LEARNING HIGH-DEGREE PARITIES: THE CRUCIAL ROLE OF THE INITIALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Parities have become a standard benchmark for evaluating learning algorithms. Recent works show that regular neural networks trained by gradient descent can efficiently learn degree k parities on uniform inputs for constant k, but fail to do so when k and $d-k$ grow with d (here d is the ambient dimension). However, the case where $k = d - O_d(1)$, including the degree d parity (the *full parity*), has remained unsettled. This paper shows that for gradient descent on regular neural networks, learnability depends on the initial weight distribution. On one hand, the discrete Rademacher initialization enables efficient learning, while on the other hand, its Gaussian perturbation with large enough constant standard deviation σ prevents it. The positive result is shown to hold up to $\sigma = O(d^{-1})$, pointing to questions about a sharper threshold phenomenon. Unlike statistical query (SQ) learning, where a singleton function class like the full parity is trivially learnable, our negative result applies to a fixed function and relies on an *initial gradient alignment* measure of potential broader relevance to neural networks learning.

1 INTRODUCTION

Initialization plays a crucial role in the performance of neural network training algorithms. It has been shown that a proper initialization can help avoiding issues such as vanishing or exploding gradients, or set the foundation for efficient convergence and improved generalization (He et al. (2015); Glorot & Bengio (2010); Sutskever et al. (2013); Kumar (2017)). In this work, we show that the choice of initialization can be critical when learning complex functions, such as high-degree parities.

Parity functions are a well-known class of challenging problems for differentiable learning models, where the task is to determine the parity of bits belonging to an unknown subset of input coordinates. Due to their inherent non-linearity and extreme sensitivity to small input changes, parity functions often serve as a challenging benchmark for evaluating and comparing learning algorithms, including gradient descent on neural networks (Abbe & Sandon (2020); Daniely & Malach (2020)). For instance, they have been used for showing the advantages of using convolutional architectures over fully connected ones (Malach & Shalev-Shwartz (2020)), the superiority of differentiable models compared to kernel methods (Abbe et al. (2021)), and the efficacy of curriculum learning in contrast to standard training (Abbe et al. (2024b); Cornacchia & Mossel (2023)).

Previous research has mainly focused on the family of *sparse* parities, also known as k-parities, where the size of the support of the target parity, k, is bounded, i.e., it does not grow with input dimension d. It has been shown that on uniform inputs, k-parities can be learned by gradient descent algorithms (GD/SGD) on standard architectures, such as 2-layer fully connected (Barak et al. (2022); Abbe & Boix-Adsera (2022); Glasgow (2023); Kou et al. (2024)), with a sample complexity of $\tilde{O}(d^{k-1})$[1] (Kou et al. (2024)).

In contrast, for *dense* parities, where the support of the target parity is unbounded ($k = \omega_d(1)$), the picture is less clear. It has been shown that when both k and $d-k$ are unbounded, stochastic gradient descent (SGD) with large batch size and limited gradient precision on fully connected architectures cannot learn dense parities *with any initialization*[2] (Abbe & Sandon (2020)). The

---

[1] Where $\tilde{O}(d^c) = O(d^c \operatorname{poly}\log(d))$, for $c \in \mathbb{R}$.

[2] Assuming the initialization is invariant to permutation of the input neurons.

difficulty in learning parities stems from their orthogonality on uniform inputs, leading to a low cross-predictability (CP) (Abbe & Sandon (2020)). However, this only occurs if a given class of k-parities is sufficiently large. Since the cardinality of this class is $\binom{d}{k} = \binom{d}{d-k}$, this hardness result does not extend to *almost-full* parities, where $k = d - O_d(1)$, including the special case of the single d-parity (the *full parity*).

In fact, it is known that the full parity, as a symmetric function, is learnable by gradient descent methods with specific initialization (Nachum & Yehudayoff (2020)), such as setting all first layer weights to 1. For random and symmetric initializations, Abbe & Boix-Adsera (2022) showed that almost-full parities are weakly-learnable[3] by gradient descent on a two-layer fully connected network with discrete Rademacher initialization.

In this paper, we focus on almost-full parity functions and provide a deeper understanding of how the initialization impacts their learning. First, we show that SGD on a two-layer fully connected ReLU network with Rademacher initialization can achieve perfect accuracy, thus going beyond weak learning. Next, we investigate the robustness of this positive result and argue that it is a special case. In particular, we prove that with Gaussian initialization GD with limited gradient precision with the correlation loss cannot learn high degree parities on two layer ReLU networks. We then introduce an intermediate case of *perturbed*-Rademacher initialization, where the weights are initialized from a mixture of two Gaussian distributions with means $+1$ and -1 and a standard deviation of σ. We prove that when $\sigma = O(d^{-1})$, the positive result still holds, while we argue that if σ is a large enough constant, learning does not occur. We leave the analysis for the remaining range of σ and the investigation of a potential threshold to future work. While our theoretical analysis focuses on Gaussian perturbations, our experiments also explore other perturbations, both discrete and continuous, supporting our claim that the success of the Rademacher initialization is a special case. In our experiments, we also explore other settings beyond our theoretical analysis in order to justify the robustness of our findings.

Crucially, the proof technique for our negative result does not rely on constructing an orbit class or using measures for function classes (as in the cross-predictability case). Instead, it introduces a new approach centered on a novel measure, the *initial gradient alignment*, which may be relevant for evaluating the suitability of an initialization for a target distribution beyond the specific parity setting discussed in this paper.

2 RELATED WORK

Learning Parities. Learning parities on uniform inputs is easy with specialized techniques like Gaussian elimination over two-element fields or through emulation networks trained with Stochastic Gradient Descent (SGD) using small batch sizes (Abbe & Sandon (2020)). However, in the statistical query (SQ) setting (Kearns (1998)) and with gradient descent methods that have limited gradient precision (Abbe & Sandon (2020)), learning parities presents computational barriers. Recent works have focused on sparse parities, or k-parities (where $k = O_d(1)$), as a classical benchmark for evaluating learning algorithms (Suzuki et al. (2024); Edelman et al. (2023); Barak et al. (2022); Daniely & Malach (2020); Malach et al. (2021); Abbe & Boix-Adsera (2022); Malach & Shalev-Shwartz (2020); Abbe et al. (2024b); Cornacchia & Mossel (2023); Wei et al. (2019); Ji & Telgarsky (2019)). In particular, in the special case of $k = 2$ (i.e. the XOR problem), Glasgow (2023) proved a sample complexity upper bound of $\tilde{O}(d)$ on a 2-layer network of logarithmic width, while for general k, Kou et al. (2024) proved a sample complexity of $\tilde{O}(d^{k-1})$, matching the SQ lower bounds in both cases. For dense parities, it has been established that if both k and $d - k$ grow with d, SGD with large batch sizes fail at learning in polynomial time (Abbe & Sandon (2020)). We build on top of Abbe & Boix-Adsera (2022), which showed that almost-full parities are weakly-learnable by gradient descent on a two-layer fully connected network with Rademacher initialization. We provide a more complete picture on the role of the initialization for learning the full parity, and we argue that the Rademacher initialization is in some sense a special case.

The Role of the Initialization. Several studies have shown that initialization is crucial for optimizing neural networks, preventing vanishing or exploding gradients (Glorot & Bengio (2010)), speed-

---

[3] i.e., an inverse polynomial edge over the trivial estimator is achieved with constant probability.

ing up convergence (He et al. (2015)), ensuring informative gradient flow in early layers (Sutskever et al. (2013)), and enabling learning challenging targets (Zhang et al. (2019); Hanin & Rolnick (2018)). While these works focus on improving learning through tailored initializations, our paper addresses the more fundamental question of what can gradient descent learn with standard initializations. Thus, our work aligns more closely with (Abbe & Boix-Adsera (2022); Abbe & Sandon (2020)), which characterize functions learnable by gradient descent on shallow networks, but without exploring initialization. Another work (Edelman et al. (2023)) shows that sparse initialization aids in learning sparse parities. However, the main challenge in their case is identifying the support of the sparse parity. In contrast, when learning the full parity, sparsifying the Rademacher initialization does not aid in learning the full parity (see Figure 3, Section 6).

Complexity Measures. Previous works have studied the sample and time complexity of learning with SGD on neural networks, proposing various measures, such as: the noise sensitivity (O'Donnell (2014); Zhang et al. (2021); Abbe et al. (2022b); Hahn & Rofin (2024)), which applies mostly to settings with i.i.d. inputs and is related to the degree of the functions, is known to be loose for strong learning (Abbe et al. (2022a; 2023)); the *globality degree* (Abbe et al. (2024a)), which generalizes the degree and sensitivity notions to non-i.i.d. settings but remains focused on weak rather than strong learning; the statistical query (SQ) dimension (Kearns (1998); Feldman (2016)) and the cross-predictability (Abbe & Sandon (2020)), which are usually defined for a class of targets/distributions rather than a single distribution (in particular the full parity is efficiently SQ learnable since there is a single function); the neural tangent kernel (NTK) alignment (Jacot et al. (2018); Cortes et al. (2012)) that are limited to the NTK framework; the information exponent (Arous et al. (2021); Bruna et al. (2023)), generative exponent (Damian et al. (2024)) and leap (Abbe et al. (2023)), which measure when fully connected neural networks can strongly learn target functions on i.i.d. or isotropic input distributions and sparse or single/multi-index functions. In particular, few works studied measures based on the alignment between the networks initialization and the target distribution, as in this paper. (Mok et al. (2022); Ortiz-Jiménez et al. (2021)) studied the label-gradient-alignment (LGA), defined as the norm of the target function in the RKHS induced by the NTK (Jacot et al. (2018)) at initialization, showing its empirical relevance for predicting network performance. In contrast, we focus on a theoretical analysis, with our measure of initial gradient alignment being loss-dependent. Abbe et al. (2022c) defined the initial alignment (INAL) as the maximum average correlation of any neuron with the target, providing a lower bound for functions with small INAL, though their result relies on input embedding and orbit hardness, which does not apply to almost-full parities.

3 SETTING AND INFORMAL CONTRIBUTIONS

We consider learning with a neural network of P parameters, $\mathrm{NN}(x;\theta)$, $\theta \in \mathbb{R}^P$, initialized as $\theta^0 \sim \mathcal{D}^0$, for some distribution $\mathcal{D}^0$, and trained using noisy stochastic gradient descent (noisy-SGD, see Def. 3). We assume that the network has access to data samples $(x, f(x))$, where $x \sim \mathcal{D}$, for $\mathcal{D}$ being a distribution in $\mathbb{R}^d$ and $f : \mathbb{R}^d \to \{\pm 1\}$ is an unknown target function. We focus on learning parity functions on uniform inputs ($\mathcal{D} = \mathrm{Unif}\{\pm 1\}^d$). A parity function over a subset S of the input coordinates $[d] := \{1, 2, \dots, d\}$ is a function $\chi_S : \{\pm 1\}^d \to \{\pm 1\}$, defined as $\chi_S(x) := \prod_{i \in S} x_i$, where $S \subseteq [d]$. We will focus on the case where $S = [d]$ (full parity) or $|S| = d - O_d(1)$ (almost full parity). Let us define our notion of perturbed initialization.

Definition 1 (Perturbed Initialization). *Consider a neural network with parameters $\theta \in \mathbb{R}^P$ and two independent random vectors $A, H_\sigma \in \mathbb{R}^P$ with independent coordinates where A is arbitrary and H_σ has independent entries $(H_\sigma)_p \sim \mathcal{N}(0, \sigma^2 \cdot \mathbb{I}_P)$. We say that a neural network* $\mathrm{NN}(x;\theta)$ *has a (A, σ)*-perturbed initialization *with noise level σ if its parameters are initialized to $\theta^0_p = A_p + \sqrt{\mathrm{Var} A_p}(H_\sigma)_p$.*

We will mostly consider the case where $A \sim \mathrm{Unif}\{\pm 1\}^P$ (Rademacher initialization). In this scenario, we refer to the initialization as *σ-perturbed* Rademacher.

Theorem 1 (Informal, Positive Full Parity). *Let $f(x) = \chi_d(x)$. A two-layer* ReLU *network with some* $\mathrm{poly}(d)$ *hidden units and σ-perturbed Rademacher initialization with $\sigma = O(d^{-1})$, trained by GD or SGD with any batch-size with the correlation*[4] *or the hinge loss, will learn f to perfect accuracy in* $\mathrm{poly}(d)$ *steps.*

---

[4] The correlation loss is defined as $L_{\mathrm{corr}}(y, \hat{y}) = -y\hat{y}$.

For our negative result, we introduce the following notion of Gradient Alignment.

Definition 2 (Gradient Alignment). *For a neural network* $\mathrm{NN}(x;\theta)$*, an input distribution* $\mathcal{D}$*, a target function* $f:\mathbb{R}^d\to\mathbb{R}$*, and a loss function* $L:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$*, we denote the population gradient as*

$$\Gamma_f(\theta) := \mathbb{E}_x\left[\nabla_\theta L(\mathrm{NN}(x;\theta), f(x))\right] \ . \tag{1}$$

If θ *is a random initialization then we define the* gradient alignment *of* θ *as*

$$\mathrm{GAL}_f(\theta) := \mathbb{E}_\theta\|\Gamma_f(\theta)-\Gamma_r(\theta)\|_2^2 \ , \tag{2}$$

where $\Gamma_r(\theta) := \mathbb{E}_{x,y}[\nabla_\theta L(\mathrm{NN}(x;\theta), y)]$ *for* $y\sim\mathrm{Rad}(1/2)$ *and independent of* x*. That is,* $\Gamma_r(\theta)$ *is the gradient of a random classification task.*

We remark that for the squared and the correlation loss, the Gradient Alignment at initialization corresponds to the Label-Gradient-Alignment of Ortiz-Jiménez et al. (2021); Mok et al. (2022).

We first prove that, under some conditions, if the Gradient Alignment at initialization is small, the network does not learn. We remark that this result holds for general input distributions (beyond Boolean and uniform) and for all networks with a linear output layer (see Section 5.1 for details).

Theorem 2 (Informal, Negative General). *Let* $f:\mathbb{R}^d\to\{\pm1\}$ *be a target function, and let* $\mathrm{NN}(x;\theta)$ *be a neural network with a linear output layer, trained by noisy-GD with noise level* τ *and the correlation loss. Assume either: 1) Gaussian initialization of the weights and homogeneous activation, or 2)* (A,σ)*-perturbed initialization, polynomially bounded gradients, and* τ *small enough (see details in Corollary 1). If* $\mathrm{GAL}_f(\theta^0) < \exp(-\Omega(d))$*, then after* $\mathrm{poly}(d)$ *training steps, the network will achieve an accuracy of at most* $\frac{1}{2}+O(\exp(-\Omega(d)))$*.*

We then apply this result to the case of almost-full parities on uniform inputs.

Theorem 3 (Informal, Negative Almost-Full Parities). *Let* $f(x)=\chi_S(x)$*, for* $S\subseteq[d]$ *such that* $|S|\geq d/2$*. Noisy-GD with correlation loss and any noise level* $\tau=\Omega(1/\mathrm{poly}(d))$ *on any two-layer fully connected* ReLU *network of* $\mathrm{poly}(d)$ *size, initialized with Gaussian initialization will not achieve accuracy better than random guessing in* $\mathrm{poly}(d)$ *training steps.*

We expect Theorem 3 to hold also in case of σ-perturbed Rademacher initialization for $\sigma>\sigma^*$ for some fixed $\sigma^*>0$. To that end in Section 5.2.2 we prove the gradient alignment bound for the hidden layer weights in the perturbed case. Together with a similar bound for the output layer weights (which we omit from this version of the paper) that would give the statement of Theorem 3 also for the σ-perturbed initialization, with $\sigma>\sigma^*$.

Full versions of Theorems 1 and 3 presented in the following sections provide the following rigorous separation between Rademacher and Gaussian initializations: Noisy-GD for correlation loss, when applied to a two-layer fully connected ReLU network with some $\mathrm{poly}(d)$ hidden neurons, can learn the full parity function in $\mathrm{poly}(d)$ steps if the network is initialized with Rademacher weights. However, using Gaussian initialization while leaving all other aspects of the algorithm unchanged requires exponential time to learn. Furthermore, the negative result is robust to details like changing hyperparameters, and as discussed above, both positive and negative results are also valid for some ranges of σ-perturbed Rademacher initializations.

4 POSITIVE RESULT FOR RADEMACHER INITIALIZATION

In both positive and negative results we will be working with the noisy SGD and GD algorithm specified below:

Definition 3 (Noisy-SGD). *Consider a neural network* $\mathrm{NN}(.;\theta)$*, with initialization of the weights* θ^0*. Let* $f:\mathcal{X}\to\mathbb{R}$ *be a target function defined on an input space* $\mathcal{X}$*. Assume we are given fresh samples* $x\sim\mathcal{D}$*, for some input distribution* $\mathcal{D}$ *defined on* $\mathcal{X}$*. Given a weakly differentiable loss function* L*, the updates of the noisy-SGD algorithm with learning rate* γ *are defined by*

$$\theta^{t+1} = \theta^t - \gamma\left(\frac{1}{B}\sum_{s=1}^{B}\nabla_{\theta^t}L(\mathrm{NN}(x^s;\theta^t), f(x^s)) + Z^t\right), \tag{3}$$

where for all $t\in\{0,\dots,T-1\}$*,* Z^t *are i.i.d.* $\mathcal{N}(0,\tau^2)$*, for some* noise level τ*, and they are independent from other variables, and* B *is the batch size. If the average* *over the batch size* $\frac{1}{B}\sum_{s=1}^{B}\nabla_{\theta^t}L(\mathrm{NN}(x^s;\theta^t), f(x^s))$ *is replaced by the population mean* $\mathbb{E}_{x\sim\mathcal{D}}\left[\nabla_{\theta^t}L(\mathrm{NN}(x;\theta^t), f(x))\right]$, *we refer to the algorithm as (full batch)* noisy-GD.

In this section we consider two layer ReLU neural networks with Rademacher initialization for the hidden layer weights. Our results imply that with large enough $\mathrm{poly}(d)$ number of hidden neurons, the hidden layer embedding induced by the Rademacher distribution makes the full parity function linearly separable. Then:

1. When trained with the correlation loss on the uniform input distribution, the network achieves *perfect accuracy in one step of full GD* or in $\mathrm{poly}(d)$ steps of SGD.
2. When trained with the hinge loss on *any input distribution*, the neural network achieves classification error ϵ in $\mathrm{poly}(d)/\epsilon$ steps of SGD.

As mentioned our positive result holds also for a perturbed Rademacher initialization with deviation up to C/d for some constant $C > 0$. We demonstrate this for hinge loss in Section 4.2.

4.1 GD AND SGD WITH CORRELATION LOSS

We consider a fully connected network $N(x) = \sum_{i=1}^{n} v_i \,\mathrm{ReLU}(w_i x + b_i)$. The network is trained with correlation loss $L(y,\hat{y}) = -y\hat{y}$ where only the output layer weights v are trained. This is in contrast to the hinge loss result in Section 4.2 where we allow training of both layers. The gradient of output weights on input $x \in \{\pm 1\}^d$ is given by $\nabla_v L = -f(x)\,\mathrm{ReLU}(Wx+b)$, where W is $n \times d$ matrix with rows $w_1,\ldots,w_n$ and $f(x) = \prod_{i=1}^{d} x_i$ is the full parity function. During training, the inputs are sampled from the uniform distribution on $\{\pm 1\}^d$.

The hidden layer weights w_i are initialized as i.i.d. Rademacher and output weights as $v_i = 0$. The biases are set as $b_i = 0$ if d is even or $b_i = -1$ if d is odd. The precise bias values are not crucial except for “unlucky” choices where the $\Delta_{d,b}$ value defined in equation 15 in Appendix A is too close to zero. In particular learning still holds for random biases for most reasonable distributions (which could be continuous, e.g., standard normal). However, we omit the details in the interest of simplicity.

Theorem 4. 1. *Consider network as above trained for one step with the GD algorithm. If* $n \geq \Omega(d^4)$, *then, except with probability at most* $2\exp(-d)$ *over the choice of initialization, we have* $\mathrm{sign}(N^1(x)) = f(x)$ *for every* $x \in \{\pm 1\}^d$, *where* $N^t(x)$ *denotes the output of the network at time* t. *This conclusion holds also in the presence of GD noise of magnitude* τ *up to* $O(1/d^2)$.

2. *More precisely, if* v^1 *are the output weights after one step of noiseless GD algorithm, and we take* $v^* = v^1/\|v^1\|$, *then,* $|v_i^*| = \frac{1}{\sqrt{n}}$ *for every* $1 \leq i \leq n$ *and, except with probability* $\exp(-d)$, *for all* $x \in \{\pm 1\}^d$,

$$f(x)\sum_{i=1}^{n} v_i^* \,\mathrm{ReLU}(w_i \cdot x + b_i) \geq \frac{\sqrt{n}}{18\sqrt{d}}\,. \tag{4}$$

3. *Consider the above network trained with SGD of any batch size. Let* $n \geq \Omega(d^4)$. *Then, except with probability* $3\exp(-d)$, *after some* $T \geq \mathrm{poly}(d)$ *steps, the network predicts correctly* $\mathrm{sign}(N^T(x)) = f(x)$ *for every* $x \in \{\pm 1\}^d$ *in the presence of GD noise of magnitude* τ *up to* $O(\sqrt{T}/d^2)$.

Remark 1. *The bound for the number of neurons* $n \geq \Omega(d^4)$ *is somewhat loose as we use crude bounds on* ReLU. *In the interest of simplicity we do not attempt to optimize the bound. At the same time, we present the result for ReLU as it is a common choice for the activation function. A similar argument computing an expression analogous to equation 15 for a bounded activation would give* $n \geq \Omega(d^2)$ *bound. In particular, this can be made rigorous for a clipped ReLU activation, e.g.,* $\sigma(x) = \max(0, \min(x, 5))$. *We omit the details in this version of the paper.*

4.2 SGD ANALYSIS FOR HINGE LOSS

One of the implications of Theorem 4 is that under Rademacher initialization, with high probability the hidden layer embeddings of the parity function are linearly separable. We use known techniques (in particular, we borrow parts of the analysis from Nachum & Yehudayoff (2020)) to show that this implies learning for SGD under the hinge loss.

As before, we consider the two layer architecture, this time with possibly perturbed Rademacher hidden layer initialization $N(x) = \sum_{i=1}^{n} v_i \operatorname{ReLU}((w_i + g_i) \cdot x + b_i)$, that is $w_{ij} \sim \operatorname{Rad}(1/2)$ and $g_{ij} \sim \mathcal{N}(0, \sigma^2)$. Other weights are initialized as before, i.e., hidden layer biases are $b_i = 0$ for d even and $b_i = -1$ for d odd, and output layer weights are $v_i = 0$. As in the case of the correlation loss, the exact bias values are not crucial.

The training is with hinge loss $L_\beta(y, \hat{y}) = \max(0, \beta - y\hat{y})$ for some $\beta \geq 0$ under i.i.d. samples from *any fixed probability distribution* on $\{\pm 1\}^d$. For simplicity we consider batch size 1 SGD, though larger batches could also be used. This time we allow a more realistic setting where both layers are trained.

Theorem 5. *For the network described above, for $\sigma \leq C/d$ for sufficiently small $C > 0$, except with probability $3\exp(-d)$ over the choice of initialization the following holds:*

Let $\mathcal{D}$ be a distribution on $\{\pm 1\}^d$, $\epsilon > 0$ and $0 < \delta \leq 1/2$. If $n \geq \Omega(d^4)$ and $n \leq \operatorname{poly}(d)$, then after training with batch size one SGD for some choices of $T = \operatorname{poly}(d)\frac{1}{\epsilon}\ln\frac{1}{\delta}$ and learning rate $\gamma = 1/\operatorname{poly}(d)$, using hinge loss L_β for $0 \leq \beta \leq O(d^2 n\gamma)$, except with probability δ over the choice of i.i.d. training samples from $\mathcal{D}$, it holds $\Pr_{x\sim\mathcal{D}}\left[\operatorname{sign}(N^T(x)) \neq f(x)\right] \leq \epsilon$.

Theorem 5 follows from the bound on the number of nonzero SGD updates:

Theorem 6. *If $\sigma \leq C/d$ for sufficiently small $C > 0$, $n \geq \Omega(d^4)$ and $n \leq \operatorname{poly}(d)$, then, except with probability $3\exp(-d)$ over the choice of initialization, the above network trained with batch size one SGD algorithm on the full parity function on any sequence of samples from $\{\pm 1\}^d$ with learning rate $0 < \gamma \leq O(d^{-3.5})$ and the hinge loss L_β for $0 \leq \beta \leq 16d^2 n\gamma$, will perform at most $O(d^3)$ nonzero updates.*

5 NEGATIVE RESULTS

5.1 NEGATIVE RESULTS FOR GENERAL TARGETS

In this section we prove a negative result that holds for all neural networks with a linear output layer:

Definition 4 (Linear Output Layer)**.** *We say that a neural network $\operatorname{NN}(x;\theta)$ has* linear output layer *if its output can be written as $\operatorname{NN}(x;\theta) = \sum_{i=1}^{n} v_i \operatorname{NN}_i(x;\psi)$, where $\theta = (v, \psi)$ are the trainable weights of the network, and n denotes the number of neurons in the last hidden layer.*

In the context of binary classification, the network's ±1 label prediction is given by $\operatorname{sign}(\operatorname{NN}(x;\theta))$. Let us state our main negative result.

Theorem 7 (Negative Result for General Targets)**.** *Let $\operatorname{NN}(x;\theta)$ be a network with a linear output layer. Let the weights θ^0 be initialized according to an (A,σ)-perturbed initialization (Def. 1), for $A \in \mathbb{R}^P$ with independent coordinates with distributions symmetric around 0. Assume the network is given samples $(x, f(x))$ where $x \sim \mathcal{D}$, for $\mathcal{D}$ being a distribution on $\mathbb{R}^d$. Let $\operatorname{NN}(x;\theta^T)$ be the output of the noisy-GD algorithm with noise level τ and learning rate γ after T steps of training with the correlation loss. Assume that there exists some bound $\varepsilon > 0$ such that for every $0 \leq \lambda^2 \leq T\gamma^2\tau^2$ we have*

$$\operatorname{GAL}_f(\theta^0 + \lambda H) \leq \varepsilon \;, \tag{5}$$

where $H \sim \mathcal{N}(0, \mathbb{I}_P)$. Then, $\mathbb{P}\Big[\operatorname{sign}(\operatorname{NN}(x;\theta^T)) = f(x)\Big] \leq \frac{1}{2} + \frac{T\sqrt{\varepsilon}}{2\tau}$.

The proof is deferred to Appendix B. In words, this theorem states that if equation 5 holds for ε which is small compared to the noise level τ, then noisy gradient descent will require a large number of training steps to achieve performance better than random guessing. Therefore, even the weakest form of learning is impossible. Let us make a few remarks.

Remark 2. *For simplicity, we present Theorem 7 in the context of full batch noisy-GD. However, we note that the proof can be extended to noisy-SGD with a sufficiently large batch size, by leveraging the concentration of the effective gradient around the population mean, similarly to e.g. (Abbe & Sandon (2020), Theorem 3).*

Remark 3. *We propose using* GAL_f *as a measure for hardness of learning. However, the condition in equation 5 requires verifying that* GAL_f *remains small for all Gaussian perturbations of the initialization, with variance within the specified range. In Corollaries 1 and 2, we demonstrate that, in some settings, the condition in equation 5 can be simplified and expressed uniquely in terms of* $\mathrm{GAL}_f(\theta^0)$.

Remark 4. *We emphasize that Theorem 7 applies to any binary classification task and network architecture with a linear output layer, unlike, for example, Abbe et al. (2022c), which is specific to Boolean functions and product measures. Importantly, our result is restricted to the correlation loss, as the proof relies on coupling the gradient descent dynamics with the 'junk flow' — i.e., the dynamics of a network trained on random noise (def . 6). This coupling requires tracking the distribution of the junk flow weights, which simplifies under the correlation loss assumption. We empirically observe that also for hinge loss, the* GAL_f *remains small along the junk flow over time (see Figure 2 in Section 6).*

As a first corollary, we show that when the GD noise level τ is small compared to the variances in the initial H_σ, the distributions of H_σ and $H_\sigma + \lambda H$ are similar. As a result, equation 5 can be expressed in terms of $\mathrm{GAL}_f(\theta^0)$.

Corollary 1. *Let* $f : \mathbb{R}^d \to \{\pm 1\}$ *be a target function under a given input distribution* $\mathcal{D}$. *Let* $\mathrm{NN}(x;\theta)$ *be network with linear output layer, with weights initialized according to an* (A,σ)-*perturbed initialization, for 0-symmetric independent* $A \in \mathbb{R}^P$. *Assume that* $\|\mathbb{E}_x|\nabla\,\mathrm{NN}(x;\theta)|\|_2^2 \leq R$ *for all* θ.[5] *Let* $\mathrm{NN}(x;\theta^T)$ *be the output of the noisy-GD algorithm with noise level* τ *and learning rate* γ *such that* $\tau^2 \leq \frac{\sigma^2 \min_p \mathrm{Var} A_p}{PT\gamma^2}$, *after* T *steps with the correlation loss. Then,*

$$\mathbb{P}(\mathrm{NN}(x;\theta^T) = f(x)) \leq \frac{1}{2} + \frac{T\sqrt{4R+1}}{2\tau} \cdot \mathrm{GAL}_f(\theta^0)^{1/18}. \tag{6}$$

The proof of Corollary 1 is deferred to Appendix B.3. While the above corollary applies to general perturbed initializations, it relies on the assumption that the GD noise level τ is sufficiently small. However, we also show that in the specific case of Gaussian initialization and assuming a homogeneous architecture, this assumption can be removed.

Gaussian Initialization. Let us restrict ourselves to the special case of Gaussian initialization, i.e. when $A = 0_P$. We assume that the activation h satisfies the following homogeneity property.

Definition 5 (H-Weakly Homogeneous.)**.** *Let* $h : \mathbb{R} \to \mathbb{R}$ *be an activation function. We say that* h *is* H*-weakly homogeneous if for all* $x \in \mathbb{R}$ *and* $C \geq 0$, $h(Cx) = C^H h(x)$.

For example, $\mathrm{ReLU}(x) = \max\{0, x\}$ is 1-weakly homogeneous. x^k is k-weakly homogeneous, for all $k \in \mathbb{N}$. We prove the following result.

Corollary 2. *Let* $\mathrm{NN}(x;\theta)$ *be a fully connected network of depth* L, *with* H*-weakly homogeneous activation and with weights initialized as* $\theta_p^0 \sim \mathcal{N}(0, \sigma_{l_p}^2)$ *where* l_p *denotes the layer of parameter* θ_p, *for* $p \in [P]$. *Let* $f : \mathbb{R}^d \to \{\pm 1\}$ *be a balanced target function. Let* $\mathrm{NN}(x;\theta^T)$ *be the output of the noisy-GD algorithm with noise-level* τ, *after* T *steps of training with the correlation loss. Then,*

$$\mathbb{P}(\mathrm{NN}(x;\theta^T) = f(x)) \leq \frac{1}{2} + \frac{T}{2\tau}\prod_{l=1}^{L}\left(1 + \frac{T\gamma^2\tau^2}{\sigma_l^2}\right)^H \cdot \mathrm{GAL}_f(\theta^0)^{1/2}. \tag{7}$$

5.2 NEGATIVE RESULTS FOR HIGH-DEGREE PARITIES

5.2.1 SMALL ALIGNMENT FOR GAUSSIAN INITIALIZATION

In this section we state a rigorous lower bound for learning of large degree parities with pure Gaussian initialization. This is in the setting of two layer ReLU neural networks. Then we will discuss extending this result to a perturbed Rademacher initialization for a large enough constant perturbation.

---

[5] This always holds if we assume gradient clipping.

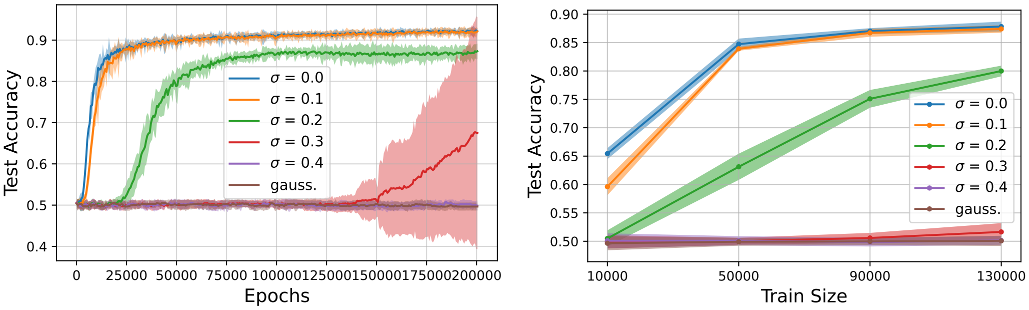

Figure 1: Learning the full parity with σ-perturbed initialization by SGD with the hinge loss on a 4-layer MLP, with $d = 50$, with online fresh samples (left) and with an offline fixed dataset (right).

Theorem 8. *Let $\theta = (w, b, v)$ for $w \in \mathbb{R}^{n\times d}, b \in \mathbb{R}^n, v \in \mathbb{R}^n$ and $\mathrm{NN}(x;\theta) = \sum_{i=1}^n v_i \,\mathrm{ReLU}(w_i \cdot x + b_i)$. Let $a = a(d) \leq d/2$ and $f_a(x) = \prod_{i=1}^{d-a} x_i$. Consider the i.i.d. initialization $w \sim \mathcal{N}\left(0, \frac{1}{d}\,\mathrm{Id}_{n\times d}\right)$, $b \sim \mathcal{N}(0, \sigma^2\,\mathrm{Id}_n)$ for any $\sigma^2 = O(1)$, $v \sim \mathcal{N}\left(0, \frac{1}{n}\,\mathrm{Id}_n\right)$.*

Then, for any number of hidden neurons $n = \exp(o(d))$, any number of time steps $T = \exp(o(d))$, any learning rate $0 \leq \gamma \leq \exp(o(d))$, any noise level $\exp(-o(d)) \leq \tau \leq \exp(o(d))$, after T steps of the noisy GD algorithm with correlation loss,

$$\Pr\left[\mathrm{sign}(\mathrm{NN}^T(x;\theta)) = f(x)\right] \leq \frac{1}{2} + \exp(-\Omega(d)) \ . \tag{8}$$

Theorem 8 follows from Theorem 7 and the following bound on the gradient alignment:

Proposition 1. *Let a neural network be as in Theorem 8. Then, for every $\sigma_0^2 > 0$, there exists $C, C' > 0$ such that for any network with $\sigma^2 \leq \sigma_0^2$ we have a gradient alignment bound*

$$\mathrm{GAL}_{f_a}(\theta) \leq PC' \exp(-Cd) \ , \tag{9}$$

where $P := nd + 2n$ is the total number of parameters.

The proofs of Theorem 8 and Proposition 1 can be found in Appendix C.

5.2.2 SMALL ALIGNMENT FOR PERTURBED INITIALIZATION

Consider the perturbed Rademacher initialization $r + g$ for $g \sim \mathcal{N}(0, \sigma^2)$ for some constant $\sigma > 0$. In order to prove a rigorous lower bound like in Theorem 8 for this initialization, we need to establish the alignment bound for $\mathrm{GAL}_f(r + g)$. Once this bound is proved, the remaining steps of the proof are the same as for Theorem 8.

Here, we show this alignment bound for large enough $\sigma \geq \sigma_0 > 0$ for some universal σ_0, in the case of the gradient of hidden layer weights. According to correlation loss gradient formulas for the full parity function, the bound boils down to calculating the following quantity (assuming zero biases for simplicity):

$$\mathrm{GAL}(\sigma, d) = \mathbb{E}_{r,g}\left[\left(\mathbb{E}_x\left[\left(\prod_{i=1}^d x_i\right)\cdot \mathbb{1}[(r+g)\cdot x \geq 0]\right]\right)^2\right] \ . \tag{10}$$

Theorem 9. *There exists some $\sigma_0, C > 0$ and D_0 such that, for $d \geq D_0$, and σ-perturbed Rademacher initialization $w = r + g$ for $\sigma \geq \sigma_0$, it holds $\mathrm{GAL}(\sigma, d) \leq \exp(-Cd)$.*

The proof of Theorem 9 is deferred to Appendix D.

6 EXPERIMENTS

In this section, we show empirical results on the impact of the initialization in learning the full parity. As our model, we use a multi-layer perceptron (MLP) with 3 hidden layers of neurons sizes 512,

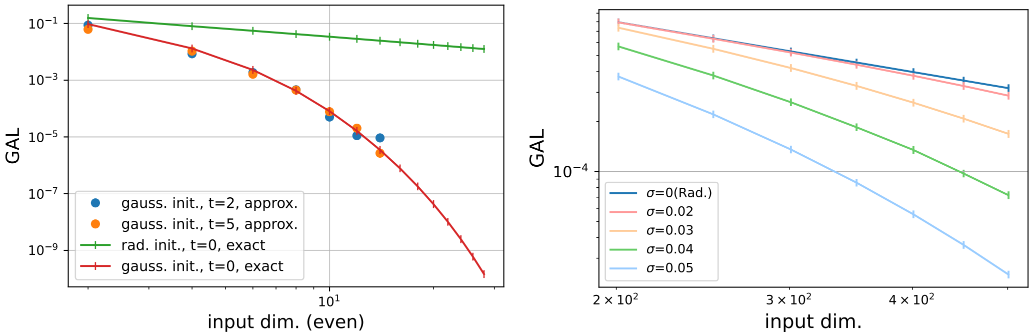

Figure 2: Computing numerically the alignment GAL_f with the hinge loss (left) and the correlation loss (right), for a one-neuron network.

512 and 64 with ReLU activation, and we train it with SGD with batch size 64 on the hinge loss, training all layers simultaneously. Each experiment is repeated for 7 random seeds and we report the 95% confidence intervals. In Appendix E, we report further experiment details, as well as additional experiments.

σ-Perturbed Initialization. We first consider learning the full parity function with σ-perturbed initializations and investigate the effect of varying σ (Figure1). To make different initializations comparable, we normalize them such that the variance entering each neuron is 1 (see Appendix E for details). We observe that the test accuracy after training decreases as σ increases. This pattern is seen in both the online setting (left plot), where fresh batches are sampled at each iteration, and the offline setting (right plot), where the network is trained on a fixed dataset until the training loss decreases to 10^{-2}, and evaluated on a separate test set. For input dimension $d = 50$, as in Figure 1, we find that some learning occurs for $\sigma \in \{0.1, 0.2\}$. However, in the Appendix, we report experiments with larger input dimensions, where learning does not occur for these values of σ (Figure5).

Gradient Alignment. In Figure 2 we compute the gradient alignment for a one-neuron ReLU network under different initializations and losses, which are not covered by our theoretical results. The left plot shows the GAL_f at initialization for the hinge loss with Rademacher and Gaussian initializations, across input dimensions up to 30 (solid lines). We observe that GAL_f decreases at an inverse-polynomial rate for Rademacher, but super-polynomially fast for Gaussian initialization. We also estimate, with Montecarlo, the GAL_f after training the neuron for a few steps ($t = 2$ and $t = 5$) on random labels (dots). We observe that training on random labels does not increase the GAL_f. A theoretical understanding of this observation would allow to extend our negative result to the hinge loss.

In the right plot, we estimate numerically the initial GAL_f for the correlation loss for a one-layer threshold neuron (equation 10). We consider σ-perturbed initializations with small σ, contrasting Theorem 9 which bounds GAL_f only for large σ. For small σ, GAL_f deviates from the Rademacher case, suggesting that it could be super-polynomially small for all constants $\sigma > 0$. Further investigation for small σ is left for future work, and Appendix E shows that GAL_f remains super-polynomially small for larger values of σ, confirming our theory.

Other Perturbed Initializations. We next explore perturbations beyond mixtures of Gaussians. In Figure 3, we consider three types: 1) a mixture of two continuous uniform distributions with means $+1$ and -1, and standard deviations $\sigma \in \{0.1, 1.0\}$; 2) a sparsified Rademacher initialization, where a fraction $s \in \{1/2, 1/3, 1/5\}$ of the weights are set to 0, and the rest follow a $\mathrm{Rad}(1/2)$ distribution; and 3) a symmetric discrete initialization, where the weights are randomly chosen from $\{-2, -1, 1, 2\}$. We find that the mixture of continuous uniforms behaves similarly to the mixture of Gaussians: for $\sigma = 0.1$ and input dimension $d = 50$, the network successfully learns, but learning is prevented at larger σ. Additionally, we observe that all other discrete initializations fail to enable learning, suggesting that the Rademacher initialization is a special case.

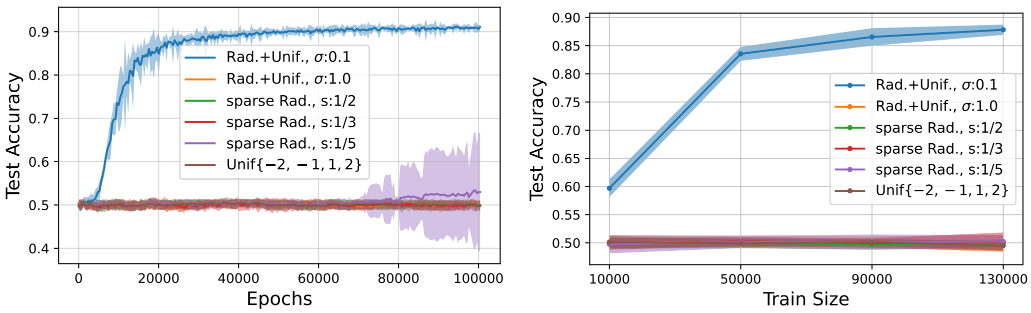

Figure 3: Learning the full parity with perturbations of the Rad. initialization by SGD with the hinge loss on a 4-layer MLP, with $d = 50$, with online fresh samples (left) and with an offline dataset (right).

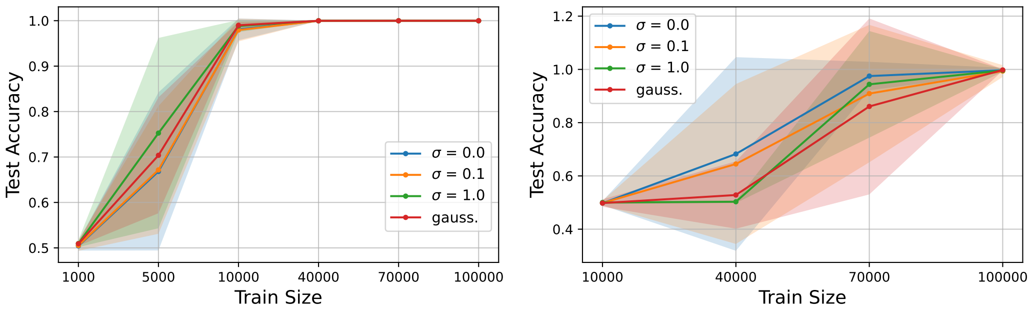

Figure 4: Learning 3-parity (left) and 5-parity (right) with Rademacher, σ-perturbed and Gaussian initializations, with SGD with the hinge loss on a 4-layer MLP, with $d = 50$. We plot the test accuracy, for several training set sizes.

Sparse Parities. In Figure 4 we train a 4-layer MLP with Rademacher initialization and σ-perturbation ($\sigma \in 0.1, 1$) on two sparse parities: degree 3 (left) and degree 5 (right). We observe no significant difference between these initializations, unlike the full parity case. This is because, for sparse parities, the learning bottleneck lies in recovering the support, which takes $d^{\Omega(k)}$ time for any i.i.d. initialization. Hence, the initial embedding does not play the same role as in the full parity scenario.

7 CONCLUSION

In this paper, we advance the understanding of whether high degree parities can be learned using noisy-GD on standard neural networks with i.i.d. initializations. Specifically, we show that while the full parity is easily learnable with Rademacher initialization, it becomes challenging when Gaussian perturbations with large variance are introduced. This constitutes a separation between SQ algorithms and gradient descent on neural networks: the full parity is an example of a function that is trivially learnable in the statistical query (SQ) framework but difficult for noisy-GD on neural networks with most typical initializations, with the Rademacher being a special case. It raises interesting questions about a threshold where learning behavior changes based on the perturbation level σ. Additionally, we propose a novel, loss-dependent measure for assessing alignment between the initialization and the target distribution, and prove a negative result for the correlation loss that applies to general input distributions, beyond the specific case of full parity and Boolean inputs. We leave to future work strengthening of that result, e.g., to hinge loss and/or deeper architectures.

REFERENCES

Mathoverflow. `https://mathoverflow.net/questions/351523/gaussian-concentration-inequality/`, 2020.

Emmanuel Abbe and Enric Boix-Adsera. On the non-universality of deep learning: quantifying the cost of symmetry. *arXiv preprint arXiv:2208.03113*, 2022.

Emmanuel Abbe and Colin Sandon. On the universality of deep learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20061–20072, 2020.

Emmanuel Abbe, Pritish Kamath, Eran Malach, Colin Sandon, and Nathan Srebro. On the power of differentiable learning versus PAC and SQ learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks. In *Conference on Learning Theory*, pp. 4782–4887. PMLR, 2022a.

Emmanuel Abbe, Samy Bengio, Elisabetta Cornacchia, Jon Kleinberg, Aryo Lotfi, Maithra Raghu, and Chiyuan Zhang. Learning to reason with neural networks: Generalization, unseen data and boolean measures. *Advances in Neural Information Processing Systems*, 35:2709–2722, 2022b.

Emmanuel Abbe, Elisabetta Cornacchia, Jan Hazla, and Christopher Marquis. An initial alignment between neural network and target is needed for gradient descent to learn. In *International Conference on Machine Learning*, pp. 33–52. PMLR, 2022c.

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 2552–2623. PMLR, 2023.

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, Colin Sandon, and Omid Saremi. How far can transformers reason? the locality barrier and inductive scratchpad. *arXiv preprint arXiv:2406.06467*, 2024a.

Emmanuel Abbe, Elisabetta Cornacchia, and Aryo Lotfi. Provable advantage of curriculum learning on parity targets with mixed inputs. *Advances in Neural Information Processing Systems*, 36, 2024b.

Gerard Ben Arous, Reza Gheissari, and Aukosh Jagannath. Online stochastic gradient descent on non-convex losses from high-dimensional inference. *Journal of Machine Learning Research*, 22 (106):1–51, 2021.

Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

Stanford S Bonan and Dean S Clark. Estimates of the hermite and the freud polynomials. *Journal of Approximation Theory*, 63(2):210–224, 1990.

Joan Bruna, Loucas Pillaud-Vivien, and Aaron Zweig. On single index models beyond gaussian data. *arXiv preprint arXiv:2307.15804*, 2023.

Elisabetta Cornacchia and Elchanan Mossel. A mathematical model for curriculum learning for parities. In *International Conference on Machine Learning*, pp. 6402–6423. PMLR, 2023.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *J. Mach. Learn. Res.*, 13(1):795–828, mar 2012. ISSN 1532-4435.

Alex Damian, Loucas Pillaud-Vivien, Jason D Lee, and Joan Bruna. The computational complexity of learning gaussian single-index models. *arXiv preprint arXiv:2403.05529*, 2024.

Amit Daniely and Eran Malach. Learning parities with neural networks. *Advances in Neural Information Processing Systems*, 33:20356–20365, 2020.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Pareto frontiers in neural feature learning: Data, compute, width, and luck. *arXiv preprint arXiv:2309.03800*, 2023.

Vitaly Feldman. A general characterization of the statistical query complexity. *arXiv preprint arXiv:1608.02198*, 2016.

Margalit Glasgow. Sgd finds then tunes features in two-layer neural networks with near-optimal sample complexity: A case study in the xor problem. *arXiv preprint arXiv:2309.15111*, 2023.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Michael Hahn and Mark Rofin. Why are sensitive functions hard for transformers? *arXiv preprint arXiv:2402.09963*, 2024.

Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. *Advances in neural information processing systems*, 31, 2018.

Bernard Harris and Andrew P Soms. The use of the tetrachoric series for evaluating multivariate normal probabilities. *Journal of Multivariate Analysis*, 10(2):252–267, 1980.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. *arXiv preprint arXiv:1909.12292*, 2019.

Nirmit Joshi, Theodor Misiakiewicz, and Nathan Srebro. On the complexity of learning sparse functions with statistical and gradient queries. *arXiv preprint arXiv:2407.05622*, 2024.

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

Yiwen Kou, Zixiang Chen, Quanquan Gu, and Sham M Kakade. Matching the statistical query lower bound for k-sparse parity problems with stochastic gradient descent. *arXiv preprint arXiv:2404.12376*, 2024.

Siddharth Krishna Kumar. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*, 2017.

N N Lebedev. *Special functions and their applications*. Courier Corporation, 1972. Translated by Richard A Silverman.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer, 2013.

Eran Malach and Shai Shalev-Shwartz. Computational separation between convolutional and fully-connected networks, 2020. URL `https://arxiv.org/abs/2010.01369`.

Eran Malach, Pritish Kamath, Emmanuel Abbe, and Nathan Srebro. Quantifying the benefit of using differentiable learning over tangent kernels. In *International Conference on Machine Learning*, pp. 7379–7389. PMLR, 2021.

Jisoo Mok, Byunggook Na, Ji-Hoon Kim, Dongyoon Han, and Sungroh Yoon. Demystifying the neural tangent kernel from a practical perspective: Can it be trusted for neural architecture search without training?, 2022. URL `https://arxiv.org/abs/2203.14577`.

Ido Nachum and Amir Yehudayoff. On symmetry and initialization for neural networks. In *LATIN 2020: Theoretical Informatics: 14th Latin American Symposium, São Paulo, Brazil, January 5-8, 2021, Proceedings 14*, pp. 401–412, 2020.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. doi: 10.1017/CBO9781139814782.

Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. What can linearized neural networks actually say about generalization? 2021. doi: 10.48550/ARXIV.2106.06770. URL `https://arxiv.org/abs/2106.06770`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147. PMLR, 2013.

Taiji Suzuki, Denny Wu, Kazusato Oko, and Atsushi Nitanda. Feature learning via mean-field langevin dynamics: classifying sparse parities and beyond. *Advances in Neural Information Processing Systems*, 36, 2024.

Oldrich Alfons Vasicek. A series expansion for the bivariate normal integral. *Journal of Computational Finance*, 1(4):5–10, 1998.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Advances in Neural Information Processing Systems*, 32, 2019.

Chiyuan Zhang, Maithra Raghu, Jon M. Kleinberg, and Samy Bengio. Pointer value retrieval: A new benchmark for understanding the limits of neural network generalization. *ArXiv*, abs/2107.12580, 2021.

Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019.

A PROOFS FOR SECTION 4

A.1 PROOF OF THEOREM 4

A useful identity to be remembered for later is, for every $x, w \in \{\pm 1\}^d$:

$$\prod_{j=1}^{d} x_j w_j = (-1)^{(d - w\cdot x)/2} \,. \tag{11}$$

1. First, consider the case without noise. One step of GD with learning rate γ results in the following update:

$$v_i^{t+1} = v_i^t + \gamma \mathbb{E}_x \left(\prod_{j=1}^{d} x_j\right) \mathrm{ReLU}(w_i \cdot x + b_i) \tag{12}$$

$$= v_i^t + \gamma \mathbb{E}_x \left(\prod_{j=1}^{d} w_{ij}\right) (-1)^{(d - w_i\cdot x)/2}\, \mathrm{ReLU}(w_i \cdot x + b_i) \tag{13}$$

$$= v_i^t + \gamma \left(\prod_{j=1}^{d} w_{ij}\right) \Delta_{d,b_i} \,, \tag{14}$$

where we define

$$\Delta_{d,b} := \mathbb{E}_x \left[(-1)^{(d-\sum_j x_j)/2} \operatorname{ReLU}\left(\sum_{j=1}^{d} x_j + b \right) \right] . \tag{15}$$

Since $w_i \cdot x$ is distributed as a sum of i.i.d. Rademachers regardless of w_i, the value in equation 13 indeed can be replaced with the factor Δ_{d,b_i} which does not depend on w_i.

Accordingly, after one step of GD for starting zero weights $v^0 = 0$ and fixed $b \in \mathbb{R}^n$ the output of the network is given by

$$N^1(x) = \sum_{i=1}^{n} \gamma \Delta_{d,b_i} \prod_{j=1}^{d} w_{ij} \operatorname{ReLU}(w_i \cdot x + b_i) . \tag{16}$$

For fixed $x \in \{\pm 1\}^d$ and in expectation over w, this is, using equation 11,

$$\begin{aligned} \mathbb{E}_w N^1(x) &= \gamma \sum_{i=1}^{n} \Delta_{d,b_i} \mathbb{E}_w \left[\left(\prod_{j=1}^{d} w_{ij} \right) \operatorname{ReLU}(w_i \cdot x + b_i) \right] && (17) \\ &= \gamma \left(\prod_{j=1}^{d} x_j \right) \sum_{i=1}^{n} \Delta_{d,b_i} \mathbb{E}_w \left[(-1)^{(d-w_i \cdot x)/2} \operatorname{ReLU}(w_i \cdot x + b_i) \right] && (18) \\ &= \gamma \left(\prod_{j=1}^{d} x_j \right) \sum_{i=1}^{n} \Delta_{d,b_i}^2 . && (19) \end{aligned}$$

We turn to developing a formula for $\Delta_{d,b}$:

Lemma 1. *Let $d > 1$, $b \in \mathbb{R}$ and $c := \lceil (d-b)/2 \rceil$. Then,*

$$\Delta_{d,b} = (-1)^{d+c} 2^{-d} \left(d\binom{d-2}{c-2} - d\binom{d-2}{c-1} + b\binom{d-1}{c-1} \right) . \tag{20}$$

Proof. Throughout this proof, we follow the convention $\binom{d}{k} = 0$ for $k < 0$ or $k > d$. First, observe that for any integer $d, c > 1$:

$$\sum_{k=c}^{d} (-1)^k \binom{d}{k} = \sum_{k=c}^{d} (-1)^k \left(\binom{d-1}{k} + \binom{d-1}{k-1} \right) = (-1)^c \binom{d-1}{c-1} , \tag{21}$$

$$\sum_{k=c}^{d} (-1)^k k \binom{d}{k} = d \sum_{k=c}^{d} (-1)^k \binom{d-1}{k-1} = (-1)^c d \binom{d-2}{c-2} . \tag{22}$$

Recall that x in the definition of $\Delta_{d,b}$ is distributed as i.i.d. uniform Rademachers. Therefore, we can write $x_j = -1 + 2z_j$, where z are i.i.d uniform Bernoullis. Using that, the equations above and the definition of $\Delta_{d,b}$:

$$\begin{aligned} \Delta_{d,b} &= (-1)^d \mathbb{E}_z \left[(-1)^{\sum_j z_j} \operatorname{ReLU}\left(b - d + 2 \sum_{j=1}^{d} z_j \right) \right] && (23) \\ &= (-1)^d 2^{-d} \sum_{k=c}^{d} (-1)^k \binom{d}{k} (b - d + 2k) , && (24) \\ &= (-1)^{d+c} 2^{-d} \left((b-d)\binom{d-1}{c-1} + 2d\binom{d-2}{c-2} \right) && (25) \\ &= (-1)^{d+c} 2^{-d} \left(d\binom{d-2}{c-2} - d\binom{d-2}{c-1} + b\binom{d-1}{c-1} \right) . \end{aligned}$$

□

Corollary 3. *For every $d > 1$, let $b(d) = 0$ if d is even and $b(d) = -1$ if d is odd. It holds*

$$\frac{1}{3\sqrt{d}} \le \left|\Delta_{d,b(d)}\right| \le \frac{1}{\sqrt{d}}. \tag{26}$$

Proof. In the even case, $c = d/2$ and:

$$|\Delta_{d,0}| = \frac{d}{2^d}\left(\binom{d-2}{d/2-1} - \binom{d-2}{d/2-2}\right) = \frac{4}{2^d}\binom{d-3}{d/2-1} \tag{27}$$

$$\in \left[\frac{1}{3\sqrt{d}}, \frac{1}{\sqrt{d}}\right] , \tag{28}$$

where in the last line we applied an estimate $\frac{2^d}{8}\frac{2}{3\sqrt{d}} \le \binom{d-3}{d/2-1} \le \frac{2^d}{8}\frac{2}{\sqrt{d}}$. In the odd case it holds $c = (d+1)/2$, and we proceed similarly

$$|\Delta_{d,-1}| = \frac{1}{2^d}\binom{d-1}{(d-1)/2} \in \left[\frac{1}{3\sqrt{d}}, \frac{1}{\sqrt{d}}\right] . \qquad \square$$

Let us come back to the expression $f(x)N^1(x)$ for a fixed $x \in \{\pm 1\}^d$. Its value is a random variable depending on the hidden layer initialization W. By equation 16, it can be written as a sum of n i.i.d. random variables, and by Corollary 3 each of them has absolute value at most $2\gamma\sqrt{d}$. Furthermore, it follows from equation 19 and Corollary 3 that $\mathbb{E}_w f(x)N^1(x) \ge \frac{n\gamma}{9d}$. Therefore, we can upper bound the prediction error probability by Hoeffding's inequality:

$$\Pr_w[f(x)N^1(x) \le 0] \le \Pr_w\left[f(x)N^1(x) \le \frac{\gamma n}{18d}\right] \le \exp\left(-\frac{n}{9 \cdot 2^5 \cdot d^3}\right) \le \exp(-2d) , \tag{29}$$

where the last inequality holds for $n > \Omega(d^4)$. Therefore, by union bound, the network will make correct predictions $f(x)N^1(x) > 0$ for all $x \in \{\pm 1\}^d$ except with probability $\exp(-d)$.

In the presence of gradient noise, the weights are given as $\tilde{v}^1 = v^1 + \gamma\xi$, where $\xi \sim \mathcal{N}(0, \tau^2 \operatorname{Id})$. Then,

$$f(x)\tilde{N}^1(x) = f(x)N^1(x) + \gamma f(x)\sum_{i=1}^{n} \xi_i \operatorname{ReLU}(w_i \cdot x + b_i) \ge f(x)N^1(x) - 2\gamma d\sum_{i=1}^{n} |\xi_i| . \tag{30}$$

Using equation 29, except with probability $\exp(-d)$, we will have $f(x)\tilde{N}^1(x) > 0$ for every x as long as $2d\sum_i |\xi_i| \le n/18d$, or equivalently $\sum_i |\xi_i| \le n/36d^2$. Note that $\mathbb{E}|\xi_i| = \tau\sqrt{2/\pi}$, so by assumption $\tau \le O(1/d^2)$ we have $\mathbb{E}\sum_i |\xi_i| \le n/72d^2$. Furthermore, as ξ_i has Gaussian distribution, its absolute value $|\xi_i|$ is sub-Gaussian (see, e.g., Proposition 2.5.2 in Vershynin (2018)). Therefore, by sub-Gaussian concentration, we can estimate

$$\Pr\left[\sum_{i=1}^{n} |\xi_i| \ge \frac{n}{36d^2}\right] \le \Pr\left[\sum_{i=1}^{n} |\xi_i| - \mathbb{E}|\xi_i| \ge \frac{n}{72d^2}\right] \le \exp\left(-\Omega\left(\frac{n}{d^4\tau^2}\right)\right) \le \exp(-d) , \tag{31}$$

where the last inequality holds as $n = \Omega(d^4)$ and $\tau^2 = O(1/d^4)$. All in all, the noisy network classifies all inputs correctly except with probability at most $2\exp(-d)$.

2. By equation 29, except with probability $\exp(-d)$ for every $x \in \{\pm 1\}^d$ we have

$$f(x)N^1(x) = f(x)\sum_{i=1}^{n} v_i^1 \operatorname{ReLU}(w_i \cdot x + b_i) \ge \frac{\gamma n}{18d} .$$

Recall that $v_i^1 = \gamma\Delta_{d,b}\prod_{j=1}^{d} w_{ij}$ and $v^* = v^1/\|v^1\|$. In particular, it follows $\|v_i^*\| = 1/\sqrt{n}$ for every i. By Corollary 3 it follows that $\|v\| \le \frac{\gamma\sqrt{n}}{\sqrt{d}}$. Finally,

$$f(x)\sum_{i=1}^{n} v_i^* \operatorname{ReLU}(w_i \cdot x + b_i) \ge \frac{\gamma n}{18d\|v\|} \ge \frac{\sqrt{n}}{18\sqrt{d}} .$$

3. In the general case of noisy SGD, let $v = v^1 \in \mathbb{R}^n$ be the update given by GD, that is $v_i = \mathbb{E}_x f(x)\,\mathrm{ReLU}(w_i \cdot x + b_i)$. The SGD update can be written as

$$\hat{v}_i^{t+1} = \hat{v}_i^t + \gamma e_i^t + \gamma \xi_i^t \; , \tag{32}$$

where: (a) e_i^t for $1 \le i \le n$ is a random variable with expectation $\mathbb{E} e_i^t = v_i$ and bounded by $|e_i^t| \le 2d$; (b) $\xi^t \sim \mathcal{N}(0, \tau^2 \,\mathrm{Id})$; and where those random variables are independent across time.

From equation 29, if $n > \Omega(d^4)$, except with probability $\exp(-d)$ over the choice of hidden layer weights w, for every $x \in \{\pm 1\}^d$ it holds

$$f(x) \sum_{i=1}^n v_i \,\mathrm{ReLU}(w_i \cdot x + b_i) > \frac{\gamma n}{18d} \; . \tag{33}$$

Let $x \in \{\pm 1\}^d$. We estimate

$$f(x)\hat{N}^T(x) = f(x)\gamma \sum_{i=1}^n \Big(T v_i + \sum_{t=1}^T e_i^t - v_i + \xi_i^t\Big)\,\mathrm{ReLU}(w_i \cdot x + b_i) \tag{34}$$

$$> \frac{\gamma T n}{18d} - 2\gamma d \left(\sum_{i=1}^n \left| \sum_{t=1}^T e_i^t - v_i \right| + \sum_{i=1}^n \left| \sum_{t=1}^T \xi_i^t \right| \right) \; . \tag{35}$$

Accordingly, if $\sum_i |\sum_t \xi_i^t| \le \frac{Tn}{72d^2}$ and $|\sum_t e_i^t - v_i| \le \frac{T}{72d^2}$ for every $1 \le i \le n$, then $f(x)\hat{N}^T(x) > 0$. We now show that each of those two events fails to occur with only exponentially small probability.

First, recall that we have almost surely $|e_i^t| \le 2d$. By Hoeffding's inequality,

$$\Pr\left[\sum_{t=1}^T |e_i^t - v_i| \ge \frac{T}{72d^2} \right] \le 2\exp\left(-\frac{T}{2^9 \cdot 3^4 \cdot d^6} \right) \le \exp(-d)/n \; ,$$

as soon as $T \ge \Omega(d^6(d + \log n)) = \mathrm{poly}(d)$. By union bound, $|\sum_t e_i^t - v_i| \le T/72d^2$ holds for every $1 \le i \le n$, except with probability $\exp(-d)$.

As for the additional Gaussian noise, observe that for $\tau = O(\sqrt{T}/d^2)$ we have

$$\mathbb{E}\left[\sum_{i=1}^n \left| \sum_{t=1}^T \xi_i^t \right| \right] = n\tau\sqrt{T}\sqrt{2/\pi} \le \frac{Tn}{144d^2} \; . \tag{36}$$

Similarly as in the GD case, $\left| \sum_t \frac{\xi_i^t}{\sqrt{T}\tau} \right|$ is a sub-Gaussian random variable. Therefore, we have concentration

$$\Pr\left[\sum_{i=1}^n \left| \sum_{t=1}^T \xi_i^t \right| \ge \frac{Tn}{72d^2} \right] \le \Pr\left[\sum_{i=1}^n \left| \sum_{t=1}^T \xi_i^t \right| - n\tau\sqrt{T}\sqrt{2/\pi} \ge \frac{Tn}{144d^2} \right] \tag{37}$$

$$\le \exp\left(-\Omega\left(\frac{Tn}{\tau^2 d^4} \right) \right) \tag{38}$$

$$\le \exp(-d) \; , \tag{39}$$

since $\tau = O(\sqrt{T}/d^2)$ and $n = \Omega(d^4)$.

A.2 PROOF OF THEOREM 6

In this proof we will apply the following result about hinge loss SGD:

Lemma 2 (Lemma 4 in Nachum & Yehudayoff (2020)). *Let $f : \mathcal{X} \to \{-1, 1\}$ be a function from some finite domain $\mathcal{X} \subseteq \mathbb{R}^d$ such that $\|x\| \le R$ for every $x \in \mathcal{X}$ and some $R \ge 1$. Consider a one layer* ReLU *neural network at initialization. For $x \in \mathcal{X}$, let $z_x \in \mathbb{R}^n$ be the embedding vector $z_{x,i} = \mathrm{ReLU}(w_i \cdot x + b_i)$ and assume that $\|z_x\| \le R_z$ for every $x \in \mathcal{X}$.*

Furthermore, assume that there exists $c > 0$ *and a choice of output layer weights* $v^* \in \mathbb{R}^n$ *with* $\|v^*\| = 1$ *such that* $f(x) \sum_{i=1}^n v_i^* \operatorname{ReLU}(w_i \cdot x + b_i) \geq c$ *for every* $x \in \mathcal{X}$*.*

Then, using learning rate $0 < \gamma \leq \frac{1}{500R} \cdot \frac{c^2}{R_z^2}$ *and* $0 \leq \beta \leq 4R_z^2\gamma$*, the batch size one SGD algorithm using hinge loss* $L(x, y) = \max(0, \beta - N(x)y)$ *run on any sequence of samples from* $\mathcal{X}$ *will perform at most* $20R_z^2/c^2$ *nonzero updates.*

Let $\mathcal{X} := \{\pm 1\}^d$, for all $x \in \mathcal{X}$ we have $\|x\| = \sqrt{d}$. First, let us consider the case of non-perturbed Rademacher initialization.

For $x \in \mathcal{X}$, let $z_x \in \mathbb{R}^n$ be its embedding vector i.e., $z_{x,i} = \operatorname{ReLU}(w_i \cdot x + b_i)$, we have $\|z_x\| \leq (d+1)\sqrt{n} \leq 2d\sqrt{n}$. By Theorem 4 (see equation 4), except with probability $\exp(-d)$ over the choice of w, there exists $v^* \in \mathbb{R}^n$ with $|v_i^*| = 1/\sqrt{n}$ such that, for all $x \in \mathcal{X}$ we have

$$f(x) \sum_{i=1}^n v_i^* \operatorname{ReLU}(w_i \cdot x + b_i) \geq \frac{\sqrt{n}}{18\sqrt{d}} \,. \tag{40}$$

Now consider the perturbed initialization $\operatorname{ReLU}((w_i + g_i) \cdot x + b_i)$, where $g \sim \mathcal{N}\left(0, \frac{C^2}{d^2} \cdot \mathbb{I}\right)$ for some $C \leq \frac{1}{72}$. Let $\mathcal{E}_1$ be the event that there exists $1 \leq i \leq n$ such that $\|g_i\| \geq \sqrt{d}$ and $\mathcal{E}_2$ that there exists x such that $\sum_{i=1}^n |g_i \cdot x| \geq \frac{n}{36\sqrt{d}}$. First, let us establish that each of these events occurs with probability at most $\exp(-d)$.

Let us start with $\mathcal{E}_1$. Since $\mathbb{E}\|g_i\|^2 = \frac{C^2}{d}$, by subgaussian concentration we have

$$\Pr\left[\|g_i\|^2 \geq \frac{C^2}{d} + t\right] \leq \exp\left(-\Omega\left(\frac{d^2 t^2}{C^4}\right)\right) \,. \tag{41}$$

Substituting $t = d/2$, we have in particular $\Pr[\|g_i\|^2 \geq d] \leq \exp(-\Omega(d^4))$. Taking union bound over $n = \operatorname{poly}(d)$, we have $\Pr[\exists i : \|g_i\|^2 \geq d] \leq \exp(-d)$.

As for $\mathcal{E}_2$, note that $\mathbb{E}|g_i \cdot x| = \frac{\sqrt{2}C}{\sqrt{\pi d}} \leq \frac{1}{72\sqrt{d}}$. Therefore, again by subgaussian concentration, for any fixed x,

$$\Pr\left[\sum_{i=1}^n |g_i \cdot x| \geq \frac{n}{36\sqrt{d}}\right] \leq \Pr\left[\sum_{i=1}^n |g_i \cdot x| \geq \mathbb{E}\left[\sum_{i=1}^n |g_i \cdot x|\right] + \frac{n}{72\sqrt{d}}\right] \leq \exp(-\Omega(n)) \,, \tag{42}$$

which is smaller than $\exp(-d)$ for $n \geq \Omega(d^4)$.

If neither $\mathcal{E}_1$ or $\mathcal{E}_2$ happens, then for every x we have

$$f(x) \sum_{i=1}^n v_i^* \operatorname{ReLU}((w_i + g_i) \cdot x + b_i) \geq f(x) \sum_{i=1}^n v_i^* \operatorname{ReLU}(w_i \cdot x + b_i) - \frac{1}{\sqrt{n}} \sum_{i=1}^n |g_i \cdot x| \tag{43}$$

$$\geq \frac{\sqrt{n}}{36\sqrt{d}} \,. \tag{44}$$

Furthermore, for every x and i it holds $|(w_i + g_i) \cdot x + b_i| \leq (\|w_i\| + \|g_i\|)\sqrt{d} + 1 \leq 3d$ and consequently $\|z_x\| \leq 3d\sqrt{n}$.

Therefore by applying Lemma 2 with $R := \sqrt{d}, R_z := 3d\sqrt{n}$ and $c := \frac{\sqrt{n}}{36\sqrt{d}}$, we conclude that using learning rate $0 < \gamma \leq \frac{1}{500\sqrt{d}} \cdot \frac{1}{9 \cdot 36^2 d^3} = O\left(\frac{1}{d^{3.5}}\right)$, the SGD algorithm using the hinge loss $L(y, \hat{y}) = \max\{0, \beta - y\hat{y}\}$, with $0 \leq \beta \leq 36d^2 n\gamma$, will perform at most $O(d^3)$ nonzero updates after which all samples will be classified correctly. □

A.3 PROOF OF THEOREM 5

Proof of Theorem 5. First, note that we can choose values of $\gamma = 1/\operatorname{poly}(d)$ and $0 \leq \beta \leq O(d^2 n\gamma)$ such that Theorem 6 applies. In line with Theorem 6, fix an initialization such that the SGD algorithm running on i.i.d. samples from $\mathcal{D}$ performs at most $C_0 := Cd^3$ nonzero updates, where C is a universal constant.

Let us run the training until there are $K := \frac{1}{\epsilon}(\ln 1/\delta + \ln C_0)$ zero updates in a row. As the number of nonzero updates is at most C_0, the algorithm runs for at most $C_0(1+K) = \mathrm{poly}(d)\frac{1}{\epsilon}\ln\frac{1}{\delta}$ steps.

Finally, let us argue that that the classification error does not exceed ϵ except with probability δ. To that end, define a "bad event" $\mathcal{E}$ as follows: There exists t such that:

1. A nonzero update occurs at time t.
2. There are K zero updates in a row immediately following t.
3. $\Pr_{x\sim\mathcal{D}}[\mathrm{sign}(N^{t+1}(x)) \neq f(x)] > \epsilon$.

It should be clear that if $\mathcal{E}$ does not occur, then at the final time T it holds $\Pr_{x\sim\mathcal{D}}[\mathrm{sign}(N^T(x)) \neq f(x)] \leq \epsilon$.

Fix some time t such that the first and third condition above are satisfied. Clearly, if the error probability exceeds ϵ, then so does the probability of a nonzero update. By independence (and the fact that only a nonzero update can change the network), the probability that there will be K zero updates in a row is at most $(1-\epsilon)^K$. By union bound over at most C_0 nonzero updates,

$$\Pr[\mathcal{E}] \leq C_0(1-\epsilon)^K \leq \delta\ . \qquad \square$$

B PROOFS FOR SECTION 5.1

B.1 PROOF OF THEOREM 7

For brevity, we denote the population gradient at θ for a target function f by

$$\Gamma_f(\theta) := \mathbb{E}_x\left[\nabla_\theta L(f(x), \theta, x)\right]. \tag{45}$$

To prove our results we couple the dynamics of the network's weights θ^t with the dynamics of the 'Junk-Flow'. The junk-flow is the dynamics of the parameters of a network trained on random labels. For that purpose let

$$\Gamma_r(\theta) := \mathbb{E}_x\left[\frac{1}{2}\left(\nabla_\theta L(1,\theta,x) + \nabla_\theta L(-1,\theta,x)\right)\right]\ . \tag{46}$$

In other words, $\Gamma_r(\theta)$ is the expected population gradient of random classification problem where $r(x) \sim \mathrm{Rad}(1/2)$ independently for every input x.

Definition 6 (Junk-Flow). *Let us define the junk-flow as the sequence $\psi^t \in \mathbb{R}^P$ that satisfies the following iterations:*

$$\psi^0 = \theta^0, \tag{47}$$

$$\psi^{t+1} = \psi^t - \gamma\left(\Gamma_r(\psi^t) + \xi^t\right), \tag{48}$$

where $\xi^t \overset{iid}{\sim} \mathcal{N}(0, \mathbb{I}\tau^2)$ We call γ the learning rate and τ the noise-level of the noisy-GD algorithm used to train the network $\mathrm{NN}(x;\theta)$.

We show that θ^T and ψ^T are close in terms of the total variation distance. Let us look at the total variation distance between the law of θ^T and ψ^T, which, by abuse of notation, we denote by $\mathrm{TV}(\theta^T;\psi^T)$.

Lemma 3. *Let $\mathrm{TV}(\theta^T;\psi^T)$ be the total variation distance between the law of θ^T and ψ^T. Then,*

$$\mathrm{TV}(\theta^T;\psi^T) \leq \frac{1}{2\tau}\sum_{t=0}^{T-1}\sqrt{\mathrm{GAL}_f(\psi^t)}. \tag{49}$$

The proof of Lemma 3 can be found in Section B.2.

Recalling that $f : \mathbb{R}^P \to \{\pm 1\}$, we have

$$\begin{aligned}\mathbb{P}\Big[\operatorname{sign}(\mathrm{NN}(x;\theta^T)) = f(x)\Big] &\le \mathbb{P}\Big[\operatorname{sign}(\mathrm{NN}(x;\psi^T)) = f(x))\Big] + \mathrm{TV}(\theta^T;\psi^T) && (50)\\ &\le \frac{1}{2} + \mathrm{TV}(\theta^T;\psi^T), && (51)\\ &\le \frac{1}{2} + \frac{1}{2\tau}\sum_{t=0}^{T-1}\sqrt{\mathrm{GAL}_f(\psi^t)}\,. && (52)\end{aligned}$$

In equation 51 we used the fact that the initialization is symmetric around 0. Since for the correlation loss $\Gamma_r(\theta) = 0$, the junk flow just adds independent Gaussian noise and the distribution of the output layer weights ψ^T is also symmetric around 0 (and independent of other weights). Therefore, the distribution of $\operatorname{sign}(\mathrm{NN}(x;\psi^T))$ is also symmetric around 0 for every fixed x. Finally, in equation 52 we used Lemma 3.

We are now left with showing that the right-hand-side of equation 49 is small, i.e. that the junk-flow dynamics does not pick correlation with f along its trajectory. Again, for the correlation loss, $\Gamma_r(\psi^t) = 0$ for all t, thus for all t, $\psi^t = A + H_\sigma + \sqrt{t}\gamma\tau H$, where $H \sim \mathcal{N}(0, \mathbb{I}_P)$. Thus, the result follows by the assumption in equation 5.

B.2 PROOF OF LEMMA 3

This proof follows a similar argument that is used in (Abbe & Sandon (2020); Abbe & Boix-Adsera (2022)). In the following let us write $\theta := \theta^{T-1}$ and $\psi := \psi^{T-1}$ for readability. The total variation distance $\mathrm{TV}(\theta^T;\psi^T)$ can be bounded in terms of θ and ψ as follows:

$$\begin{aligned}\mathrm{TV}(\theta^T;\psi^T) &= \mathrm{TV}\left(\theta - \gamma(\Gamma_f(\theta) + Z^t); \psi - \gamma(\Gamma_r(\psi) + \xi^t)\right) && (53)\\ &\overset{a)}{\le} \mathrm{TV}\left(\theta - \gamma(\Gamma_f(\theta) + Z^t); \psi - \gamma(\Gamma_f(\psi) + Z^t)\right) && (54)\\ &\quad + \mathrm{TV}\left(\psi - \gamma(\Gamma_f(\psi) + Z^t); \psi - \gamma(\Gamma_r(\psi) + \xi^t)\right) && (55)\\ &\overset{b)}{\le} \mathrm{TV}\left(\theta;\psi\right) && (56)\\ &\quad + \mathbb{E}_\psi\, \mathrm{TV}\left(\psi - \gamma(\Gamma_f(\psi) + Z^t); \psi - \gamma(\Gamma_r(\psi) + \xi^t) \mid \psi\right) && (57)\\ &\overset{c)}{\le} \mathrm{TV}(\theta;\psi) && (58)\\ &\quad + \mathbb{E}_\psi\sqrt{\frac{1}{2}D_{\mathrm{KL}}\left(\psi - \gamma(\Gamma_f(\psi) + Z^t) || \psi - \gamma(\Gamma_r(\psi) + \xi^t) \mid \psi\right)} && (59)\\ &\overset{d)}{\le} \mathrm{TV}\left(\theta^{T-1};\psi^{T-1}\right) + \frac{1}{2\tau\gamma}\mathbb{E}_\psi\|\gamma\Gamma_f(\psi) - \gamma\Gamma_r(\psi)\|_2 && (60)\\ &= \mathrm{TV}\left(\theta^{T-1};\psi^{T-1}\right) + \frac{1}{2\tau}\mathbb{E}_\psi\|\Gamma_f(\psi) - \Gamma_r(\psi)\|_2 && (61)\end{aligned}$$

where in $a)$ we used the triangle inequality, in $b)$ the data processing inequality (DPI) and triangle inequality again, in $c)$ Pinsker's inequality. Finally, $d)$ follows since, conditional on ψ, both distributions in the KL divergence are Gaussian, and due to the known formula $D_{\mathrm{KL}}(\mathcal{N}(\mu, \sigma\,\mathrm{Id}), \mathcal{N}(\mu', \sigma\,\mathrm{Id})) = \frac{\|\mu-\mu'\|^2}{2\sigma^2}$. Thus,

$$\begin{aligned}\mathrm{TV}(\theta^T;\psi^T) &\le \frac{1}{2\tau}\sum_{t=0}^{T-1}\mathbb{E}_{\psi^t}\|\Gamma_f(\psi^t) - \Gamma_r(\psi^t)\|_2 && (62)\\ &\overset{(a)}{\le} \frac{1}{2\tau}\sum_{t=0}^{T-1}\sqrt{\mathrm{GAL}_f(\psi^t)}, && (63)\end{aligned}$$

where in (a) we used Cauchy-Schwartz.

B.3 PROOF OF COROLLARY 1

Let us state a claim about Gaussians with slightly different variances:

Claim 1. *Let $F : \mathbb{R}^P \to \mathbb{R}$ be a function such that $0 \le F(x) \le R$ for all $x \in \mathbb{R}^P$. Let $\theta \sim \mathcal{N}(\mu, D)$, for some $\mu \in \mathbb{R}^P$ and D a diagonal matrix with diagonal entries $(\sigma_1^2, \ldots, \sigma_P^2)$, and let $\psi \sim \mathcal{N}(\mu, D')$ for some other diagonal D' with entries $((\sigma_1')^2, \ldots, (\sigma_P')^2)$ such that $(\sigma_i')^2 \le \sigma_i^2(1 + 1/P)$ for every $1 \le i \le P$.*

If $\mathbb{E}F(\theta) \le \epsilon$, for some ϵ, then $\mathbb{E}F(\psi) \le (4R+1)\epsilon^{1/9}$.

Proof. Let $M > 0$ and define the event $\mathcal{E}_M$ as $\sqrt{\sum_{i=1}^P \left(\frac{\psi_i - \mu_i}{\sigma_i}\right)^2} > M$. By Gaussian concentration (formula (3.5) in Ledoux & Talagrand (2013), see also MO2 (2020)):

$$\Pr\left[\mathcal{E}_M\right] \le 4\exp\left(-\frac{M^2}{8\mathbb{E}\sum_{i=1}^P \left(\frac{\psi_i - \mu_i}{\sigma_i}\right)^2}\right) \le 4\exp\left(-\frac{M^2}{16P}\right) \ . \tag{64}$$

At the same time, if $\sum_{i=1}^P \left(\frac{x_i - \mu_i}{\sigma_i}\right)^2 \le M^2$, then the density functions φ_θ and φ_ψ satisfy

$$\begin{aligned}
\varphi_\psi(x) &= \prod_{i=1}^P \frac{1}{\sqrt{2\pi}\sigma_i'} \exp\left(-\frac{(x_i - \mu_i)^2}{2(\sigma_i')^2}\right) && (65)\\
&\le \exp\left(\sum_{i=1}^P \frac{(x_i - \mu_i)^2}{2\sigma_i^2} \cdot \frac{(\sigma_i')^2 - \sigma_i^2}{(\sigma_i')^2}\right) \prod_{i=1}^P \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right) && (66)\\
&\le \exp\left(\frac{M^2}{2P}\right) \varphi_\theta(x) \ . && (67)
\end{aligned}$$

So,

$$\begin{aligned}
\mathbb{E}F(\psi) &= \int_{x \in \mathcal{E}_M} F(x)\varphi_\psi(x) + \int_{x \notin \mathcal{E}_M} F(x)\varphi_\psi(x) && (68)\\
&\le \exp\left(\frac{M^2}{2P}\right)\epsilon + 4R\exp\left(-\frac{M^2}{16P}\right) \ . && (69)
\end{aligned}$$

Substituting $M := \sqrt{\frac{16P}{9} \ln 1/\epsilon}$, we get the bound. ☐

Let $F(\theta) := \|\Gamma_f(\theta) - \Gamma_r(\theta)\|_2^2$. Conditional on the value of A, the distribution of θ^0 is Gaussian $\theta^0 \sim \mathcal{N}(A, \sigma^2 D_A)$ where D_A is diagonal with entries $(D_A)_{pp} = \mathrm{Var}A_p$. Let $0 \le \lambda \le \gamma^2\tau^2 T$. Then, the distribution of $\theta^0 + \lambda H$ for H standard gaussian is $\theta^0 + \lambda H \sim \mathcal{N}(A, \sigma^2 D_A + \lambda^2 \mathbb{I}_P)$. Therefore, by assumption for every $1 \le p \le P$ it holds

$$\sigma^2 \mathrm{Var}A_p + \lambda^2 \le \sigma^2 \mathrm{Var}A_p + \gamma^2\tau^2 T \le \sigma^2 \mathrm{Var}A_p \left(1 + \frac{1}{P}\right) \ . \tag{70}$$

By Claim 1 (and averaging over A), it follows

$$\mathrm{GAL}_f(\theta^0 + \lambda H) = \mathbb{E}F(\sigma^0 + \lambda H) \le (4R+1)\mathbb{E}F(\theta^0)^{1/9} = (4R+1)\,\mathrm{GAL}_f(\theta^0)^{1/9} \ . \tag{71}$$

Equation 6 now follows directly by applying Theorem 7.

B.4 PROOF OF COROLLARY 2

For Corollary 2 we focus on fully-connected networks of bounded depth. For simplicity, we consider fully connected networks with one bias vector in the first layer, but we believe that, with a more involved argument, one could extend the proof and include bias vectors in all layers. In particular, we use the following notation:

$$\begin{aligned}
x^{(1)}(\theta) &= W^{(1)}x + b^{(1)} && (72)\\
x^{(l)}(\theta) &= W^{(l)}\sigma(x^{(l-1)}(\theta)), \qquad l = 2, ..., L, && (73)
\end{aligned}$$

and we denote the network function as $\mathrm{NN}(x;\theta) = x^{(L)}(\theta)$. We assume that the activation σ satisfies the H-weak homogeneity assumption of Def. 5. We assume that each parameter of the network is independently initialized as $\theta_p^0 \sim \mathcal{N}(0, v_{l_p}^2)$, where l_p denotes the layer of parameter θ_p, for $p \in [P]$.

Corollary 2 follows from the following Proposition.

Proposition 2. *Let* $\mathrm{NN}(x;\theta)$ *be a network that satisfies the assumptions of Corollary 2. Then, if* $\mathrm{GAL}_f(\theta^0) < \epsilon$,

$$\mathrm{GAL}_f(\theta^0 + \gamma\lambda H) \leq \prod_{l=1}^{L} \left(1 + \frac{\gamma^2\lambda^2}{v_l^2}\right)^H \epsilon, \tag{74}$$

where $H \sim \mathcal{N}(0, \mathbb{I}_P)$.

B.5 PROOF OF PROPOSITION 2

Recall, that $\theta^0 \sim \mathcal{N}(0, V)$, where V is a $P \times P$ diagonal matrix such that $V_{pp} = v_{l_p}^2$, where l_p is the layer of parameter p, and $\psi_p^t \sim \mathcal{N}(0, U)$, where U is a $P \times P$ diagonal matrix such that $U_{pp} = v_{l_p}^2 + t\gamma^2\tau^2$. Thus, $U = \overline{C}V\overline{C}^T$, where $\overline{C}$ is a $P \times P$ diagonal matrix such that

$$\overline{C}_{pp} = \sqrt{1 + \frac{t\gamma^2\tau^2}{v_{l_p}^2}}. \tag{75}$$

Definition 7 ($\overline{C}$-Rescaling)**.** *Let* $\mathrm{NN}(x;\theta)$ *be an L-layers network, with parameters* $\theta \in \mathbb{R}^P$. *Let* $C^{(1)}, ..., C^{(L)}$ *be L positive constants, and let* $\overline{C}$ *be a* $P \times P$ *diagonal matrix such that* $\overline{C}_{pp} = C^{(l_p)}$ *where* l_p *is the layer of parameter* θ_p. *We say that the vector* $\overline{C} \cdot \theta$ *is a* $\overline{C}$*-rescaling of* θ.

Definition 8 (Weak Positive Homogeneity (SPH))**.** *We say that an architecture is* H-weakly homogeneous *(H-SPH) if for all* $\overline{C}$*-rescaling such that* $\min_{p\in[P]} \overline{C}_{pp} > 1$*, it holds:*

$$\mathrm{NN}(x; \overline{C} \cdot \theta) = \prod_{l=1}^{L} (C^{(l)})^H \cdot \mathrm{NN}(x;\theta), \tag{76}$$

$$\partial_{(\overline{C}\theta)_p} \mathrm{NN}(x; \overline{C} \cdot \theta) = D_{p,H} \cdot \partial_{\theta_p} \mathrm{NN}(x;\theta), \tag{77}$$

where $D_{p,H}$ *is such that* $D_{p,H} \leq \prod_{l=1}^{l_p} \left(C^{(l)}\right)^H$.

Lemma 4. *Let* $\mathrm{NN}(x;\theta)$ *be a fully connected network as in equation 72-equation 73. Assume that the activation* σ *is* H*-weakly homogeneous (as defined in Def. 5), with* $H \geq 1$. *Then,* $\mathrm{NN}(x;\theta)$ *is* H*-SPH.*

The proof of Lemma 4 is in Appendix B.6.

If we optimize over the Correlation Loss, i.e. $L_{\mathrm{corr}}(y, \hat{y}) := -y\hat{y}$, then the gradients of interest are given by:

$$\Gamma_f(\theta) = -\mathbb{E}_x \left[f(x) \cdot \nabla_\theta \mathrm{NN}(x;\theta)\right]; \tag{78}$$

$$\Gamma_r(\theta) = 0. \tag{79}$$

Thus,

$$\mathbb{E}_{\psi^t} \|\Gamma_f(\psi^t) - \Gamma_r(\psi^t)\|_2^2 = \sum_{p=1}^{P} \mathbb{E}_{\psi^t}\mathbb{E}_x \left[\partial_{\psi_p^t} \mathrm{NN}(x;\psi^t) \cdot f(x)\right]^2$$

Let $\overline{C}$ be a $P \times P$ matrix such that $\overline{C}_{pp} = \sqrt{1 + \frac{t\gamma^2\tau^2}{v_{l_p}^2}}$, where l_p is the layer of θ_p^0. One can verify that the $\overline{C}$-rescaling of θ^0 has the same distribution as ψ^t. We can thus rewrite each term in the sum above as:

$$\begin{aligned}\mathbb{E}_{\psi^t}\mathbb{E}_x\left[\partial_{\psi_p^t}\operatorname{NN}(x;\psi^t)\cdot f(x)\right]^2 &= \mathbb{E}_{\overline{C}\theta^0}\mathbb{E}_x\left[\partial_{(\overline{C}\theta^0)_p}\operatorname{NN}(x;\overline{C}\theta^0)\cdot f(x)\right]^2\\ &\overset{(a)}{=} D_{p,H}^2\cdot\mathbb{E}_{\theta^0}\mathbb{E}_x\left[\partial_{\theta_p^0}\operatorname{NN}(x;\theta^0)\cdot f(x)\right]^2\end{aligned}$$

where in (a) we used Lemma 4. Thus,

$$\begin{aligned}\mathbb{E}_{\psi^t}\|\Gamma_f(\psi^t)-\Gamma_r(\psi^t)\|_2^2 &= \mathbb{E}_{\theta^0}\sum_{p=1}^{P}D_{p,H}^2\mathbb{E}_x\left[\partial_{\theta_p^0}\operatorname{NN}(x;\theta^0)\cdot f(x)\right]^2\\ &\overset{(a)}{\leq} K\cdot\mathbb{E}_{\theta^0}\|G_f(\theta^0)\|_2^2,\end{aligned}$$

where $K=\prod_{l=1}^{L}\left(1+\frac{t\gamma^2\tau^2}{v_l^2}\right)^H$, and where in (a) we used that $|D_{p,H}|\leq C_{p,H}$.

B.6 PROOF OF LEMMA 4

We proceed by induction on the network depth. As a base case, we consider a 2-layer network. Let us write explicitly the gradients of the network.

$$\nabla_{W_i^{(2)}}\operatorname{NN}(x;\theta)=\sigma(x_i^{(1)}(\theta)), \tag{80}$$

$$\nabla_{W_{ij}^{(1)}}\operatorname{NN}(x;\theta)=W_i^{(2)}\sigma'(x_i^{(1)}(\theta))x_j, \tag{81}$$

$$\nabla_{b_i^{(1)}}\operatorname{NN}(x;\theta)=W_i^{(2)}\sigma'(x_i^{(1)}(\theta)). \tag{82}$$

Notice that the weak homogeneity assumption on the activation σ (Def. 5), we have for $l\in\{1,2\}$:

$$x_i^{(l)}(\overline{C}\cdot\theta)=\prod_{h=1}^{l}(C^{(h)})^H\cdot x_i^{(l)}(\theta), \tag{83}$$

thus equation 76 holds. Moreover,

$$\partial_{W_i^{(2)}}\operatorname{NN}(x;\overline{C}\cdot\theta)=(C^{(1)})^H\partial_{W_i^{(2)}}\operatorname{NN}(x;\theta), \tag{84}$$

$$\partial_{W_{ij}^{(1)}}\operatorname{NN}(x;\overline{C}\cdot\theta)=(C^{(2)})^H\partial_{W_{ij}^{(1)}}\operatorname{NN}(x;\theta), \tag{85}$$

$$\partial_{b_i^{(1)}}\operatorname{NN}(x;\overline{C}\cdot\theta)=(C^{(2)})^H\partial_{b_i^{(1)}}\operatorname{NN}(x;\theta). \tag{86}$$

Therefore, for any parameter θ_p, $p\in[P]$,

$$\partial_{\theta_p}\operatorname{NN}(x;\overline{C}\cdot\theta)=D_{p,H}\partial_{\theta_p}\operatorname{NN}(x;\theta), \tag{87}$$

with $1<D_{p,H}\leq\max\{(C^{(1)})^H,(C^{(2)})^H\}\leq\prod_{l=1}^{2}(C^{(l)})^H$.

For the induction step, assume that for a network of depth $L-1$, for all parameters θ_p,

$$\partial_{\theta_p}\operatorname{NN}(x;\overline{C}(\theta))=D_{p,H}\cdot\partial_{\theta_p}\operatorname{NN}(x;\theta), \tag{88}$$

with $1<D_{p,H}\leq\prod_{l=1}^{L-1}(C^{(l)})^H$. Let us consider a neural network of depth L, and let us write the gradients,

$$\partial_{W_i^{(L)}}\operatorname{NN}(x;\theta)=\sigma(x_i^{(L-1)}(\theta)), \tag{89}$$

$$\partial_{W_{ij}^{(l)}}\operatorname{NN}(x;\theta)=\sum_{k=1}^{N_{L-1}}W_k^{(L)}\sigma'(x_k^{(L-1)}(\theta))\cdot\partial_{W_{ij}^{(l)}}x_k^{(L-1)}(\theta),\qquad l=1,...,L-1, \tag{90}$$

$$\partial_{b_i^{(1)}}\operatorname{NN}(x;\theta)=\sum_{k=1}^{N_{L-1}}W_k^{(L)}\sigma'(x_k^{(L-1)}(\theta))\cdot\partial_{b_i^{(1)}}x_k^{(L-1)}(\theta), \tag{91}$$

where N_{L-1} denotes the width of the $(L-1)$-th hidden layer. One can observe that $x_k^{(L-1)}(\theta)$ corresponds to the output of a fully connected network of depth $L-1$, and thus we can use the induction hypothesis for bounding $\partial_{\theta_p} x_k^{(L-1)}(\theta)$, for all parameters θ_p in the first $L-1$ layers. Thus,

$$\partial_{W_i^{(L)}} \mathrm{NN}(x;\overline{C}(\theta)) = (C^{(L-1)})^H \cdot D_{W_i^{(L)},H} \cdot \partial_{W_i^{(L)}} \mathrm{NN}(x;\theta), \tag{92}$$

$$\partial_{W_{ij}^{(l)}} \mathrm{NN}(x;\overline{C}(\theta)) = \sum_{k=1}^{N_{L-1}} C^{(L)} W_k^{(L)} \sigma'(x_k^{(L-1)}(\theta)) \cdot D_{W_{ij}^{(l)},H} \partial_{W_{ij}^{(l)}} x_k^{(L-1)}(\theta), \qquad l = 1, ..., L-1, \tag{93}$$

$$\partial_{b_i^{(1)}} \mathrm{NN}(x;\overline{C}(\theta)) = \sum_{k=1}^{N_{L-1}} C^{(L)} W_k^{(L)} \sigma'(x_k^{(L-1)}(\theta)) \cdot D_{b_i^{(1)},H} \partial_{b_i^{(l)}} x_k^{(L-1)}(\theta). \tag{94}$$

Thus, the result follows.

C SMALL ALIGNMENT FOR GAUSSIAN INITIALIZATION: PROOF OF PROPOSITION 1

Proposition 3 (Restatement of Proposition 1)**.** *Let a neural network be as in Theorem 8. Then, for every $\sigma_0^2 > 0$, there exists $C, C' > 0$ such that for any network with $\sigma^2 \leq \sigma_0^2$ we have a gradient alignment bound*

$$\mathrm{GAL}_{f_a}(\theta) \leq PC' \exp(-Cd) \ , \tag{95}$$

where $P := nd + 2n$ is the total number of parameters.

In order to establish Proposition 1, we will need two calculations arising from the gradient formulas.

Definition 9. *Let $d \in \mathbb{N}$ and $\alpha \geq 0$ and β be such that $\alpha + |\beta| \leq 1$. We say that random variables (k, G_1, G_2) are (d, α, β)-*alternating Gaussians *if:*

- $k \sim \mathrm{Bin}(d, 1/2)$.
- *Conditioned on k, the pair (G_1, G_2) are joint centered unit variance Gaussians with covariance $(1 - 2k/d)\alpha + \beta$.*

Lemma 5. *For each $\alpha_0 > 0$ there exist $C', C > 0$ such that if (k, G_1, G_2) are (d, α, β)-alternating Gaussians for $\alpha \geq \alpha_0$, then*

$$\mathbb{E}\Big[(-1)^k \mathbb{1}(G_1 \geq 0)\mathbb{1}(G_2 \geq 0)\Big] \leq C' \exp(-Cd) \ . \tag{96}$$

Lemma 6. *For each $\alpha_0 > 0$ there exist $C', C > 0$ such that if (k, G_1, G_2) are (d, α, β)-alternating Gaussians for $\alpha \geq \alpha_0$, then*

$$\mathbb{E}\Big[(-1)^k \mathrm{ReLU}(G_1) \, \mathrm{ReLU}(G_2)\Big] \leq C' \exp(-Cd) \ . \tag{97}$$

A crucial element of both calculations is the following claim:

Claim 2. *Let $d \in \mathbb{N}$. For all $n < d$, for any polynomial P of degree n,*

$$\sum_{k=0}^{d} (-1)^k \binom{d}{k} P(k) = 0. \tag{98}$$

Proof. We prove the statement by induction on n. If $n = 0$, then

$$\sum_{k=0}^{d} (-1)^k \binom{d}{k} = (1-1)^d = 0 \ , \tag{99}$$

and therefore the sum equation 98 indeed vanishes for every constant polynomial. Assume that the claim holds for some $n \geq 0$. By linearity, it is enough that we only prove

$$\sum_{k=0}^{d} (-1)^k \binom{d}{k} k^{n+1} = 0 \ . \tag{100}$$

To that end, calculate

$$\begin{aligned} \sum_{k=0}^{d} (-1)^k \binom{d}{k} k^{n+1} &= \sum_{k=1}^{d} (-1)^k \binom{d}{k} k \cdot k^n \qquad (101) \\ &\overset{(a)}{=} d \sum_{k=1}^{d} (-1)^k \binom{d-1}{k-1} k^n \qquad (102) \\ &\overset{(b)}{=} -d \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} (k+1)^n = 0, \qquad (103) \end{aligned}$$

where (a) applied $\binom{d}{k} \cdot k = \binom{d-1}{k-1} \cdot d$ and (b) is a change of variables and applying the induction. ☐

C.1 PROPOSITION 1 IMPLIES THEOREM 8

Let $0 \leq \lambda^2 \leq \gamma^2 \tau^2 T$. In order to apply Theorem 7 for $A = 0$, we need to check the gradient alignment for initializations $\theta + \lambda H$, where $H \sim \mathcal{N}(0, \mathrm{Id}_P)$. More precisely, that means we have initialization with independent coordinates where

$$w_{ij} \sim \mathcal{N}\left(0, \frac{1}{d} + \lambda^2\right), b_i \sim \mathcal{N}\left(0, \sigma^2 + \lambda^2\right), v_i \sim \mathcal{N}\left(0, \frac{1}{n} + \lambda^2\right) \ . \tag{104}$$

Let us normalize by dividing w and b by $\sqrt{1 + d\lambda^2}$ and v by $\sqrt{1 + n\lambda^2}$. That gives new initialization $\widetilde{\theta}_\lambda = (\widetilde{w}, \widetilde{b}_\lambda, \widetilde{v})$ such that

$$\widetilde{w}_{ij} \sim \mathcal{N}\left(0, \frac{1}{d}\right), \widetilde{b}_{\lambda,i} \sim \mathcal{N}\left(0, \frac{\sigma^2 + \lambda^2}{1 + d\lambda^2}\right), \widetilde{v}_i \sim \mathcal{N}\left(0, \frac{1}{n}\right) \ . \tag{105}$$

In particular, the variance of $\widetilde{b}_{\lambda,i}$ is $\frac{\sigma^2+\lambda^2}{1+d\lambda^2} \leq \sigma^2 + \frac{\lambda^2}{1+\lambda^2} \leq \sigma^2 + O(1)$. By Proposition 1, we have a uniform bound

$$\mathrm{GAL}_{f_a}(\widetilde{\theta}_\lambda) \leq 2C' nd \exp(-Cd) \ . \tag{106}$$

By homogenity $\mathrm{ReLU}(cx) = c\,\mathrm{ReLU}(x)$ for $c \geq 0$, it is easy to check that

$$\mathrm{GAL}_{f_a}(\theta + \lambda H) \leq (1 + d\lambda^2)(1 + n\lambda^2)\,\mathrm{GAL}_{f_a}(\widetilde{\theta}_\lambda) \leq \exp(-\Omega(d)) \ . \tag{107}$$

The result now follows directly from Theorem 7. ☐

C.2 LEMMA 5 AND LEMMA 6 IMPLY PROPOSITION 1

Recall that $\mathrm{GAL}_{f_a} = \mathbb{E}_\theta \big\| \left(\mathbb{E}_x f_a(x) \nabla_\theta \mathrm{NN}(x;\theta)\right)^2 \big\|^2$. We will estimate the expectation of each squared coordinate of this vector by $O(\exp(-Cd))$. Then, equation 95 follows by summing up. Let us first write the neural network gradients for all types of weights $\theta = (w, b, v)$:

$$\begin{aligned} \nabla_{w_{ij}} \mathrm{NN} &= v_i \mathbb{1}(w_i \cdot x + b_i \geq 0) x_j \ , \qquad (108) \\ \nabla_{b_i} \mathrm{NN} &= v_i \mathbb{1}(w_i \cdot x + b_i \geq 0) \ , \qquad (109) \\ \nabla_{v_i} \mathrm{NN} &= \mathrm{ReLU}(w_i \cdot x + b_i) \ . \qquad (110) \end{aligned}$$

The square of the expected gradient $(\mathbb{E}_x f_a(x) \nabla_{\theta_i} NN(x;\theta)^2$ can be also written as the expectation over two independent input samples x, x'. In particular, in the case of w_{ij} from equation 108, we have

$$\mathbb{E}_\theta \left(\mathbb{E}_x f_a(x) \nabla_{w_{ij}} \mathrm{NN}\right)^2 = \mathbb{E}_{x,x'} \left(\prod_{\ell=1}^{d-a} x_\ell x'_\ell\right) \left(\mathbb{E}_{v_i} v_i^2\right) \left(\mathbb{E}_{w_i,b_i} \mathbb{1}(w_i \cdot x + b_i \geq 0) \mathbb{1}(w_i \cdot x' + b_i \geq 0)\right) x_j x'_j \ . \tag{111}$$

Consider the set $S := \{1, \ldots, d-a\} \triangle \{j\}$, where $\triangle$ denotes the symmetric difference. Abusing notation, let us write $x = (y, z)$ and $x' = (y', z')$ where y, y' containt the coordinates in S and z, z' the coordinates from $[d] \setminus S$. Let k be the Hamming distance $k := d_H(y, y')$. Note that the distribution of k is binomial $k \sim \mathrm{Bin}(|S|, 1/2)$. Then, continuing from equation 111,

$$\mathbb{E}_\theta \left(\mathbb{E}_x f_a(x) \nabla_{w_{ij}} \mathrm{NN}\right)^2 = \frac{1}{n} \mathbb{E}_{z,z',k} \left[(-1)^k \mathbb{E}_{w_i} \left[\mathbb{1}(w_i \cdot x + b_i \geq 0) \mathbb{1}(w_i \cdot x' + b_i \geq 0)\right]\right] . \tag{112}$$

Fix some values of z, z' and k. Let $G_1 := w_i \cdot x + b_i$ and $G_2 := w_i \cdot x' + b_i$. Notice that, conditionally on k, z, z', random variables G_1 and G_2 are joint centered Gaussian with $\mathrm{Var} G_1 = \mathrm{Var} G_2 = 1 + \sigma^2$ and

$$\mathrm{Cov}[G_1, G_2] = \frac{1}{d} \left(d - 2k - 2d_H(z, z')\right) + \sigma^2 . \tag{113}$$

Let $\widetilde{G}_i := G_i / \sqrt{1 + \sigma^2}$ for $i = 1, 2$. Then, $\widetilde{G}_1$ and $\widetilde{G}_2$ are two joint centered unit variancce Gaussians with correlation

$$\begin{aligned} \mathrm{Cov}[\widetilde{G}_1, \widetilde{G}_2] &= \frac{1}{d(1+\sigma^2)} \left(d - 2k - 2d_H(z, z')\right) + \frac{\sigma^2}{1+\sigma^2} && (114) \\ &= \left(1 - \frac{2k}{|S|}\right) \frac{|S|}{d(1+\sigma^2)} + \frac{d - |S| - 2d_H(z, z') + d\sigma^2}{d(1+\sigma^2)} . && (115) \end{aligned}$$

Therefore, conditioned on z and z', random variables (k, G_1, G_2) are (d, α, β)-alternating Gaussians for $\alpha = \frac{|S|}{d(1+\sigma^2)} \geq \frac{1}{3(1+\sigma_0^2)} > 0$. It is also easy to check that $\alpha + |\beta| \leq \frac{|S| + d - |S| + d\sigma^2}{d(1+\sigma^2)} = 1$. By Lemma 5, for some uniform constant $C > 0$ it holds

$$\begin{aligned} \mathbb{E}_{k, G_1, G_2} \left[(-1)^k \mathbb{1}(G_1 \geq 0) \mathbb{1}(G_2 \geq 0)\right] &= \mathbb{E}_{k, \widetilde{G}_1, \widetilde{G}_2} \left[(-1)^k \mathbb{1}(\widetilde{G}_1 \geq 0) \mathbb{1}(\widetilde{G}_2 \geq 0)\right] && (116) \\ &\leq C' \exp(-Cd) . && (117) \end{aligned}$$

Plugging this into equation 111 and equation 112, we get the desired bound. The case of the hidden layer bias b_i proceeds by the same argument with $S := \{1, \ldots, d-a\}$.

Finally, in case of v_i we set $S := \{1, \ldots, d-a\}$ and proceed with a similar calculation

$$\begin{aligned} \mathbb{E}_\theta \left(\mathbb{E}_x f_a(x) \nabla_{v_i} \mathrm{NN}\right)^2 &= (1 + \sigma^2) \mathbb{E}_{z,z',k} \left[(-1)^k \mathbb{E}_{\widetilde{G}_1, \widetilde{G}_2} [\mathrm{ReLU}(\widetilde{G}_1) \mathrm{ReLU}(\widetilde{G}_2)]\right] && (118) \\ &\leq (1 + \sigma_0^2) C' \exp(-Cd) \leq C'' \exp(-Cd) , && (119) \end{aligned}$$

where in the last line we applied Lemma 6. □

C.3 PROOF OF LEMMA 5

It is well-known (see, e.g., Chapter 11 in O'Donnell (2014)), that for two ρ-correlated unit variance centered joint Gaussians it holds $\mathbb{E}[\mathbb{1}(G_1 \geq 0) \mathbb{1}(G_2 \geq 0)] = f(\rho)$ where $f(x) = \frac{1}{2} - \frac{1}{2\pi} \arccos(x)$. By definition of (k, G_1, G_2), conditioned on k, random variables G_1 and G_2 have correlation $\rho = \rho(k) = \left(1 - \frac{2k}{d}\right) \alpha + \beta$.

Hence,

$$\begin{aligned} \left|\mathbb{E}(-1)^k \mathbb{1}(G_1 \geq 0) \mathbb{1}(G_2 \geq 0)\right| &= \left|\mathbb{E}_k (-1)^k f(\rho(k))\right| && (120) \\ &\leq \mathbb{P}(|k - d/2| \geq d/4) \cdot \sup_{x \in [-1,1]} |f(x)| + \left|\frac{1}{2^d} \sum_{k=\lceil d/4 \rceil}^{\lfloor 3d/4 \rfloor} (-1)^k \binom{d}{k} f(\rho)\right| && (121) \\ &\overset{(a)}{\leq} 2 \exp(-d/10) + \left|\frac{1}{2^d} \sum_{k=\lceil d/4 \rceil}^{\lfloor 3d/4 \rfloor} (-1)^k \binom{d}{k} f(\rho)\right| , && (122) \end{aligned}$$

where (a) follows by Hoeffding's inequality.

It remains to bound the last term in equation 122. Consider the Taylor expansion of f:

$$f(x) = \frac{1}{2} - \frac{1}{2\pi}\left[\frac{\pi}{2} - \sum_{n=0}^{\infty} \frac{(2n)!}{4^n (n!)^2 (2n+1)} x^{2n+1}\right] \tag{123}$$

$$= \frac{1}{4} + \frac{1}{2\pi}\sum_{n=0}^{\infty} \frac{(2n)!}{4^n (n!)^2 (2n+1)} x^{2n+1} \tag{124}$$

$$= \frac{1}{4} + \frac{1}{2\pi}\sum_{n=0}^{\infty} \frac{\binom{2n}{n}}{4^n (2n+1)} x^{2n+1} \tag{125}$$

$$= \frac{1}{4} + \sum_{2n+1<d} a_n x^{2n+1} + \sum_{2n+1\geq d} a_n x^{2n+1}\ , \tag{126}$$

where $a_n := \frac{\binom{2n}{n}}{2\pi 4^n (2n+1)}$. For future reference let us note that $0 \leq a_n \leq 1$ for every n. So the second part of the RHS of equation 122 is upper bounded by:

$$\underbrace{\Big| \frac{1}{2^d} \sum_{k=\lceil d/4 \rceil}^{\lfloor 3d/4 \rfloor} (-1)^k \binom{d}{k} \left(\frac{1}{4} + \sum_{2n+1<d} a_n \rho^{2n+1}\right) \Big|}_{:=T_1} \tag{127}$$

$$+ \underbrace{\Big| \frac{1}{2^d} \sum_{k=\lceil d/4 \rceil}^{\lfloor 3d/4 \rfloor} (-1)^k \binom{d}{k} \sum_{2n+1\geq d} a_n \rho^{2n+1} \Big|}_{:=T_2} \tag{128}$$

We are going to show that $|T_1| \leq 2\exp(-d/10)$ and $|T_2| \leq \frac{2}{\alpha_0}(1-\alpha_0/2)^d$. These two bounds together with equation 122 imply the theorem statement.

Let us start with T_2. In the sum in equation 128 we have $d/4 \leq k \leq 3d/4$, and we can check that

$$|\rho| = \left|\left(1 - \frac{2k}{d}\right)\alpha + \beta\right| \leq \frac{1}{2}\alpha + |\beta| \leq 1 - \frac{\alpha_0}{2}. \tag{129}$$

Therefore,

$$|T_2| = \Big| \frac{1}{2^d} \sum_{k=\lceil d/4 \rceil}^{\lfloor 3d/4 \rfloor} (-1)^k \binom{d}{k} \sum_{2n+1\geq d} a_n \rho^{2n+1} \Big| \tag{130}$$

$$\leq \frac{1}{2^d} \cdot \sum_{k=\lceil d/4 \rceil}^{\lfloor 3d/4 \rfloor} \binom{d}{k} \sum_{2n+1\geq d} a_n \left(1 - \frac{\alpha_0}{2}\right)^{2n+1} \tag{131}$$

$$\leq \sum_{2n+1\geq d} \left(1 - \frac{\alpha_0}{2}\right)^{2n+1} \leq \frac{2}{\alpha_0}\left(1 - \frac{\alpha_0}{2}\right)^d\ . \tag{132}$$

For T_1, we follow two steps. First,

$$|T_1| \leq \sum_{2n+1<d} \Big| \frac{1}{2^d} \sum_{k=\lceil d/4 \rceil}^{\lfloor 3d/4 \rfloor} (-1)^k \binom{d}{k} \rho^{2n+1} \Big|. \tag{133}$$

Applying Claim 2 (for this note that ρ is a linear function of k, and therefore ρ^{2n+1} is a polynomial in k of degree $2n+1$):

$$\begin{aligned}
&\left|\frac{1}{2^d}\sum_{k=\lceil d/4\rceil}^{\lfloor 3d/4\rfloor}(-1)^k\binom{d}{k}\rho^{2n+1}\right| && (134)\\
&\qquad\le\left|\frac{1}{2^d}\sum_{k=0}^{d}(-1)^k\binom{d}{k}\rho^{2n+1}\right|+\left|\frac{1}{2^d}\sum_{k:|k-d/2|\ge d/4}(-1)^k\binom{d}{k}\rho^{2n+1}\right| && (135)\\
&\qquad\le\sum_{k:|k-d/2|\ge d/4}\binom{d}{k}2^{-d} && (136)\\
&\qquad=\mathbb{P}(|k-d/2|\ge d/4) && (137)\\
&\qquad\le 2\exp\left(-d/10\right)\ . && (138)
\end{aligned}$$

Finally, we substitute into equation 133 and conclude $|T_1|\le 2\exp\left(-d/10\right)$. $\square$

C.4 PROOF OF LEMMA 6

In this proof we will use the probabilist's Hermite polynomials $H_k(x)=\frac{(-1)^k}{\varphi(x)}\frac{\mathrm{d}^k}{\mathrm{d}x^k}\varphi(x)$, where $\varphi(x)=\frac{1}{\sqrt{2\pi}}\exp(-x^2/2)$ is the standard Gaussian density, see, e.g., Lebedev (1972) for more details. One property that we will need is that for two centered ρ-correlated unit variance joint Gaussians G_1, G_2 it holds

$$\mathbb{E}H_m(G_1)H_n(G_2)=\begin{cases}m! & \text{if } m=n,\\ 0 & \text{otherwise.}\end{cases} \tag{139}$$

We will also make use of the ReLU Hermite expansion, see, e.g., Proposition 6 in Abbe et al. (2022c). That is, $\mathrm{ReLU}(x)=\frac{1}{\sqrt{2\pi}}+\frac{1}{2}x+\sum_{m=1}^{\infty}a_mH_{2m}(x)$ for $a_m:=\frac{(-1)^{m+1}}{\sqrt{2\pi}2^m(2m-1)m!}$ and consequently, applying equation 139,

$$\mathbb{E}\,\mathrm{ReLU}(G_1)\,\mathrm{ReLU}(G_2)=\frac{1}{2\pi}+\frac{1}{4}\rho+\sum_{m=1}^{\infty}a_m^2(2m)!\rho^{2m}\ . \tag{140}$$

Furthermore, in any case we always have

$$\mathbb{E}\,\mathrm{ReLU}(G_1)\,\mathrm{ReLU}(G_2)\le\mathbb{E}\,\mathrm{ReLU}^2(G_1)=\frac{1}{2}. \tag{141}$$

As in Lemma 5, conditioned on k, random variables G_1, G_2 are centered unit variance Gaussians with correlation $\rho=\rho(k)=\left(1-\frac{2k}{d}\right)\alpha+\beta$. In particular, by equation 129, as long as $d/4\le k\le 3d/4$, then $|\rho|\le 1-\frac{\alpha_0}{2}$. Now we estimate, for $d\ge 2$, applying Claim 2 in equation 143 and again in equation 148:

$$
\begin{aligned}
&\left|\mathbb{E}(-1)^k \operatorname{ReLU}(G_1)\operatorname{ReLU}(G_2)\right| && (142)\\
&\quad= \left|\mathbb{E}(-1)^k\left(\operatorname{ReLU}(G_1)\operatorname{ReLU}(G_2) - \frac{1}{2\pi} - \frac{1}{4}\rho\right)\right| && (143)\\
&\quad\le \Pr[|k-d/2| > d/4] + \left|\sum_{k=\lceil d/4\rceil}^{\lfloor 3d/4\rfloor}(-1)^k\binom{d}{k}\sum_{m=1}^{\infty} a_m^2(2m)!\rho^{2m}\right| && (144)\\
&\quad\le 2\exp(-d/10) + \sum_{2m<d} a_m^2(2m!)\left|\sum_{k=\lceil d/4\rceil}^{\lfloor 3d/4\rfloor}(-1)^k\binom{d}{k}\rho^{2m}\right| && (145)\\
&\qquad + \sum_{2m\ge d} a_m^2(2m!)\left|\sum_{k=\lceil d/4\rceil}^{\lfloor 3d/4\rfloor}(-1)^k\binom{d}{k}\rho^{2m}\right| && (146)\\
&\quad\le 2\exp(-d/10) + \sum_{2m<d}\left|\sum_{k=\lceil d/4\rceil}^{\lfloor 3d/4\rfloor}(-1)^k\binom{d}{k}\rho^{2m}\right| + \sum_{2m\ge d}\left(1-\frac{\alpha_0}{2}\right)^{2m} && (147)\\
&\quad\le C'\exp(-Cd) + \sum_{2m<d}\left(\left|\sum_{k=0}^{d}(-1)^k\binom{d}{k}\rho^{2m}\right| + \Pr[|k-d/2|>d/4]\right) && (148)\\
&\quad\le C'\exp(-Cd)\ . && (149)
\end{aligned}
$$

□

D SMALL ALIGNMENT FOR PERTURBED INITIALIZATION: PROOF OF THEOREM 9

As we will make use of Hermite polynomials, it will be convenient to rescale the initialization: Let $w' = g + \mu r$, where $g \sim \mathcal{N}\left(0, \frac{1}{d}\operatorname{Id}\right)$ and the coordinates of r are i.i.d Rademacher. Let's write

$$
\begin{aligned}
\operatorname{GAL}'(\mu, d) &= \mathbb{E}_{g,r}\left[\left(\mathbb{E}_x\left[\prod_{i=1}^{d} x_i \mathbb{1}[(g+\mu r)\cdot x \ge 0]\right]\right)^2\right]\\
&= \mathbb{E}_{g,x,x',r}\left[\prod_{i=1}^{d} x_i x_i' \mathbb{1}[(g+\mu r)\cdot x, (g+\mu r)\cdot x' \ge 0]\right]\ .
\end{aligned}
$$

Using the identity $\operatorname{GAL}'(\mu, d) = \operatorname{GAL}\left(\frac{1}{\mu\sqrt{d}}, d\right)$, it is straightforward to see that the statement below is equivalent to Theorem 9:

Theorem 10. *There exists some $\alpha_0, C > 0$ and D_0 such that, for $d \ge D_0$, and $\mu \le \frac{\alpha_0}{\sqrt{d}}$, it holds $\operatorname{GAL}'(\mu, d) \le \exp(-Cd)$.*

Proof. Let us write $\mu = \alpha/\sqrt{d}$, so that by assumption $\alpha \le \alpha_0$. We have,

$$
\begin{aligned}
\operatorname{GAL}' &= \mathbb{E}_{g,x,x',r}\left[\prod_i x_i x_i' \mathbb{1}[g\cdot x + \mu r\cdot x, g\cdot x' + \mu r \cdot x' \ge 0]\right] && (150)\\
&= \mathbb{E}_{x,x',r}\left[\prod_i x_i x_i' \Pr_g[g\cdot x + \mu r\cdot x, g\cdot x' + \mu r\cdot x' \ge 0]\right] =: \mathbb{E}_{x,x',r}F(x,x',r)\ . && (151)
\end{aligned}
$$

As for every $x, x', r, s \in \{-1,1\}^d$ we have $F(x,x',r) = F(x\odot s, x'\odot s, r\odot s)$ (where $\odot$ denotes the Hadamard product), it follows

$$
\mathbb{E}_{x,x',r}F(x,x',r) = \mathbb{E}_{x,x'}F(x,x',1^d)\ , \tag{152}
$$

so we can rewrite

$$\mathrm{GAL}' = \mathbb{E}_{x,x'}\left[\prod_i x_i x'_i \Pr_g[g\cdot x + \mu\cdot x, g\cdot x' + \mu\cdot x' \geq 0]\right] = \mathbb{E}_{x,x'} F(x, x', 1^d)\ .$$

Fix x and assume w.l.o.g. that $x = (1^{d-d'}, -1^{d'})$ for some $0 \leq d' \leq d$. Furthermore, divide $x' = (y, z)$ such that $y \in \{-1,1\}^{d-d'}$ and $z \in \{-1,1\}^{d'}$. Assume that $d' \geq d/2$ and fix y. (If $d' < d/2$ we exchange the roles of y and z and proceed with an entirely symmetric argument.) Let $G(x,y,z) = F(x,(y,z),1^d)$. We want to analyze $\mathbb{E}_z G(x,y,z)$ so that the bound on $\mathbb{E}_{x,x'} F(x,x',1^d) = \mathbb{E}_{x,y,z} G(x,y,z)$ will follow by averaging. Let $\rho = \frac{1}{d} x \cdot x'$ and k be the number of -1 entries in z. Note that we have $\rho = \frac{1\cdot y + 2k - d'}{d}$. Continuing:

$$\begin{aligned}
|\mathbb{E}_z G(x,y,z)| &= \left|(-1)^{d'} \prod_{i=1}^{d-d'} y_i \mathbb{E}_z\left[(-1)^k \Pr_g[g\cdot x + \mu\cdot x, g\cdot x' + \mu\cdot x' \geq 0]\right]\right| && (153)\\
&= \left|\mathbb{E}_k\left[(-1)^k \Lambda_\rho(\mu(d-2d'), \mu(1\cdot y + d' - 2k))\right]\right| && (154)\\
&= \left|\mathbb{E}_k\left[(-1)^k \Lambda_\rho(\mu(d-2d'), -\mu(d\rho - 2\cdot y))\right]\right|\ , && (155)
\end{aligned}$$

where $\Lambda_\rho(a,b) = \Pr_{g,g'}[g + a, g' + b \geq 0] = \Pr_{g,g'}[g \leq a, g' \leq b]$, where g, g' are two standard ρ-correlated joint Gaussians. Note that the distribution of k is binomial, that is $\Pr[k = k^*] = 2^{-d'}\binom{d'}{k^*}$ for $0 \leq k^* \leq d'$.

In particular, conditioned on x, y, the expectation in equation 155 can be written as $|\mathbb{E}_k G(x,y,z)| = |\sum_{k=0}^{d'}(-1)^k\binom{d'}{k}W(\rho)|$ for some function W that depends only on ρ. Since ρ is a linear function of k, as in the Gaussian case, we will now expand W as a power series and apply Claim 2.

Let

$$A := \mu(d - 2d')\, , B := 2\mu\cdot y\, , C := -\mu d\ \text{, and } w := B + C\rho\ . \qquad (156)$$

Take some $\beta > 0$, where later on we will choose it to be a small enough universal constant (in fact $\beta = 0.005$ will be enough). Let us define two "bad" events: $\mathcal{E}_1$ is $|\rho| \geq 1/2$ and $\mathcal{E}_2$ is $|w| \geq \beta\sqrt{d}$ and let $\mathcal{F}$ be the complement of $\mathcal{E}_1 \cup \mathcal{E}_2$.

First, let us argue that $\Pr[\mathcal{E}_1 \cup \mathcal{E}_2] \leq \exp(-c\beta^2 d)$ for some universal $c > 0$ and d large enough:

$$\begin{aligned}
\Pr[\mathcal{E}_1 \cup \mathcal{E}_2] &\leq \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_2] && (157)\\
&= \Pr[|\rho| \geq 1/2] + \Pr\left[|w| \geq \beta\sqrt{d}\right] && (158)\\
&\leq \Pr\left[\left|\sum_{i=1}^d x_i x'_i\right| \geq \frac{d}{2}\right] + \Pr\left[|B| \geq \beta\frac{\sqrt{d}}{2}\right] + \Pr\left[|C\rho| \geq \beta\frac{\sqrt{d}}{2}\right] && (159)\\
&\leq \Pr\left[\left|\sum_{i=1}^d x_i x'_i\right| \geq \frac{d}{2}\right] + \Pr\left[\left|\sum_{i=1}^{d-d'} y_i\right| \geq \beta\frac{d}{4\alpha}\right] + \Pr\left[\left|\sum_{i=1}^d x_i x'_i\right| \geq \frac{\beta d}{2\alpha}\right] && (160)\\
&\leq 2\exp(-\frac{d}{8}) + 2\exp\left(-\frac{\beta^2 d^2}{32\alpha^2(d-d')}\right) + 2\exp\left(-\frac{\beta^2 d}{8\alpha^2}\right) && (161)\\
&\leq \exp(-c\beta^2 d)\ , && (162)
\end{aligned}$$

where equation 161 is by Hoeffding's inequality. Using equation 155, our bound becomes

$$\begin{aligned}
\mathrm{GAL}' &\leq \mathbb{E}_{x,y}\left|\mathbb{E}_k(-1)^k\Lambda_\rho(A,w)\right| && (163)\\
&\leq \Pr[\mathcal{E}_1 \cup \mathcal{E}_2] + \mathbb{E}_{x,y}\left|\mathbb{E}_k(-1)^k\Lambda_\rho(A,w)\mathbb{1}_{\mathcal{F}}\right| && (164)\\
&\leq \exp(-c\beta^2 d) + \mathbb{E}_{x,y}\left|\mathbb{E}_k(-1)^k\Lambda_\rho(A,w)\mathbb{1}_{\mathcal{F}}\right|\ . && (165)
\end{aligned}$$

To study the expression $\Lambda_\rho(A, w)$, let us recall some facts about the Gaussians. We have the following expansions:

$$\Phi(z) = \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k}{2^k k!(2k+1)} z^{2k+1} \tag{166}$$

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k}{2^k k!} z^{2k} \ , \tag{167}$$

as well as the tetrachoric series for Λ (convergent for every $a, b \in \mathbb{R}$ and $|\rho| < 1$) Harris & Soms (1980), Vasicek (1998):

$$\Lambda_\rho(a, b) = \Phi(a)\Phi(b) + \phi(a)\phi(b) \sum_{k=0}^{\infty} H_k(a) H_k(b) \frac{1}{(k+1)!} \rho^{k+1} \ . \tag{168}$$

Substituting into equation 165,

$$\mathrm{GAL}' \le \exp(-c\beta^2 d) + \mathbb{E}_{x,y} \left| \mathbb{E}_k (-1)^k \mathbb{1}_{\mathcal{F}} \left(\Phi(A)\Phi(w) + \phi(A)\phi(w) \sum_{\ell=0}^{\infty} H_\ell(A) H_\ell(w) \frac{\rho^{\ell+1}}{(\ell+1)!} \right) \right| \tag{169}$$

$$\le \exp(-c\beta^2 d) + \mathbb{E}_{x,y} \left| \mathbb{E}_k (-1)^k \mathbb{1}_{\mathcal{F}} \Phi(A)\Phi(w) \right| \tag{170}$$

$$+ \sum_{\ell=0}^{\infty} \mathbb{E}_{x,y} \left| \mathbb{E}_k (-1)^k \mathbb{1}_{\mathcal{F}} \phi(A)\phi(w) H_\ell(A) H_\ell(w) \frac{\rho^{\ell+1}}{(\ell+1)!} \right| \tag{171}$$

$$\le \exp(-c\beta^2 d) + \underbrace{\mathbb{E}_{x,y} \left| \mathbb{E}_k (-1)^k \mathbb{1}_{\mathcal{F}} \Phi(w) \right|}_{=:T_1} \tag{172}$$

$$+ \underbrace{\sum_{\ell=0}^{\infty} \mathbb{E}_{x,y} \left| \mathbb{E}_k (-1)^k \mathbb{1}_{\mathcal{F}} \phi(w) H_\ell(w) \frac{\rho^{\ell+1}}{\sqrt{\ell!}} \right|}_{=:T_2} , \tag{173}$$

where in the last line we used the estimate from (Harris & Soms, 1980, proof of Theorem 2),

$$|H_\ell(A)| \le 2\exp(A^2/4)\sqrt{\ell!} \ , \tag{174}$$

which implies

$$|\phi(A) H_\ell(A)| \le \sqrt{\ell!} \ . \tag{175}$$

For tighter estimates on Hermite polynomials, see also Bonan & Clark (1990).

It remains to show that both T_1 and T_2 are exponentially small.

Let us start with T_1. Recall equation 166 and let $a_\ell = \frac{(-1)^\ell}{\sqrt{2\pi} 2^\ell \ell!(2\ell+1)}$. Using equation 166 and triangle inequality,

$$T_1 \le \mathbb{E}_{x,y}\left|\mathbb{E}_k(-1)^k \mathbb{1}_{\mathcal{F}}\left(\frac{1}{2}+\sum_{\ell<d/10} a_\ell w^{2\ell+1}\right)\right| + \sum_{\ell\ge d/10}|a_\ell|(\beta\sqrt{d})^{2\ell+1} \tag{176}$$

$$\le \mathbb{E}_{x,y}\left|\mathbb{E}_k(-1)^k \left(\frac{1}{2}+\sum_{\ell<d/10} a_\ell w^{2\ell+1}\right)\right| + \mathbb{E}_{x,y}\left|\mathbb{E}_k(-1)^k \mathbb{1}_{\mathcal{E}_1\cup\mathcal{E}_2}\left(\frac{1}{2}+\sum_{\ell<d/10} a_\ell w^{2\ell+1}\right)\right| \tag{177}$$

$$+ \sum_{\ell\ge d/10}|a_\ell|(\beta\sqrt{d})^{2\ell+1} \tag{178}$$

$$\le \Pr[\mathcal{E}_1\cup\mathcal{E}_2]\left(\frac{1}{2}+\sum_{\ell<d/10}|a_\ell|(\alpha\sqrt{d})^{2\ell+1}\right) + \sum_{\ell\ge d/10}|a_\ell|(\beta\sqrt{d})^{2\ell+1} \tag{179}$$

$$\le \exp(-c\beta^2 d)\left(\frac{1}{2}+\sum_{\ell<d/10}|a_\ell|(\alpha\sqrt{d})^{2\ell+1}\right) + \sum_{\ell\ge d/10}|a_\ell|(\beta\sqrt{d})^{2\ell+1}\ . \tag{180}$$

In the right term in equation 176 we used that event $\mathcal{F}$ implies $|w| \le \beta\sqrt{d}$. In equation 177, we apply Claim 2 to the first term. This is valid since w is a linear function of k, and since $2\ell+1 < 2d/10+1 \le d/2 \le d'$, which holds for $d \ge 4$. In bounding the second term in equation 177, we used a uniform bound $|w| = |\mu x'| \le \alpha\sqrt{d}$.

We will now argue that both terms in equation 180 are exponentially small. Let us start with the second term:

$$\sum_{\ell\ge d/10}|a_\ell|(\beta\sqrt{d})^{2\ell+1} \le \sum_{\ell\ge d/10}\frac{(\beta\sqrt{d})^{2\ell+1}}{\ell!} \le \beta\sqrt{d}\sum_{\ell\ge d/10}\exp(\ell\ln d + 2\ell\ln\beta - \ell\ln\ell + \ell) \tag{181}$$

$$\le \beta\sqrt{d}\sum_{\ell\ge d/10}\exp(2\ell\ln\beta + \ell\ln 10 + \ell) \tag{182}$$

$$= \beta\sqrt{d}\sum_{\ell\ge d/10}(10e\beta^2)^\ell \le \beta\sqrt{d}\sum_{\ell\ge d/10}2^{-\ell} \le 2\beta\sqrt{d}2^{-d/10} \le \exp(-c'd)\ , \tag{183}$$

where the first inequality in equation 183 follows if β satisfies $10e\beta^2 \le 1/2$.

Now let us move to the left-hand side term in equation 180. It is sufficient to prove $1/2 + \sum_{\ell<d/10}|a_\ell|(\alpha\sqrt{d})^{2\ell+1} \le \exp(c\beta^2 d/2)$ and this is what we are going to show. Indeed,

$$\sum_{\ell<d/10}|a_\ell|(\alpha\sqrt{d})^{2\ell+1} \le \sum_{\ell<d/10}\frac{(\alpha\sqrt{d})^{2\ell+1}}{\ell!} \le \alpha\sqrt{d}\sum_{\ell<d/10}\frac{(e\alpha\sqrt{d})^{2\ell}}{\ell^\ell}\ . \tag{184}$$

Consider the function $f(\ell) = \frac{(e\alpha\sqrt{d})^{2\ell}}{\ell^\ell}$. We check that its derivative is $f'(\ell) = f(\ell)\big(\ln\big((e\alpha)^2 d\big) - 1 - \ln\ell\big)$. Therefore, f achieves its maximum at $\ell^* = \alpha^2 ed$ and we have

$$\frac{(e\alpha\sqrt{d})^{2\ell}}{\ell^\ell} = f(\ell) \le f(\ell^*) = \exp(e\alpha^2 d) \tag{185}$$

for every $\ell \ge 0$. For α small enough, for example if $\alpha^2 e \le c\beta^2/2$, we can substitute into equation 184 to get $\sum_{\ell<d/10}|a_\ell|(\alpha\sqrt{d})^{2\ell+1} \le \alpha d\sqrt{d}\exp(e\alpha^2 d)$ and consequently

$$1/2 + \sum_{\ell<d/10}|a_\ell|(\alpha\sqrt{d})^{2\ell+1} \le \exp(c\beta^2 d/2)\ . \tag{186}$$

In summary, by combining equation 180, equation 183, and equation 186, the inequality $T_1 \leq \exp(-\Omega(d))$ is established for large enough d.

We now turn to bounding T_2. The idea is essentially the same with a more complicated calculation. Recall equation 167, let $b_m := \frac{1}{\sqrt{2\pi}} \frac{(-1)^m}{2^m m!}$ and note for later that $|b_m| \leq 1/m!$. We write down

$$T_2 = \sum_{\ell=0}^{\infty} \left| \mathbb{E}_k (-1)^k \mathbb{1}_{\mathcal{F}} \phi(w) H_\ell(w) \frac{\rho^{\ell+1}}{\sqrt{\ell!}} \right| \tag{187}$$

$$\leq \underbrace{\sum_{\ell<d/10} \left| \mathbb{E}_k (-1)^k \mathbb{1}_{\mathcal{F}} \left(\sum_{m<d/10} b_m w^{2m} \right) H_\ell(w) \frac{\rho^{\ell+1}}{\sqrt{\ell!}} \right|}_{=:T_3} \tag{188}$$

$$+ \underbrace{\sum_{\ell<d/10, m\geq d/10} \frac{1}{m!} \mathbb{E}_{x,y,k} \left| \mathbb{1}_{\mathcal{F}} w^{2m} \frac{H_\ell(w)}{\sqrt{\ell!}} \right|}_{=:T_4} \tag{189}$$

$$+ \underbrace{\sum_{\ell\geq d/10} \mathbb{E}_{x,y,k} \left| \mathbb{1}_{\mathcal{F}} \phi(w) H_\ell(w) \frac{\rho^{\ell+1}}{\sqrt{\ell!}} \right|}_{=:T_5} . \tag{190}$$

Let us argue in turns that each of T_3, T_4, T_5 is exponentially small proceeding in the reverse order. For T_5, we use equation 175 and the fact that event $\mathcal{F}$ implies $|\rho| \leq 1/2$:

$$T_5 \leq \sum_{\ell\geq d/10} 2^{-\ell+1} \leq 2^{-d/10} . \tag{191}$$

For T_4, we invoke equation 174 and event $\mathcal{F}$ implying $|w| \leq \beta\sqrt{d}$:

$$T_4 \leq 2d \exp(\beta^2 d/4) \sum_{m\geq d/10} \frac{(\beta^2 d)^m}{m!} \leq 2d \exp(\beta^2 d/4) \sum_{m\geq d/10} (10e\beta^2)^m . \tag{192}$$

If β is chosen such that $(10e\beta^2)^{1/10} \leq 1/2$ and $\exp(\beta^2/4) \leq 1.01$, then we can continue and obtain the desired bound

$$T_4 \leq 2d(1.01)^d 2^{-d} \leq \exp(-c'd) . \tag{193}$$

Finally, we turn to T_3:

$$T_3 \leq \sum_{\ell<d/10} \mathbb{E}_{x,y} \left| \mathbb{E}_k (-1)^k \left(\sum_{m<d/10} b_m w^{2m} \right) H_\ell(w) \frac{\rho^{\ell+1}}{\sqrt{\ell!}} \right| \tag{194}$$

$$+ \sum_{\ell<d/10} \mathbb{E}_{x,y} \left| \mathbb{E}_k (-1)^k \mathbb{1}_{\mathcal{E}_1 \cup \mathcal{E}_2} \left(\sum_{m<d/10} b_m w^{2m} \right) H_\ell(w) \frac{\rho^{\ell+1}}{\sqrt{\ell!}} \right| \tag{195}$$

$$\leq 2d \Pr[\mathcal{E}_1 \cup \mathcal{E}_2] \exp(\alpha^2 d/4) \sum_{m<d/10} \frac{(\alpha^2 d)^m}{m!} \tag{196}$$

$$\leq 2d^2 \exp(-c\beta^2 d) \exp(\alpha^2 d/4) \exp(e\alpha^2 d) \leq \exp(-c'd) . \tag{197}$$

The sum in equation 194 is equal zero by Claim 2: Indeed both w and ρ are linear functions of k, so the expression inside the absolute value is a polynomial of degree at most $2m + \ell + (\ell+1) < 4d/10 + 1 \leq d/2 \leq d'$. To bound the sum in equation 195, we applied $|b_m| \leq 1/m!$, $|w| \leq 3\alpha\sqrt{d}$, equation 174 and $|\rho| \leq 1$. Finally, to bound equation 196 we applied equation 162 and equation 185 and the final inequality follows if we choose α_0 small enough so that, e.g., $\alpha^2/4 + e\alpha^2 \leq c\beta^2/2$ (recall that β is already chosen to be a small enough absolute constant).

Summing up, equation 191, equation 193 and equation 197 substituted into equation 190 give $T_2 \leq \exp(-\Omega(d))$. Together with $T_1 \leq \exp(-\Omega(d))$, substituted into equation 173, we established $\mathrm{GAL}'(\mu, d) \leq \exp(-\Omega(d))$, which is what we set out to prove. □

E EXPERIMENT DETAILS AND ADDITIONAL EXPERIMENTS

E.1 EXPERIMENT DETAILS

All experiments were performed using the PyTorch framework (Paszke et al. (2019)) and they were executed on NVIDIA Volta V100 GPUs.

Architectures. For the results presented in the main, we used mainly a 4-layer MLP architecture trained by SGD with the hinge loss. In this Section, we also present some experiments obtained with a 2-layer MLP trained by SGD with the squared loss.

- **4-layer MLP.** This is a fully-connected architecture of 3 hidden layers of neurons of size $512, 512, 64$, and ReLU activation.
- **2-layer MLP.** This is again a fully-connected architecture, with 1 hidden layer of 512 neurons, and ReLU activation,

Initializations. We compare few initialization schemes. In the following, $\dim$ denotes the input dimension of the layer of the corresponding parameter. All layers weights and biases are independently initialized according to:

- **σ-perturbed Rademacher:** $\big(\mathrm{Rad}(1/2) + \mathcal{N}(0, \sigma^2)\big) \cdot \frac{1}{\sqrt{\dim \cdot (1+\sigma^2)}}$.
- **Gaussian:** $\mathcal{N}(0, \frac{1}{\dim})$.
- **s-sparsified Rademacher:** $\mathrm{Ber}(1-s) \cdot \mathrm{Rad}(1/2) \cdot \frac{1}{\sqrt{\dim \cdot (1-s)}}$.
- **Uniform σ-perturbed Rademacher:** $\big(\mathrm{Rad}(1/2) + \mathrm{Unif}[-\sqrt{3}\sigma, \sqrt{3}\sigma]\big) \cdot \frac{1}{\sqrt{\dim \cdot (1+\sigma^2)}}$.
- **Discrete perturbed Rademacher:** $\mathrm{Unif}\{-2, -1, 1, 2\} \cdot \sqrt{\frac{2}{5 \cdot \dim}}$.

Training procedure. We consider mainly the hinge loss: $L_{\text{hinge}}(\hat{y}, y) := \max(0, 1 - \hat{y}y)$. In some experiments we consider the ℓ_2 loss: $L_{\ell_2}(\hat{y}, y) := (\hat{y} - y)^2$. We train the architectures using SGD with batch size 64. In the online setting, we sample fresh batches of samples at each iterations. In the offline setting, we sample batches from a fixed dataset and we stop training when the training loss is less than 0.01.

Hyperparameter tuning. The primary goal of our experiments is to conduct a fair comparison of different initialization methods. Thus, we did not engage in extensive hyperparameter tuning. We tried different batch sizes and learning rates, and we did not observe significant qualitative difference. We chose to report the experiments obtained for a standard batch size of 64 and a learning rate of 0.01.

Additional details for Figure 2. In the left plot of Figure 2, we are computing the quantity $\mathbb{E}_w \left[\mathbb{E}_{x,r} \left[\frac{\partial L(w,x,f(x))}{\partial w_d} - \frac{\partial L(w,x,r)}{\partial w_d} \right]^2 \right]$, where $w \sim \mathcal{N}(0, \frac{1}{d}\,\mathrm{Id}_d)$ for one case and $w \sim \mathrm{Rad}(1/2)$ for the other case, f is the full parity, $r \sim \mathrm{Rad}(1/2)$ and $L(w, x, y) := \max\left(0, 1 - y\,\mathrm{ReLU}(w.x)\right)$ is the hinge loss. For the approximated part we update the weights according to $\psi^{t+1} = \psi^t - \gamma\left(\Gamma_r(\psi^t)\right)$, with $\psi^0 \sim \mathcal{N}(0, \frac{1}{d}\,\mathrm{Id}_d)$ and $\gamma = 1$, and we calculate $\mathbb{E}_{\psi^t} \left[\mathbb{E}_{x,r} \left[\frac{\partial L(\psi^t,x,f(x))}{\partial \psi^t_d} - \frac{\partial L(\psi^t,x,r)}{\partial \psi^t_d} \right]^2 \right]$.

E.2 ADDITIONAL EXPERIMENTS

Larger input dimension. In Figure 5, we plot the test accuracy achieved by a 4-layer MLP trained with the hinge loss on the full parity task, with different σ-perturbed initializations. We report only the curves for small σ. We observe that for fixed σ, learning becomes hard as d increases.

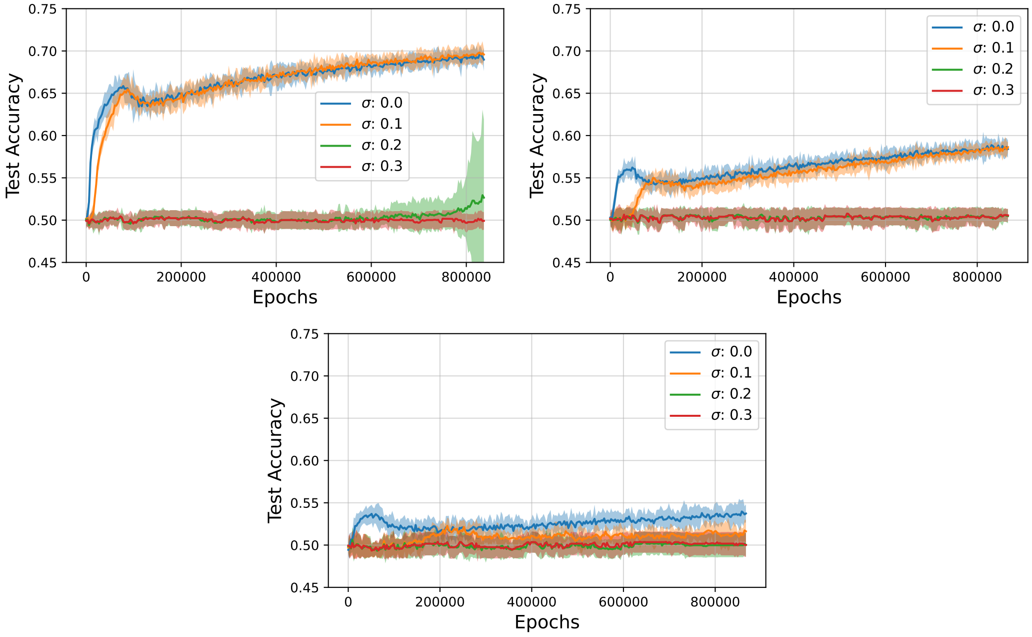

Figure 5: Learning the full parity with σ-perturbed initialization by SGD with the hinge loss on a 4-layer MLP, with input dimension $d = 100$ (top-left), $d = 150$ (top-right) and $d = 200$ (bottom), with online fresh samples.

Alignment for correlation loss. Figure 6 completes Figure 2 (right) in the main. Here we plot the numerically computed GAL_f for larger values of σ. We observe that the GAL_f becomes consistently smaller as σ increases. Moreover, from the plot the decay seems super-polynomially small for all $\sigma > 0$.

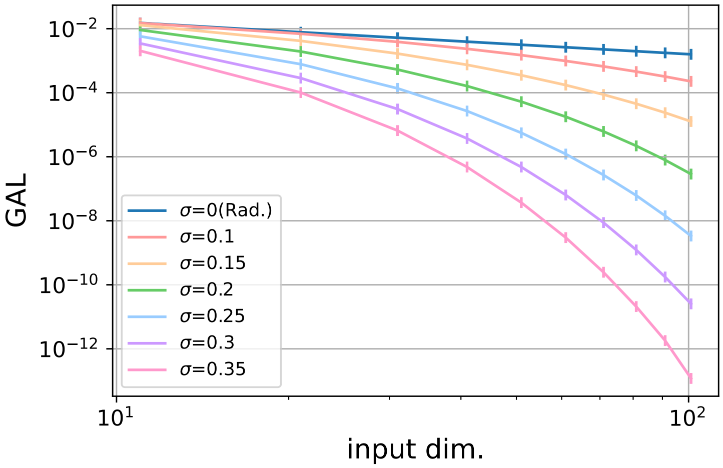

Figure 6: Computing numerically GAL_f for correlation loss for one-neuron with threshold activation. We report the estimated GAL_f for different values of the input dimension, in a log-log plot.

Two-layer MLP and squared loss. In Figure 7 we train a 2-layer MLP with the squared loss and online fresh samples. In the left plot, we initialize the weights according to σ-perturbed Rademacher, for different values of σ. In the right plot, we initialize with other perturbations of the Rademacher initialization, namely a mixture of (continuous) uniform distributions of mean $+1$ and -1 and standard deviation σ and s-sparsified Rademacher with $s = 1/3$. We observe in both plots a similar behavior as for the 4-layer MLP with the hinge loss.

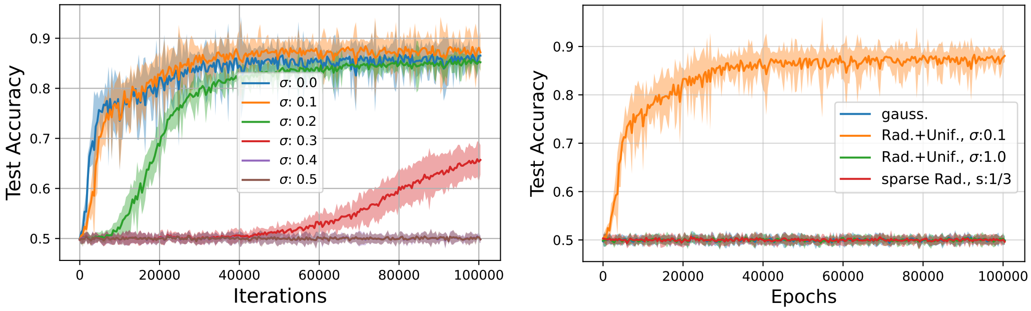

Figure 7: Learning the full parity with σ-perturbed Rademacher (left) and uniform and sparse perturbed Rademacher (right) with a 2-layer MLP trained with the squared loss, with online fresh samples.

Effect of the Loss. We consider the following Boolean function:

$$f(x) := \frac{1}{8}x_1x_2x_3 + \frac{3}{8}x_1x_2x_4 + \frac{1}{4}x_1x_3x_4 + \frac{1}{4}x_2x_3x_4. \tag{198}$$

In (Joshi et al. (2024)), the authors show that this function is learned more efficiently by SGD with L1-loss than with L2-loss (see Section 7.1 therein). In Figure 8, we observe that such difference is captured by our loss-dependent notion of Initial Gradient Alignment (GAL). This motivates future work in comparing our GAL with previously defined measures (e.g. LGA (Mok et al. (2022))) in a broader setting.

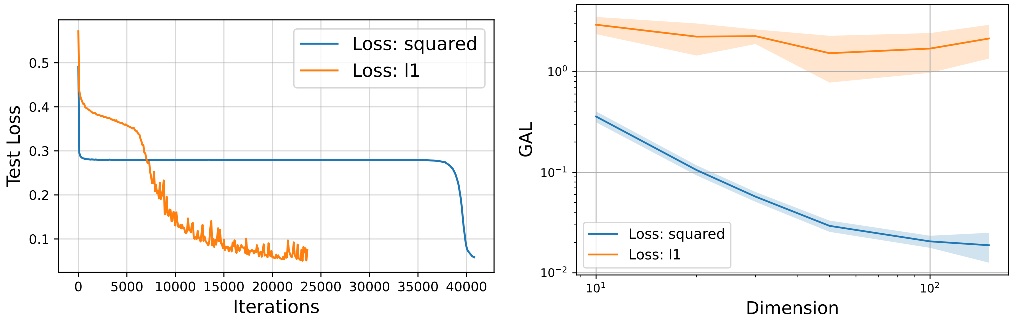

Figure 8: (left) Learning f (Eq. equation 198) with SGD with the L1 and L2 (squared) loss on a 4-layer MLP, with input dimension $d = 50$. (right) Initial GAL for f on the same architecture, with the two losses.