On the non-universality of deep learning: quantifying the cost of symmetry

Emmanuel Abbe
EPFL

Enric Boix-Adserà
MIT

Abstract

We prove limitations on what neural networks trained by noisy gradient descent (GD) can efficiently learn. Our results apply whenever GD training is equivariant, which holds for many standard architectures and initializations. As applications, (i) we characterize the functions that fully-connected networks can weak-learn on the binary hypercube and unit sphere, demonstrating that depth-2 is as powerful as any other depth for this task; (ii) we extend the merged-staircase necessity result for learning with latent low-dimensional structure [ABM22] to beyond the mean-field regime. Under cryptographic assumptions, we also show hardness results for learning with fully-connected networks trained by stochastic gradient descent (SGD).

1 Introduction

Over the last decade, deep learning has made advances in areas as diverse as image classification [KSH12], language translation [BCB14], classical board games [SHS+18], and programming [LCC+22]. Neural networks trained with gradient-based optimizers have surpassed classical methods for these tasks, raising the question: can we hope for deep learning methods to eventually replace all other learning algorithms? In other words, is deep learning a universal learning paradigm? Recently, [AS20, AKM+21] proved that in a certain sense the answer is yes: any PAC-learning algorithm [Val84] can be efficiently implemented as a neural network trained by stochastic gradient descent; analogously, any Statistical Query algorithm [Kea98] can be efficiently implemented as a neural network trained by noisy gradient descent.

However, there is a catch: the result of [AS20] relies on a carefully crafted network architecture with memory and computation modules, which is capable of emulating an arbitrary learning algorithm. This is far from the architectures which have been shown to be successful in practice. Neural networks in practice do incorporate domain knowledge, but they have more "regularity" than the architectures of [AS20], in the sense that they do not rely on heterogeneous and carefully assigned initial weights (e.g., convolutional networks and transformers for image recognition and language processing [LB+95, LKF10, VSP+17], graph neural networks for analyzing graph data [GMS05, BZSL13, VCC+17], and networks specialized for particle physics [BAO+20]). We therefore refine our question:

Is deep learning with "regular" architectures and initializations a universal learning paradigm? If not, can we quantify its limitations when architectures and data are not well aligned?

We would like an answer applicable to a wide range of architectures. In order to formalize the problem and develop a general theory, we take an approach similar to [Ng04, Sha18, LZA21] of understanding deep learning through the *equivariance group* G (a.k.a., symmetry group) of the learning algorithm.

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

Definition 1.1 (G-equivariant algorithm)**.** *A randomized algorithm $\mathcal{A}$ that takes in a data distribution $\mathcal{D} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$[1] and outputs a function $\mathcal{A}(\mathcal{D}) : \mathcal{X} \to \mathcal{Y}$ is said to be G-equivariant if for all $g \in G$*

$$\mathcal{A}(\mathcal{D}) \stackrel{d}{=} \mathcal{A}(g(\mathcal{D})) \circ g. \qquad (G\text{-equivariance})$$

Here g is a group element that acts on the data space $\mathcal{X}$, and so is viewed as a function $g : \mathcal{X} \to \mathcal{X}$, and $g(\mathcal{D})$ is the distribution of $(g(\boldsymbol{x}), y)$, where $(\boldsymbol{x}, y) \sim \mathcal{D}$.

In the case that the algorithm $\mathcal{A}$ is deep learning on the distribution $\mathcal{D}$, the equivariance group depends on the optimizer, the architecture, and the network initialization [Ng04, LZA21].[2]

Examples of G-equivariant algorithms in deep learning In many deep learning settings, the equivariance group of the learning algorithm is large. Thus, in this paper, we call an algorithm "regular" if it has a large equivariance group. For example, SGD training of fully-connected networks with Gaussian initialization is orthogonally-equivariant [Ng04]; and is permutation-equivariant if we add skip connections [HZRS16]. SGD training of convolutional networks is translationally-equivariant if circular convolutions are used [SNPP19], and SGD training of i.i.d.-initialized transformers without positional embeddings is equivariant to permutations of tokens [VSP+17]. Furthermore, [LZA21, Theorem C.1] provides general conditions under which a deep learning algorithm is equivariant. See also the preliminaries in Section 2.

Summary of this work Based off of G-equivariance, we prove limitations on what "regular" neural networks trained by noisy gradient descent (GD) or stochastic gradient descent (SGD) can efficiently learn, implying a separation with the initializations and architectures considered in [AS20]. For GD, we prove a master theorem that enables two novel applications: (a) characterizing which functions can be efficiently weak-learned by fully-connected (FC) networks on both the hypercube and the unit sphere; and (b) a necessity result for which functions on the hypercube with latent low-dimensional structure can be efficiently learned. See Sections 1.2 and 1.3 for more details.

1.1 Related work

Most prior work on computational lower bounds for deep learning has focused on proving limitations of kernel methods (a.k.a. linear methods). Starting with [Bar93] and more recently with [WLLM19, AL19, KMS20, AL20, Hsu, HSSV21, ABM22] it is known that there are problems on which kernel methods provably fail. These results apply to training neural networks in the Neural Tangent Kernel (NTK) regime [JGH18], but do not apply to more general nonlinear training. Furthermore, for specific architectures such as FC architectures [GMMM21, Mis22] and convolutional architectures [MM21], the kernel and random features models at initialization are well understood, yielding stronger lower bounds for training in the NTK regime.

For nonlinear training, which is the setting of this paper, considerably less is known. In the context of sample complexity, [Ng04] introduced the study of the equivariance group of SGD, and constructed a distribution on d dimensions with a $\Omega(d)$ versus $O(1)$ sample complexity separation for learning with an SGD-trained FC architecture versus an arbitrary algorithm. More recently, [LZA21] built on [Ng04] to show a $O(1)$ versus $\Omega(d^2)$ sample-complexity separation between SGD-trained convolutional and FC architectures. In this paper, we also analyze the equivariance group of the training algorithm, but with the goal of proving superpolynomial computational lower bounds.

In the context of computational lower bounds, it is known that networks trained with noisy[3] gradient descent (GD) fall under the Statistical Query (SQ) framework [Kea98], which allows showing computational limitations for GD training based on SQ lower bounds. This has been combined in [AS20, SSS17, MS20, ACHM22] with the permutation symmetry of GD-training of i.i.d. FC networks to prove impossibility of efficiently learning high-degree parities and polynomials. In

[1] The set of probability distributions on Ω is denoted by $\mathcal{P}(\Omega)$. You should think of $\mathcal{D} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ as a distribution of pairs $(\boldsymbol{x}, y)$ of covariates and labels.

[2] Note that the equivariance group of a *training algorithm* should not be confused with the equivariance group of an *architecture* in the context of geometric deep learning [BBCV21]. In that context, G-equivariance refers to the property of a neural network architecture $f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}) : \mathcal{X} \to \mathcal{Y}$ that $f_{\mathsf{NN}}(g(\boldsymbol{x}); \boldsymbol{\theta}) = g(f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}))$ for all $\boldsymbol{x} \in \mathcal{X}$ and all group elements $g \in G$. In that case, G acts on both the input in $\mathcal{X}$ and output in $\mathcal{Y}$.

[3] Here the noise is used to control the gradients' precision as in [AS20, AKM+21].

our work, we show that these arguments can be viewed in the broader context of more general group symmetries, yielding stronger lower bounds than previously known. For stochastic gradient descent (SGD) training, [ABM22] proves a computational limitation for training of two-layer mean-field networks, but their result applies only when SGD converges to the mean-field limit, and does not apply to more general architectures beyond two-layer networks. Finally, most related to our SGD hardness result is [Sha18], which shows limitations of SGD-trained FC networks under a cryptographic assumption. However, the argument of [Sha18] relies on training being equivariant to linear transformations of the data, and therefore requires that data be whitened or preconditioned. Instead, our result for SGD does not require any preprocessing steps.

There is also recent work showing sample complexity benefits of invariant/equivariant neural network *architectures* [MMM21, EZ21, Ele21, BVB21, Ele22]. In contrast, we study equivariant training *algorithms*. These are distinct concepts: a deep learning algorithm can be G-equivariant, while the neural network architecture is neither G-invariant nor G-equivariant. For example, a FC network is not invariant to orthogonal transformations of the input. However, if we initialize it with Gaussian weights and train with SGD, then the learning algorithm is equivariant to orthogonal transformations of the input (see Proposition 2.5 below).

1.2 Contribution 1: Lower bounds for noisy gradient descent (GD)

Consider the supervised learning setup where we train a neural network $f_{\mathsf{NN}}(\cdot;\boldsymbol{\theta}) : \mathcal{X} \to \mathbb{R}$ parametrized by $\boldsymbol{\theta} \in \mathbb{R}^p$ to minimize the mean-squared error on a data distribution $\mathcal{D} \in \mathcal{P}(\mathcal{X} \times \mathbb{R})$,

$$\ell_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[(y - f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}))^2]. \tag{1}$$

The noisy Gradient Descent (GD) training algorithm randomly initializes $\boldsymbol{\theta}^0 \sim \mu_{\boldsymbol{\theta}}$ for some initialization distribution $\mu_{\boldsymbol{\theta}} \in \mathcal{P}(\mathbb{R}^p)$, and then iteratively updates the parameters with step size $\eta > 0$ in a direction $\boldsymbol{g}_{\mathcal{D}}(\boldsymbol{\theta}^k)$ approximating the population loss gradient, plus Gaussian noise $\boldsymbol{\xi}^k \sim \mathcal{N}(0, \tau^2 \boldsymbol{I})$,

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \eta \boldsymbol{g}_{\mathcal{D}}(\boldsymbol{\theta}^k) + \boldsymbol{\xi}^k. \tag{GD}$$

Up to a constant factor, $\boldsymbol{g}_{\mathcal{D}}(\boldsymbol{\theta})$ is the population loss gradient, except we have clipped the gradients of the network with the projection operator $\Pi_{B(0,R)}$ to lie in the ball $B(0,R) = \{\boldsymbol{z} : \|\boldsymbol{z}\|_2 \le R\} \subset \mathbb{R}^p$,[4]

$$\boldsymbol{g}_{\mathcal{D}}(\boldsymbol{\theta}) = -\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[(y - f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}))(\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}))].$$

Clipping the gradients is often used in practice to avoid instability from exploding gradients (see, e.g., [ZHSJ19] and references within). In our context, clipping ensures that the injected noise $\boldsymbol{\xi}^k$ is on the same scale as the gradient $\nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}$ of the network and so it controls the gradients' precision. Similarly to the works [AS20, AKM+21, ACHM22], we consider noisy gradient descent training to be efficient if the following conditions are met.

Definition 1.2 (Efficiency of GD, informal)**.** *GD training is efficient if the clipping radius R, step size η, and inverse noise magnitude $1/\tau$ are all polynomially-bounded in d, since then* (GD) *can be efficiently implemented using noisy minibatch SGD*[5].

We prove that some data distributions cannot be efficiently learned by G-equivariant GD training. For this, we introduce the G-alignment:

Definition 1.3 (G-alignment)**.** *Let G be a compact group, let $\mu_{\mathcal{X}} \in \mathcal{P}(\mathcal{X})$ be a distribution over data points, and let $f \in L^2(\mu_{\mathcal{X}})$ be a labeling function. The G-alignment of $(\mu_{\mathcal{X}}, f)$ is:*

$$\mathcal{C}((\mu_{\mathcal{X}}, f); G) = \sup_h \mathbb{E}_{g\sim\mu_G}[\mathbb{E}_{\boldsymbol{x}\sim\mu_{\mathcal{X}}}[f(g(\boldsymbol{x}))h(\boldsymbol{x})]^2],$$

where μ_G is the Haar measure of G and the supremum is over $h \in L^2(\mu_{\mathcal{X}})$ such that $\|h\|^2 = 1$.

In our applications, we use tools from representation theory (see e.g., [Kna96]) to evaluate the G-alignment. Using the G-alignment, we can prove a master theorem for lower bounds:

Theorem 1.4 (GD lower bound, informal statement of Theorem 3.1)**.** *Let $\mathcal{D}_f \in \mathcal{P}(\mathcal{X} \times \mathbb{R})$ be the distribution of $(\boldsymbol{x}, f(\boldsymbol{x}))$ for $\boldsymbol{x} \sim \mu_{\mathcal{X}}$. If $\mu_{\mathcal{X}}$ is G-invariant*[6] *and the G-alignment of $(\mu_{\mathcal{X}}, f)$ is small, then f cannot be efficiently learned by a G-equivariant GD algorithm.*

[4] Note that if f_{NN} is an R-Lipschitz model, then $\boldsymbol{g}_{\mathcal{D}}(\boldsymbol{\theta})$ will simply be the population gradient of the loss.

[5] Efficient implementability by minibatch SGD assumes bounded residual errors.

[6] Meaning that if $\boldsymbol{x} \sim \mu_{\mathcal{X}}$, then for any $g \in G$, we also have $g(\boldsymbol{x}) \sim \mu_{\mathcal{X}}$.

Proof ideas We first make an observation of [Ng04]: if a G-equivariant algorithm can learn the function f by training on the distribution $\mathcal{D}_f$, then, for any group element $g \in G$, it can learn $f \circ g$ by training on the distribution $\mathcal{D}_{f \circ g}$. In other words, the algorithm can learn the class of functions $\mathcal{F} = \{f \circ g : g \in g\}$, which can potentially be much larger than just the singleton set $\{f\}$. We conclude by showing that the class of functions $\mathcal{F}$ cannot be efficiently learned by GD training. The intuition is that the G-alignment measures the diversity of the functions in $\mathcal{F}$. If the G-alignment is small, then there is no function h that correlates with most of the functions in $\mathcal{F}$, which can be used to show $\mathcal{F}$ is hard to learn by gradient descent.

This type of argument appears in [AS20, ACHM22] in the specific case of Boolean functions and for permutation equivariance; our proof both applies to a more general setting (beyond Boolean functions and permutations) and yields sharper bounds; see Appendix A.3. Our bound can also be interpreted in terms of the Statistical Query framework, as we discuss in Appendix A.4. While Theorem 1.4 is intuitively simple, we demonstrate its power and ease-of-use by deriving two new applications.

Application: Characterization of weak-learnability by fully-connected (FC) networks In our first application, we consider weak-learnability: when can a function be learned non-negligibly better than just outputting the estimate $f_{\mathsf{NN}} \equiv 0$? Using Theorem 1.4, we characterize which functions over the binary hypercube $f : \{+1, -1\}^d \to \mathbb{R}$ and over the sphere $f : \mathbb{S}^{d-1} \to \mathbb{R}$ are efficiently weak-learnable by GD-trained FC networks with i.i.d. symmetric and i.i.d. Gaussian initialization, respectively. The takeaway is that a function $f : \{+1, -1\}^d \to \mathbb{R}$ is weak-learnable if and only if it has a nonnegligible Fourier coefficient of order $O(1)$ or $d - O(1)$. Similarly, a function $f : \mathbb{S}^{d-1} \to \mathbb{R}$ is weak-learnable if and only if it has nonnegligible projection onto the degree-$O(1)$ spherical harmonics. Perhaps surprisingly, such functions can be efficiently weak-learned by 2-layer fully-connected networks, which shows that adding more depth does not help. This application is presented in Section 3.1.

Application: Evidence for the staircase property In our second application, we consider learning a target function $f : \{+1, -1\}^d \to \mathbb{R}$ that only depends on the first P coordinates, $f(\boldsymbol{x}) = h(x_1, \ldots, x_P)$. Our regime of interest here is when the function $hand : \{+1, -1\}^P \to \mathbb{R}$ remains fixed and the dimension d grows, since this models the situation where a latent low-dimensional space determines the labels in a high-dimensional dataset. Recently, [ABM22] studied SGD-training of mean-field two-layer networks, and gave a near-characterization of which functions can be learned to arbitrary accuracy ϵ in $O_{h,\epsilon}(d)$ samples, in terms of the *merged-staircase property* (MSP). Using Theorem 1.4, we prove that the MSP is necessary for GD-learnability whenever training is permutation-equivariant (which applies beyond the 2-layer mean-field regime) and we also generalize it beyond leaps of size 1. Details are in Section 3.2.

1.3 Contribution 2: Hardness for stochastic gradient descent (SGD)

The second part of this paper concerns Stochastic Gradient Descent (SGD) training, which randomly initializes the weights $\boldsymbol{\theta}^0 \sim \mu_{\boldsymbol{\theta}}$, and then iteratively trains the parameters with the following update rule to try to minimize the loss (1):

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \eta \nabla_{\boldsymbol{\theta}} (y - f_{\mathsf{NN}}(\boldsymbol{x}_{k+1}; \boldsymbol{\theta}))^2 \mid_{\boldsymbol{\theta} = \boldsymbol{\theta}^k}, \tag{SGD}$$

where $(y_{k+1}, \boldsymbol{x}_{k+1}) \sim \mathcal{D}$ is a fresh sample on each iteration, and $\eta > 0$ is the learning rate.[7]

Proving computational lower bounds for SGD is a notoriously difficult problem [AKM+21], exacerbated by the fact that for general architectures SGD can be used to simulate any polynomial-time learning algorithm [AS20]. However, we demonstrate that one can prove hardness results for SGD training based off of cryptographic assumptions when the training algorithm has a large equivariance group. We demonstrate the non-universality of SGD on a standard FC architecture.

Theorem 1.5 (Hardness for SGD, informal statement of Theorem 4.4)**.** *Under the assumption that the Learning Parities with Noise (LPN) problem*[8] *is hard, FC neural networks with Gaussian initialization*

[7] For brevity, we focus on one-pass SGD with a single fresh sample per iteration. Our results extend to empirical risk minimization (ERM) setting and to mini-batch SGD, see Remark E.1.

[8] See Section 4 and Appendix D.3 for definitions and discussion on LPN.

trained by SGD cannot learn $f_{\text{mod8}} : \{+1, -1\}^d \to \{0, \ldots, 7\}$,

$$f_{\text{mod8}}(\boldsymbol{x}) \equiv \sum_{i=1}^{d} x_i \pmod 8,$$

in polynomial time from noisy samples $(\boldsymbol{x}, f_{\text{mod8}}(\boldsymbol{x}) + \xi)$ *where* $\boldsymbol{x} \sim \{+1, -1\}^d$ *and* $\xi \sim \mathcal{N}(0, 1)$.

This result shows a limitation of SGD training based on an average-case reduction from a cryptographic problem. The closest prior result is in [Sha18], which proved hardness results for learning with SGD on FC networks, but required preprocessing the data with a whitening transformation.

Proof idea The FC architecture and Gaussian initialization are necessary: an architecture that outputted $f_{\text{mod8}}(\boldsymbol{x})$ at initialization would trivially achieve zero loss. However, SGD on Gaussian-initialized FC networks is *sign-flip* equivariant, and this symmetry makes f_{mod8} hard to learn. If a sign-flip equivariant algorithm can learn the function $f_{\text{mod8}}(\boldsymbol{x})$ from noisy samples, then it can learn the function $f_{\text{mod8}}(\boldsymbol{x} \odot \boldsymbol{s})$ from noisy samples, where $\boldsymbol{s} \in \{+1, -1\}^d$ is an unknown sign-flip vector, and $\odot$ denotes elementwise product. However, this latter problem is hard under standard cryptographic assumptions. More details in Section 4.

2 Preliminaries

Notation Let $\mathcal{H}_d = \{+1, -1\}^d$ be the binary hypercube, and $\mathbb{S}^{d-1} = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 = 1\}$ be the unit sphere. The law of a random variable X is $\mathcal{L}(X)$. If S is a finite set, then $X \sim S$ stands for $X \sim \text{Unif}[S]$. Also let $\boldsymbol{x} \sim \mathbb{S}^{d-1}$ denote $\boldsymbol{x}$ drawn from the uniform Haar measure on $\mathbb{S}^{d-1}$. For a set Ω, let $\mathcal{P}(\Omega)$ be the set of distributions on Ω. Let $\odot$ be the elementwise product. For any $\mu_{\mathcal{X}} \in \mathcal{P}(\mathcal{X})$, and group G acting on $\mathcal{X}$, we say $\mu_{\mathcal{X}}$ is G-invariant if $g(\boldsymbol{x}) \overset{d}{=} \boldsymbol{x}$ for $\boldsymbol{x} \sim \mu_{\mathcal{X}}$ and any $g \in G$.

2.1 Equivariance of GD and SGD

We define GD and SGD equivariance separately.

Definition 2.1. *Let* $\mathcal{A}^{\text{GD}}$ *be the algorithm that takes in data distribution* $\mathcal{D} \in \mathcal{P}(\mathcal{X} \times \mathbb{R})$*, runs* (GD) *on initialization* $\boldsymbol{\theta}^0 \sim \mu_{\boldsymbol{\theta}}$ *for* k *steps, and outputs the function* $\mathcal{A}^{\text{GD}}(\mathcal{D}) = f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}^k)$

We say "$(f_{\mathsf{NN}}, \mu_{\boldsymbol{\theta}})$*-GD is* G*-equivariant" if* $\mathcal{A}^{\text{GD}}$ *is* G*-equivariant in the sense of Definition 1.1.*

Definition 2.2. *Let* $\mathcal{A}^{\text{SGD}}$ *be the algorithm that takes in samples* $(\boldsymbol{x}_i, y_i)_{i \in [n]}$*, runs* (SGD) *on initialization* $\boldsymbol{\theta}^0 \sim \mu_{\boldsymbol{\theta}}$ *for* n *steps, and outputs* $\mathcal{A}^{\text{SGD}}((\boldsymbol{x}_i, y_i)_{i \in [n]}) = f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}^k)$.

We say "$(f_{\mathsf{NN}}, \mu_{\boldsymbol{\theta}})$*-SGD is* G*-equivariant" if* $\mathcal{A}^{\text{SGD}}((\boldsymbol{x}_i, y_i)_{i \in [n]}) \overset{d}{=} \mathcal{A}^{\text{SGD}}((g(\boldsymbol{x}_i), y_i)_{i \in [n]}) \circ g$ *for any* $g \in G$*, and any samples* $(\boldsymbol{x}_i, y_i)_{i \in [n]}$.

2.2 Regularity conditions on networks imply equivariances of GD and SGD

We take a data space $\mathcal{X} \subseteq \mathbb{R}^d$, and consider the following groups that act on $\mathbb{R}^d$.

Definition 2.3. *Define the following groups and actions:*

- *Let* $G_{perm} = S_d$ *denote the group of permutations on* $[d]$*. An element* $\sigma \in G_{perm}$ *acts on* $\boldsymbol{x} \in \mathbb{R}^d$ *in the standard way:* $\sigma(\boldsymbol{x}) = (x_{\sigma(1)}, \ldots, x_{\sigma(d)})$.
- *Let* $G_{\text{sign,perm}}$ *denote the group of signed permutations, an element* $g = (\boldsymbol{s}, \sigma) \in G_{\text{sign,perm}}$ *is given by a sign-flip vector* $\boldsymbol{s} \in \mathcal{H}_d$ *and a permutation* $\sigma \in G_{perm}$*. It acts on* $\boldsymbol{x} \in \mathbb{R}^d$ *by* $g(\boldsymbol{x}) = \boldsymbol{s} \odot \sigma(\boldsymbol{x}) = (s_1 x_{\sigma(1)}, \ldots, s_d x_{\sigma(d)})$.[9]
- *Let* $G_{\text{rot}} = SO(d) \subseteq GL(d, \mathbb{R})$ *denote the rotation group. An element* $g \in G_{\text{rot}}$ *is a rotation matrix that acts on* $\boldsymbol{x} \in \mathbb{R}^d$ *by matrix multiplication.*

[9] The group product is $g_1 g_2 = (\boldsymbol{s}_1, \sigma_1)(\boldsymbol{s}_2, \sigma_2) = (\boldsymbol{s}_1 \odot \sigma_1(\boldsymbol{s}_2), \sigma_1 \circ \sigma_2)$.

Under mild conditions on the neural network architecture and initialization, GD and SGD training are known to be G_{perm}-, $G_{\text{sign,perm}}$-, or G_{rot}-equivariant [Ng04, LZA21].

Assumption 2.4 (Fully-connected i.i.d. first layer and no skip connections from the input)**.** *We can decompose the parameters as $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{\psi})$, where $\boldsymbol{W} \in \mathbb{R}^{m \times d}$ is the matrix of the first-layer weights, and there is a function $g_{\mathsf{NN}}(\cdot; \boldsymbol{\psi}) : \mathbb{R}^m \to \mathbb{R}$ such that $f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}) = g_{\mathsf{NN}}(\boldsymbol{W}\boldsymbol{x}; \boldsymbol{\psi})$. Furthermore, the initialization distribution is $\mu_{\boldsymbol{\theta}} = \mu_{\boldsymbol{W}} \times \mu_{\boldsymbol{\psi}}$, where $\mu_{\boldsymbol{W}} = \mu_w^{\otimes(m \times d)}$ for $\mu_w \in \mathcal{P}(\mathbb{R})$.*

Notice that Assumption 2.4 is satisfied by FC networks with i.i.d. initialization. Under assumptions on μ_w, we obtain equivariances of GD and SGD (see Appendix E for proofs.)

Proposition 2.5 ([Ng04, LZA21])**.** *Under Assumption 2.4, GD and SGD are G_{perm}-equivariant. If μ_w is sign-flip symmetric, then GD and SGD are $G_{\text{sign,perm}}$-equivariant. If $\mu_w = \mathcal{N}(0, \sigma^2)$ for some σ, then GD and SGD are G_{rot}-equivariant.*

3 Lower bounds for learning with GD

In this section, let $\mathcal{D}(f, \mu_{\mathcal{X}}) \in \mathcal{P}(\mathcal{X} \times \mathbb{R})$ denote the distribution of $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x} \sim \mu_{\mathcal{X}}$.

We give a master theorem for computational lower bounds for learning with G-equivariant GD.

Theorem 3.1 (GD lower bound using G-alignment)**.** *Let G be a compact group, and let $f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}) : \mathcal{X} \to \mathbb{R}$ be an architecture and $\mu_{\boldsymbol{\theta}} \in \mathcal{P}(\mathbb{R}^p)$ be an initialization such that GD is G-equivariant.*

Fix any G-invariant distribution $\mu_{\mathcal{X}} \in \mathcal{P}(\mathcal{X})$, any label function $f_* \in L^2(\mu_{\mathcal{X}})$, and any baseline function $\alpha \in L^2(\mu_{\mathcal{X}})$ satisfying $\alpha \circ g = \alpha$ for all $g \in G$. Let $\boldsymbol{\theta}^k$ be the random weights after k time-steps of GD training with noise parameter $\tau > 0$, step size $\eta > 0$, and clipping radius $R > 0$ on the distribution $\mathcal{D} = \mathcal{D}(f_*, \mu_{\mathcal{X}})$. Then, for any $\epsilon > 0$,

$$\mathbb{P}_{\boldsymbol{\theta}^k}[\ell_{\mathcal{D}}(\boldsymbol{\theta}^k) \leq \|f_* - \alpha\|^2_{L^2(\mu_{\mathcal{X}})} - \epsilon] \leq \frac{\eta R \sqrt{k\mathcal{C}}}{2\tau} + \frac{\mathcal{C}}{\epsilon},$$

where $\mathcal{C} = \mathcal{C}((f_* - \alpha, \mu_{\mathcal{X}}); G)$ is the G-alignment of Definition 1.3.

As discussed in Section 1.2, the theorem states that if the G-alignment $\mathcal{C}$ is very small, then GD training cannot efficiently improve on the trivial loss from outputting α: either the number of steps k, the gradient precision R/τ, or the step size η have to be very large in order to learn. Appendix A shows a generalization of the theorem for learning a class of functions $\mathcal{F} = \{f_1, \ldots, f_m\}$ instead of just a single function f_*. This result goes beyond the lower bound of [AS20] even when G is the trivial group with one element: the main improvement is that Theorem 3.1 proves hardness for learning real-valued functions beyond just Boolean-valued functions. We demonstrate the usefulness of the theorem through two new applications in Sections 3.1 and 3.2.

3.1 Application: Characterizing weak-learnability by FC networks

In our first application of Theorem 3.1, we consider FC architectures with i.i.d. initialization, and show how to use their training equivariances to characterize what functions they can weak-learn: i.e., for what target functions f_* they can efficiently achieve a non-negligible correlation after training.

Definition 3.2 (Weak learnability)**.** *Let $\{\mu_d\}_{d \in \mathbb{N}}$ be a family of distributions $\mu_d \in \mathcal{P}(\mathcal{X}_d)$, and let $\{f_d\}_{d \in \mathbb{N}}$ be a family of functions $f_d \in L^2(\mu_d)$. Finally, let $\{\tilde{f}_d\}_{d \in \mathbb{N}}$ be a family of estimators, where $\tilde{f}_d$ is a random function in $L^2(\mu_d)$. We say that $\{f_d, \mu_d\}_{d \in \mathbb{N}}$ is "weak-learned" by the family of estimators $\{\tilde{f}_d\}_{d \in \mathbb{N}}$ if there are constants $d_0, C > 0$ such that for all $d > d_0$,*

$$\mathbb{P}_{\tilde{f}_d}[\|f_d - \tilde{f}_d\|^2_{L^2(\mu_d)} \leq \|f_d\|^2_{L^2(\mu_d)} - d^{-C}] \geq 9/10. \tag{2}$$

The constant $9/10$ in the definition is arbitrary. In words, weak-learning measures whether the family of estimators $\{\tilde{f}_d\}$ has a non-negligible edge over simply estimating with the identically zero functions $\tilde{f}_d \equiv 0$. We study weak-learnability by GD-trained FC networks.

Definition 3.3. *We say that $\{f_d, \mu_d\}_{d \in \mathbb{N}}$ is efficiently weak-learnable by GD-trained FC networks if there are FC networks and initializations $\{f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d}\}$, and hyperparameters $\{\eta_d, k_d, R_d, \tau_d\}$ such that for some constant $c > 0$,*

- *Hyperparameters are polynomial size:* $0 \le \eta_d, k_d, R_d, 1/\tau_d \le O(d^c)$;
- $\{\tilde{f}_d\}$ *weak-learns* $\{f_d, \mu_d\}$ *in the sense of Definition 3.2, where* $\tilde{f}_d = f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}_d)$ *for weights* $\boldsymbol{\theta}_d$ *that are GD-trained on* $\mathcal{D}(f_d, \mu_d)$ *for* k_d *steps with step size* η_d*, clipping radius* R_d*, and noise* τ_d*, starting from initialization* $\mu_{\boldsymbol{\theta},d}$.

If $\mu_{\boldsymbol{\theta},d}$ *is i.i.d copies of a symmetric distribution, we say that the FC networks are symmetrically-initialized, and Gaussian-initialized if* $\mu_{\boldsymbol{\theta},d}$ *is i.i.d. copies of a Gaussian distribution.*

3.1.1 Functions on hypercube, FC networks with i.i.d. symmetric initialization

Let us first consider functions on the Boolean hypercube $f : \mathcal{H}_d \to \mathbb{R}$. These can be uniquely written as a multilinear polynomial

$$f(\boldsymbol{x}) = \sum_{S \subseteq [d]} \hat{f}(S) \prod_{i \in S} x_i,$$

where $\hat{f}(S)$ are the Fourier coefficients of f [O'D14]. We characterize weak learnability of functions on the hypercube in terms of their Fourier coefficients. The full proof is deferred to Appendix B.1.

Theorem 3.4. *Let* $\{f_d\}_{d \in \mathbb{N}}$ *be a family of functions* $f_d : \mathcal{H}_d \to \mathbb{R}$ *with* $\|f_d\|_{L^2(\mathcal{H}_d)} \le 1$*. Then* $\{f_d, \mathcal{H}_d\}$ *is efficiently weak-learnable by GD-trained symmetrically-initialized FC networks if and only if there is a constant* $C > 0$ *such that for each* $d \in \mathbb{N}$ *there is* $S_d \subseteq [d]$ *with* $|S_d| \le C$ *or* $|S_d| \ge d - C$*, and* $|\hat{f}_d(S_d)| \ge \Omega(d^{-C})$.

The algorithmic result can be achieved by two-layer FC networks, and relies on random features analysis where each network weight is initialized to 0 with probability $1 - p$, and $+1$ or -1 with equal probability $p/2$.[10] Therefore, for weak learning on the hypercube, two-layer networks are as good as networks of any depth. For the converse impossibility result, we apply Theorem 3.1, recalling that GD is $G_{\text{sign,perm}}$-equivariant by Proposition 2.5, and noting that $G_{\text{sign,perm}}$-alignment is:

Lemma 3.5. *Let* $f : \mathcal{H}_d \to \mathbb{R}$*. Then* $\mathcal{C}((f, \mathcal{H}_d); G_{\text{sign,perm}}) = \max_{k \in [d]} \binom{d}{k}^{-1} \sum_{\substack{S \subseteq [d] \\ |S| = k}} \hat{f}(S)^2$.

Proof. In the following, let $\boldsymbol{s} \sim \mathcal{H}_d$ and $\sigma \sim G_{perm}$, so that $g = (\boldsymbol{s}, \sigma) \sim G_{\text{sign,perm}}$. Also let $\boldsymbol{x}, \boldsymbol{x}' \sim \mathcal{H}_d$ be independent. For any $h : \mathcal{H}_d \to \mathbb{R}$, by (a) tensorizing, (b) expanding f in the Fourier basis, (c) the orthogonality relation $\mathbb{E}_{\boldsymbol{s}}[\chi_S(\boldsymbol{s})\chi_{S'}(\boldsymbol{s})] = \delta_{S,S'}$, and (d) tensorizing,

$$\begin{aligned}
\mathbb{E}_g[\mathbb{E}_{\boldsymbol{x}}[f(g(\boldsymbol{x}))h(\boldsymbol{x})]^2] &= \mathbb{E}_{\sigma,\boldsymbol{s}}[\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{s} \odot \sigma(\boldsymbol{x}))h(\boldsymbol{x})]^2] \\
&\overset{(a)}{=} \mathbb{E}_{\sigma,\boldsymbol{s},\boldsymbol{x},\boldsymbol{x}'}[f(\boldsymbol{s} \odot \sigma(\boldsymbol{x}))f(\boldsymbol{s} \odot \sigma(\boldsymbol{x}'))h(\boldsymbol{x})h(\boldsymbol{x}')] \\
&\overset{(b)}{=} \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}',\sigma}[\sum_{S,S' \subseteq [d]} \hat{f}(S)\hat{f}(S')h(\boldsymbol{x})h(\boldsymbol{x}')\chi_S(\sigma(\boldsymbol{x}))\chi_{S'}(\sigma(\boldsymbol{x}'))\,\mathbb{E}_{\boldsymbol{s}}[\chi_S(\boldsymbol{s})\chi_{S'}(\boldsymbol{s})]] \\
&\overset{(c)}{=} \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}',\sigma}[\sum_{S \subseteq [d]} \hat{f}(S)^2 h(\boldsymbol{x})h(\boldsymbol{x}')\chi_S(\sigma(\boldsymbol{x}))\chi_S(\sigma(\boldsymbol{x}'))] \\
&\overset{(d)}{=} \mathbb{E}_{\sigma}[\sum_{S \subseteq [d]} \hat{f}(S)^2\,\mathbb{E}_{\boldsymbol{x}}[h(\boldsymbol{x})\chi_S(\sigma(\boldsymbol{x}))]^2] \\
&= \sum_{S \subseteq [d]} \hat{f}(S)^2\,\mathbb{E}_{\sigma}[\hat{h}(\sigma^{-1}(S))^2] \\
&= \sum_{S \subseteq [d]} \hat{f}(S)^2 \binom{d}{|S|}^{-1} \sum_{S', |S'| = |S|} \hat{h}(S')^2.
\end{aligned}$$

And since $\sum_{S', |S'| = |S|} \hat{h}(S')^2 \le \|h\|^2_{L^2(\mathcal{H}_d)}$, the supremum over h such that $\|h\|_{L^2(\mathcal{H}_d)} = 1$ is achieved by taking $h(\boldsymbol{x}) = \chi_S(\boldsymbol{x})$ for some S. □

[10] Surprisingly, this means that the full parity function $f_*(\boldsymbol{x}) = \prod_{i=1}^d x_i$ can be efficiently learned with such initializations. See Appendix B.

So if the Fourier coefficients of f are negligible for all S s.t. $\min(|S|, d-|S|) \leq O(1)$, then the $G_{\text{sign,perm}}$-alignment of f is negligible. By Theorem 3.1, this means f cannot be learned efficiently. In Appendix B.1.2 we give a concrete example of a hard function, that was not previously known.

3.1.2 Functions on sphere, FC networks with i.i.d. Gaussian initialization

We now study learning a target function on the unit sphere, $f \in L^2(\mathbb{S}^{d-1})$, where we take the standard Lebesgue measure on $\mathbb{S}^{d-1}$. A key fact in harmonic analysis is that $L^2(\mathbb{S}^{d-1})$ can be written as the direct sum of subspaces spanned by spherical harmonics of each degree (see, e.g., [Hoc12]).

$$L^2(\mathbb{S}^{d-1}) = \bigoplus_{l=0}^{\infty} \mathcal{V}_{d,l},$$

where $\mathcal{V}_{d,l} \subseteq L^2(\mathbb{S}^{d-1})$ is the space of degree-l spherical harmonics, which is of dimension

$$\dim(\mathcal{V}_{d,l}) = \frac{2l+d-2}{l}\binom{l+d-3}{l-1}.$$

Let $\Pi_{\mathcal{V}_{d,l}} : L^2(\mathbb{S}^{d-1}) \to \mathcal{V}_{d,l}$ be the projection operator to the space of degree-l spherical harmonics. In Appendix B.2, we prove this characterization of weak-learnability for functions on the sphere:

Theorem 3.6. *Let $\{f_d\}_{d\in\mathbb{N}}$ be a family of functions $f_d : \mathbb{S}^{d-1} \to \mathbb{R}$ with $\|f_d\|_{L^2(\mathbb{S}^{d-1})} \leq 1$. Then $\{f_d, \mathbb{S}^{d-1}\}$ is efficiently weak-learnable by GD-trained Gaussian-initialized FC networks if and only if there is a constant $C > 0$ such that $\sum_{l=0}^{C} \|\Pi_{\mathcal{V}_{d,l}} f_d\|^2 \geq d^{-C}$.*

The algorithmic result can again be achieved by two-layer FC networks, and is a consequence of the analysis of the random feature kernel in [GMMM21], which shows that the projection of f_d onto the low-degree spherical harmonics can be efficiently learned. For the impossibility result, we apply Theorem 3.1, noting that GD is G_{rot}-equivariant by Proposition 2.5, and the G_{rot}-alignment is:

Lemma 3.7. *Let $f \in L^2(\mathbb{S}^{d-1})$. Then $\mathcal{C}((f, \mathbb{S}^{d-1}); G_{\text{rot}}) = \max_{l\in\mathbb{Z}_{\geq 0}} \|\Pi_{\mathcal{V}_{d,l}} f\|^2 / \dim(\mathcal{V}_{d,l})$.*

Proof. The G_{rot}-alignment is computed using the representation theory of G_{rot}, specifically the Schur orthogonality theorem (see, e.g., [Ser77, Kna96]). For any l, the subspace $\mathcal{V}_{d,l}$ is invariant to action by G_{rot}, meaning that we may define the representation Φ_l of G_{rot}, which for any $g \in G_{\text{rot}}, f \in \mathcal{V}_{d,l}$ is given by $\Phi_l(g) : \mathcal{V}_{d,l} \to \mathcal{V}_{d,l}$ and $\Phi_l(g)f = f \circ g^{-1}$. Furthermore, Φ_l is a unitary, irreducible representation, and Φ_l is not equivalent to $\Phi_{l'}$, for any $l \neq l'$ (see e.g., [Sta90, Theorem 1]). Therefore, by the Schur orthogonality relations [Kna96, Corollary 4.10], for any $v_1, w_1 \in \mathcal{V}_{d,l_1}$ and $v_2, w_2 \in \mathcal{V}_{d,l_2}$, we have

$$\begin{aligned}\mathbb{E}_{g\sim G_{\text{rot}}}[\langle \phi_{l_1}(g)v_1, w_1\rangle_{L^2(\mathbb{S}^{d-1})} \langle \phi_{l_2}(g)v_2, w_2\rangle_{L^2(\mathbb{S}^{d-1})}] \\ = \delta_{l_1 l_2} \langle v_1, v_2\rangle_{L^2(\mathbb{S}^{d-1})} \langle w_1, w_2\rangle_{L^2(\mathbb{S}^{d-1})} / \dim(\mathcal{V}_{d,l_1}). \qquad (3)\end{aligned}$$

Let $g \sim G_{\text{rot}}$, drawn from the Haar probability measure. For any $h \in L^2(\mathbb{S}^{d-1})$ such that $\|h\|^2_{L^2(\mathbb{S}^{d-1})} = 1$, by (a) the decomposition of $L^2(\mathbb{S}^{d-1})$ into subspaces of spherical harmonics, (b) the G_{rot}-invariance of each subspace $\mathcal{V}_{d,l}$, and (c) the Schur orthogonality relations in (3),

$$\begin{aligned}\mathbb{E}_g[\langle f\circ g, h\rangle^2_{L^2(\mathbb{S}^{d-1})}] &\overset{(a)}{=} \sum_{l_1,l_2=0}^{\infty} \mathbb{E}_g[\langle \Pi_{\mathcal{V}_{d,l_1}}(f\circ g), \Pi_{\mathcal{V}_{d,l_1}} h\rangle_{L^2(\mathbb{S}^{d-1})} \langle \Pi_{\mathcal{V}_{d,l_2}}(f\circ g), \Pi_{\mathcal{V}_{d,l_2}} h\rangle_{L^2(\mathbb{S}^{d-1})}] \\ &\overset{(b)}{=} \sum_{l_1,l_2=0}^{\infty} \mathbb{E}_g[\langle (\Pi_{\mathcal{V}_{d,l_1}} f)\circ g, \Pi_{\mathcal{V}_{d,l_1}} h\rangle_{L^2(\mathbb{S}^{d-1})} \langle (\Pi_{\mathcal{V}_{d,l_2}} f)\circ g, \Pi_{\mathcal{V}_{d,l_2}} h\rangle_{L^2(\mathbb{S}^{d-1})}] \\ &\overset{(c)}{=} \sum_{l=0}^{\infty} \frac{1}{\dim(\mathcal{V}_{d,l})} \|\Pi_{\mathcal{V}_{d,l}} f\|^2_{L^2(\mathbb{S}^{d-1})} \|\Pi_{\mathcal{V}_{d,l}} h\|^2_{L^2(\mathbb{S}^{d-1})} \\ &\leq \left(\sum_{l=0}^{\infty} \|\Pi_{\mathcal{V}_{d,l}} h\|^2_{L^2(\mathbb{S}^{d-1})}\right) \max_{l\in\mathbb{Z}_{\geq 0}} \frac{1}{\dim(\mathcal{V}_{d,l})} \|\Pi_{\mathcal{V}_{d,l}} f\|^2_{L^2(\mathbb{S}^{d-1})} \\ &= \max_{l\in\mathbb{Z}_{\geq 0}} \frac{1}{\dim(\mathcal{V}_{d,l})} \|\Pi_{\mathcal{V}_{d,l}} f\|^2_{L^2(\mathbb{S}^{d-1})}.\end{aligned}$$

Let l^* be the optimal value of l in the last line, which is known to exist by the fact that $\|\Pi_{\mathcal{V}_{d,l}} f\|^2 \leq \|f\|^2$ and $\dim(\mathcal{V}_{d,l}) \to \infty$ as $l \to \infty$. The inequality is achieved by $h = \Pi_{\mathcal{V}_{d,l^*}} f / \|\Pi_{\mathcal{V}_{d,l^*}} f\|$. $\square$

This implies that the G_{rot}-alignment of f is negligible if and only if its projection to the low-order spherical harmonics is negligible. By Theorem 3.1, this implies the necessity result of Theorem 3.6.

3.2 Application: Extending the merged-staircase property necessity result

In our second application, we study the setting of learning a sparse function on the binary hypercube (a.k.a. a junta) that depends on only $P \leq d$ coordinates of the input $\boldsymbol{x}$, i.e.,

$$f_*(\boldsymbol{x}) = h_*(x_1, \ldots, x_P),$$

where $h_* : \mathcal{H}_P \to \mathbb{R}$. The regime of interest to us is when h_* is fixed and $d \to \infty$, representing a hidden signal in a high-dimensional dataset. This setting was studied by [ABM22], who identified the "merged-staircase property" (MSP) as an extension of [ABB+21]. We generalize the MSP below.

Definition 3.8 (l-MSP)**.** *For $l \in \mathbb{Z}_+$ and $h_* : \mathcal{H}_P \to \mathbb{R}$, we say that h_* satisfies the merged staircase property with leap l (i.e., l-MSP) if its set of nonzero Fourier coefficients $\mathcal{S} = \{S : \hat{h}_*(S) \neq \emptyset\}$ can be ordered as $\mathcal{S} = \{S_1, \ldots, S_m\}$ such that for all $i \in [m]$, $|S_i \setminus \cup_{j<i} S_j| \leq l$.*

For example, $h_*(\boldsymbol{x}) = x_1 + x_1x_2 + x_1x_2x_3$ satisfies 1-MSP; $h_*(\boldsymbol{x}) = x_1x_2 + x_1x_2x_3$ satisfies 2-MSP, but not 1-MSP because of the leap required to learn x_1x_2; similarly $h_*(\boldsymbol{x}) = x_1x_2x_3 + x_4$ satisfies 3-MSP but not 2-MSP. If h_* satisfies l-MSP for some small l, then the function f_* can be learned greedily in an efficient manner, by iteratively discovering the coordinates on which it depends. In [ABM22] it was proved that the 1-MSP property nearly characterized which sparse functions could be ϵ-learned in $O_{\epsilon,h_*}(d)$ samples by one-pass SGD training in the mean-field regime.

We prove the MSP necessity result for GD training. On the one hand, our necessity result is for a different training algorithm, GD, which injects noise during training. On the other, our result is much more general since it applies whenever GD is permutation-equivariant, which includes training of FC networks and ResNets of any depth (whereas the necessity result of [ABM22] applies only to two-layer architectures in the mean-field regime). We also generalize the result to any leap l.

Theorem 3.9 (l-MSP necessity)**.** *Let $f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}) : \mathcal{H}_d \to \mathbb{R}$ be an architecture and $\mu_{\boldsymbol{\theta}} \in \mathcal{P}(\mathbb{R}^p)$ be an initialization such that GD is G_{perm}-equivariant. Let $\boldsymbol{\theta}^k$ be the random weights after k steps of GD training with noise parameter $\tau > 0$, step size η, and clipping radius R on the distribution $\mathcal{D} = \mathcal{D}(f_*, \mathcal{H}_d)$. Suppose that $f_*(\boldsymbol{x}) = h_*(\boldsymbol{z})$ where $h_* : \mathcal{H}_P \to \mathbb{R}$ does not satisfy l-MSP for some $l \in \mathbb{Z}_+$. Then there are constants $C, \epsilon_0 > 0$ depending on h_* such that*

$$\mathbb{P}_{\boldsymbol{\theta}^k}[\ell_{\mathcal{D}}(\boldsymbol{\theta}^k) \leq \epsilon_0] \leq \frac{C\eta R}{2\tau}\sqrt{\frac{k}{d^{l+1}}} + \frac{C}{d^{l+1}}.$$

The interpretation is that if h_* does not satisfy l-MSP, then to learn f_* to better than ϵ_0 error with constant probability, we need at least $\Omega_{h_*,\epsilon}(d^{l+1})$ steps of (GD) on a network with step size $\eta = O_{h_*,\epsilon}(1)$, clipping radius $R = O_{h_*,\epsilon}(1)$, and noise level $\tau = \Omega_{h_*,\epsilon}(1)$. The proof is deferred to Appendix C. It proceeds by first isolating the "easily-reachable" coordinates $T \subseteq [P]$, and subtracting their contribution from f_*. We then bound G-alignment of the resulting function, where G is the permutation group on $[d] \setminus T$.

4 Hardness for learning with SGD

In this section, for $\gamma > 0$, we let $\mathcal{D}(f, \mu_{\mathcal{X}}, \gamma) \in \mathcal{P}(\mathcal{X} \times \mathbb{R})$ denote the distribution of $(\boldsymbol{x}, f(\boldsymbol{x}) + \xi)$ where $\boldsymbol{x} \sim \mu_{\mathcal{X}}$ and $\xi \sim \mathcal{N}(0, \gamma^2)$ is independent noise.

We show that the equivariance of SGD on certain architectures implies that the function $f_{\text{mod8}} : \mathcal{H}_d \to \{0, \ldots, 7\}$ given by

$$f_{\text{mod8}}(\boldsymbol{x}) \equiv \sum_i x_i \pmod 8 \tag{4}$$

is hard for SGD-trained, i.i.d. symmetrically-initialized FC networks. Our hardness result relies on a cryptographic assumption to prove superpolynomial lower bounds for SGD learning. For any $S \subseteq [d]$, let $\chi_S : \mathcal{H}_d \to \{+1, -1\}$ be the parity function $\chi_S(\boldsymbol{x}) = \prod_{i \in S} x_i$.

Definition 4.1. *The learning parities with Gaussian noise, (d, n, γ)-LPGN, problem is parametrized by $d, n \in \mathbb{Z}_{>0}$ and $\gamma \in \mathbb{R}_{>0}$. An instance $(S, \boldsymbol{q}, (\boldsymbol{x}_i, y_i)_{i\in[n]})$ consists of (i) an unknown subset $S \subseteq [d]$ of size $|S| = \lfloor d/2 \rfloor$, and (ii) a known query vector $\boldsymbol{q} \sim \mathcal{H}_d$, and i.i.d. samples $(\boldsymbol{x}_i, y_i)_{i\in[n]} \sim \mathcal{D}(\chi_S, \mathcal{H}_d, \gamma)$. The task is to return $\chi_S(\boldsymbol{q}) \in \{+1, -1\}$.*[11]

Our cryptographic assumption is that $\mathrm{poly}(d)$-size circuits cannot succeed on LPGN.

Definition 4.2. *Let $\gamma > 0$. We say γ-LPGN is $\mathrm{poly}(d)$-time solvable if there is a sequence of sample sizes $\{n_d\}_{d\in\mathbb{N}}$ and circuits $\{\mathcal{A}_d\}_{d\in\mathbb{N}}$ such that $n_d, \mathrm{size}(\mathcal{A}_d) \leq \mathrm{poly}(d)$, and $\mathcal{A}_d$ solves (d, n_d, γ)-LPGN with success probability at least $9/10$, when inputs are rounded to $\mathrm{poly}(d)$ bits.*

Assumption 4.3. *Fix γ. The γ-LPGN-hardness assumption is: γ-LPGN is not $\mathrm{poly}(d)$-time solvable.*

The LPGN problem is the simply standard Learning Parities with Noise problem (LPN) [BKW03], except with Gaussian noise instead of binary classification noise, and we are also promised that $|S| = \lfloor d/2 \rfloor$. In Appendix D.3, we derive Assumption 4.3 from the standard hardness of LPN. We now state our SGD hardness result.

Theorem 4.4. *Let $\{f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d}\}_{d\in\mathbb{N}}$ be a family of networks and initializations satisfying Assumption 2.4 (fully-connected) with i.i.d. symmetric initialization. Let $\gamma > 0$, and let $\{n_d\}$ be sample sizes such that $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-SGD training on n_d samples from $\mathcal{D}(f_{\mathrm{mod8}}, \mathcal{H}_d, \gamma)$ rounded to $\mathrm{poly}(d)$ bits yields parameters $\boldsymbol{\theta}_d$ with*

$$\mathbb{E}_{\boldsymbol{\theta}_d}[\|f_{\mathrm{mod8}} - f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}_d)\|^2] \leq 0.0001.$$

Then, under $(\gamma/2)$-LPGN hardness, $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-SGD on n_d samples cannot run in $\mathrm{poly}(d)$ time.

In order to prove Theorem 4.4, we use the sign-flip equivariance of gradient descent guaranteed by the symmetry in the initialization. A sign-flip equivariant network that learns $f_{\mathrm{mod8}}(\boldsymbol{x})$ from γ-noisy samples, is capable of solving the harder problem of learning $f_{\mathrm{mod8}}(\boldsymbol{x} \odot \boldsymbol{s})$ from γ-noisy samples, where $\boldsymbol{s} \in \mathcal{H}_d$ is an unknown sign-flip vector. However, through an average-case reduction we show that this problem is $(\gamma/2)$-LPGN-hard. Therefore the theorem follows by contradiction.

5 Discussion

The general GD lower bound in Theorem 3.1 and the approach for basing hardness of SGD training on cryptographic assumptions in Theorem 4.4 could be further developed to other settings.

There are limitations of the results to address in future work. First, the GD lower bound requires adding noise to the gradients, which can hinder training. Second, real-world data distributions are typically not invariant to a group of transformations, so the results obtained by this work may not apply. It is open to develop results for distributions that are approximately invariant.

Finally, it is open whether computational lower bounds for SGD/GD training can be shown beyond those implied by equivariance. For example, consider the function $f : \mathcal{H}_d \to \{+1, -1\}$ that computes the "full parity", i.e., the parity of all of the inputs $f(\boldsymbol{x}) = \prod_{i=1}^d x_i$. Past work has empirically shown that SGD on FC networks with Gaussian initialization [SSS17, AS20, NY21] fails to learn this function. Proving this would represent a significant advance, since there is no obvious equivariance that implies that the full parity is hard to learn — in fact we have shown weak-learnability with symmetric $\mathrm{Rad}(1/2)$ initialization, in which case training is $G_{\mathrm{sign,perm}}$-equivariant.

Acknowledgements

We thank Jason Altschuler, Guy Bresler, Elisabetta Cornacchia, Sonia Hashim, Jan Hazla, Hannah Lawrence, Theodor Misiakiewicz, Dheeraj Nagaraj, and Philippe Rigollet for stimulating discussions. We thank the Simons Foundation and the NSF for supporting us through the Collaboration on the Theoretical Foundations of Deep Learning (deepfoundations.ai). This work was done in part while E.B. was visiting the Simons Institute for the Theory of Computing and the Bernoulli Center at EPFL, and was generously supported by Apple with an AI/ML fellowship.

[11] More formally, one would express this as a probabilistic promise problem [Ale03].

References

[ABB+21] E. Abbe, E. Boix-Adsera, M. S. Brennan, G. Bresler, and D. Nagaraj. The staircase property: How hierarchical structure can guide deep learning. *Advances in Neural Information Processing Systems*, 34:26989–27002, 2021.

[ABM22] 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.

[ACHM22] E. Abbe, E. Cornacchia, J. Hązła, and C. Marquis. An initial alignment between neural network and target is needed for gradient descent to learn. *arXiv preprint arXiv:2202.12846*, 2022.

[AKM+21] E. Abbe, P. Kamath, E. Malach, C. Sandon, and N. Srebro. On the power of differentiable learning versus PAC and SQ learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[AL19] Z. Allen-Zhu and Y. Li. What can resnet learn efficiently, going beyond kernels? *Advances in Neural Information Processing Systems*, 32, 2019.

[AL20] Z. Allen-Zhu and Y. Li. Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413*, 2020.

[Ale03] M. Alekhnovich. More on average case vs approximation complexity. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 298–307. IEEE, 2003.

[AS20] E. Abbe and C. Sandon. Poly-time universality and limitations of deep learning. *arXiv preprint arXiv:2001.02992*, 2020.

[BAO+20] A. Bogatskiy, B. Anderson, J. Offermann, M. Roussi, D. Miller, and R. Kondor. Lorentz group equivariant neural network for particle physics. In *International Conference on Machine Learning*, pages 992–1002. PMLR, 2020.

[Bar93] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[BBCV21] M. M. Bronstein, J. Bruna, T. Cohen, and P. Velicković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[BBH18] M. Brennan, G. Bresler, and W. Huleihel. Reducibility and computational lower bounds for problems with planted sparse structure. In *Conference On Learning Theory*, pages 48–166. PMLR, 2018.

[BCB14] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[BF02] N. H. Bshouty and V. Feldman. On using extended statistical queries to avoid membership queries. *Journal of Machine Learning Research*, 2(Feb):359–395, 2002.

[BIK90] S. Ben-David, A. Itai, and E. Kushilevitz. Learning by distances. In *COLT*, pages 232–245, 1990.

[BKW03] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

[Boi20] E. Boix-Adsera. Average-case statistical query algorithms, 2020. Report for MDS Lab Internship.

[BVB21] A. Bietti, L. Venturi, and J. Bruna. On the sample complexity of learning under geometric stability. *Advances in Neural Information Processing Systems*, 34, 2021.

[BZSL13] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[Ele21] B. Elesedy. Provably strict generalisation benefit for invariance in kernel methods. *Advances in Neural Information Processing Systems*, 34, 2021.

[Ele22] B. Elesedy. Group symmetry in pac learning. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022.

[EZ21] B. Elesedy and S. Zaidi. Provably strict generalisation benefit for equivariant models. In *International Conference on Machine Learning*, pages 2959–2969. PMLR, 2021.

[GMMM21] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029–1054, 2021.

[GMS05] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks*, volume 2, pages 729–734, 2005.

[Hoc12] H. Hochstadt. *The functions of mathematical physics*. Courier Corporation, 2012.

[HSSV21] D. Hsu, C. Sanford, R. A. Servedio, and E. Vlatakis-Gkaragkounis. On the approximation power of two-layer networks of random relus. *arXiv preprint arXiv:2102.02336*, 2021.

[Hsu] D. Hsu. Dimension lower bounds for linear approaches to function approximation.

[HZRS16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[JGH18] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[Kea98] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[KMS20] P. Kamath, O. Montasser, and N. Srebro. Approximate is good enough: Probabilistic variants of dimensional and margin complexity. In *Conference on Learning Theory*, pages 2236–2262. PMLR, 2020.

[Kna96] A. W. Knapp. *Lie groups beyond an introduction*, volume 140. Springer, 1996.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[LB+95] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[LCC+22] Y. Li, D. Choi, J. Chung, N. Kushman, J. Schrittwieser, R. Leblond, T. Eccles, J. Keeling, F. Gimeno, A. D. Lago, et al. Competition-level code generation with alphacode. *arXiv preprint arXiv:2203.07814*, 2022.

[LKF10] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.

[LZA21] Z. Li, Y. Zhang, and S. Arora. Why are convolutional nets more sample-efficient than fully-connected nets? In *International Conference on Learning Representations*, 2021.

[Mis22] T. Misiakiewicz. Spectrum of inner-product kernel matrices in the polynomial regime and multiple descent phenomenon in kernel ridge regression. *arXiv preprint arXiv:2204.10425*, 2022.

[MM21] T. Misiakiewicz and S. Mei. Learning with convolution and pooling operations in kernel methods. *arXiv preprint arXiv:2111.08308*, 2021.

[MMM21] S. Mei, T. Misiakiewicz, and A. Montanari. Learning with invariances in random features and kernel models. In *Conference on Learning Theory*, pages 3351–3418. PMLR, 2021.

[MS20] E. Malach and S. Shalev-Shwartz. Computational separation between convolutional and fully-connected networks. *arXiv preprint arXiv:2010.01369*, 2020.

[Ng04] A. Y. Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78, 2004.

[NY21] I. Nachum and A. Yehudayoff. On symmetry and initialization for neural networks. In *Latin American Symposium on Theoretical Informatics*, pages 401–412. Springer, 2021.

[O'D14] R. O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[Rey20] L. Reyzin. Statistical queries and statistical algorithms: Foundations and applications. *arXiv preprint arXiv:2004.00557*, 2020.

[Ser77] J.-P. Serre. *Linear representations of finite groups*, volume 42. Springer, 1977.

[Sha18] O. Shamir. Distribution-specific hardness of learning neural networks. *The Journal of Machine Learning Research*, 19(1):1135–1163, 2018.

[SHS+18] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[SNPP19] S. Schubert, P. Neubert, J. Pöschmann, and P. Protzel. Circular convolutional neural networks for panoramic images and laser data. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 653–660. IEEE, 2019.

[SSS17] S. Shalev-Shwartz, O. Shamir, and S. Shammah. Failures of gradient-based deep learning. In *International Conference on Machine Learning*, pages 3067–3075. PMLR, 2017.

[Sta90] D. Stanton. An introduction to group representations and orthogonal polynomials. In *Orthogonal Polynomials*, pages 419–433. Springer, 1990.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[VCC+17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *stat*, 1050:20, 2017.

[VSP+17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[WLLM19] C. Wei, J. D. Lee, Q. Liu, and T. Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Advances in Neural Information Processing Systems*, 32, 2019.

[ZHSJ19] J. Zhang, T. He, S. Sra, and A. Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (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]
   (b) Did you include complete proofs of all theoretical results? [Yes]
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)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(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)? [N/A]

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]

Contents

1 Introduction **1**

1.1 Related work 2

1.2 Contribution 1: Lower bounds for noisy gradient descent (GD) 3

1.3 Contribution 2: Hardness for stochastic gradient descent (SGD) 4

2 Preliminaries **5**

2.1 Equivariance of GD and SGD 5

2.2 Regularity conditions on networks imply equivariances of GD and SGD 5

3 Lower bounds for learning with GD **6**

3.1 Application: Characterizing weak-learnability by FC networks 6

3.2 Application: Extending the merged-staircase property necessity result 9

4 Hardness for learning with SGD **9**

5 Discussion **10**

A Lower bound for GD, Proof of Theorem 3.1 and generalization **15**

A.1 Proof of Theorem A.2 16

A.2 Proof of Theorem A.4 17

A.3 Remark: relation to bound based on cross-predictability, and efficiently verifying G-alignment is small 19

A.4 Remark: alternative proof using the statistical query lens 20

B Characterization of weak-learnability by GD **21**

B.1 Functions on Boolean hypercube: proof of Theorem 3.4 21

B.2 Functions on unit sphere: proof of Theorem 3.6 26

C Extension of MSP necessity result, proof of Theorem 3.9 **28**

D Hardness results for SGD **29**

D.1 Warm-up: hardness from permutation equivariance 29

D.2 Hardness from sign-flip equivariance, proof of Theorem 4.4 30

D.3 On the cryptographic assumption that LPGN is hard 32

E On the equivariance of SGD and GD **34**

A Lower bound for GD, Proof of Theorem 3.1 and generalization

We prove a generalization of Theorem 3.1, which applies to learning a random function $f_* \sim \mu_{\mathcal{F}}$, where $\mu_{\mathcal{F}}$ is a distribution over functions in $L^2(\mu_{\mathcal{X}})$. In order to state the theorem, let us first define the G-alignment for a distribution over functions:

Definition A.1. *Let G be a compact group that acts on $\mathcal{X}$, and let $\mu_{\mathcal{X}} \in \mathcal{P}(\mathcal{X})$ be a G-invariant distribution. For any distribution of functions $\mu_{\mathcal{F}} \in \mathcal{P}(L^2(\mu_{\mathcal{X}}))$, we define the G-alignment as:*

$$\mathcal{C}((\mu_{\mathcal{F}}, \mu_{\mathcal{X}}); G) = \sup_h \mathbb{E}_{g\sim\mu_G}[\mathbb{E}_{f\sim\mu_{\mathcal{F}}}[\langle f \circ g, h\rangle^2_{L^2(\mu_{\mathcal{X}})}]],$$

where μ_G is the Haar probability measure over G, and the supremum is over $h \in L^2(\mu_{\mathcal{X}})$ such that $\|h\|^2 = 1$.

This is a generalization of the G-alignment of Definition 1.3, since we can take $\mu_{\mathcal{F}}$ to be the probability distribution that has all mass on a deterministic function f_*. We use it to prove the following generalization of Theorem 3.1:

Theorem A.2 (GD lower bound for distribution of functions, using G-alignment)**.** *Let G be a compact group that acts on $\mathcal{X}$, and let $f_{\mathsf{NN}}(\cdot;\boldsymbol{\theta}) : \mathcal{X} \to \mathbb{R}$ be an architecture and $\mu_{\boldsymbol{\theta}} \in \mathcal{P}(\mathbb{R}^p)$ be an initialization distribution, such that GD is G-equivariant.*

Fix any G-invariant distribution $\mu_{\mathcal{X}} \in \mathcal{P}(\mathcal{X})$. For any $f \in L^2(\mu_{\mathcal{X}})$, let $\boldsymbol{\theta}_f^k$ be the random weights after k time-steps of (GD) training with noise parameter $\tau > 0$, step size $\eta > 0$, and clipping radius $R > 0$ on the distribution $\mathcal{D}_f = \mathcal{D}(f, \mu_{\mathcal{X}})$. Fix any baseline function $\alpha \in L^2(\mu_{\mathcal{X}})$ such that $\alpha \circ g = \alpha$ for all $g \in G$, and let $\mu_{\mathcal{F}} \in \mathcal{P}(L^2(\mu_{\mathcal{X}}))$ be any distribution of target functions. Then, for any $\epsilon > 0$,

$$\mathbb{P}_{f\sim\mu_{\mathcal{F}},\boldsymbol{\theta}_f^k}[\ell_{\mathcal{D}_f}(\boldsymbol{\theta}_f^k) \le \|f-\alpha\|^2_{L^2(\mu_{\mathcal{X}})} - \epsilon] \le \frac{\eta R\sqrt{k\mathcal{C}}}{2\tau} + \frac{\mathcal{C}}{\epsilon},$$

where $\mathcal{C} = \mathcal{C}((\bar{\mu}_{\mathcal{F}}, \mu_{\mathcal{X}}); G)$, and $\bar{\mu}_{\mathcal{F}}$ is the distribution of $f - \alpha$ for $f \sim \mu_{\mathcal{F}}$.

Theorem 3.1 is the special case in which $\mu_{\mathcal{F}}$ has all probability mass on one atom f_*.

A.1 Proof of Theorem A.2

We derive Theorem A.2 from the following theorem, which is the same bound but without the G-equivariance assumption (think of G as being the trivial group). We first define the alignment of a distribution of functions:

Definition A.3. *Let $\mu_{\mathcal{X}} \in \mathcal{P}(\mathcal{X})$. For any distribution of functions $\mu_{\mathcal{F}} \in \mathcal{P}(L^2(\mu_{\mathcal{X}}))$, we define:*

$$\mathcal{C}(\mu_{\mathcal{F}}, \mu_{\mathcal{X}}) = \sup_h \mathbb{E}_{f\sim\mu_{\mathcal{F}}}[\langle f, h\rangle^2],$$

where the supremum is over $h \in L^2(\mu_{\mathcal{X}})$ such that $\|h\|^2 = 1$.

Theorem A.4 (GD lower bound for distribution of functions)**.** *Let $f_{\mathsf{NN}}(\cdot;\boldsymbol{\theta}) : \mathcal{X} \to \mathbb{R}$ be an architecture, and let $\mu_{\boldsymbol{\theta}} \in \mathcal{P}(\mathbb{R}^p)$ be an initialization distribution.*

Fix $\mu_{\mathcal{X}} \in \mathcal{P}(\mathcal{X})$. For any $f \in L^2(\mu_{\mathcal{X}})$, let $\boldsymbol{\theta}_f^k$ be the random weights after k time-steps of (GD) training with noise parameter $\tau > 0$, step size $\eta > 0$, and clipping radius $R > 0$ on the distribution $\mathcal{D}_f = \mathcal{D}(f, \mu_{\mathcal{X}})$. Then, for any $\alpha \in L^2(\mu_{\mathcal{X}})$ any $\mu_{\mathcal{F}} \in \mathcal{P}(L^2(\mu_{\mathcal{X}}))$, and any $\epsilon > 0$,

$$\mathbb{P}_{f\sim\mu_{\mathcal{F}},\boldsymbol{\theta}_f^k}[\ell_{\mathcal{D}_f}(\boldsymbol{\theta}_f^k) \le \|f-\alpha\|^2_{L^2(\mu_{\mathcal{X}})} - \epsilon] \le \frac{\eta R\sqrt{k\mathcal{C}}}{2\tau} + \frac{\mathcal{C}}{\epsilon},$$

where $\mathcal{C} = \mathcal{C}(\bar{\mu}_{\mathcal{F}}, \mu_{\mathcal{X}})$, and $\bar{\mu}_{\mathcal{F}}$ is the distribution of $f - \alpha$ for $f \sim \mu_{\mathcal{F}}$.

We defer the proof of this theorem, and first use it to prove Theorem A.2.

Proof of Theorem A.2. Let μ_G be the Haar probability measure on G. Define the distribution of functions $\nu_{\mathcal{F}} \in \mathcal{P}(L^2(\mu_{\mathcal{X}}))$ as the distribution of $f \circ g \in L^2(\mu_{\mathcal{X}})$, where $f \sim \mu_{\mathcal{F}}$ and $g \sim \mu_G$, independently. Notice that $f \circ g$ is in $L^2(\mu_{\mathcal{X}})$ since $\mu_{\mathcal{X}}$ is G-invariant and $f \in L^2(\mu_{\mathcal{X}})$, so this is well defined.

Now define $\bar{\mu}_{\mathcal{F}}$ and $\bar{\nu}_{\mathcal{F}}$ to be the distribution of $f - \alpha$ for $f \sim \mu_{\mathcal{F}}$ and $f \sim \nu_{\mathcal{F}}$, respectively. Since $\alpha \circ g = \alpha$ for all $g \in G$, we have

$$\begin{aligned}\mathcal{C}((\bar{\mu}_{\mathcal{F}}, \mu_{\mathcal{X}}); G) &= \sup_h \mathbb{E}_{g\sim\mu_G}[\mathbb{E}_{f\sim\mu_{\mathcal{F}}}[\langle (f-\alpha)\circ g, h\rangle^2]] = \sup_h \mathbb{E}_{g\sim\mu_G}[\mathbb{E}_{f\sim\mu_{\mathcal{F}}}[\langle (f\circ g)-\alpha, h\rangle^2]] \\ &= \sup_h \mathbb{E}_{f\sim\bar{\nu}_{\mathcal{F}}}[\langle f, h\rangle^2] = \mathcal{C}(\bar{\nu}_{\mathcal{F}}, \mu_{\mathcal{X}})\,.\end{aligned}$$

We conclude by Theorem A.4 that

$$\mathbb{P}_{f\sim\mu_{\mathcal{F}},g\sim\mu_G,\boldsymbol{\theta}^k_{f\circ g}}[\ell_{\mathcal{D}_{f\circ g}}(\boldsymbol{\theta}^k_{f\circ g}) \le \|f\circ g-\alpha\|^2-\epsilon] \le \frac{\eta R\sqrt{k\mathcal{C}}}{2\tau}+\frac{\mathcal{C}}{\epsilon},$$

for $\mathcal{C}=\mathcal{C}(\bar{\mu}_{\mathcal{F}},\mu_{\mathcal{X}},G)$. To derive Theorem A.2, note for any $f\in L^2(\mu_{\mathcal{X}})$ and $g\in G$, by the G-equivariance of GD training we have $\ell_{\mathcal{D}_{f\circ g}}(\boldsymbol{\theta}^k_{f\circ g}) \overset{d}{=} \ell_{\mathcal{D}_f}(\boldsymbol{\theta}^k_f)$. Finally, conclude by noting $\|f\circ g-\alpha\|^2_{L^2(\mu_{\mathcal{X}})}=\|(f-\alpha)\circ g\|^2_{L^2(\mu_{\mathcal{X}})}=\|f-\alpha\|^2_{L^2(\mu_{\mathcal{X}})}$, where we first use the G-invariance of α and then that of $\mu_{\mathcal{X}}$. □

A.2 Proof of Theorem A.4

The proof of Theorem A.4 is a variation on the junk flow argument of [AS20]. There are two significant differences. First, the junk data distribution is not chosen so that labels are independent of data, but rather chosen so that the labels are given by the function α. This allows us to prove lower bounds beyond weak learning (cf. allowing the merged-staircase property necessity result of Section 3.2). Second, instead of using cross-predictability, we use a tighter bound based on the quantity $\mathcal{C}(\bar{\mu}_{\mathcal{F}},\mu_{\mathcal{X}})$. This allows a $\Omega(d^{l+1})$-lower bound instead of a $\Omega(d^{(l+1)/2})$-lower bound for functions that are not l-MSP in Section 3.2. However, for this tighter bound during training we need to clip the gradients of the neural network instead of the gradients of the loss as in [AS20]. In Appendix A.3, we show how to recover the bound of [AS20] based on cross-predictability if we instead clip the gradients of the loss.

For the analysis, we define the following gradient descent trajectories:

- *GD trajectory on $\mathcal{D}_f$*: for any $f:\mathcal{X}\to\mathbb{R}$, we let $\boldsymbol{\theta}^0_f,\dots,\boldsymbol{\theta}^k_f$ be the trajectory of (GD) on the data distribution $\mathcal{D}_f=\mathcal{D}(f,\mu_{\mathcal{X}})\in\mathcal{P}(\mathcal{X}\times\mathbb{R})$. I.e., we initialize $\boldsymbol{\theta}^0\sim\mu_{\boldsymbol{\theta}}$, and update
$$\boldsymbol{\theta}^{k+1}_f=\boldsymbol{\theta}^k_f-\eta\boldsymbol{g}_{\mathcal{D}_f}(\boldsymbol{\theta}^k_f)+\boldsymbol{\xi}^k\,,\qquad \text{where } \boldsymbol{\xi}^k\sim\mathcal{N}(0,\tau^2\boldsymbol{I})\,.$$
- *Junk GD trajectory*: define $\mathcal{D}_{\text{junk}}=\mathcal{D}(\alpha,\mu_{\mathcal{X}})$. We let $\boldsymbol{\theta}^0_\alpha,\dots,\boldsymbol{\theta}^k_\alpha$ be the trajectory of (GD) on $\mathcal{D}_{\text{junk}}$, which we call the *junk trajectory*. I.e., we initialize $\boldsymbol{\theta}^0_\alpha\sim\mu_{\boldsymbol{\theta}}$, and update
$$\boldsymbol{\theta}^{k+1}_\alpha=\boldsymbol{\theta}^k_\alpha-\eta\boldsymbol{g}_{\mathcal{D}_{\text{junk}}}(\boldsymbol{\theta}^k_\alpha)+\tilde{\boldsymbol{\xi}}^k\,,\qquad \text{where } \tilde{\boldsymbol{\xi}}^k\sim\mathcal{N}(0,\tau^2\boldsymbol{I})\,.$$

We now prove that the junk trajectory $\boldsymbol{\theta}^k_\alpha$ stays close to the trajectory $\boldsymbol{\theta}^k_f$, for most functions $f\sim\mu_{\mathcal{F}}$. The bound depends on the quantity $\mathcal{C}(\bar{\mu}_{\mathcal{F}},\mu_{\mathcal{X}})$.

Lemma A.5 (Junk trajectory close to most GD trajectories)**.** *Under the assumptions of Theorem A.4,*

$$\mathbb{E}_{f\sim\mu_{\mathcal{F}}}[\mathrm{TV}(\mathcal{L}(\boldsymbol{\theta}^k_\alpha),\mathcal{L}(\boldsymbol{\theta}^k_f))]\le\frac{\eta R}{2\tau}\sqrt{k\mathcal{C}(\bar{\mu}_{\mathcal{F}},\mu_{\mathcal{X}})}.$$

Finally, we show that the junk GD trajectory is not correlated with most random $f\sim\mu_{\mathcal{F}}$. This bound again depends on the quantity $\mathcal{C}(\bar{\mu}_{\mathcal{F}},\mu_{\mathcal{X}})$.

Lemma A.6 (Junk trajectory does not learn)**.** *Under the assumptions of Theorem A.4, for any $\epsilon>0$,*

$$\mathbb{P}_{f\sim\mu_{\mathcal{F}},\boldsymbol{\theta}^k_\alpha}[\ell_{\mathcal{D}_f}(\boldsymbol{\theta}^k_\alpha)\le\|f-\alpha\|^2-\epsilon]\le\mathcal{C}(\bar{\mu}_{\mathcal{F}},\mu_{\mathcal{X}})/\epsilon.$$

Combining the above two lemmas, we prove the theorem.

Proof of Theorem A.4. By Lemma A.5, we can couple $\boldsymbol{\theta}^k_\alpha$ with $\boldsymbol{\theta}^k_f$ so that

$$\mathbb{P}_{f\sim\mu_{\mathcal{F}},\boldsymbol{\theta}^k_\alpha,\boldsymbol{\theta}^k_f}[\boldsymbol{\theta}^k_\alpha\ne\boldsymbol{\theta}^k_f]\le\frac{\eta R}{2\tau}\sqrt{k\mathcal{C}(\bar{\mu}_{\mathcal{F}},\mu_{\mathcal{X}})}.$$

So, for any $t>0$,

$$\mathbb{P}_{f\sim\mu_{\mathcal{F}},\boldsymbol{\theta}^k_f}[\ell_{\mathcal{D}_f}(\boldsymbol{\theta}^k_f)\le t]\le\mathbb{P}_{f\sim\mu_{\mathcal{F}},\boldsymbol{\theta}^k_\alpha}[\ell_{\mathcal{D}_f}(\boldsymbol{\theta}^k_\alpha)\le t]+\frac{\eta R}{2\tau}\sqrt{k\mathcal{C}(\bar{\mu}_{\mathcal{F}},\mu_{\mathcal{X}})},$$

and the theorem follows by Lemma A.6. □

A.2.1 Proofs of auxiliary lemmas

Proof of Lemma A.5. For brevity, write $\boldsymbol{\theta}_{\widetilde{\alpha}}^{\leq k} = (\boldsymbol{\theta}_{\alpha}^{0}, \ldots, \boldsymbol{\theta}_{\alpha}^{k})$, and similarly for $\boldsymbol{\theta}_{\widetilde{f}}^{\leq k}$. We bound the KL-divergence between the junk trajectory $\boldsymbol{\theta}_{\widetilde{\alpha}}^{\leq k}$ and the trajectory $\boldsymbol{\theta}_{\widetilde{f}}^{\leq k}$ by (a) using the chain rule for KL-divergence, (b) the fact that $\mathcal{L}(\boldsymbol{\theta}_{\alpha}^{0}) = \mathcal{L}(\boldsymbol{\theta}_{f}^{0}) = \mu_{\boldsymbol{\theta}}$, (c) the Markov property of GD training, (d) the definition of the update step (GD), and (e) the KL divergence between two Gaussians:

$$
\begin{aligned}
&\mathrm{KL}(\mathcal{L}(\boldsymbol{\theta}_{\widetilde{\alpha}}^{\leq k}) || \mathcal{L}(\boldsymbol{\theta}_{\widetilde{f}}^{\leq k})) \qquad (5)\\
&\overset{(a)}{=} \mathrm{KL}(\mathcal{L}(\boldsymbol{\theta}_{\alpha}^{0}) || \mathcal{L}(\boldsymbol{\theta}_{f}^{0})) + \sum_{k'=1}^{k} \mathrm{KL}(\mathcal{L}(\boldsymbol{\theta}_{\alpha}^{k'} | \boldsymbol{\theta}_{\widetilde{\alpha}}^{\leq k'-1}) || \mathcal{L}(\boldsymbol{\theta}_{f}^{k'} | \boldsymbol{\theta}_{\widetilde{f}}^{\leq k'-1}))\\
&\overset{(b)}{=} \sum_{k'=1}^{k} \mathrm{KL}(\mathcal{L}(\boldsymbol{\theta}_{\alpha}^{k'} | \boldsymbol{\theta}_{\widetilde{\alpha}}^{\leq k'-1}) || \mathcal{L}(\boldsymbol{\theta}_{f}^{k'} | \boldsymbol{\theta}_{\widetilde{f}}^{\leq k'-1}))\\
&\overset{(c)}{=} \sum_{k'=1}^{k} \mathrm{KL}(\mathcal{L}(\boldsymbol{\theta}_{\alpha}^{k'} | \boldsymbol{\theta}_{\alpha}^{k'-1}) || \mathcal{L}(\boldsymbol{\theta}_{f}^{k'} | \boldsymbol{\theta}_{f}^{k'-1}))\\
&\overset{(d)}{=} \sum_{k'=1}^{k} \mathbb{E}_{\boldsymbol{\theta}' \sim \mathcal{L}(\boldsymbol{\theta}_{\alpha}^{k'-1})} [\mathrm{KL}(\mathcal{N}(\boldsymbol{\theta}' - \eta \boldsymbol{g}_{\mathcal{D}_{\mathrm{junk}}}(\boldsymbol{\theta}'), \tau^2 \boldsymbol{I}) || \mathcal{N}(\boldsymbol{\theta}' - \eta \boldsymbol{g}_{\mathcal{D}_f}(\boldsymbol{\theta}'), \tau^2 \boldsymbol{I}))]\\
&\overset{(e)}{=} \sum_{k'=1}^{k} \frac{\eta^2}{2\tau^2} \mathbb{E}_{\boldsymbol{\theta}' \sim \mathcal{L}(\boldsymbol{\theta}_{\alpha}^{k'-1})} [\| \boldsymbol{g}_{\mathcal{D}_{\mathrm{junk}}}(\boldsymbol{\theta}') - \boldsymbol{g}_{\mathcal{D}_f}(\boldsymbol{\theta}') \|^2]. \qquad (6)
\end{aligned}
$$

In order to analyze this, let us simplify the following quantity. Throughout $\boldsymbol{x}, \boldsymbol{x}' \sim \mu_{\mathcal{X}}$ are i.i.d. We use that (a) $\mathcal{D}_{\mathrm{junk}}$ and $\mathcal{D}_f$ have the same marginal distribution of $\boldsymbol{x} \sim \mu_{\mathcal{X}}$, (b) the definitions of $\mathcal{D}_{\mathrm{junk}}$ and $\mathcal{D}_f$,

$$
\begin{aligned}
&\boldsymbol{g}_{\mathcal{D}_{\mathrm{junk}}}(\boldsymbol{\theta}') - \boldsymbol{g}_{\mathcal{D}_f}(\boldsymbol{\theta}')\\
&\quad = -\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\mathrm{junk}}}[(y - f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}'))\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}')]\\
&\qquad + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_f}[(y - f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}'))\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}')]\\
&\quad \overset{(a)}{=} -\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\mathrm{junk}}}[y\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}')] + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_f}[y\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}')]\\
&\quad \overset{(b)}{=} -\mathbb{E}_{\boldsymbol{x}}[\alpha(\boldsymbol{x})\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}')] + \mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}')]\\
&\quad = \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \alpha(\boldsymbol{x}))\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}')].
\end{aligned}
$$

Now let us draw $f \sim \mu_{\mathcal{F}}$, and bound the expected KL divergence of $\boldsymbol{\theta}_{\widetilde{\alpha}}^{\leq k}$ with $\boldsymbol{\theta}_{\widetilde{f}}^{\leq k}$. By (a) plugging the above equation into (6), (b) using independence of $\boldsymbol{\theta}_{\alpha}^{k'-1}$ from f, (c) using the definition of $\mathcal{C}(\bar{\mu}_{\mathcal{F}}, \mu_{\mathcal{X}})$ to bound each coordinate, and (d) using the fact that $\Pi_{B(0,R)}$ is projection to the ball of radius R,

$$
\begin{aligned}
\mathbb{E}_f[\mathrm{KL}(\mathcal{L}(\boldsymbol{\theta}_{\widetilde{\alpha}}^{\leq k}) || \mathcal{L}(\boldsymbol{\theta}_{\widetilde{f}}^{\leq k}))] &\overset{(a)}{=} \mathbb{E}_f[\sum_{k'=1}^{k} \frac{\eta^2}{2\tau^2} \mathbb{E}_{\boldsymbol{\theta}' \sim \mathcal{L}(\boldsymbol{\theta}_{\alpha}^{k'-1})} [\| \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \alpha(\boldsymbol{x}))\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}')]\|^2]]\\
&\overset{(b)}{=} \sum_{k'=1}^{k} \frac{\eta^2}{2\tau^2} \mathbb{E}_{\boldsymbol{\theta}' \sim \mathcal{L}(\boldsymbol{\theta}_{\alpha}^{k'-1})} [\mathbb{E}_f[\| \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \alpha(\boldsymbol{x}))\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}')]\|^2]]\\
&\overset{(c)}{\leq} \sum_{k'=1}^{k} \frac{\eta^2}{2\tau^2} \mathbb{E}_{\boldsymbol{\theta}' \sim \mathcal{L}(\boldsymbol{\theta}_{\alpha}^{k'-1})} [\mathcal{C}(\bar{\mu}_{\mathcal{F}}, \mu_{\mathcal{X}}) \mathbb{E}_{\boldsymbol{x}}[\|\Pi_{B(0,R)} \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}')\|^2]]\\
&\overset{(d)}{\leq} \frac{k\eta^2 R^2}{2\tau^2} \mathcal{C}(\bar{\mu}_{\mathcal{F}}, \mu_{\mathcal{X}}). \qquad (7)
\end{aligned}
$$

Finally we apply (a) the data processing inequality for total variation distance, (b) Pinsker's inequality, (c) Jensen's inequality, and (d) the bound in (7):

$$\begin{aligned}
\mathbb{E}_f[\mathrm{TV}(\mathcal{L}(\boldsymbol{\theta}_\alpha^k), \mathcal{L}(\boldsymbol{\theta}_f^k))] &\overset{(a)}{\leq} \mathbb{E}_f[\mathrm{TV}(\mathcal{L}(\boldsymbol{\theta}_\alpha^{\leq k}), \mathcal{L}(\boldsymbol{\theta}_f^{\leq k}))] \\
&\overset{(b)}{\leq} \mathbb{E}_f[\sqrt{\frac{1}{2}\mathrm{KL}(\mathcal{L}(\boldsymbol{\theta}_\alpha^{\leq k})||\mathcal{L}(\boldsymbol{\theta}_f^{\leq k}))}] \\
&\overset{(c)}{\leq} \sqrt{\mathbb{E}_f[\frac{1}{2}\mathrm{KL}(\mathcal{L}(\boldsymbol{\theta}_\alpha^{\leq k})||\mathcal{L}(\boldsymbol{\theta}_f^{\leq k}))]} \\
&\overset{(d)}{\leq} \frac{\eta L}{2\tau}\sqrt{k\mathcal{C}(\bar{\mu}_{\mathcal{F}}; \mu_{\mathcal{X}})}.
\end{aligned}$$

□

Finally, we prove Lemma A.6, which is the last remaining lemma.

Proof. Define $\rho = \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \alpha(\boldsymbol{x}))f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_\alpha^k)]$. Recall the junk trajectory $\boldsymbol{\theta}_\alpha^k$ is drawn independently of $f \sim \mu_{\mathcal{F}}$, so, by definition of $\mathcal{C}(\bar{\mu}_{\mathcal{F}}, \mu_{\mathcal{X}})$,

$$\mathbb{E}_f[\rho^2] \leq \mathcal{C}(\bar{\mu}_{\mathcal{F}}, \mu_{\mathcal{X}})\, \mathbb{E}_{\boldsymbol{x}}[f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_\alpha^k)^2].$$

Let E be the event that $\rho^2 \leq \epsilon\, \mathbb{E}_{\boldsymbol{x}}[f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_\alpha^k)^2]$. By a Markov bound, $\mathbb{P}[E] \geq 1 - \mathcal{C}(\bar{\mu}_{\mathcal{F}}, \mu_{\mathcal{X}})/\epsilon$. Finally, under event E we have

$$\begin{aligned}
\ell_{\mathcal{D}_f}(\boldsymbol{\theta}_\alpha^k) &= \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_\alpha^k))^2] \\
&= \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \alpha(\boldsymbol{x}))^2] - 2\,\mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \alpha(\boldsymbol{x}))(f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_\alpha^k) - \alpha(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{x}}[(f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_\alpha^k) - \alpha(\boldsymbol{x}))^2] \\
&= \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \alpha(\boldsymbol{x}))^2] - 2\rho + \mathbb{E}_{\boldsymbol{x}}[(f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_\alpha^k) - \alpha(\boldsymbol{x}))^2] \\
&\geq \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \alpha(\boldsymbol{x}))^2] - 2\sqrt{\epsilon}\sqrt{\mathbb{E}_{\boldsymbol{x}}[(f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_\alpha^k) - \alpha(\boldsymbol{x}))^2]} + \mathbb{E}_{\boldsymbol{x}}[(f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_\alpha^k) - \alpha(\boldsymbol{x}))^2] \\
&\geq \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \alpha(\boldsymbol{x}))^2] - \epsilon \\
&= \|f - \alpha\|_{L^2(\mu_{\mathcal{X}})}^2 - \epsilon
\end{aligned}$$

where in the last line we optimize over the quantity $\mathbb{E}_{\boldsymbol{x}}[(f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_\alpha^k) - \alpha(\boldsymbol{x}))^2]$. □

A.3 Remark: relation to bound based on cross-predictability, and efficiently verifying G-alignment is small

In [AS20], a similar bound to Theorem A.4 was proved. The first main difference is that bound of [AS20] bound applied only to learning functions with binary output alphabet, $\{+1, -1\}$. The second difference is that [AS20] clips the gradient of the loss instead of clipping the gradient of the network. The third difference is that the bound of [AS20] was in terms of the cross-predictability, instead of the G-alignment of Definition A.1:

Definition A.7 (Cross-predictability)**.** *For any distribution over the inputs $\mu_{\mathcal{X}} \in \mathcal{P}(\mathcal{X})$ and any distribution over functions $\mu_{\mathcal{F}} \in \mathcal{P}(L^2(\mu_{\mathcal{X}}))$, the cross-predictability is*

$$\mathcal{CP}(\mu_{\mathcal{F}}, \mu_{\mathcal{X}}) = \mathbb{E}_{f, f' \sim \mu_{\mathcal{F}}}[\langle f, f' \rangle_{L^2(\mu_{\mathcal{X}})}^2].$$

Nevertheless, we show that the cross-predictability is an upper bound on the alignment.

Lemma A.8. *For any $\mu_{\mathcal{X}}$ and $\mu_{\mathcal{F}}$, we have $\mathcal{C}(\mu_{\mathcal{F}}, \mu_{\mathcal{X}}) \leq \sqrt{\mathcal{CP}(\mu_{\mathcal{F}}, \mu_{\mathcal{X}})}$.*

Proof. For any $h \in L^2(\mu_{\mathcal{X}})$, we show $\mathbb{E}_{f \sim \mu_{\mathcal{F}}}[\langle f, h \rangle^2] \leq \sqrt{\mathcal{CP}(\mu_{\mathcal{F}}, \mu_{\mathcal{X}})}\|h\|_{L^2(\mu_{\mathcal{X}})}^2$. In the following, let $f, f' \sim \mu_{\mathcal{F}}$ and $\boldsymbol{x}, \boldsymbol{x}' \sim \mu_{\mathcal{X}}$ be independent. By (a) a tensorization trick for $\boldsymbol{x}$, (b)

Cauchy-Schwarz, (c) the tensorization trick for f, and (d) reversing the tensorization trick for $\boldsymbol{x}$,

$$\begin{aligned}
\mathbb{E}_{f\sim\mu_{\mathcal{F}}}[\langle f,h\rangle^2] &= \mathbb{E}_f[\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})h(\boldsymbol{x})]^2] \\
&\overset{(a)}{=} \mathbb{E}_{f,\boldsymbol{x},\boldsymbol{x}'}[f(\boldsymbol{x})f(\boldsymbol{x}')h(\boldsymbol{x})h(\boldsymbol{x}')] \\
&= \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[\mathbb{E}_f[f(\boldsymbol{x})f(\boldsymbol{x}')]h(\boldsymbol{x})h(\boldsymbol{x}')] \\
&\overset{(b)}{\leq} \sqrt{\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[\mathbb{E}_f[f(\boldsymbol{x})f(\boldsymbol{x}')]^2]}\sqrt{\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[h(\boldsymbol{x})^2h(\boldsymbol{x}')^2]} \\
&= \sqrt{\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[\mathbb{E}_f[f(\boldsymbol{x})f(\boldsymbol{x}')]^2]}\,\mathbb{E}_{\boldsymbol{x}}[h(\boldsymbol{x})^2] \\
&\overset{(c)}{=} \sqrt{\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}',f,f'}[f(\boldsymbol{x})f(\boldsymbol{x}')f'(\boldsymbol{x})f'(\boldsymbol{x}')]}\,\mathbb{E}_{\boldsymbol{x}}[h(\boldsymbol{x})^2] \\
&\overset{(d)}{=} \sqrt{\mathbb{E}_{f,f'}[\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})f'(\boldsymbol{x})]^2]}\,\mathbb{E}_{\boldsymbol{x}}[h(\boldsymbol{x})^2] \\
&= \sqrt{\mathcal{CP}(\mu_{\mathcal{F}},\mu_{\mathcal{X}})}\|h\|^2_{L^2(\mu_{\mathcal{X}})}.
\end{aligned}$$

□

Interestingly, the above lemma provides an algorithmically-efficient way to verify that the G-alignment of a function is small. Namely, the "G-cross-predictability", is an upper bound on the G-alignment.

Corollary A.9 (Efficiently verifying G-alignment is small)**.** *For any compact group G, any G-invariant distribution $\mu_{\mathcal{X}} \in \mathcal{P}(\mathcal{X})$, and any $\mu_{\mathcal{F}} \in \mathcal{P}(L^2(\mu_{\mathcal{X}}))$, we have*

$$\mathcal{C}((\mu_{\mathcal{F}},\mu_{\mathcal{X}});G) \leq \sqrt{\mathcal{CP}((\mu_{\mathcal{F}},\mu_X);G)} := \sqrt{\mathbb{E}_{f,f'\sim\mu_{\mathcal{F}},g,g'\sim\mu_G}[\mathbb{E}_{\boldsymbol{x}\sim\mu_{\mathcal{X}}}[f(g(\boldsymbol{x}))f'(g'(\boldsymbol{x}))]^2]}$$

The right-hand-side in this inequality can be approximated efficiently by taking a large enough empirical sample of f, f' and g, g' and $\boldsymbol{x}$. By McDiarmid's inequality, the plug-in empirical estimate will concentrate well for large enough number of samples assuming that the functions $f \sim \mu_{\mathcal{F}}$ are bounded. Thus, given a distribution of functions $\mu_{\mathcal{F}}$ and covariates $\mu_{\mathcal{X}}$, one can empirically show that no G-equivariant network can learn them.

However, Corollary A.9 is loose, which is why we prove our bounds in terms of the G-alignment instead of the G-cross-predictability, and means that we can obtain tighter bounds than [AS20] for learning sparse parities (although with a different scheme for clipping the gradients).

Lemma A.10. *Let $\mu_{\mathcal{X}} = \mathrm{Unif}[\mathcal{H}_d]$. Let $k \in \{0,\ldots,d\}$, and let $\mu_{\mathcal{F}}$ be the uniform distribution on $\{\chi_S : S \subseteq [d], |S| = k\}$, where $\chi_S : \mathcal{H}_d \to \{+1,-1\}$ is given by $\chi_S(\boldsymbol{x}) = \prod_{i\in S} x_i$. Then*

$$\binom{d}{k}^{-1} = \mathcal{C}(\mu_{\mathcal{F}},\mu_{\mathcal{X}}) \ll \sqrt{\mathcal{CP}(\mu_{\mathcal{F}},\mu_{\mathcal{X}})} = \binom{d}{k}^{-1/2}$$

Proof. Let $S, S' \subseteq [d]$ be independent subsets of size $|S| = |S'| = k$. The cross-predictability is

$$\mathcal{CP}(\mu_{\mathcal{F}},\mu_{\mathcal{X}}) = \mathbb{E}_{S,S'}[\langle\chi_S,\chi_{S'}\rangle^2_{L^2(\mathcal{H}_d)}] = \mathbb{E}_{S,S'}[\delta_{S,S'}] = \binom{d}{k}^{-1}.$$

On the other hand, the alignment of $\mu_{\mathcal{F}}$ is:

$$\mathcal{C}(\mu_{\mathcal{F}},\mu_{\mathcal{X}}) = \sup_h \mathbb{E}_S[\langle\chi_S,h\rangle^2] = \sup_h \binom{d}{k}^{-1} \sum_{S''\subseteq[d],|S''|=k} \hat{h}(S'')^2 \leq \binom{d}{k}^{-1},$$

where the last line is by Parseval's theorem. □

A.4 Remark: alternative proof using the statistical query lens

It is possible to prove the impossibility results for learning with GD by using arguments from the Statistical Query literature (see e.g., [Kea98, Rey20]). In particular, (GD) fits under the Correlational

Statistical Query (CSQ) model of computation because it accesses the data distribution $\mathcal{D}$ only through noisy evaluations of $\boldsymbol{g}_{\mathcal{D}}(\boldsymbol{\theta})$, where

$$\boldsymbol{g}_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[y(\Pi_{B(0,R)}\nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}))] - \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta})(\Pi_{B(0,R)}\nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}))],$$

for different $\boldsymbol{\theta}^k$. The left-hand term is an expectation over the distribution $\mathcal{D}$ of a bounded function times y, since $\|\Pi_{B(0,R)}\nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta})\|^2 \leq R^2$. And the right-hand term is known since we may assume the distribution $\mu_{\mathcal{X}}$ is known. Therefore each noisy evaluation of $\boldsymbol{g}_{\mathcal{D}}(\boldsymbol{\theta})$ can be made with one CSQ [BIK90, BF02].[12]

We provide a sketch of the proof of the impossibility result using the CSQ formalism. By the G-equivariance of the GD algorithm, if GD can learn $f \in L^2(\mu_{\mathcal{X}})$ via statistical queries on the distribution $\mathcal{D}_f$, then for any $g \in G$ it can learn any $f \circ g$ via statistical queries on the distribution $\mathcal{D}_{f\circ g}$. But, if $\{\mathcal{D}_{f\circ g}\}_{g\in G}$ has high CSQ dimension then this is impossible, and therefore this proves that GD cannot efficiently learn the function f. The high CSQ dimension can ensured if the G-alignment of Definition 1.3 is small. For further comparison between lower bounds based on CSQ and lower bounds based on a junk-flow argument, see [AS20].

B Characterization of weak-learnability by GD

In this section, we give the deferred proofs from Section 3.1, characterizing weak-learnability for functions on the Boolean hypercube and functions on the unit sphere by FC networks with i.i.d. symmetric and i.i.d. initialization, respectively.

B.1 Functions on Boolean hypercube: proof of Theorem 3.4

We first prove Theorem 3.4, which concerns functions on the hypercube, learned by FC networks with i.i.d. symmetric initialization. We show the impossibility and achievability results separately.

Recall that for functions $f : \mathcal{H}_d \to \mathbb{R}$ we may write

$$f(\boldsymbol{x}) = \sum_{S\subseteq[d]} \hat{f}(S)\chi_S(\boldsymbol{x}),$$

where for each $S \subseteq [d]$, we define the monomial $\chi_S : \mathcal{H}_d \to \{+1, -1\}$ by

$$\chi_S(\boldsymbol{x}) = \prod_{i\in S} x_i$$

and the Fourier coefficient

$$\hat{f}(S) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{H}_d}[f(\boldsymbol{x})\chi_S(\boldsymbol{x})].$$

These monomial functions form an orthogonal basis over the Boolean hypercube, which we will use throughout this section, i.e., for any $S, S' \subseteq [d]$,

$$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{H}_d}[\chi_S(\boldsymbol{x})\chi_{S'}(\boldsymbol{x})] = \delta_{S,S'}.$$

B.1.1 Proof of impossibility result of Theorem 3.4

The impossibility result is proved using Theorem 3.1, and the fact from Proposition 2.5 that GD-training of FC networks with i.i.d. symmetric initialization is $G_{\text{sign,perm}}$-equivariant. First, recall the computation of the $G_{\text{sign,perm}}$-alignment.

Lemma B.1 (Restatement of Lemma 3.5)**.** *For any $f : \mathcal{H}_d \to \mathbb{R}$,*

$$\mathcal{C}((f,\mathcal{H}_d); G_{\text{sign,perm}}) = \max_{k\in[d]} \binom{d}{k}^{-1} \sum_{\substack{S\subseteq[d]\\|S|=k}} \hat{f}(S)^2.$$

[12] Although correlational statistical queries are typically defined with adversarially-chosen noise, analogous correlational statistical query lower bounds can be proved with random noise (see, e.g., Section 3.2 of [Boi20]).

Remark B.2 (Alternative proof of Lemma B.1 using the Schur orthogonality theorem)**.** *The same representation theory approach used to prove Lemma 3.7 for the G_{rot}-alignment could have been used to prove Lemma B.1 for $G_{\mathrm{sign,perm}}$, instead of the ad hoc calculation. One would decompose the functions over the Boolean hypercube as $L^2(\mathcal{H}_d) = \oplus_{l=0}^{d} \mathcal{W}_{d,l}$, where $\mathcal{W}_{d,l}$ is the subspace of homogeneous degree-l polynomials. The representations of $G_{\mathrm{sign,perm}}$ given by $\Phi_l(g) : \mathcal{W}_{d,l} \to \mathcal{W}_{d,l}$ mapping $\Phi_l(g)f = f \circ g^{-1}$ are all irreducible and inequivalent for distinct $l \neq l'$, and $\dim(\mathcal{W}_{d,l}) = \binom{d}{l}$.*

Using the previous lemma, we may relate a high $G_{\mathrm{sign,perm}}$-alignment to the existence of a non-negligible Fourier coefficient of $O(1)$ or $d - O(1)$ degree.

Lemma B.3. *Suppose $\{f_d\}_{d\in\mathbb{N}}$ is such that $\mathcal{C}((f_d, \mathcal{H}_d); G_{\mathrm{sign,perm}}) \geq \Omega(d^{-C})$ for some constant $C > 0$. Then there is a constant $C' > 0$ such that for each d there is $S_d \subseteq [d]$ with $|S_d| \leq C'$ or $|S_d| \geq d - C'$ and $|\hat{f}_d(S_d)| \geq \Omega(d^{-C'})$.*

Proof. By Lemma B.1, we have

$$\begin{aligned}
\mathcal{C}_d((f_d, \mathcal{H}_d); G_{\mathrm{sign,perm}}) &\leq \sum_{\substack{S\subseteq[d] \\ \min(|S|, d-|S|)\leq C}} \hat{f}_d(S)^2 \binom{d}{|S|}^{-1} + \sum_{\substack{S\subseteq[d] \\ \min(|S|, d-|S|)> C}} \hat{f}_d(S)^2 \binom{d}{|S|}^{-1} \\
&\leq \sum_{\substack{S\subseteq[d] \\ \min(|S|, d-|S|)\leq C}} \hat{f}_d(S)^2 + \sum_{\substack{S\subseteq[d] \\ \min(|S|, d-|S|)> C}} \hat{f}_d(S)^2 \binom{d}{C+1}^{-1} \\
&\leq \sum_{\substack{S\subseteq[d] \\ \min(|S|, d-|S|)\leq C}} \hat{f}_d(S)^2 + O(d^{-C-1}) \\
&\leq \max_{\substack{S\subseteq[d] \\ \min(|S|, d-|S|)\leq C}} d^C \hat{f}_d(S)^2 + O(d^{-C-1}).
\end{aligned}$$

So if $\mathcal{C}((f_d, \mathcal{H}_d); G_{\mathrm{sign,perm}}) \geq \Omega(d^{-C})$, then

$$\max_{\substack{S\subseteq[d] \\ \min(|S|, d-|S|)\leq C}} \hat{f}_d(S)^2 \geq \Omega(d^{-2C}).$$

Letting $C' = 2C$ proves the lemma. □

Now we are ready to prove the impossibility result of Theorem 3.4.

Proof of impossibility result of Theorem 3.4. Suppose that $\{f_d, \mathcal{H}_d\}$ is efficiently weak-learnable by GD-trained FC networks with symmetric initialization; let $\{f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d}, \eta_d, k_d, R_d\tau_d\}$ be the sequence of networks, initializations, and GD hyperparameters satisfying Definition 3.3, and in particular (2) for some constant $C > 0$. By Proposition 2.5, $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-GD training is $G_{perm,sign}$-equivariant, so by Theorem 3.1,

$$\mathbb{P}[\|f_d - f_{\mathsf{NN},d}(\cdot; \boldsymbol{\theta}_d)\|^2_{L^2(\mu_d)} \leq \|f_d\|^2_{L^2(\mu_d)} - d^{-C}] \leq \frac{\eta_d R_d \sqrt{k_d \mathcal{C}_d}}{2\tau_d} + \frac{\mathcal{C}_d}{d^{-C}} \leq O(d^{4c}\sqrt{\mathcal{C}_d} + d^C \mathcal{C}_d),$$

where $\mathcal{C}_d = \mathcal{C}((f_d, \mathcal{H}_d); G_{\mathrm{sign,perm}})$, and we have used the bounds on the parameters guaranteed by Definition 3.3.

On the other hand, since the network weak-learns (2), the right-hand side must be greater than or equal to $9/10$, so $\mathcal{C}_d \geq \Omega(d^{-\max(8c,C)})$. By Lemma B.3, this implies there is a constant C' and sequence $\{S_d\}$ such that $\min(|S_d|, d - |S_d|) \leq C'$ and $|\hat{f}_d(S_d)| \geq \Omega(d^{-C'})$. □

B.1.2 Example: a concrete function that is not efficiently weak-learnable

As a concrete example of a bounded function on the Boolean hypercube that is not efficiently weak-learnable by (GD), which was previously not known, consider the function $f \in L^2(\mathcal{H}_d)$:

$$f(\boldsymbol{x}) = \begin{cases} 0, & |\{j : x_j = 1\}| \equiv 0 \pmod 2 \\ 1, & |\{j : x_j = 1\}| \equiv 1 \pmod 4 \\ -1, & |\{j : x_j = 1\}| \equiv 3 \pmod 4 \end{cases}.$$

This function is invariant to permutations of the input, so its G_{perm}-alignment is large. Therefore previous work that considers permutation symmetry of GD training of networks [SSS17, MS20, AS20, ACHM22] cannot imply that this is a hard function to learn. Nevertheless, we will show that its G_{sign}-alignment is small. First, let us rewrite f as:

$$f(\boldsymbol{x}) = \text{Im}(\zeta(\boldsymbol{x})), \text{ where } \zeta(\boldsymbol{x}) = \exp((\pi i/2)\sum_{j=1}^{d}(x_j+1)/2)$$

Let $\boldsymbol{s}, \boldsymbol{x} \sim \mathcal{H}_d$ and take the supremum over $h \in L^2(\mathcal{H}_d)$ such that $\|h\|^2 = 1$ in the calculations below. By (a) permutation invariance of f, (b) using that $\text{Im}(z)^2 \le |z|^2$ for any complex $z \in \mathbb{C}$, (c) a tensorization trick,

$$\begin{aligned}
&\mathcal{C}((f, \mathcal{H}_d); G_{\text{sign,perm}}) \\
&\quad \overset{(a)}{=} \mathcal{C}((f, \mathcal{H}_d); G_{\text{sign}}) \\
&\quad = \sup_h \mathbb{E}_{\boldsymbol{s}}[\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x} \odot \boldsymbol{s})h(\boldsymbol{x})]^2] \\
&\quad \overset{(b)}{\le} \sup_h \mathbb{E}_{\boldsymbol{s}}[|\,\mathbb{E}_{\boldsymbol{x}}[\zeta(\boldsymbol{x} \odot \boldsymbol{s})h(\boldsymbol{x})]|^2] \\
&\quad \overset{(c)}{\le} \sup_h \mathbb{E}_{\boldsymbol{s}}[\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[\zeta(\boldsymbol{x} \odot \boldsymbol{s})\bar{\zeta}(\boldsymbol{x}' \odot \boldsymbol{s})h(\boldsymbol{x})h(\boldsymbol{x}')]] \\
&\quad \le \sup_h \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[h(\boldsymbol{x})h(\boldsymbol{x}')\,\mathbb{E}_{\boldsymbol{s}}[\zeta(\boldsymbol{x} \odot \boldsymbol{s})\bar{\zeta}(\boldsymbol{x}' \odot \boldsymbol{s})]] \\
&\quad = \sup_h \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[h(\boldsymbol{x})h(\boldsymbol{x}')\,\mathbb{E}_{\boldsymbol{s}}[\exp((\pi i/2)(\sum_{j=1}^{d}(x_j s_j + 1)/2 - \sum_{j=1}^{d}(x'_j s_j + 1)/2)]] \\
&\quad = \sup_h \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[h(\boldsymbol{x})h(\boldsymbol{x}')\prod_{j=1}^{d}\mathbb{E}_{s_j}[\exp((\pi i/2)(s_j(x_j - x'_j)/2)]] \\
&\quad = \sup_h \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[h(\boldsymbol{x})h(\boldsymbol{x}')\prod_{j=1}^{d}(\frac{1}{2}\exp((\pi i/2)((x_j - x'_j)/2)) + \frac{1}{2}\exp((\pi i/2)((-x_j + x'_j)/2)))] \\
&\quad = \sup_h \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[h(\boldsymbol{x})h(\boldsymbol{x}')\prod_{j=1}^{d}\delta_{x_j x'_j}] \\
&\quad = \sup_h \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[h(\boldsymbol{x})h(\boldsymbol{x}')\delta_{\boldsymbol{x},\boldsymbol{x}'}] \\
&\quad = \sup_h \mathbb{E}_{\boldsymbol{x}}[h(\boldsymbol{x})^2]/2^d \\
&\quad = \frac{1}{2^d}.
\end{aligned}$$

So, $\mathcal{C}((f, \mathcal{H}_d); G_{\text{sign,perm}}) \le 1/2^d$, which is negligible. Plugging this into Theorem 3.1 implies that f cannot be efficiently learned by FC networks with sign-flip symmetric initialization trained by (GD).

B.1.3 Proof of achievability result of Theorem 3.4

We prove the converse achievability result with the following training setup. Consider a two-layer FC network architecture with parameters $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{a})$ where $\boldsymbol{W} \in \mathbb{R}^{d \times m}$ and $\boldsymbol{a} \in \mathbb{R}^m$ and

$f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{a}^\top \sigma(\boldsymbol{W}\boldsymbol{x})$, where the activation function σ is a bump function applied elementwise:

$$\sigma(z) = \begin{cases} 1, & -0.5 \le z \le 1.5 \\ 0, & z \notin (-0.5, 1.5) \end{cases}.$$

Set the initialization distribution $\mu_{\boldsymbol{\theta}} = \mu_{\boldsymbol{W}} \times \mu_{\boldsymbol{a}}$, where $\mu_{\boldsymbol{a}} = \delta_{\boldsymbol{0}}$, and $\boldsymbol{W} \sim \mu_w^{\otimes(m\times d)}$, where

$$\mu_w = (1-p)\delta_0 + \frac{p}{2}\delta_1 + \frac{p}{2}\delta_{-1},$$

for some parameter $p \in [0,1]$. Note that the above is a FC architecture with an i.i.d. sign-flip symmetric initialization.[13] We obtain the following guarantee for training the network with GD.

Lemma B.4. *Let $f : \mathcal{H}_d \to \mathbb{R}$, and let $\mathcal{D} \in \mathcal{P}(\mathcal{H}_d \times \mathbb{R})$ be the distribution of $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x} \sim \mathcal{H}_d$. Let $s \in \{0,\dots,d\}$. Define $r_s = \max_{S,|S|=s} |\hat{f}(S)|/(d^2\binom{d}{s})$.*

Consider training the above architecture and initialization with parameter $p = s/d$, width $m \ge 100/(r_s)^4$. If we run (GD) for one step, with learning rate $0 < \eta < 1/(4m)$ and noise level $\tau \le \eta(r_s)^4/(100m^2 d)$, then with probability at least 9/10 we have

$$\ell_{\mathcal{D}}(\boldsymbol{\theta}^1) \le \|f\|^2_{L^2(\mathcal{H}_d)} - \eta(r_s)^4/4.$$

Proof. Let us prove that the loss is smooth in a small ball around initialization: For any $\boldsymbol{x} \in \mathcal{H}_d$ and $\tilde{\boldsymbol{W}}$ such that $\|\tilde{\boldsymbol{W}} - \boldsymbol{W}^0\|_\infty \le 1/(4d)$, we have $\tilde{\boldsymbol{W}}\boldsymbol{x} \in \mathbb{Z} + [-1/4, 1/4]$ almost surely. Therefore $\sigma'(\tilde{\boldsymbol{W}}\boldsymbol{x}) = \sigma''(\tilde{\boldsymbol{W}}\boldsymbol{x}) = \boldsymbol{0}$. So for any $\tilde{\boldsymbol{a}}$, one can show that the loss is smooth at $\tilde{\boldsymbol{\theta}} = (\tilde{\boldsymbol{W}}, \tilde{\boldsymbol{a}})$:

$$\nabla^2_{\boldsymbol{\theta}} \ell_{\mathcal{D}}(\tilde{\boldsymbol{\theta}}) \lesssim m\|\sigma\|^2_\infty \boldsymbol{I} \lesssim m\boldsymbol{I}. \tag{8}$$

Let us lower-bound the magnitude of the gradient. Since $\sigma'(\boldsymbol{W}^0\boldsymbol{x}) = 0$, we have

$$\|\nabla_{\boldsymbol{W}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2_F = \|\boldsymbol{0}\|^2_F = 0.$$

However, the gradient with respect to $\boldsymbol{a}$ is nonzero. Writing $\boldsymbol{W}^0 = [\boldsymbol{w}_1^0, \dots, \boldsymbol{w}_m^0]$, since $\boldsymbol{a}^0 = \boldsymbol{0}$,

$$\|\nabla_{\boldsymbol{a}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2 = \|\mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - (\boldsymbol{a}^0)^\top \sigma(\boldsymbol{W}^0\boldsymbol{x}))\sigma(\boldsymbol{W}^0\boldsymbol{x})]\|^2 = \sum_{i\in[m]} \mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{w}_i^0, \boldsymbol{x}\rangle)]^2. \tag{9}$$

Our main claim is that each term in this sum is lower-bounded in expectation.

Claim B.5. *For any $i \in [m]$, $\mathbb{E}_{\boldsymbol{w}_i^0}[\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{w}_i^0, \boldsymbol{x}\rangle)]^2] \ge \frac{1}{d^4}\binom{d}{s}^{-2} \sum_{S,|S|=s} |\hat{f}(S)|^2 \ge (r_s)^2$.*

Proof of claim. For ease of notation, let $\boldsymbol{w} \sim \mathcal{L}(\boldsymbol{w}_i^0)$. Write $\boldsymbol{w} = \boldsymbol{a} \odot \boldsymbol{1}_S$ for $\boldsymbol{a} \sim \mathcal{H}_d$ and $S \subset [d]$ where each element of $[d]$ is in S independently with probability p. Then

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{w}}[\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle)]^2] && (10)\\
&\quad = \mathbb{E}_{S,\boldsymbol{a}}[\mathbb{E}_{\boldsymbol{x}}[(-1)^{\lfloor |S|/2\rfloor} \operatorname{sgn}(\hat{f}(S))\chi_S(\boldsymbol{a}) f(\boldsymbol{x})\sigma(\langle \boldsymbol{a}\odot \boldsymbol{1}_S, \boldsymbol{x}\rangle)]^2]\\
&\quad \ge \mathbb{E}_{S,\boldsymbol{a},\boldsymbol{x}}[(-1)^{\lfloor |S|/2\rfloor} \operatorname{sgn}(\hat{f}(S))\chi_S(\boldsymbol{a}) f(\boldsymbol{x})\sigma(\langle \boldsymbol{a}\odot \boldsymbol{1}_S, \boldsymbol{x}\rangle)]^2\\
&\quad = \Big(\sum_{S'\subseteq[d]} \mathbb{E}_S[(-1)^{\lfloor |S|/2\rfloor} \operatorname{sgn}(\hat{f}(S))\hat{f}(S')\, \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}[\chi_S(\boldsymbol{a})\chi_{S'}(\boldsymbol{x})\sigma(\langle \boldsymbol{a}\odot \boldsymbol{1}_S, \boldsymbol{x}\rangle)]]\Big)^2\\
&\quad = \Big(\sum_{S'\subseteq[d]} \mathbb{E}_S[(-1)^{\lfloor |S|/2\rfloor} \operatorname{sgn}(\hat{f}(S))\hat{f}(S') c_{S,S'}]\Big)^2, && (11)
\end{aligned}$$

where in the last line we define $c_{S,S'} = \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}[\chi_S(\boldsymbol{a})\chi_{S'}(\boldsymbol{x})\sigma(\langle \boldsymbol{a}\odot \boldsymbol{1}_S, \boldsymbol{x}\rangle)]$. If $S \not\subseteq S'$, then we have $c_{S,S'} = 0$, since $\mathbb{E}_{\boldsymbol{a}}[\chi_S(\boldsymbol{a})\sigma(\langle \boldsymbol{a}\odot \boldsymbol{1}_S, \boldsymbol{x}\rangle)]$ depends only on $\boldsymbol{x}_S$ and not on $\boldsymbol{x}_{[d]\setminus S}$. Furthermore, for any $S' \subseteq S \subseteq [d]$,

$$c_{S,S'} = \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}[\chi_S(\boldsymbol{a})\chi_{S'}(\boldsymbol{x})\delta(\langle \boldsymbol{a}\odot \boldsymbol{1}_S, \boldsymbol{x}\rangle \in \{0,1\})]$$

[13] To fully ensure i.i.d. initialization for both layers, we could take $\mu_{\boldsymbol{a}} = \mu_w^{\otimes m}$. However it is notationally simpler to take $\mu_{\boldsymbol{a}} = \delta_{\boldsymbol{0}}$, and furthermore any bounded initialization works for $\mu_{\boldsymbol{a}}$, because the same random features analysis goes through.

$$= \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}[\chi_S(\boldsymbol{a})\chi_{S'}(\boldsymbol{x}) \sum_{k=-d}^{d} \delta(\langle \boldsymbol{a} \odot \mathbf{1}_S, \boldsymbol{x}\rangle \in \{0,1\})\delta(\langle \boldsymbol{a} \odot \mathbf{1}_{S'}, \boldsymbol{x}\rangle = k)]$$

$$= \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}[\chi_S(\boldsymbol{a})\chi_{S'}(\boldsymbol{x}) \sum_{k=-d}^{d} \delta(\langle \boldsymbol{a} \odot \mathbf{1}_{S\setminus S'}, \boldsymbol{x}\rangle \in \{-k,-k+1\})\delta(\langle \boldsymbol{a} \odot \mathbf{1}_{S'}, \boldsymbol{x}\rangle = k)].$$

Notice that $\langle \boldsymbol{a} \odot \mathbf{1}_{S'}, \boldsymbol{x}\rangle = \sum_{i\in S'} a_i x_i = |S'| - 2|\{i \in S' : a_i x_i = -1\}|$. Also, $\chi_{S'}(\boldsymbol{a})\chi_{S'}(\boldsymbol{x}) = (-1)^{|\{i\in S' : a_i x_i = -1\}|}$. Therefore, $\chi_{S'}(\boldsymbol{a})\chi_{S'}(\boldsymbol{x}) = (-1)^{(|S'| - \langle \boldsymbol{a}\odot \mathbf{1}_{S'}, \boldsymbol{x}\rangle)/2}$, so plugging this into the above,

$$c_{S,S'} = \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}[\chi_{S\setminus S'}(\boldsymbol{a}) \sum_{k=-d}^{d} (-1)^{(|S'|-k)/2}\delta(\langle \boldsymbol{a} \odot \mathbf{1}_{S\setminus S'}, \boldsymbol{x}\rangle \in \{-k,-k+1\})\delta(\langle \boldsymbol{a} \odot \mathbf{1}_{S'}, \boldsymbol{x}\rangle = k)]$$

$$= \mathbb{E}_{\boldsymbol{a}}[\chi_{S\setminus S'}(\boldsymbol{a}) \sum_{k=-d}^{d} (-1)^{(|S'|-k)/2}\mathbb{P}_{\boldsymbol{x}_{S\setminus S'}}[\langle \boldsymbol{a} \odot \mathbf{1}_{S\setminus S'}, \boldsymbol{x}\rangle \in \{-k,-k+1\}]\mathbb{P}_{\boldsymbol{x}_{S'}}[\langle \boldsymbol{a} \odot \mathbf{1}_{S'}, \boldsymbol{x}\rangle = k]]$$

$$= \mathbb{E}_{\boldsymbol{a}}[\chi_{S\setminus S'}(\boldsymbol{a})] \sum_{k=-d}^{d} (-1)^{(|S'|-k)/2}\mathbb{P}_{\boldsymbol{x}_{S\setminus S'}}[\langle \mathbf{1}_{S\setminus S'}, \boldsymbol{x}\rangle \in \{-k,-k+1\}]\mathbb{P}_{\boldsymbol{x}_{S'}}[\langle \mathbf{1}_{S'}, \boldsymbol{x}\rangle = k],$$

where the last line uses the sign-flip symmetry of the distributions $\boldsymbol{a}, \boldsymbol{x} \sim \mathcal{H}_d$. Therefore if $S \neq S'$, we have $c_{S,S'} = 0$. Otherwise, if $S = S'$, we have

$$c_{S,S} = \sum_{k\in\{0,1\}} (-1)^{(|S|-k)/2}\mathbb{P}_{\boldsymbol{x}}[\langle \mathbf{1}_S, \boldsymbol{x}\rangle = k] = (-1)^{\lfloor |S|/2\rfloor}\mathbb{P}_{\boldsymbol{x}}[\langle \mathbf{1}_S, \boldsymbol{x}\rangle \in \{0,1\}],$$

where we have used that $\langle \mathbf{1}_S, \boldsymbol{x}\rangle$ has the same parity as $|S|$ almost surely over $\boldsymbol{x}$, so either $\mathbb{P}_{\boldsymbol{x}}[\langle \mathbf{1}_S, \boldsymbol{x}\rangle = 0] = 0$ or $\mathbb{P}_{\boldsymbol{x}}[\langle \mathbf{1}_S, \boldsymbol{x}\rangle = 1] = 0$. Plugging this into (11),

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{w}}[\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle)]^2] &= \mathbb{E}_S[|\hat{f}(S)|\mathbb{P}_{\boldsymbol{x}}[\langle \mathbf{1}_S, \boldsymbol{x}\rangle \in \{0,1\}]]^2 \\
&= \mathbb{E}_S[|\hat{f}(S)|\binom{|S|}{\lfloor |S|/2\rfloor}/2^{|S|}]^2 \\
&\geq \left(\frac{1}{d}\sum_{S,|S|=s}\binom{d}{s}^{-1}|\hat{f}(S)|\binom{|S|}{\lfloor |S|/2\rfloor}/2^{|S|}\right)^2 \\
&\geq \frac{1}{d^4}\binom{d}{s}^{-2}\sum_{S,|S|=s}|\hat{f}(S)|^2 \\
&\geq (r_s)^2,
\end{aligned}$$

where recall that $s = pd$ and we use that with probability at least $1/d$ the random set S where each element is included with probability p has size $|S| = s$. □

Combining the above claim with a Hoeffding bound, we can bound the gradient with high probability.

Claim B.6. *Let E_1 denote the event that $(r_s)^2/2 \leq \|\nabla_{\boldsymbol{a}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2/m \leq 1$. Then $\mathbb{P}_{\boldsymbol{\theta}^0}[E_1] \geq 19/20$.*

Proof of claim. In (9), $\|\nabla_{\boldsymbol{a}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2$ is written as a sum of m i.i.d. random variables of the form $\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{w}_i^0, \boldsymbol{x}\rangle)]^2$. These are almost surely in the range $[0,1]$ since $\|f(\boldsymbol{x})\|_\infty \leq 1$ and $\|\sigma\|_\infty \leq 1$. This proves the upper bound almost surely.

For the lower bound, by Hoeffding's inequality,

$$\mathbb{P}[\|\nabla_{\boldsymbol{a}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2/m \leq \mathbb{E}[\|\nabla_{\boldsymbol{a}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2/m] - 5/\sqrt{m}] \leq \exp(-2\cdot 25) < 1/20.$$

Finally, recall that

$$\mathbb{E}[\|\nabla_{\boldsymbol{a}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2/m] = \mathbb{E}_{\boldsymbol{w}_i}[\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{w}_i^0, \boldsymbol{x}\rangle)]^2] \geq (r_s)^2$$

by Claim B.5. Since we have taken $m \geq 100/(r_s)^4$, we have $(r_s)^2 - 5/\sqrt{m} \geq (r_s)^2/2$, which concludes the claim. □

We also bound the magnitude of the added noise.

Claim B.7. *Let E_2 denote the event that $\|\boldsymbol{\xi}^0\|_\infty \le \min(1/(4d), \frac{\eta(r_s)^4}{4m^2 d})$. Then $\mathbb{P}[E_2] \ge 19/20$.*

Proof of claim. By a union bound over the Gaussian tail bounds for all $md + m$ entries of $\boldsymbol{\xi}^0$. □

Condition on events E_1 and E_2, which hold with probability at least $9/10$ by the above two claims. Since, $\boldsymbol{W}^1 = \boldsymbol{W}^0 + \boldsymbol{\xi}^0$, we have $\|\boldsymbol{W}^1 - \boldsymbol{W}^0\|_\infty \le 1/(4d)$. By (a) the smoothness in (8), and (b) the learning rate choice $\eta \le 1/(4m)$, (c) by the above bounds,

$$\begin{aligned}
\ell_{\mathcal{D}}(\boldsymbol{\theta}^1) &\overset{(a)}{\le} \ell_{\mathcal{D}}(\boldsymbol{\theta}^0) - \eta\|\nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2 + |\langle \nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0), \boldsymbol{\xi}^0\rangle| + m\|\eta^2\nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0) + \boldsymbol{\xi}^0\|^2 \\
&\le \|f\|^2_{L^2(\mathcal{H}_d)} - \eta\|\nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2(1 - 2m\eta) + \sqrt{m}\sqrt{(md+m)}\|\boldsymbol{\xi}^0\|_\infty + 2m(md+m)\|\boldsymbol{\xi}^0\|^2_\infty \\
&\overset{(b)}{\le} \|f\|^2_{L^2(\mathcal{H}_d)} - \frac{\eta}{2}\|\nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2 + \sqrt{m}\sqrt{(md+m)}\|\boldsymbol{\xi}^0\|_\infty + 2m(md+m)\|\boldsymbol{\xi}^0\|^2_\infty \\
&\le \|f\|^2_{L^2(\mathcal{H}_d)} - \frac{\eta}{4}(r_s)^4,
\end{aligned}$$

which proves the lemma. □

Lemma B.4 implies the achievability result of Theorem 3.4:

Proof of achievability result of Theorem 3.4. Suppose that $\{f_d\}$ is a sequence of functions $f_d \in L^2(\mathcal{H}_d)$ such that $\|f_d\|_{L^2(\mathcal{H}_d)} \le 1$ and there is a constant $C > 0$ such that

$$\max_{\substack{S\subseteq[d] \\ \min(|S|, d-|S|)\le C}} \hat{f}_d(S)^2 \ge \Omega(d^{-C}).$$

Then for each $d \in \mathbb{N}$ there is $s_d \in \{0, \dots, d\}$ such that $\max_{S\subseteq[d],|S|=s_d} |\hat{f}(S)|/(d^2\binom{d}{s_d}) \ge \Omega(d^{-2C-2})$. So by Lemma B.4 there is a FC architecture with width $m \le O(d^{8C+8})$ such that initializing i.i.d. from a symmetric distribution and GD-training with learning rate $\eta \le 1$ and noise level $\tau \ge \Omega(d^{-40C-41})$ reaches a loss $\le \|f_d\|^2_{L^2(\mathcal{H}_d)} - \Omega(d^{-12C-12})$ with probability at least $9/10$. One can take $R = \mathrm{poly}(d)$, since under events E_1 and E_2 in the analysis of Lemma B.4 we only optimize in parameter regions with $\mathrm{poly}(d)$-size gradients. □

B.2 Functions on unit sphere: proof of Theorem 3.6

We now prove the analogous result for functions on the unit sphere, again proving the impossibility and achievability results separately. We use that we can decompose $L^2(\mathbb{S}^{d-1})$ as

$$L^2(\mathbb{S}^{d-1}) = \oplus_{l=0}^{\infty}\mathcal{V}_{d,l},$$

where $\mathcal{V}_{d,l}$ is the space of degree-l spherical harmonics, which has dimension $\dim(\mathcal{V}_{d,l}) = \frac{2l+d-2}{l}\binom{l+d-3}{l-1}$ (see, e.g., [Hoc12]). Let $\Pi_{\mathcal{V}_{d,l}} : L^2(\mathbb{S}^{d-1}) \to \mathcal{V}_{d,l}$ denote the projection onto the degree-l spherical harmonics.

B.2.1 Proof of impossibility result of Theorem 3.6

Again, we apply Theorem 3.1 to prove the impossibility result. Since the FC networks are i.i.d. Gaussian-initialized, GD training is G_{rot}-equivariant by Proposition 2.5. So recall the bound on the G_{rot}-alignment proved in the main text.

Lemma B.8 (Restatement of Lemma 3.7)**.** *Let $f \in L^2(\mathbb{S}^{d-1})$. Then*

$$\mathcal{C}((f, \mathbb{S}^{d-1}); G_{\mathrm{rot}}) = \max_{l\in\mathbb{Z}_{\ge 0}} \|\Pi_{\mathcal{V}_{d,l}} f\|^2/\dim(\mathcal{V}_{d,l}).$$

We may now conclude the proof of impossibility of learning.

Proof of impossibility result of Theorem 3.6. Suppose $\{f_d, \mathbb{S}^{d-1}\}$ is efficiently weak-learnable by GD-trained FC networks with Gaussian initialization; let $\{f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d}, \eta_d, k_d, R_d, \tau_d\}$ be the sequence of networks, initializations, and GD hyperparameters achieving (2) for some constant C. On the other hand, by Proposition 2.5, GD training is G_{rot}-equivariant, so by Theorem 3.1

$$\mathbb{P}[\|f_d - \tilde{f}_d\|^2_{L^2(\mathbb{S}^{d-1})} \leq \|f_d\|^2_{L^2(\mathbb{S}^{d-1})} - d^{-C}] \leq \frac{\eta_d R_d \sqrt{k_d \mathcal{C}_d}}{2\tau_d} + \frac{\mathcal{C}_d}{d^{-C}} \leq O(d^{4c}\sqrt{\mathcal{C}_d} + d^C \mathcal{C}_d),$$

where $\mathcal{C}_d = \mathcal{C}(f_d; \mathbb{S}^{d-1}, G_{\mathrm{rot}})$. By (2), the right-hand side must be greater than $9/10$, so $\mathcal{C}_d \geq \Omega(d^{-\max(8c,C)})$. By Lemma B.8, this implies that for each $d \in \mathbb{N}$, there is $l_d \in \mathbb{Z}_{\geq 0}$ such that $\|\Pi_{\mathcal{V}_{d,l_d}} f_d\|^2 / \dim(\mathcal{V}_{d,l_d}) \geq \Omega(d^{-\max(8c,C)})$. Since $\dim(\mathcal{V}_{d,l_d}) \geq 1$, this means $\|\Pi_{\mathcal{V}_{d,l_d}} f_d\|^2 \geq \Omega(d^{-\max(8c,C)})$. Now let us prove that l_d is upper-bounded by a constant. Since $\|\Pi_{\mathcal{V}_{d,l_d}} f_d\|^2 \leq \|f_d\|^2 \leq 1$, we must have $\dim(\mathcal{V}_{d,l_d}) \leq O(d^{\max(8c,C)})$. Since $\dim(\mathcal{V}_{d,l}) \geq (d/l) \cdot ((l+d-3)/(l-1))^{l-1}$, it follows that $l_d \leq \min(8c, C)$ for all sufficiently large d. □

B.2.2 Proof of achievability result of Theorem 3.6

The positive weak learning result can again be achieved by two-layer FC networks. Indeed, in Theorem 1(b) of [GMMM21] it is proved that for any $l \in \mathbb{Z}_{\geq 1}$, a two-layer neural network random features model with $d^{\omega(l)}$ neurons will learn the projection of the target function to the degree-l spherical harmonics. The 2-layer model is given by $f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{a}^\top \sigma(\boldsymbol{W}\boldsymbol{x})$, where only $\boldsymbol{a} \in \mathbb{R}^m$ is trained, and $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m] \in \mathbb{R}^{m \times d}$ is fixed to its initialization. Unfortunately, the first-layer initialization used in [GMMM21] is not Gaussian initialization but rather uniform over the unit sphere $\boldsymbol{w}_i^0 \sim \mathbb{S}^{d-1}$. Therefore, we must modify their analysis, and for simplicity we analyze only one step of GD training on the second-layer weights, when we have $\boldsymbol{w}_i^0 \overset{i.i.d.}{\sim} \mathcal{N}(0, I_d/\sqrt{d})$, and $\boldsymbol{a}^0 = \boldsymbol{0}$.

Lemma B.9. *Let $f \in L^2(\mathbb{S}^{d-1})$, and let $\mathcal{D} \in \mathcal{P}(\mathbb{S}^{d-1} \times \mathbb{R})$ be the distribution of $(\boldsymbol{x}, f(\boldsymbol{x}))$, where $\boldsymbol{x} \sim \mathbb{S}^{d-1}$.*

Let $l \in \mathbb{Z}_{\geq 0}$ and $c > 0$. Suppose that $\|\Pi_{\mathcal{V}_{d,l}} f\|^2 \geq d^{-c}$. Consider training the above architecture and initialization with activation function $\sigma(t) = t^l$, width $m \geq d^{4l+4c+1}$. If we run (GD) for one step, with learning rate $0 < \eta < d^{-8l+8c-2}/m$ and noise level $\tau \leq \eta d^{-8l+8c-3}/(100m^2 d)$, then with probability at least $9/10$ for large enough d we have

$$\ell_{\mathcal{D}}(\boldsymbol{\theta}^1) \leq \|f\|^2_{L^2(\mathbb{S}^{d-1})} - \eta d^{-8l+8c+2}$$

Proof. Similarly to the proof of Lemma B.4, the proof relies on upper-bounding the smoothness of f_{NN} in a ball around initialization, as well as lower-bounding the gradient at initialization.

First, lower-bound the gradient:

Claim B.10. $\mathbb{P}[\|\nabla_{\boldsymbol{a}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2 \geq md^{-2l-2c-1/2}] \geq 49/50$.

Proof.

$$\nabla_{\boldsymbol{a}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^0) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[-y \nabla_{\boldsymbol{a}} f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\theta}^0)] = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{S}^{d-1}}[-f(\boldsymbol{x})\sigma(\boldsymbol{W}^0\boldsymbol{x})]$$

Let $C > 0$ be a constant. For any $i \in [m]$, let E_i be the event that $\|\boldsymbol{w}_i^0\| \in [1 - d^{-C}, 1 + d^{-C}]$. Note $\mathbb{P}[E_i] \geq \Omega(d^{-C})$. And conditioned on the event E_i, we have

$$|\mathbb{E}_{\boldsymbol{x}\sim\mathbb{S}^{d-1}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{w}_i^0, \boldsymbol{x}\rangle) \mid E_i] - \mathbb{E}_{\boldsymbol{x}\sim\mathbb{S}^{d-1}}[f(\boldsymbol{x})\sigma(\langle \frac{\boldsymbol{w}_i^0}{\|\boldsymbol{w}_i^0\|}, \boldsymbol{x}\rangle) \mid E_i]| \leq \|f\| l 2^{l-1} d^{-C} = O(d^{-C}).$$

So letting $\boldsymbol{v}_i^0 = \boldsymbol{w}_i^0 / \|\boldsymbol{w}_i^0\|$ we have

$$\mathbb{E}_{\boldsymbol{x}\sim\mathbb{S}^{d-1}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{w}_i^0, \boldsymbol{x}\rangle) \mid E_i]^2 \geq \mathbb{E}_{\boldsymbol{x}\sim\mathbb{S}^{d-1}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{v}_i^0, \boldsymbol{x}\rangle)]^2 - O(d^{-C})$$

Finally, by (a) a tensorization trick, (b) the rotational invariance of the distribution $\boldsymbol{v}_i^0$, (c) the Schur orthogonality relations for G_{rot},

$$\mathbb{E}_{\boldsymbol{v}_i^0}[\,\mathbb{E}_{\boldsymbol{x}\sim\mathbb{S}^{d-1}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{v}_i^0, \boldsymbol{x}\rangle)]^2]$$

$$
\begin{aligned}
&\overset{(a)}{=} \mathbb{E}_{\boldsymbol{v}_i^0}[\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim\mathbb{S}^{d-1}}[f(\boldsymbol{x})f(\boldsymbol{x}')\sigma(\langle \boldsymbol{v}_i^0,\boldsymbol{x}\rangle)\sigma(\langle \boldsymbol{v}_i^0,\boldsymbol{x}'\rangle)]]\\
&\overset{(b)}{=} \mathbb{E}_{g\sim G_{\mathrm{rot}},\boldsymbol{v}_i^0}[\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim\mathbb{S}^{d-1}}[f(\boldsymbol{x})f(\boldsymbol{x}')\sigma(\langle g(\boldsymbol{v}_i^0),\boldsymbol{x}\rangle)\sigma(\langle g(\boldsymbol{v}_i^0),\boldsymbol{x}'\rangle)]]\\
&= \mathbb{E}_{g\sim G_{\mathrm{rot}},\boldsymbol{v}_i^0}[\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim\mathbb{S}^{d-1}}[f(\boldsymbol{x})f(\boldsymbol{x}')\sigma(\langle \boldsymbol{v}_i^0,g^{-1}(\boldsymbol{x})\rangle)\sigma(\langle \boldsymbol{v}_i^0,g^{-1}(\boldsymbol{x}')\rangle)]]\\
&= \mathbb{E}_{\boldsymbol{v}_i^0}[\mathbb{E}_{g\sim G_{\mathrm{rot}}}[\langle f,\sigma(\langle \boldsymbol{v}_i^0,g^{-1}(\cdot)\rangle)\rangle_{L^2(\mathbb{S}^{d-1})}\langle f,\sigma(\langle \boldsymbol{v}_i^0,g^{-1}(\cdot)\rangle)\rangle_{L^2(\mathbb{S}^{d-1})}]]\\
&\overset{(c)}{=} \mathbb{E}_{\boldsymbol{v}_i^0}[\sum_{r=0}^{\infty}\frac{1}{\dim(\mathcal{V}_{d,r})}\|\Pi_{\mathcal{V}_{d,r}}f\|^2_{L^2(\mathbb{S}^{d-1})}\|\Pi_{\mathcal{V}_{d,r}}\sigma(\langle \boldsymbol{v}_i^0,\cdot\rangle)\|^2_{L^2(\mathbb{S}^{d-1})}]\\
&\geq \Omega(d^{-2l}\cdot\|\Pi_{\mathcal{V}_{d,l}}f\|^2)
\end{aligned}
$$

So combined we have

$$
\mathbb{E}_{\boldsymbol{v}_i^0}[\mathbb{E}_{\boldsymbol{x}\sim\mathbb{S}^{d-1}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{w}_i^0,\boldsymbol{x}\rangle)\mid E_i]^2]\geq \Omega(d^{-C-2l}\cdot\|\Pi_{\mathcal{V}_{d,l}}f\|^2)-O(d^{-2C}).
$$

Taking $C=2l+2c$, we get

$$
\mathbb{E}_{\boldsymbol{v}_i^0}[\mathbb{E}_{\boldsymbol{x}\sim\mathbb{S}^{d-1}}[f(\boldsymbol{x})\sigma(\langle \boldsymbol{w}_i^0,\boldsymbol{x}\rangle)\mid E_i]^2]\geq d^{-C}.
$$

So since $m\geq 2C+1$, by Hoeffding's inequality we get

$$
\mathbb{P}_{\boldsymbol{W}^0}[\|\nabla_{\boldsymbol{a}}\ell_{\mathcal{D}}(\boldsymbol{\theta}^0)\|^2\leq md^{-C-1/2}]\ll 1,
$$

proving the claim. □

The rest of the proof proceeds similarly to the achievability proof of Theorem 3.4. We can upper-bound the smoothness under the event that $\boldsymbol{W}^0$ lies in a ball of radius 2, which occurs with high probability. And we can upper-bound the magnitude of the noise by Gaussian tail bounds. Therefore, the $1/\operatorname{poly}(d)$ lower bound on the gradient implies that a $\geq 1/\operatorname{poly}(d)$ amount of progress is made towards learning when training $\boldsymbol{a}$ for one step. Finally, we can again take $R=\operatorname{poly}(d)$ since we can ensure that the gradients remain $\operatorname{poly}(d)$-bounded.

□

Proof of achievability result of Theorem 3.6. The proof is analogous to the achievability result of Theorem 3.4, but using Lemma B.9 instead of Lemma B.4. □

C Extension of MSP necessity result, proof of Theorem 3.9

Proof of Theorem 3.9. For any set $T\subseteq[P]$, let $\mathcal{S}^{\not\subseteq T}=\{S\subseteq[P]:\hat{f}_*(S)\neq 0, S\not\subseteq T\}$. Since h_* is not ℓ-MSP, we can choose T such that $\mathcal{S}^{\not\subseteq T}\neq\emptyset$, and for all $S\in\mathcal{S}^{\not\subseteq T}$, we have $|S\setminus T|\geq \ell+1$.

Let $G'\subseteq G_{\mathrm{perm}}$ be the subgroup of permutations on indices $[d]\setminus T$. Furthermore, let $\alpha(\boldsymbol{x})=\sum_{S\subseteq T}\hat{f}_*(S)\chi_S(\boldsymbol{x})$. Notice that α is invariant to the action of G', since each permutation $\sigma\in G'$ keeps the indices T fixed. Furthermore, $f_*-\alpha=\sum_{S\in\mathcal{S}^{\not\subseteq T}}\hat{f}_*(S)\chi_S(\boldsymbol{x})$, so

$$
\|f_*-\alpha\|^2>c_{h_*},
$$

where $c_{h_*}=\min_{S\subseteq[P],\hat{h}_*(S)\neq 0}|\hat{h}_*(S)|^2>0$ is a constant depending only on h_*.

Let us now bound the G'-alignment of $f_*-\alpha$. For any $\beta\in L^2(\mathcal{H}_d)$, by Cauchy-Schwarz

$$
\begin{aligned}
\mathbb{E}_{\sigma\sim G'}[\langle f_*\circ\sigma-\alpha\circ\sigma,\beta\rangle^2_{L^2(\mathcal{H}_d)}] &= \mathbb{E}_{\sigma\sim G'}[\langle\sum_{S\in\mathcal{S}^{\not\subseteq T}}\hat{f}_*(S)\chi_S\circ\sigma,\beta\rangle^2_{L^2(\mathcal{H}_d)}]\\
&\leq 2^P\sum_{S\in\mathcal{S}^{\not\subseteq T}}\hat{f}_*(S)^2\,\mathbb{E}_{\sigma\sim G'}[\langle\chi_S\circ\sigma,\beta\rangle^2_{L^2(\mathcal{H}_d)}]
\end{aligned}
$$

And for any $S \in \mathcal{S}^{\not\subseteq T}$, define $G'(S) = \{\sigma(S) : \sigma \in G'\}$. Then (a) by Parseval's, using the orthogonality of the Fourier coefficients, and (b) since $|S \setminus T| \geq \ell + 1$ by construction for all $S \in \mathcal{S}^{\not\subseteq T}$, we have $|G'(S)| = (d - |T|)!/(d - |T| - |S \setminus T|)! \geq d^{\ell+1}/2^P$, for any $d \geq 2P$,

$$\mathbb{E}_{\sigma \sim G'}[\langle \chi_S \circ \sigma, \beta \rangle^2_{L^2(\mathcal{H}_d)}] = \frac{1}{|G'(S)|} \sum_{S' \in G'(S)} \langle \chi_{S'}, \beta \rangle^2_{L^2(\mathcal{H}_d)} \overset{(a)}{\leq} \frac{1}{|G'(S)|} \|\beta\|^2_{L^2(\mathcal{H}_d)} \overset{(b)}{\leq} \|\beta\|^2_{L^2(\mathcal{H}_d)} 2^P / d^{\ell+1}.$$

We conclude that

$$\mathbb{E}_{\sigma \sim G'}[\langle f_* \circ \sigma - \alpha \circ \sigma, \beta \rangle^2_{L^2(\mathcal{H}_d)}] \leq \|\beta\|^2_{L^2(\mathcal{H}_d)} 2^{2P} \sum_{S \in \mathcal{S}^{\not\subseteq T}} \hat{f}_*(S)^2 / d^{\ell+1} \leq \|\beta\|^2_{L^2(\mathcal{H}_d)} C_{h_*} / d^{\ell+1},$$

where C_{h_*} is a constant depending only on h_*. So

$$\mathcal{C}(f_* - \alpha; \mathcal{H}_d, G') \leq C_{h_*} / d^{\ell+1}.$$

Set $\epsilon_0 = c_{h_*}/2$. By Theorem 3.1 have

$$\mathbb{P}_{\boldsymbol{\theta}^k}[\ell_{\mathcal{D}}(\boldsymbol{\theta}^k) \leq \epsilon_0] \leq \frac{\eta L}{2\tau} \sqrt{\frac{C_{h_*} k}{d^{\ell+1}}} + \frac{2C_{h_*}}{d^{\ell+1} c_{h_*}} \leq \frac{C_{h_*} \eta L}{\tau} \sqrt{\frac{k}{d^{\ell+1}}} + \frac{2C_{h_*}}{d^{\ell+1}}.$$

□

D Hardness results for SGD

In this section, for $\gamma > 0$, we let $\mathcal{D}(f, \mu_{\mathcal{X}}, \gamma) \in \mathcal{P}(\mathcal{X} \times \mathbb{R})$ denote the distribution of $(\boldsymbol{x}, f(\boldsymbol{x}) + \xi)$ where $\boldsymbol{x} \sim \mu_{\mathcal{X}}$ and $\xi \sim \mathcal{N}(0, \gamma^2)$ is independent noise.

D.1 Warm-up: hardness from permutation equivariance

Before proving Theorem 4.4, we warm up with a simple observation that exploits permutation equivariance.

Theorem D.1. *For any d, define $f_d : \mathcal{H}_d \to \{+1, -1\}$ to be the parity of the first $\lfloor d/2 \rfloor$ bits, i.e.,*

$$f_d(\boldsymbol{x}) = \prod_{i=1}^{\lfloor d/2 \rfloor} x_i.$$

Let $\{f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d}\}_{d \in \mathbb{N}}$ be a family of networks and initializations satisfying Assumption 2.4 (FC, i.i.d. initialization). Let $\gamma > 0$, and let $\{n_d\}$ be sample sizes such that $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta}_d})$-SGD training on n_d samples from $\mathcal{D}(f_d, \mathcal{H}_d, \gamma)$ rounded to $\mathrm{poly}(d)$ bits yields parameters $\boldsymbol{\theta}_d$ with

$$\mathbb{E}_{\boldsymbol{\theta}_d}[\|f_d - f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}_d)\|^2] \leq 0.01. \tag{12}$$

Then, under γ-LPGN hardness, it is not possible to run $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-SGD on n_d samples and evaluate $f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}_d)$ in $\mathrm{poly}(d)$ time.

This result follows straightforwardly from G_{perm}-equivariance of SGD in Proposition 2.5, which states that SGD on FC networks with i.i.d. initialization is blind to the order of the input vectors' coordinates, up to an unknown shared permutation. So if $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-SGD can learn $f(\boldsymbol{x}) = \prod_{i=1}^{\lfloor d/2 \rfloor} x_i$ from γ-noisy samples, it can learn $f(\boldsymbol{x}) = \chi_S(\boldsymbol{x})$ from noisy samples, for any $S \subseteq [d]$ with $|S| = \lfloor d/2 \rfloor$. But that is a γ-LPGN-hard problem.

Proof of Theorem D.1. Formally, consider the algorithm $\mathcal{A}_d^{\mathrm{SGD}}$ that runs $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-SGD on n_d samples $(\boldsymbol{x}_i, y_i)_{i \in [n_d]}$ and outputs $\mathcal{A}_d^{\mathrm{SGD}}((\boldsymbol{x}_i, y_i)_{i \in [n_d]}) : \mathcal{H}_d \to \mathbb{R}$ given by $[\mathcal{A}_d^{\mathrm{SGD}}((\boldsymbol{x}_i, y_i)_{i \in [n_d]})](\cdot) = f_{\mathsf{NN},d}(\cdot; \boldsymbol{\theta}_d)$. Let us prove that $\mathcal{A}_d^{\mathrm{SGD}}$ achieves probability of error ≤ 0.01 on the γ-LPGN problem.

Let $(S, \boldsymbol{q}, (x_i, y_i)_{i \in [n_d]})$ be a (d, n_d, γ)-LPGN instance, where $S \subseteq [d]$ is unknown, $\boldsymbol{q} \sim \mathcal{H}_d$ is random, and $(\boldsymbol{x}_i, y_i)_{i \in [n_d]}$ is random with $\boldsymbol{x}_i \overset{i.i.d}{\sim} \mathcal{H}_d$ and $y_i = \prod_{i \in S} x_i + \xi_i$, for $\xi_i \sim \mathcal{N}(0, \gamma^2)$. Let $\sigma \in S_d$ be a permutation that sends $\{1, \ldots, \lfloor d/2 \rfloor\}$ to S. Then by (a) permutation equivariance

of $\mathcal{A}_d^{\mathrm{SGD}}$ from Proposition 2.5; (b) letting $\tilde{\boldsymbol{q}} = \sigma(\boldsymbol{q})$ and using that $\chi_S(\boldsymbol{q}) = \prod_{i=1}^{\lfloor d/2 \rfloor} \tilde{\boldsymbol{q}}_i$; (c) letting $(\tilde{\boldsymbol{x}}_i, \tilde{y}_i)_{i\in[n_d]}$ be such that $\tilde{\boldsymbol{x}}_i = \sigma(\boldsymbol{x}_i)$ and $\tilde{y}_i = y_i$; and finally (d) using that $\tilde{\boldsymbol{q}} \sim \mathcal{H}_d$, and $(\tilde{\boldsymbol{x}}_i, \tilde{y}_i)_{i\in[n_d]} \overset{i.i.d.}{\sim} \mathcal{D}(f_*^{(d)}, \mu^{(d)}, \gamma)$ and a Markov bound on (12), ,

$$\begin{aligned}
\mathbb{P}\{\mathrm{sgn}([\mathcal{A}_d^{\mathrm{SGD}}((\boldsymbol{x}_i, y_i)_{i\in[n_d]})](\boldsymbol{q})) = \chi_S(\boldsymbol{q})\} &\overset{(a)}{=} \mathbb{P}\{\mathrm{sgn}([\mathcal{A}_d^{\mathrm{SGD}}((\sigma(\boldsymbol{x}_i), y_i)_{i\in[n_d]})](\sigma(\boldsymbol{q}))) = \chi_S(\boldsymbol{q})\} \\
&\overset{(b)}{=} \mathbb{P}\{\mathrm{sgn}([\mathcal{A}_d^{\mathrm{SGD}}((\sigma(\boldsymbol{x}_i), y_i)_{i\in[n_d]})](\tilde{\boldsymbol{q}})) = \prod_{i=1}^{\lfloor d/2 \rfloor} \tilde{q}_i\} \\
&\overset{(c)}{=} \mathbb{P}\{\mathrm{sgn}([\mathcal{A}_d^{\mathrm{SGD}}((\tilde{\boldsymbol{x}}_i, \tilde{y}_i)_{i\in[n_d]})](\tilde{\boldsymbol{q}})) = \prod_{i=1}^{\lfloor d/2 \rfloor} \tilde{q}_i\} \\
&\overset{(d)}{\geq} 1 - 0.01 \geq 0.99.
\end{aligned}$$

Therefore $\mathrm{sgn}(\mathcal{A}_d^{\mathrm{SGD}})$ solves the (d, n_d, γ)-LPGN problem. By the γ-LPGN-hardness assumption this algorithm cannot run in $\mathrm{poly}(d)$ time. So if the $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-SGD training can be run in $\mathrm{poly}(d)$ time and $f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}_d)$ can be evaluated in $\mathrm{poly}(d)$ time, contradicting the γ-LPGN-hardness assumption. □

D.2 Hardness from sign-flip equivariance, proof of Theorem 4.4

Our second result is not as obvious, and exploits the sign-flip equivariance of training on FC architectures with sign-flip-symmetric initialization (such as Gaussian initialization). Let $f_{\mathrm{mod8},d} : \mathcal{H}_d \to \{0, \ldots, 7\}$ denote the function given by $f_{\mathrm{mod8},d}(\boldsymbol{x}) \equiv \sum_i x_i \pmod 8$. For brevity, we drop the subscript with d and write f_{mod8}, since d is clear from context.

Restated, our main result is:

Theorem D.2 (Theorem 4.4 restated)**.** *Let $\{f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d}\}_{d\in\mathbb{N}}$ be a family of networks and initializations satisfying Assumption 2.4 (fully-connected) with i.i.d. symmetric initialization. Let $\gamma > 0$, and let $\{n_d\}$ be sample sizes such that $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-SGD training on n_d samples from $\mathcal{D}(f_{\mathrm{mod8}}, \mathcal{H}_d, \gamma)$ rounded to $\mathrm{poly}(d)$ bits yields parameters $\boldsymbol{\theta}_d$ with*

$$\mathbb{E}_{\boldsymbol{\theta}_d}[\|f_{\mathrm{mod8}} - f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}_d)\|^2] \leq 0.0001. \tag{13}$$

Then, under $(\gamma/2)$-LPGN hardness, it is not possible to run $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-SGD on n_d samples and evaluate $f_{\mathsf{NN}}(\cdot; \boldsymbol{\theta}_d)$ in $\mathrm{poly}(d)$ time.

The proof of Theorem 4.4 can be also proved via a reduction to LPGN. However, this reduction is much less straightforward.

D.2.1 The secretly-flipped sum-mod-8 (SFSM8) problem

We prove the hardness of learning sum-mod-8 with SGD by using only the sign-flip equivariance, and no other properties of the SGD algorithm. The idea is that any sign-flip-equivariant algorithm that can learn f_{mod8}, must also be capable of solving a much more difficult problem. First, define the problem of outputting sum-mod-8.

Problem D.3. *The (d, n, γ)-SM8 (sum-mod-8) problem is parametrized by $\gamma > 0$ and integers $d, n > 0$. It is as follows:*

- *Input: query vector $\boldsymbol{q} \sim \mathcal{H}_d$, samples $(\boldsymbol{x}_i, y_i)_{i\in[n]} \overset{i.i.d.}{\sim} \mathcal{D}(f_{\mathrm{mod8}}, \mathcal{H}_d, \gamma)$.*
- *Task: return $f_{\mathrm{mod8}}(\boldsymbol{q}) \in \{0, \ldots, 7\}$.*

Notice that SM8 is a trivial problem: an algorithm that was not sign-flip equivariant could ignore the samples $(\boldsymbol{x}_i, y_i)_{i\in[n]}$, and simply return $f_{\mathrm{mod8}}(\boldsymbol{q})$. However, sign-flip equivariance makes solving SM8 much more difficult: we prove that any sign-flip equivariant algorithm that can solve SM8, can also solve the problem SFSM8, defined below.

Problem D.4. *The (d, n, γ)-secretly-flipped sum-mod-8 (SFSM8) problem is parametrized by $\gamma > 0$, and integers $d, n > 0$. It is as follows:*

Algorithm REDUCE-LPGN-TO-SFSM8

Inputs: query $\boldsymbol{q} \in \mathcal{H}_d$, samples $(\boldsymbol{x}_i, y_i)_{i\in[n]}$ from an instance of $(d, n, \gamma_{\mathcal{A}}/2)$-LPGN, oracle $\mathcal{A}$ for $(d, n, \gamma_{\mathcal{A}})$-SFSM8.

1. For each $i \in [n]$, compute $t_i \in \{0, \ldots, 7\}$ by

$$t_i \equiv 2|S| - 2 - \sum_{j\in[d]} x_{i,j} \pmod 8,$$

and let

$$\tilde{y}_i = \begin{cases} t_i + 2y_i, & t_i \in \{2,3,4,5\} \\ t_i + 4 - 2y_i, & t_i \in \{0,1\} \\ t_i - 4 - 2y_i, & t_i \in \{6,7\} \end{cases}.$$

2. Let ans $= \mathcal{A}(\boldsymbol{q}, (\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)_{i\in[n]}) - 2|S| + 2 + \sum_{j\in[d]} x_j$.
3. If ans $\equiv 2 \pmod 8$, return 1. Else if ans $\equiv -2 \pmod 8$, return -1. Else, return "error".

Figure 1: Reduction from LPGN to SFSM8 (Lemma D.6). Here, $\gamma_{\mathcal{B}} = \gamma_{\mathcal{A}}/2$. Note that $|S| = \lfloor d/2 \rfloor$ is known, so the reduction is efficient.

- *Unknown: sign-flip vector* $\boldsymbol{s} \in \mathcal{H}_d$.
- *Input: query vector* $\boldsymbol{q} \sim \mathcal{H}_d$*, modified samples* $(\boldsymbol{x}_i \odot \boldsymbol{s}, y_i)_{i\in[n]}$*, where* $(\boldsymbol{x}_i, y_i)_{i\in[n]} \overset{i.i.d.}{\sim} \mathcal{D}(f_{\mathrm{mod8}}, \mathcal{H}_d, \gamma)$.
- *Task: return* $f_{\mathrm{mod8}}(\boldsymbol{q} \odot \boldsymbol{s}) \in \{0, \ldots, 7\}$.

Lemma D.5. *Let $\mathcal{A}$ be a G_{sign}-equivariant algorithm in the sense of Definition 2.2. Then $\mathcal{A}$'s error probability on SM8 equals $\mathcal{A}$'s error probability on SFSM8.*

Proof. By sign-flip equivariance, for any $\boldsymbol{s}, \boldsymbol{q}$, and any samples $(\boldsymbol{x}_i, y_i)_{i\in[n]}$,

$$[\mathcal{A}((\boldsymbol{x}_i \odot \boldsymbol{s}, y_i)_{i\in[n]})](\boldsymbol{q} \odot \boldsymbol{s}) \overset{d}{=} [\mathcal{A}((\boldsymbol{x}_i, y_i)_{i\in[n]})](\boldsymbol{q}).$$

So drawing $(\boldsymbol{x}_i, y_i)_{i\in[n]} \overset{i.i.d.}{\sim} \mathcal{D}(f_{\mathrm{mod8}}, \mathcal{H}_d, \gamma)$,

$$\mathbb{P}[[\mathcal{A}((\boldsymbol{x}_i, y_i)_{i\in[n]})](\boldsymbol{q}) = f_{\mathrm{mod8}}(\boldsymbol{q})] = \mathbb{P}[[\mathcal{A}((\boldsymbol{x}_i \odot \boldsymbol{s}, y_i)_{i\in[n]})](\boldsymbol{q} \odot \boldsymbol{s}) = f_{\mathrm{mod8}}(\boldsymbol{q} \odot \boldsymbol{s})].$$

□

D.2.2 Reduction from LPGN to SFSM8

Now we show that there is no algorithm that solves SFSM8 in polynomial time and samples, under the cryptographic assumption that learning parities with Gaussian noise (LPGN) is hard.

Lemma D.6 (Reduction from LPGN to SFSM8)**.** *Given access to an oracle $\mathcal{A}$ for $(d, n, \gamma_{\mathcal{A}})$-SFSM8 that is correct with probability $> 9/10$, one can construct an algorithm $\mathcal{B}$ for $(d, n, \gamma_{\mathcal{A}}/2)$-LPGN that is correct with probability $> 9/10$ and runs in one call to $\mathcal{A}$ and $\mathrm{poly}(n, d)$ additional time. Furthermore, $\gamma_{\mathcal{B}} = \gamma_{\mathcal{A}}/2$.*

Proof. The pseudocode for the reduction is given in Figure 1. We prove correctness. Let $S \subseteq [d]$ be the unknown subset for the LPGN problem. Define $\boldsymbol{s} \in \mathcal{H}_d$ where $s_j = 1$ for all $j \in S$, and $s_j = -1$ for all $j \notin S$. Then, for any $\boldsymbol{x} \in \mathcal{H}_d$,

$$\begin{aligned} \sum_{j\in[d]} x_j s_j &= \sum_{j\in[d]} x_j(1 + s_j) - \sum_{j\in[d]} x_j \\ &= 2|\{j \in S : x_j = 1\}| - 2|\{j \in S : x_j = -1\}| - \sum_{j\in[d]} x_j \end{aligned}$$

$$= 2(|S| - 2|\{j \in S : x_j = -1\}|) - \sum_{j\in[d]} x_j$$
$$\equiv 2\prod_{j\in S} x_j + 2|S| - 2 - \sum_{j\in[d]} x_j \pmod 8$$

where in the last line we use $|\{j \in S : x_j = -1\}| \equiv (\prod_{j\in S} x_j - 1)/2 \pmod 2$.

This guarantees that after Step 1 of the algorithm, the samples $(\boldsymbol{x}_i, \tilde{y}_i)_{i\in n}$ are distributed as if they were drawn from an instance of $(d, n, \gamma_{\mathcal{A}})$-SFSM8 with secret sign-flip vector $\boldsymbol{s}$. I.e., $\boldsymbol{x}_i \overset{i.i.d.}{\sim} \mathcal{H}_d$, and $\tilde{y}_i = f_{\text{mod8}}(\boldsymbol{x}_i) + \xi_i$ for $\xi_i \overset{i.i.d}{\sim} \mathcal{N}(0, \gamma_{\mathcal{A}}^2)$.

By the correctness guarantee of $\mathcal{A}$, we know that with probability at least 9/10, the output $\mathcal{A}(\boldsymbol{q}, (\boldsymbol{x}_i, \tilde{y}_i)_{i\in[n]}) \in \{0, \ldots, 7\}$ satisfies $\mathcal{A}(\boldsymbol{q}, (\boldsymbol{x}_i, \tilde{y}_i)_{i\in[n]}) \equiv \sum_i q_i s_i \pmod 8$. So by the above calculations, ans $\equiv 2\prod_{j\in S} q_S$, proving correctness. □

D.2.3 Proof of Theorem 4.4

Proof of Theorem 4.4. Suppose by contradiction that $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-SGD can be run in $\text{poly}(d)$ time and the learned function can be evaluated in $\text{poly}(d)$ time. Let $\mathcal{A}_d = \text{round}(f_{\mathsf{NN},d}(\cdot; \boldsymbol{\theta}_d))$ be the algorithm that runs $(f_{\mathsf{NN},d}, \mu_{\boldsymbol{\theta},d})$-SGD on n_d samples $(\boldsymbol{x}_i, y_i)_{i\in[n_d]}$ and returns the learned function, rounded to the nearest integer. By a Markov bound on (13), $\mathbb{P}\{[\mathcal{A}_d((\boldsymbol{x}_i, y_i)_{i\in[n_d]})](\boldsymbol{q}) = f_{\text{mod8}}(\boldsymbol{q})] \geq 1 - 0.004 \geq 0.99$. Furthermore, $\mathcal{A}_d$ is G_{sign}-equivariant by Proposition 2.5. So by Lemma D.5, $\mathcal{A}_d$ gives a $\text{poly}(d)$-size circuit for the (d, n_d, γ)-SFSM8 problem. By the LPGN-to-SFSM8 reduction in Lemma D.6, we see that this contradicts the $(\gamma/2)$-LPGN-hardness assumption. □

D.3 On the cryptographic assumption that LPGN is hard

In our SGD hardness results, we assume hardness of the LPGN (Learning Parities with Gaussian Noise) problem from Definition 4.1. The standard hardness assumption in the literature is hardness of LPN (Learning Parities with Noise). The differences are that in LPN: (1) the noise is classification noise instead of Gaussian noise; (2) there is no promise that the unknown subset has size $|S| = \lfloor d/2 \rfloor$. In this appendix, we show that our LPGN assumption can be derived from the LPN assumption, with slightly different parameters.

Definition D.7. *The learning parities with noise, (d, n, ρ)-LPN, problem is parametrized by $d, n \in \mathbb{Z}_{>0}$ and $\rho \in (0, 1]$. An instance $(S, \boldsymbol{q}, (\boldsymbol{x}_i, y_i)_{i\in[n]})$ consists of (i) an unknown subset $S \subseteq [d]$, and (ii) a known query vector $\boldsymbol{q} \sim \mathcal{H}_d$, and i.i.d. samples $(\boldsymbol{x}_i, y_i)_{i\in[n]}$ such that $\boldsymbol{x}_i \sim \mathcal{H}_d$ and $y_i = \chi_S(\boldsymbol{x}_i)\zeta_i$, where*

$$\zeta_i \sim \begin{cases} 1, & \text{w.p. } (1+\rho)/2 \\ -1, & \text{w.p. } (1-\rho)/2 \end{cases}.$$

The task is to return $\chi_S(\boldsymbol{q}) \in \{+1, -1\}$.

In order to reduce from LPN to LPGN, let us define promise-LPN:

Definition D.8. *The promise-LPN problem is the LPN problem with the promise that $|S| = \lfloor d/2 \rfloor$.*

We prove the following theorem, where $\rho(d)$-LPN is the LPN problem where can take any number of samples n, and the value of ρ depends on d.

Theorem D.9. *Suppose that for all constants $C > 0$, and any $\rho(d) \leq 1 - \exp(-C\sqrt{\log(d)})$, there are no $\text{poly}(d)$-time algorithms for $\rho(d)$-LPN. Then, for any constant $\gamma > 0$, there are no $\text{poly}(d)$-time algorithms for γ-LPGN.*

Proof. This follows by combining Lemmas D.10 and D.11, as outlined in the diagram below:

$$\text{LPN} \xrightarrow{\text{Lemma } D.10} \text{promise-LPN} \xrightarrow{\text{Lemma } D.11} \text{LPGN}.$$

□

Algorithm LPN-TO-PROMISE-LPN

Inputs: query $\boldsymbol{q} \in \mathcal{H}_d$, samples $(\boldsymbol{x}_i, y_i)_{i\in[n_\mathcal{B}]}$, from an instance of $(d, \rho, n_\mathcal{B})$-LPN, oracle $\mathcal{A}$ for $(2d, \rho, n_\mathcal{A})$-promise-LPN.

1. Let $T = 10000\log(d)/\rho^2$. Relabel the samples, splitting them into groups as

$$(\boldsymbol{x}_i, y_i)_{i\in[n_\mathcal{B}]} = \left(\bigsqcup_{r\in\{0,\ldots,d\}, t\in[T]} \{(\boldsymbol{x}_i^{(r,t)}, y_i^{(r,t)})\}_{i\in[n_\mathcal{A}+1]} \right) \sqcup (\boldsymbol{x}_i^{(*)}, y_i^{(*)})_{i\in[n_\mathcal{A}]}$$

2. For each $r \in \{0, \ldots, d\}$ and $t \in [T]$, let

$$\text{ans}_{r,t} = \mathcal{A}([\boldsymbol{x}_{n_\mathcal{A}+1}^{(r,t)}, \boldsymbol{z}_{n_\mathcal{A}+1}^{(r,t)}], ([\boldsymbol{x}_i^{(r,t)}, \boldsymbol{z}_i^{(r,t)}], y_i^{(r,t)} \cdot \prod_{j=1}^{r} z_{ij}^{(r,t)})_{i\in[n_\mathcal{A}]}) \cdot \prod_{j=1}^{r} z_{n_\mathcal{A}+1,j}^{(r,t)},$$

   where $\boldsymbol{z}_i^{(r,t)} \sim \mathcal{H}_d$ is random padding and $[\boldsymbol{x}_i^{(r,t)}, \boldsymbol{z}_i^{(r,t)}] \in \mathcal{H}_{2d}$ is the concatenation.

3. For each $r \in \{0, \ldots, d\}$, let $\hat{p}_r = |\{t \in [T] : \text{ans}_{r,t} = y_{n_\mathcal{A}+1}^{(r,t)}\}|/T$

4. Let $(\boldsymbol{z}_i^{(*)})_{i\in[n_\mathcal{A}]} \sim \mathcal{H}_d$. Let $\hat{r} = \arg\max_r \hat{p}_r$. Return

$$\mathcal{A}([\boldsymbol{q}, \boldsymbol{z}_{n_\mathcal{A}+1}^{(*)}], ([\boldsymbol{x}_i^{(*)}, \boldsymbol{z}_i^{(*)}], y_i^{(*)} \cdot \prod_{j=1}^{\hat{r}} z_{ij}^{(*)})_{i\in[n_\mathcal{A}]}) \cdot \prod_{j=1}^{\hat{r}} z_{n_\mathcal{A}+1,j}^{(*)}.$$

Figure 2: Reduction from LPN to promise-LPN (Lemma D.10). Here $n_\mathcal{B} = dT(1 + n_\mathcal{A}) + n_\mathcal{A}$.

D.3.1 Reduction from LPN to promise-LPN

Lemma D.10 (LPN to promise-LPN)**.** *Given access to an oracle $\mathcal{A}$ for $(2d, n_\mathcal{A}, \rho)$-promise-LPGN that is correct with probability $> 19/20$, one can construct an algorithm $\mathcal{B}$ for $(d, n_\mathcal{B}, \rho)$-LPGN that is correct with probability $> 9/10$ and runs in $O(d\log(d)/\rho^2)$ calls to $\mathcal{A}$ and $O(n_\mathcal{A} d\log(d)/\rho^2)$ additional time. Furthermore, $n_\mathcal{B} \le d\log(d)n_\mathcal{A}/\rho^2$.*

Proof. The pseudocode for the reduction is given in Figure 2. We prove correctness. Let $S \subseteq [d]$ be the unknown subset for the LPN problem. For any $r \in \{0, \ldots, d\}$, define the success probability of running $\mathcal{A}$ with r as

$$s_r := \mathbb{P}[\text{ans}_{r,1} = \chi_S(\boldsymbol{x}_{n_\mathcal{A}+1}^{(r,1)})].$$

Notice that if we take $r^* = d - |S|$, then the samples that we feed into $\mathcal{A}$ are those of the LPN problem with unknown subset $S' = S \cup \{d+1, \ldots, d+r\}$, which has size $|S'| = d = \lfloor 2d/2 \rfloor$. Therefore, the guarantee for $\mathcal{A}$ implies that if we take $r^* = d - |S|$, then

$$s_{r^*} > 19/20.$$

However, we are not given r^* in the input of the LPGN problem, which is the main difficulty. Instead, in the first part of the algorithm we estimate the success probability for each $r \in \{0, \ldots, d\}$ using fresh samples, and at the end run $\mathcal{A}$ with the r^* that we estimate gives the best success probability.

To prove correctness, let E be the event that for all $r \in \{0, \ldots, d\}$ we have

$$|\hat{p}_r - \mathbb{P}[\text{ans}_{r,1} = y_{n_\mathcal{A}+1}^{(r,1)}]| \le 1/(50\rho).$$

By Hoeffding's inequality, E holds with probability at least $\mathbb{P}[E] \ge 1 - 2\exp(-8\log(d)) \ge 1/100$. Now notice that $y_{n_\mathcal{A}+1}^{(r,1)}$ is just $\chi_S(\boldsymbol{x}_{n_\mathcal{A}+1}^{(r,1)})$ with classification noise so under the event E

$$|(\hat{p}_r + \rho/2 - 1/(2\rho))/\rho - s_r| \le 1/(50).$$

Therefore, since $s_{r^*} \ge 19/20$, under event E we must have $s_{\hat{r}} \ge 19/20 - 1/25 = 9/10 + 1/100$. Finally, since the probability of success of the algorithm is $s_{\hat{r}}$, and $\mathbb{P}[E] \ge 1/100$, we have that the algorithm's success probability is at least $9/10$. □

Algorithm PROMISE-LPN-TO-LPGN

Inputs: query $\boldsymbol{q} \in \mathcal{H}_d$, samples $(\boldsymbol{x}_i, y_i)_{i\in[n]}$, from an instance of (d, ρ, n)-promise-LPN, oracle $\mathcal{A}$ for (d, γ, n)-LPGN.

1. For each $i \in [n]$, let $\tilde{y}_i = \mathrm{RK}(y_i)$, where RK is the rejection kernel
$$\mathrm{RK} = \mathrm{RK}((1+\rho)/2 \to \mathcal{N}(1, \gamma^2), (1-\rho)/2 \to \mathcal{N}(-1, \gamma^2))$$
from Lemma 5.1 of [BBH18].
2. Return $\mathcal{A}(\boldsymbol{q}, (\boldsymbol{x}_i, \tilde{y}_i)_{i\in n})$

Figure 3: Reduction from promise-LPN to LPGN (Lemma D.11).

D.3.2 Reduction from promise-LPN to LPGN

Here, we show that the LPGN hardness assumption follows from the standard LPN hardness assumption with classification noise. The technique is rejection kernels.

Lemma D.11. *Given access to an oracle $\mathcal{A}$ for $(2d, n, \gamma)$-LPGN that is correct with probability $> 49/50$, one can construct an algorithm $\mathcal{B}$ for (d, n, ρ)-promise-LPN that is correct with probability $> 19/20$ and runs in one call to $\mathcal{A}$ and $\mathrm{poly}(n, d)$ additional time. Furthermore, we can take $\rho \le 1 - \exp(-C\sqrt{\log(n)}\max(\gamma^2, 1/\gamma^2))$, for some universal constant $C > 0$.*

Proof. The pseudocode of the reduction is given in Figure 3. We use the rejection kernel technique developed in [BBH18] to convert the classification noise of LPN to the additive Gaussian noise of LPGN. Let $\rho = 1 - 2\delta$, where $\delta < 0.1$. Notice that if we define $p = (1+\rho)/2 = 1 - \delta$ and $q = (1-\rho)/2 = \delta$, Lemma 5.1 of [BBH18] provides a randomized map RK that runs in time and maps a $\mathrm{Rad}(p)$ random variable to $\mathcal{N}(1, \gamma^2)$ and a $\mathrm{Rad}(q)$ random variable to $\mathcal{N}(-1, \gamma^2)$. For some parameter $N > 0$, this map runs in $O(N)$ time and has the guarantee that
$$d_{\mathrm{TV}}(\mathrm{RK}(\mathrm{Rad}(p)), \mathcal{N}(1, \gamma^2)) \le \Delta \quad \text{and} \quad d_{\mathrm{TV}}(\mathrm{RK}(\mathrm{Rad}(q)), \mathcal{N}(-1, \gamma^2)) \le \Delta,$$
where
$$\Delta = \mathbb{P}_{X\sim\mathcal{N}(1,\gamma^2)}[X \notin \mathcal{S}] + (\mathbb{P}_{X\sim\mathcal{N}(-1,\gamma^2)}[X \notin \mathcal{S}] + \frac{\delta}{1-\delta})^N,$$
and
$$\mathcal{S} = \{x : \frac{\delta}{1-\delta} \le \frac{\exp(-(x-1)^2/(2\gamma^2)}{\exp(-(x+1)^2/(2\gamma^2)} \le \frac{1-\delta}{\delta}\} = \{x : \frac{\delta}{1-\delta} \le \exp(\frac{2x}{\gamma^2}) \le \frac{1-\delta}{\delta}\}.$$
By standard Gaussian tail bounds,
$$\mathbb{P}_{X\sim\mathcal{N}(1,\gamma^2)}[X \notin \mathcal{S}] = \mathbb{P}_{X\sim\mathcal{N}(-1,\gamma^2)}[X \notin \mathcal{S}] \le \exp(-(\gamma^2\log((1-\delta)/\delta)/2 - 1)^2/2\gamma^2).$$
So for any $\epsilon > 0$, as long as $\delta < \min(\exp(-10\sqrt{\log(1/\epsilon)}/\gamma^2), \exp(-10\sqrt{\log(1/\epsilon)}\gamma^2))$, we have
$$\mathbb{P}_{X\sim\mathcal{N}(1,\gamma^2)}[X \notin \mathcal{S}] \le \epsilon.$$
So letting $\epsilon \le 1/(2000n^2)$, and $N \ge \log(2000n^2)$, we have $\Delta \le n/100$.

Thus, $(\boldsymbol{q}, (\boldsymbol{x}_i, \tilde{y}_i)_{i\in[n]})$ is $\le 1/100$ total-variation distance from being drawn from LPGN with unknown set S. This means that the algorithm has success probability $49/50 - 1/100 \ge 19/20$. □

E On the equivariance of SGD and GD

Recall from Definition 1.1 that an algorithm $\mathcal{A}$ that takes in a distribution $\mathcal{D} \in \mathcal{P}(\mathcal{X} \times \mathbb{R})$ and outputs a function $\mathcal{A}(\mathcal{D}) : \mathcal{X} \to \mathbb{R}$ is said to be G-equivariant if
$$\mathcal{A}(\mathcal{D}) \stackrel{d}{=} \mathcal{A}(g(\mathcal{D})) \circ g,$$
for any $g \in G$. Here we view the group element as a function $g : \mathcal{X} \to \mathcal{X}$, since it acts on the space of inputs $\mathcal{X}$. We also define $g(\mathcal{D})$ to be the distribution of $(g(\boldsymbol{x}), y)$, where $(\boldsymbol{x}, y) \sim \mathcal{D}$. In

Definitions 2.1 and 2.2, we define the equivariance of GD and SGD, respectively. In Proposition 2.5, we claim that GD and SGD are G_{perm}-equivariant in the case of training FC networks with i.i.d. initialization. Furthermore, these algorithms are $G_{\text{sign,perm}}$-, and G_{rot}-equivariant when the initialization is symmetric and Gaussian, respectively. The equivariances of SGD in Proposition 2.5 are proved by [Ng04, LZA21]. They also show extension to other algorithms beyond SGD:

Remark E.1. *[LZA21] note that other popular optimizers on FC networks with i.i.d. symmetric initialization, such as AdaGrad and Adam, also yield $G_{\text{sign,perm}}$-equivariance, as well as SGD with a mini-batch, or gradient descent on the loss of the empirical distribution $\ell_{\hat{\mathcal{D}}}$, where $\hat{\mathcal{D}} = \frac{1}{n}\sum_{i=1}^{n}\delta_{(y_i,\boldsymbol{x}_i)}$, for $(\boldsymbol{x}_i, y_i)_{i\in[n]} \overset{i.i.d.}{\sim} \mathcal{D}$. Therefore our SGD hardness result in Theorem 4.4 applies to training with the above algorithms.*

We limit ourselves in this appendix to prove the equivariances claimed by Proposition 2.5 for GD. We provide the proofs only for completeness, since they are quite similar to the proofs of equivariance of SGD.

In the remainder of this section, suppose that we have an architecture $f_{\mathsf{NN}}(\cdot;\boldsymbol{\theta})$ and an initialization $\mu_{\boldsymbol{\theta}}$ that satisfy Assumption 2.4, so that we can write the parameters $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{\psi})$. Furthermore, let $\mathcal{D} \subseteq \mathcal{P}(\mathbb{R}^d \times \mathbb{R})$ be a data distribution.

Definition E.2. *For any invertible linear transformation $\boldsymbol{M} \in GL(d,\mathbb{R})$, let $\boldsymbol{M}$ act on $\boldsymbol{\theta}$ as $\boldsymbol{M}\square\boldsymbol{\theta} = (\boldsymbol{W}\boldsymbol{M}, \boldsymbol{\psi})$. Let $\boldsymbol{M}\square\mathcal{D}$ be the distribution of $(\boldsymbol{M}\boldsymbol{x}, y)$ for $(\boldsymbol{x}, y) \sim \mathcal{D}$.*

Lemma E.3. *Suppose that for some orthogonal transformation $\boldsymbol{M} \in O(d)$, if we draw $\boldsymbol{\theta}^0 \sim \mu_{\boldsymbol{\theta}}$, then $\boldsymbol{M}\square\boldsymbol{\theta}^0 \overset{d}{=} \boldsymbol{\theta}^0$. Let $\boldsymbol{\theta}^k$ be the weights from $(f_{\mathsf{NN}}, \mu_{\boldsymbol{\theta}})$-GD on distribution $\mathcal{D}$, for any number of steps k, with any learning rate $\eta > 0$, and with any noise level $\tau > 0$. Similarly, let $\tilde{\boldsymbol{\theta}}^k$ be the weights from running $(f_{\mathsf{NN}}, \mu_{\boldsymbol{\theta}})$-GD on distribution $\tilde{\mathcal{D}} = \boldsymbol{M}\square\mathcal{D}$.*

Then, for any $\boldsymbol{x} \in \mathbb{R}^d$,

$$f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}^k) \overset{d}{=} f_{\mathsf{NN}}(\boldsymbol{M}\boldsymbol{x};\tilde{\boldsymbol{\theta}}^k)$$

Proof. We claim that we can couple $\boldsymbol{\theta}^k$ and $\tilde{\boldsymbol{\theta}}^k$ such that $\boldsymbol{\theta}^k = \boldsymbol{M}\square\tilde{\boldsymbol{\theta}}^k$ almost surely. Therefore, for any $\boldsymbol{x}$:

$$f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}^k) = f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{M}\square\tilde{\boldsymbol{\theta}}^k) = g_{\mathsf{NN}}(\tilde{\boldsymbol{W}}^k\boldsymbol{M}\boldsymbol{x};\tilde{\boldsymbol{\psi}}^k) = f_{\mathsf{NN}}(\boldsymbol{M}\boldsymbol{x};\tilde{\boldsymbol{\theta}}^k), \tag{14}$$

which proves the lemma. It remains to show our coupling inductively on k. The base case $k = 0$ is assumed. For the inductive step, assume that the coupling is true for $k \geq 0$, and prove it for $k+1$. By (a) using (14),

$$\begin{aligned}\boldsymbol{g}_{\tilde{\mathcal{D}}}(\tilde{\boldsymbol{\theta}}^k) &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[(f_{\mathsf{NN}}(\boldsymbol{M}\boldsymbol{x};\tilde{\boldsymbol{\theta}}^k) - y)\Pi_{B(0,R)}\nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{M}\boldsymbol{x};\tilde{\boldsymbol{\theta}}^k)] \\ &\overset{(a)}{=} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[(f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}^k) - y)\Pi_{B(0,R)}\nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{M}\boldsymbol{x};\tilde{\boldsymbol{\theta}}^k)]\end{aligned}$$

Furthermore, by (a) the chain rule for differentiation combined with (14), (b) using that $\boldsymbol{M}^\top \in O(d)$ preserves the $\|\cdot\|_2$ norm,

$$\begin{aligned}\Pi_{B(0,R)}\nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{M}\boldsymbol{x};\tilde{\boldsymbol{\theta}}^k) &= \Pi_{B(0,R)}[\nabla_{\boldsymbol{W}} g_{\mathsf{NN}}(\tilde{\boldsymbol{W}}^k\boldsymbol{M}\boldsymbol{x},\tilde{\boldsymbol{\psi}}^k), \nabla_{\boldsymbol{\psi}} g_{\mathsf{NN}}(\tilde{\boldsymbol{W}}^k\boldsymbol{M}\boldsymbol{x},\tilde{\boldsymbol{\psi}}^k)] \\ &\overset{(a)}{=} \Pi_{B(0,R)}[\nabla_{\boldsymbol{W}} g_{\mathsf{NN}}(\boldsymbol{W}^k\boldsymbol{x},\boldsymbol{\psi}^k)\boldsymbol{M}^\top, \nabla_{\boldsymbol{\psi}} g_{\mathsf{NN}}(\boldsymbol{W}^k\boldsymbol{x},\boldsymbol{\psi}^k)] \\ &= \boldsymbol{M}^\top\square(\Pi_{B(0,R)}[\nabla_{\boldsymbol{W}} g_{\mathsf{NN}}(\boldsymbol{W}^k\boldsymbol{x},\boldsymbol{\psi}^k), \nabla_{\boldsymbol{\psi}} g_{\mathsf{NN}}(\boldsymbol{W}^k\boldsymbol{x},\boldsymbol{\psi}^k)])\end{aligned}$$

By linearity of the $\square$ operation, it follows that

$$\boldsymbol{g}_{\tilde{\mathcal{D}}}(\tilde{\boldsymbol{\theta}}^k) = \boldsymbol{M}^\top\square\boldsymbol{g}_{\mathcal{D}}(\boldsymbol{\theta}^k).$$

So since $\boldsymbol{M} \in O(d)$ we have $\boldsymbol{M}^{-1} = \boldsymbol{M}^\top$, implying

$$\boldsymbol{g}_{\mathcal{D}}(\boldsymbol{\theta}^k) = \boldsymbol{M}\square\boldsymbol{g}_{\tilde{\mathcal{D}}}(\tilde{\boldsymbol{\theta}}^k).$$

Also couple the added noises ξ^k and $\tilde{\xi}^k$ so that $\xi^k = \boldsymbol{M}\square\tilde{\xi}^k$, which can be done since $\boldsymbol{M} \in O(d)$ and Gaussians are orthogonal-invariant. By linearity of the $\square$ operation, and the inductive hypothesis, it holds that $\boldsymbol{\theta}^{k+1} = \boldsymbol{M}\square\tilde{\boldsymbol{\theta}}^{k+1}$ almost surely. The proof of the claim follows by induction. $\square$

The above lemma immediately implies the GD equivariances claimed in Proposition 2.5.

Proof of GD equivariance in Proposition 2.5. Notice that G_{perm}, $G_{\text{sign,perm}}$, and G_{rot} can be identified with subgroups of $O(d)$. For i.i.d. initialization $\mu_{\boldsymbol{W}} = \mu_w^{\otimes(m\times d)}$, for any permutation matrix $\boldsymbol{M}$ we have $\boldsymbol{M}\Box\boldsymbol{\theta}^0 \overset{d}{=} \boldsymbol{\theta}^0$. Similarly, if μ_w is symmetric then for any signed permutation matrix $\boldsymbol{M}$ we have $\boldsymbol{M}\Box\boldsymbol{\theta}^0 \overset{d}{=} \boldsymbol{\theta}^0$. Finally, if μ_w is Gaussian then for any rotation matrix $\boldsymbol{M} \in SO(d)$ we have $\boldsymbol{M}\Box\boldsymbol{\theta}^0 \overset{d}{=} \boldsymbol{\theta}^0$. Lemma E.3 implies the GD equivariances claimed in Proposition 2.5. $\Box$