt{\|f_*''\|_\gamma^2 + \|f_*'\|_\gamma^2} && \text{[Lemma B.7]} \\
&< \alpha_s^2 s |m|^{2s-1} \leq \sum_{j=0}^\infty \alpha_j^2 j |m|^{2j-1} = \left| \sum_{j=0}^\infty \alpha_j^2 j m^{2j-1} \right|. \ \square
\end{aligned}$$

5 Empirical Landscape and Generalization Guarantees

Now that we have established that the population landscape $L(c,\theta)$ has a particularly simple structure given sufficiently many random features $N \gtrsim O(1)$, we study the optimization properties of its finite-sample counterpart. For training samples $\{(x_i, y_i)\}_{i\in[n]}$ generated by independently drawn $x_i \sim \mathcal{N}(0, I)$ and $y_i = f_*(\langle x_i, \theta^* \rangle)$ [2] and random features $\Phi : \mathbb{R} \to \mathbb{R}^N$ with

[2] For simplicity of the analysis and exposition, we assume clean labels without additive noise, but note we may incorporate additive noise to our analysis in a straightforward manner.

$\Phi(u)_j = N^{-1/2}\phi(u - b_j)$, we define the *empirical risk* as

$$L_n(c,\theta) := \frac{1}{n}\sum_{i=1}^{n}\left(c^\top \Phi(\langle x_i,\theta\rangle) - y_i\right)^2 + \lambda\|c\|^2 \,. \tag{10}$$

We specifically consider the estimator $\hat{F}(x) := \hat{f}(\langle x, \hat{\theta}\rangle)$, where $(\hat{f}, \hat{\theta})$ are obtained by using gradient descent on c and θ to minimize $L_n(c,\theta)$. Such strategy appears to be reasonable, in light of the properties of the population landscape studied in Section 4, namely that its only local minimisers are also global, and correspond to $\tilde{f} = \hat{P}_\lambda f^*$ and $\tilde{\theta} = \theta_*$. Given a sufficiently large sample size n, one expects the empirical landscape L_n to concentrate around its expectation L and inherit its benign optimization properties. However, the presence of a degenerate saddle at $(c, m) = (0, 0)$ flattens the landscape around the "equator" $\{\theta : \langle\theta, \theta_*\rangle = 0\}$, and a sufficiently large sample complexity is ultimately required to ensure that gradient descent iterates can escape from it, given its dangerously close initialization $|\langle\theta_0, \theta_*\rangle| = \Theta(1/\sqrt{d})$.

Prior works have obtained the optimal sample complexity of $n = O(d^s)$ (recall that s is the information exponent of the target function f_*) for recovering the target direction θ_* either by employing a learning algorithm that explicitly learns individual harmonic polynomials [19] or by assuming that f^* is known a priori [3]. Their argument goes essentially as follows:

- The empirical optimization landscape (when regarded as a function only of the direction θ) near the equator $|m| \ll 1$ is of the form $L(\theta) = \Theta(m^s)$.
- In order to certify that the optimization does not converge to a suboptimal critical point on the equator (i.e. $\|\nabla L(\theta)\| \gtrsim \epsilon$) one must require that $m \gtrsim \epsilon^{1/(s-1)}$.
- A standard uniform concentration argument of the form $\|\nabla L(\theta) - \nabla L_n(\theta)\| \lesssim \sqrt{\frac{d}{n}}$ and the fact that $m \sim d^{-1/2}$ at initialization together imply that $n = O(d^s)$ samples are sufficient to escape from the "influence" of the equator.

In order to repurpose these arguments to our setting, the relative scaling of the top-layer weights c relative to the direction vector θ will turn out to be crucial, as has also been observed in the literature on *lazy*-vs-*rich* regimes [14, 53] in the context of overparametrized neural networks.

We consider an idealized version of a Gradient Descent algorithm over the empirical loss L_n in the infinitesimally small learning rate regime, resulting in the gradient flow ODE:

$$\begin{aligned} \dot{c}(t) &= -\zeta(t)\nabla_c L_n(c,\theta) \\ \dot{\theta}(t) &= -\nabla_\theta^{\mathbb{S}^{d-1}} L_n(c,\theta) \,, \end{aligned} \tag{11}$$

where ζ is the relative time-scale between first- and second-layer weights, and $\nabla_\theta^{\mathbb{S}^{d-1}}$ is the Riemannian gradient. The learning rate schedule $\zeta(t)$ is set so that only θ is trained first and after a fixed amount of time, which we denote by T_0, then c and θ is jointly optimized. This is required to guarantee that the gradient flow *escapes* the equator (see Lemma C.4), which is crucial since the (random) starting point for the flow is close to the equator with high probability.

Our main result, proved in Appendix C, establishes that this gradient flow, under a simple schedule for ζ, efficiently finds an approximate minimiser of the *population* loss, with an error explicitly quantified as a function of n:

Theorem 5.1 (Gradient flow finds approximate minimizers)**.** *For $\delta > 0$ and f_* satisfying Assumption 4.1, suppose the following are true: (i) $d = \Omega(\log\frac{1}{\delta})$, (ii) $\lambda = O(1)$ (iii) $n = \Omega(\frac{d^s}{\lambda^4}\log\frac{1}{\delta})$, (iv) $N = \Omega(\frac{1}{\lambda}\log\frac{1}{\lambda\delta})$ & $N = O(\frac{1}{\lambda}(\frac{\lambda^4 n}{d})^{1/(2(s+1))})$, (v) $N_0 = \Omega(\log\frac{1}{\delta})$ & $N_0 = O((\frac{\lambda^4 n}{d})^{1/(2(s+1))})$, (vi) $\rho = \Theta(\sqrt{N}N_0^{-(2+s)/2})$, (vii) $T_0 = \tilde{\Theta}(N_0^{1+s/2}d^{-1+s/2})$, and (viii) $T_1 = \Theta(\frac{\lambda^4 n}{d})$. Then, if we run Procedure 1 for $T = T_0 + T_1$ time steps with the above parameters, its outcome $(\hat{c}, \hat{\theta})$ satisfies with probability at least $\frac{1}{2} - \delta$ the following:*

$$\begin{aligned} \|\hat{c}^\top\Phi - \hat{P}_\lambda f_*\|_\gamma &= \widetilde{O}\left(\sqrt{\frac{d}{\lambda^4 n}}\right) + \widetilde{O}\left(\frac{d}{\lambda^5 n}\right) \,, \\ 1 - |\langle\theta_T, \theta^*\rangle| &= \widetilde{O}\left(\frac{d}{\lambda^4 n}\right) \,. \end{aligned} \tag{12}$$

Procedure 1 Gradient Flow

Require: $N_0, \rho, T_0, T_1, N, \lambda$
Initialize $\theta(0) \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, $c(0) \sim \mathrm{Unif}(\{c \in \mathbb{R}^N; \|c\|_2 = \rho; \|c\|_0 = N_0\})$.
Run Gradient Flow (11) with $\zeta(t) = \mathbf{1}(t > T_0)$ up to time $T = T_0 + T_1$.
Set $\hat{\theta} = \theta(T)$, $\hat{c} = c(T)$.

Procedure 2 Fine-Tuning

Require: $\lambda_{n'}$, $\hat{\theta}$ from Procedure 1.
Set $\hat{c} = \arg\min_c L'_{n'}(c, \hat{\theta})$ as in (14).

Corollary 5.2 (Excess risk of Procedure 1)**.** *Under the assumptions of Theorem 5.1, after optimizing the regularization parameter λ, one obtains an excess risk of the form*

$$\|\hat{F} - F_{\theta^*}\|_{\gamma_d}^2 \lesssim \left(\frac{d \log d}{n}\right)^{\frac{2\beta}{\beta+8}} . \tag{13}$$

Fine-tuning the second layer. After running Procedure 1, we may include a final fine-tuning phase of training for second layer weights c alone, using a separate training sample (x'_i, y'_i), $i = 1, \ldots, n'$ and a possibly different regularization parameter $\lambda_{n'}$. More precisely, we set

$$\hat{c} = \arg\min_c \left\{ L'_{n'}(c, \hat{\theta}) := \frac{1}{n'} \sum_{i=1}^{n'} (c^\top \Phi(\langle \hat{\theta}, x'_i \rangle) - y'_i)^2 + \lambda_{n'} \|c\|^2 \right\}, \tag{14}$$

where $\hat{\theta}$ denotes the output of the previous gradient descent phase. Note that this is a strongly convex optimization problem, and can thus be optimized efficiently using gradient methods or by solving a linear system. While this may not be needed in practice, we use a different training sample for technical reasons, namely to break the dependence between the data and the kernel, which depends on the initial training sample through $\hat{\theta}$. We note that such sample splitting strategies are commonly used in other contexts in the statistics literature (e.g., [9, 12]). We obtain the following guarantee.

Proposition 5.3 (Excess risk of fine-tuning)**.** *Let $\delta > 0$. Let $m = \langle \hat{\theta}, \theta^* \rangle$, where $\hat{\theta}$ is obtained from the previous gradient descent phase, and let $\hat{c}$ be the ridge regression estimator obtained from a fresh dataset $\mathcal{D}'$ of n' samples, N random features,*[3] *and regularization parameter $\lambda_{n'} := (\sigma^2\tau^2/\|f''_*\|_\gamma^2 n')^{1/(\beta+1)}$, and let $\hat{F}(x) = \hat{c}^\top \Phi(\langle \hat{\theta}, x \rangle)$. Assume*

$$n' \gtrsim \max\left\{\sigma^2\tau^2/\|f''_*\|_\gamma^2, (\|f''_*\|_\gamma^2/\sigma^2\tau^2)^{1/\beta}, \|f_*\|_\infty^2/(\sigma^2\tau^2)^{\beta/(\beta+1)}\right\}$$
$$N \gtrsim C_\tau \left(n' \|f''_*\|_\gamma^2/\sigma^2\tau^2\right)^{\frac{1}{\beta+1}} \ln\left(n'^{1/(\beta+1)}\delta^{-1}\right)$$

With probability at least $1 - \delta$ over the random features, we have

$$\mathop{\mathbb{E}}_{\mathcal{D}'}[\|\hat{F} - F_{\theta^*}\|_{\gamma_d}^2 | \hat{\theta}] \lesssim \|f''_*\|_\gamma^{\frac{2}{\beta+1}} \left(\frac{\sigma^2\tau^2}{n'}\right)^{\frac{\beta}{\beta+1}} + \|f_*\|_\gamma^2 (1 - |m|) \tag{15}$$

where the expectation is over the n' fresh samples, and is conditioned on the previously obtained $\hat{\theta}$.

Combining the two phases leads to the following generalization guarantee on the excess risk for Procedure 2.

Corollary 5.4. *Excess risk of Algorithm 2 Let $\delta > 0$. As in Theorem 5.1, assume $d \geq C \log(\delta^{-1})$, $\mu_s = \langle h_s, \Sigma h_s \rangle > 0$, and assumption 4.1 on f_* with a constant C_{f_*}. Let $\lambda = (s^2\alpha_s^2/2C_{f_*})^{2/\beta}$, and assume the following on the sample sizes and number of random features for the first phase (n, N, N_0) and fine-tuning phase (n', N'):*

$$n \gtrsim \lambda^{-1} d^s \log \delta^{-1}, \qquad n' \gtrsim \max\left\{\sigma^2\tau^2/\|f''_*\|_\gamma^2, (\|f''_*\|_\gamma^2/\sigma^2\tau^2)^{1/\beta}, \|f_*\|_\infty^2/(\sigma^2\tau^2)^{\frac{\beta}{\beta+1}}\right\}$$
$$N = N_0 = C\lambda^{-1}\log(\lambda^{-1}\delta^{-1}), \quad N' \gtrsim C_\tau \left(n' \|f''_*\|_\gamma^2/\sigma^2\tau^2\right)^{\frac{1}{\beta+1}} \ln\left(n'^{1/(\beta+1)}\delta^{-1}\right),$$

[3] These can be the same as in the previous phase.

and let ρ be as in Theorem 5.1. With probability at least $1/2 - 2\delta$ over the initial n samples, initialization, random features, we have

$$\mathop{\mathbb{E}}_{\mathcal{D}'}[\|\hat{F} - F_{\theta^*}\|_{\gamma_d}^2] \lesssim C_1 \frac{d}{n} + C_2 \left(\frac{\sigma^2\tau^2}{n'}\right)^{\frac{\beta}{\beta+1}}, \tag{16}$$

where C_1 and C_2 only depend on regularity of f_*, ignoring the optimization error, and logarithmic factors in n, d, and N.

We can make the following remarks:

- The empirical gradient flow therefore escapes the influence of the degenerate saddle with optimal sample complexity $n = \Theta(d^s)$ [3]. This is an instance of gradient descent successfully optimizing a non-convex objective *without the strict saddle property* as soon as $s > 2$, building from the simpler optimization landscapes of [19, 3]. This is in contrast with e.g. spiked tensor recovery problems [4] where the signal strength is substantially weaker, leading to complexity in the optimization landscape.
- While the recovery of the high-dimensional direction is optimal in statistical terms, the joint recovery of the link function is sub-optimal, by comparing Corollaries 5.2 and 5.4. In this sense, the fine-tuning stage recovers the optimal sample complexity, and where the non-parametric rate no longer depends on the ambient dimension d.
- The time-scale separation scheduling for ζ is sufficient but possibly not necessary. The analysis of vanilla dynamics ($\zeta(t) \equiv 1$) is challenging, since during the initial phase of training there may be adverse interaction effects between c and θ, which under naive analysis lead to sub-optimal sample complexity of $n \geq O(d^{2s})$.
- The time discretisation to turn Procedure 1 into a proper algorithm should follow from standard time discretisation arguments, although the case where $\phi = \text{ReLU}$ requires special care due to the non-smoothness of the loss. In such setting, such discretisation arguments do not hold for vanilla gradient descent in the worst-case [30], although may be recovered by appropriately smoothing the objective prior to computing the gradient.

6 Conclusions

This work studies the ability of shallow neural networks to learn single-index models with gradient descent. Our main results are positive, and demonstrate their ability to solve a semi-parametric problem with nearly optimal guarantees. Interestingly, this success story combines elements from the feature-learning regime —the ability to efficiently identify the hidden direction in high-dimensions under a non-convex objective, with ingredients from the lazy-regime, which offer better computational tradeoffs to solve low-dimensional non-parametric problems. Our technical analysis leverages tools from high-dimensional probability (such as uniform gradient concentration) and RKHS approximation theory, and complements the growing body of theoretical work certifying the efficiency of gradient methods on non-convex objectives. We have followed the standard approach of first establishing benign topological properties of the population loss, and then extend them to the empirical loss.

That said, there are several unanswered questions that our work has not addressed, such as:

- **Extension to Multi-Index Models.** Multi-index models are natural extensions of single-index models where the hidden direction is replaced by a hidden low-dimensional subspace. They enjoy similar statistical guarantees as single-index models [19, 5], and thus a natural question is whether the same algorithmic tools developed here extend to the multi-index setting.
- **Gradient dynamics without warmstart.** An unsatisfactory aspect of our results is the requirement that the algorithm starts by only optimizing θ for $t < T_0$. It would be interesting to understand whether the vanilla dynamics could also succeed.
- **Trainable biases and untied directions.** Our proposed neural network architecture is non-standard, in the sense that its biases are frozen at initialization and all neurons share the same inner weight. For the purposes of learning single-index models, removing these restrictions would not bring any statistical benefits. However, it would be interesting to extend our analysis to the general setting.

References

[1] E. Abbe, E. Boix-Adsera, and T. Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for SGD learning of sparse functions on two-layer neural networks. *arXiv preprint arXiv:2202.08658*, 2022.

[2] Z. Allen-Zhu and Y. Li. Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413*, 2020.

[3] G. B. Arous, R. Gheissari, and A. Jagannath. Online stochastic gradient descent on non-convex losses from high-dimensional inference. *arXiv preprint arXiv:2003.10409*, 2020.

[4] G. B. Arous, S. Mei, A. Montanari, and M. Nica. The landscape of the spiked tensor model. *Communications on Pure and Applied Mathematics*, 72(11):2282–2330, 2019.

[5] F. Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

[6] F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, 18(1):714–751, 2017.

[7] F. Bach. Learning theory from first principles, 2021.

[8] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[9] X. Bing, F. Bunea, S. Strimas-Mackey, and M. Wegkamp. Prediction under latent factor regression: Adaptive pcr, interpolating predictors and beyond. *Journal of Machine Learning Research*, 22(177):1–50, 2021.

[10] N. Boumal, V. Voroninski, and A. Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. *Advances in Neural Information Processing Systems*, 29, 2016.

[11] S.-H. Chang, P. C. Cosman, and L. B. Milstein. Chernoff-type bounds for the gaussian error function. *IEEE Transactions on Communications*, 59(11):2939–2944, 2011.

[12] V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

[13] L. Chizat and F. Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.

[14] L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming, 2018.

[15] Y. Cho and L. Saul. Kernel methods for deep learning. *Advances in neural information processing systems*, 22, 2009.

[16] A. S. Dalalyan, A. Juditsky, and V. Spokoiny. A new algorithm for estimating the effective dimension-reduction subspace. *The Journal of Machine Learning Research*, 9:1647–1678, 2008.

[17] A. Daniely and G. Vardi. Hardness of learning neural networks with natural weights. *Advances in Neural Information Processing Systems*, 33:930–940, 2020.

[18] I. Diakonikolas, D. M. Kane, V. Kontonis, and N. Zarifis. Algorithms and sq lower bounds for pac learning one-hidden-layer relu networks. In J. Abernethy and S. Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 1514–1539. PMLR, 09–12 Jul 2020.

[19] R. Dudeja and D. Hsu. Learning single-index models in gaussian space. In S. Bubeck, V. Perchet, and P. Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1887–1930. PMLR, 06–09 Jul 2018.

[20] D. J. Foster, A. Sekhari, and K. Sridharan. Uniform convergence of gradients for non-convex learning and optimization. In *Advances in Neural Information Processing Systems*, 2018.

[21] J. H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American statistical Association*, 76(376):817–823, 1981.

[22] R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.

[23] R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minimum. *Advances in neural information processing systems*, 29, 2016.

[24] R. Ge, J. D. Lee, and T. Ma. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017.

[25] R. Ge and T. Ma. On the optimization landscape of tensor decompositions. *Advances in Neural Information Processing Systems*, 30, 2017.

[26] S. Goel, A. Gollakota, Z. Jin, S. Karmalkar, and A. Klivans. Superpolynomial lower bounds for learning one-layer neural networks using gradient descent. In *International Conference on Machine Learning*, pages 3587–3596. PMLR, 2020.

[27] S. Goldt, M. Advani, A. M. Saxe, F. Krzakala, and L. Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. *Advances in neural information processing systems*, 32, 2019.

[28] M. Hristache, A. Juditsky, and V. Spokoiny. Direct estimation of the index coefficient in a single-index model. *The Annals of Statistics*, 29(3):595–623, 2001.

[29] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan. How to escape saddle points efficiently, 2017.

[30] G. Kornowski and O. Shamir. Oracle complexity in nonsmooth nonconvex optimization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 324–334. Curran Associates, Inc., 2021.

[31] K.-C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[32] K.-C. Li. On principal hessian directions for data visualization and dimension reduction: Another application of stein's lemma. *Journal of the American Statistical Association*, 87(420):1025–1039, 1992.

[33] E. Malach, P. Kamath, E. Abbe, and N. Srebro. Quantifying the benefit of using differentiable learning over tangent kernels. In *International Conference on Machine Learning*, pages 7379–7389. PMLR, 2021.

[34] E. Malach, G. Yehudai, S. Shalev-Schwartz, and O. Shamir. The connection between approximation, depth separation and learnability in neural networks. In *Conference on Learning Theory*, pages 3265–3295. PMLR, 2021.

[35] S. S. Mannelli, G. Biroli, C. Cammarota, F. Krzakala, P. Urbani, and L. Zdeborová. Marvels and pitfalls of the langevin algorithm in noisy high-dimensional inference. *Physical Review X*, 10(1):011057, 2020.

[36] S. S. Mannelli, F. Krzakala, P. Urbani, and L. Zdeborova. Passed & spurious: Descent algorithms and local minima in spiked matrix-tensor models. In *international conference on machine learning*, pages 4333–4342. PMLR, 2019.

[37] S. Mei, Y. Bai, and A. Montanari. The landscape of empirical risk for non-convex losses. *arXiv preprint arXiv:1607.06534*, 2016.

[38] S. Mei, T. Misiakiewicz, and A. Montanari. Generalization error of random feature and kernel methods: Hypercontractivity and kernel matrix concentration. *Applied and Computational Harmonic Analysis*, 59:3–84, 2022.

[39] S. Minsker. On some extensions of bernstein's inequality for self-adjoint operators. *Statistics & Probability Letters*, 127:111–119, 2017.

[40] R. O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[41] A. Pinkus. Approximation theory of the mlp model in neural networks. *Acta numerica*, 8:143–195, 1999.

[42] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[43] V. Ros, G. B. Arous, G. Biroli, and C. Cammarota. Complex energy landscapes in spiked-tensor and simple glassy models: Ruggedness, arrangements of local minima, and phase transitions. *Physical Review X*, 9(1):011003, 2019.

[44] A. Rudi and L. Rosasco. Generalization properties of learning with random features. *Advances in neural information processing systems*, 30, 2017.

[45] I. Safran and J. D. Lee. Optimization-based separations for neural networks. *arXiv preprint arXiv:2112.02393*, 2021.

[46] S. Saitoh. *Integral transforms, reproducing kernels and their applications*, volume 369. CRC Press, 1997.

[47] B. Schölkopf and A. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[48] M. J. Song, I. Zadik, and J. Bruna. On the cryptographic hardness of learning single periodic neurons. *Advances in Neural Processing Systems (NeurIPS)*, 2021.

[49] J. Sun, Q. Qu, and J. Wright. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198, 2018.

[50] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

[51] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[52] F. Williams, M. Trager, D. Panozzo, C. Silva, D. Zorin, and J. Bruna. Gradient dynamics of shallow univariate relu networks. *Advances in neural information processing systems*, 32, 2019.

[53] B. Woodworth, S. Gunasekar, J. D. Lee, E. Moroshko, P. Savarese, I. Golan, D. Soudry, and N. Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673. PMLR, 2020.

[54] K. Zhong, Z. Song, P. Jain, P. L. Bartlett, and I. S. Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *International conference on machine learning*, pages 4140–4149. PMLR, 2017.

[55] M. Zhou, R. Ge, and C. Jin. A local convergence theory for mildly over-parameterized two-layer neural network. In *Conference on Learning Theory*, pages 4577–4632. PMLR, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Conclusions.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This is a purely theoretical work with no practical applications.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sections 4 and 5.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 4.3.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Supplemental zip file.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section D.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]