# Matching the Statistical Query Lower Bound for $k$-Sparse Parity Problems with Sign Stochastic Gradient Descent

**Yiwen Kou**[*]
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA
evankou@cs.ucla.edu

**Zixiang Chen**[*]
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA
chenzx19@cs.ucla.edu

**Quanquan Gu**
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA
qgu@cs.ucla.edu

**Sham M. Kakade**
Kempner Institute at Harvard University
Harvard University
Cambridge, MA 02138, USA
sham@seas.harvard.edu

## Abstract

The $k$-sparse parity problem is a classical problem in computational complexity and algorithmic theory, serving as a key benchmark for understanding computational classes. In this paper, we solve the $k$-sparse parity problem with sign stochastic gradient descent, a variant of stochastic gradient descent (SGD) on two-layer fully-connected neural networks. We demonstrate that this approach can efficiently solve the $k$-sparse parity problem on a $d$-dimensional hypercube ($k \le O(\sqrt{d})$) with a sample complexity of $\widetilde{O}(d^{k-1})$ using $2^{\Theta(k)}$ neurons, matching the established $\Omega(d^k)$ lower bounds of Statistical Query (SQ) models[2]. Our theoretical analysis begins by constructing a *good* neural network capable of correctly solving the $k$-parity problem. We then demonstrate how a trained neural network with sign SGD can effectively approximate this good network, solving the $k$-parity problem with small statistical errors. To the best of our knowledge, this is the first result that matches the SQ lower bound for solving $k$-sparse parity problem using gradient-based methods.

## 1 Introduction

The $k$-parity problem, defined on a binary sequence of length $d$, is a fundamental problem in the field of computational complexity and algorithmic theory. This problem involves finding a subset of cardinality $k$ by assessing if the occurrence of 1's in this subset is even or odd. The complexity of the problem escalates as the parameter $k$ increases. Its significance, while evidently practical, is rooted in its theoretical implications; it serves as a vital benchmark in the study of computational complexity classes and has profound implications for our understanding of P versus NP (Vardy, 1997; Downey et al., 1999; Dumer et al., 2003) and other cornerstone questions in computational theory (Blum, 2005; Klivans et al., 2006). Furthermore, the $k$-parity problem's complexity underpins many theoretical models in error detection and information theory (Dutta et al., 2008), and is instrumental

[*] Equal contribution

[2] The $\widetilde{O}(d^{k-1})$ sample complexity implies $\widetilde{O}(d^k)$ query complexity, because each sample corresponds to $d$ scalar-valued queries.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

in delineating the limitations and power of algorithmic efficiency (Farhi et al., 1998). This paper tackles the $k$-sparse parity problem (Daniely and Malach, 2020), where the focus is on the parity of a subset with cardinality $k \ll d$.

Recent progress in computational learning theory has focused on improving the sample complexity guarantees for learning $k$-sparse parity functions using stochastic gradient descent (SGD). Under the framework of the Statistical Query (SQ) model (Kearns, 1998), it has been established that learning the $k$-sparse parity function requires a minimum of $\Omega(d^k)$ queries (Barak et al., 2022), highlighting the challenge in efficiently learning these functions. On the other hand, considerable effort has been devoted to establishing sample complexity upper bounds for the special XOR case ($k = 2$), with notable successes including $O(d)$ sample complexity using infinite-width or exponential-width (i.e., $O(2^d)$) two-layer neural networks trained via gradient flow (Wei et al., 2019; Chizat and Bach, 2020; Telgarsky, 2022), and $O(d^2)$ sample complexity with polynomial-width networks via SGD (Ji and Telgarsky, 2020; Telgarsky, 2022). A significant advancement in solving the 2-parity problem was recently introduced by Glasgow (2023). They proved a sample complexity bound of $\widetilde{O}(d)$ using a two-layer ReLU network with logarithmic width trained by SGD, thus matching the SQ lower bound when $k = 2$.

In the general case of $k \geq 2$, Barak et al. (2022) has made significant progress, achieving a sample complexity of $\widetilde{O}(d^{k+1})$ with a network width requirement of $2^{\Theta(k)}$, which is independent of the input dimension $d$. Additionally, Barak et al. (2022) demonstrated that the neural tangent kernel (NTK)-based method (Jacot et al., 2018) requires a network of polynomial width $d^{\Omega(k)}$ to solve the $k$-parity problem. Recently, Suzuki et al. (2023) achieved a sample complexity of $O(d)$ by the mean-field Langevin dynamics (MFLD) (Mei et al., 2018; Hu et al., 2019), which requires neural networks with an exponential width in $d$, i.e., $O(e^d)$, and an exponential number of iterations (i.e., $O(e^d)$ ) to converge. Thus, their method is not computationally efficient and does not match the SQ lower bound. Notably, Abbe et al. (2023a) introduced the leap-$k$ function for binary and Gaussian sequences, which extends the scope of the $k$-parity problem. They also proved Correlational Statistical Query (CSQ) (Kearns, 1998; Bshouty and Feldman, 2002) lower bounds for learning leap-$k$ function for both Gaussian and Boolean inputs. In detail, they proved CSQ lower bounds of $\Omega(d^{k-1})$ for Boolean input and $\Omega(d^{k/2})$ for Gaussian input, which suggests that learning from Boolean input can be substantially harder than learning from Gaussian input. They also proved that SGD can learn low dimensional target functions with Gaussian isotropic data and 2-layer neural networks using $n \gtrsim d^{\mathrm{Leap}-1}$ examples. However, their upper bound analysis is based on the assumption that the input data $\mathbf{x}$ follows a Gaussian distribution and relies on Hermite polynomials, making it unclear how to extend it to analyze Boolean input. Based on the above review of existing literature, it raises a natural but unresolved question:

*Is it possible to match the statistical query lower bound for $k$-sparse parity problems with stochastic gradient descent?*

In this paper, we give an affirmative answer to the above question. In particular, we consider the standard $k$-sparse parity problem, where the input $\mathbf{x}$ is drawn from a uniform distribution over $d$-dimensional hypercube $\mathrm{Unif}(\{-1, 1\}^d)$. Our approach involves training two-layer fully-connected neural networks with $m = 2^{\Theta(k)}$ width using sign SGD (Bernstein et al., 2018) with batch size $B = O(d^{k-1}\,\mathrm{polylog}(d))$. We prove that the neural network trained by SGD can achieve a constant-order positive margin with high probability after $T = O(\log d)$ iterations. Therefore, the total number of examples required in our approach is $n = BT = \widetilde{O}(d^{k-1})$. Thus, the total number of scalar-valued queries required in our paper is $m \cdot d \cdot n = 2^{\Theta(k)} \cdot d \cdot (d^{k-1} \cdot \mathrm{polylog} d) = 2^{\Theta(k)} d^k \cdot \mathrm{polylog} d$, where $m$ is the number of neurons, $d$ is the input dimension, and $n$ is the total number of fresh examples seen by the algorithm[3]. Abbe et al. (2023a) also proved a CSQ lower bound $\Omega(d^k)$ for learning $d$-dimensional $k$-parity problems, which implies the sample complexity lower bound $n \gtrsim d^{k-1}$. Thus, our sample complexity result also matches the CSQ lower bound in Abbe et al. (2023a).

### 1.1 Our Contributions

The Statistical Query (SQ) lower bound indicates that, regardless of architecture, SGD requires a query complexity of $\Omega(d^k)$ for learning $k$-sparse $d$-dimensional parities under a constant noise level.

[3] Note that $m \cdot d$ is the total number of scalar-valued queries used for one example.

We push the sample complexity frontier of $k$-sparse parity problem to $\widetilde{O}(d^{k-1})$ via SGD, specifically with online stochastic sign gradient descent. Our main result is stated in the following informal theorem:

**Theorem 1.1** (Informal)**.** *For a two-layer fully-connected neural networks of width $2^{\Theta(k)}$, online sign SGD with batch size $\widetilde{O}(d^{k-1})$ can find a solution to the $k$-parity problem with a small test error within $O(k \log d)$ iterations.*

The above theorem improves the sample complexity in Barak et al. (2022) from $\widetilde{O}(d^{k+1})$ to $\widetilde{O}(d^{k-1})$. Moreover, the total number of queries required is $\widetilde{O}(d^{k})$, which matches the SQ/CSQ lower bound up to logarithmic factors. Additionally, under the standard basis setting, our result matches the sample complexity in Glasgow (2023) for solving the XOR (i.e., 2-parity) problem with sign SGD. It is worth noting that our result only requires two-layer fully connected neural networks with $2^{\Theta(k)}$ width and sign SGD training with $O(k \log d)$ iterations, which gives a computationally efficient algorithm. Finally, we empirically verify our theory in Appendix A, showcasing the efficiency and efficacy of our approach.

**Notation.** We use $[N]$ to denote the index set $\{1, \dots, N\}$. We use lowercase letters, lowercase boldface letters, and uppercase boldface letters to denote scalars, vectors, and matrices, respectively. For a vector $\mathbf{v} = (v_1, \cdots, v_d)^\top$, we denote by $\|\mathbf{v}\|_2 := (\sum_{j=1}^{d} v_j^2)^{1/2}$ its $L_2$ norm. For a vector $\mathbf{v} = (v_1, \cdots, v_d)^\top$, we denote by $\mathbf{v}_{[i_1:i_2]} := (v_{i_1}, \cdots, v_{i_2})^\top$ its truncated vector ranging from the $i_1$-th coordinate to the $i_2$-th coordinate. We denote by $\mathbf{0}$ a vector of all zeros. For two sequence $\{a_k\}$ and $\{b_k\}$, we denote $a_k = O(b_k)$ if $|a_k| \le C|b_k|$ for some absolute constant $C$, denote $a_k = \Omega(b_k)$ if $b_k = O(a_k)$, and denote $a_k = \Theta(b_k)$ if $a_k = O(b_k)$ and $a_k = \Omega(b_k)$. We also denote $a_k = o(b_k)$ if $\lim |a_k/b_k| = 0$. We use $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ to omit logarithmic terms in the notation. Finally, we denote $x_n = \mathrm{poly}(y_n)$ if $x_n = \mathcal{O}(y_n^D)$ for some positive constant $D$, and $x_n = \mathrm{polylog}(y_n)$ if $x_n = \mathrm{poly}(\log(y_n))$.

## 2 Related Work

**XOR Problem.** The performance of two-layer neural networks in the task of learning 2-parity has been the subject of extensive research in recent years. Wei et al. (2019); Chizat and Bach (2020); Telgarsky (2022) employed margin techniques to establish the convergence toward a global margin maximization solution, utilizing gradient flow and sample complexity of $O(d)$. Notably, Wei et al. (2019) and Chizat and Bach (2020) employed infinite-width neural networks, while Telgarsky (2022) employed a more relaxed width condition of $O(d^d)$. A significant breakthrough in this domain was achieved by Glasgow (2023), who demonstrated a sample complexity of $\widetilde{O}(d)$ by employing SGD in conjunction with a ReLU network of width $\mathrm{polylog}(d)$. Furthermore, several other studies have shown that when input distribution follows Gaussian distribution, neural networks can be effectively trained to learn the XOR cluster distribution (Frei et al., 2022; Meng et al., 2023; Xu et al., 2023). A comparison between the results in this paper and those of related work solving the XOR problem can be found in Table 1.

**$k$-parity Problem.** The challenge of training neural networks to learn parities has been explored in previous research from diverse angles. Several papers studied learning the $k$-parity function by using two-layer neural networks. Daniely and Malach (2020) studied learning $k$-parity function by applying gradient descent on the population risk (infinite sample size). Notably, Barak et al. (2022) presented both empirical and theoretical evidence that the $k$-parity function can be effectively learned using SGD and a neural network of constant width, demonstrating a sample complexity of $O(d^{k+1})$ and query complexity of $O(d^{k+2})$. Edelman et al. (2024) demonstrated that sparse initialization and increased network width lead to improvements in sample efficiency. Specifically, they showed that the sample complexity can be reduced at the cost of increasing the width. However, the best statistical query complexity they can achieve is $O(d^{k+2})$, which is the same as that in Barak et al. (2022). Suzuki et al. (2023) reported achieving a sample complexity of $O(d)$ by employing mean-field Langevin dynamics (MFLD). Furthermore, Abbe et al. (2022) and Abbe et al. (2023a) introduced a novel complexity measure termed "leap" and established that leap-$k$ (with $k$-parity as a special case) functions can be learned through SGD with a sample complexity of $\widetilde{O}(d^{\max(k-1,1)})$. Additionally, Abbe et al. (2023b) demonstrated that a curriculum-based noisy-GD (or SGD) approach

|  | Activation Function | Loss Function | Algorithm | Width ($m$) Requirement | Sample ($n$) Requirement | Iterations ($t$) to Converge |
|---|---|---|---|---|---|---|
| Theorem 2.1<br>(Wei et al., 2019) | ReLU | logistic | WF with Noise | $\infty$ | $d/\epsilon$ | $\infty$ |
| Theorem 8<br>(Chizat and Bach, 2020) | 2-homogenous | logistic/hinge | WF | $\infty$ | $d/\epsilon$ | $\infty$ |
| Theorem 3.3<br>(Telgarsky, 2022) | ReLU | logistic | scalar GF | $d^d$ | $d/\epsilon$ | $d/\epsilon$ |
| Theorem 3.2<br>(Ji and Telgarsky, 2020) | ReLU | logistic | SGD | $d^8$ | $d^2/\epsilon$ | $d^2/\epsilon$ |
| Theorem 2.1<br>(Telgarsky, 2022) | ReLU | logistic | SGD | $d^2$ | $d^2/\epsilon$ | $d^2/\epsilon$ |
| Theorem 3.1<br>(Glasgow, 2023) | ReLU | logistic | SGD | $\mathrm{polylog}(d)$ | $d \cdot \mathrm{polylog}(d)$ | $\mathrm{polylog}(d)$ |
| Ours | $x^2$ | correlation | Sign SGD | $O(1)$ | $d \cdot \mathrm{polylog}(d)$ | $\log d$ |

Table 1: Comparison of existing works on the XOR (2-parity) problem. We mainly focus on the dependence on the input dimension $d$ and test error $\epsilon$ and treat other arguments as constant. Here WF denotes Wasserstein flow technique from the mean-field analysis, and GF denotes gradient flow. The sample requirement and convergence iteration in both Glasgow (2023) and our method do not explicitly depend on the test error $\epsilon$. Instead, the dependence on $\epsilon$ is implicitly incorporated within the condition for $d$. Specifically, our approach requires that $d \geq C \log^2(2m/\epsilon)$ while Glasgow (2023) requires $d \geq \exp((1/\epsilon)^C)$ where $C$ is a constant.

.

|  | Activation Function | Loss Function | Algorithm | Width ($m$) Requirement | Sample ($n$) Requirement | Iterations ($t$) to Converge |
|---|---|---|---|---|---|---|
| Theorem 4<br>(Barak et al., 2022) | ReLU | hinge | SGD | $2^{\Theta(k)}$ | $d^{k+1} \cdot \log(d/\epsilon)/\epsilon^2$ | $d/\epsilon^2$ |
| Theorem 4<br>(Edelman et al., 2024) | ReLU | hinge | SGD | $(d/s)^k$ | $(s/k)^{k-1} d^2 \log(d)/\epsilon^2$ | $1/\epsilon^2$ |
| Corollary 1<br>(Suzuki et al., 2023) | Variant of Tanh | logistic | MFLD | $e^d$ | $d/\epsilon$ | $e^d$ |
| Ours | $x^k$ | correlation | Sign SGD | $2^{\Theta(k)}$ | $d^{k-1} \cdot \mathrm{polylog}(d)$ | $\log d$ |

Table 2: Comparison of existing works for the general $k$-parity problem, focusing primarily on the dimension $d$ and error $\epsilon$, treating other parameters as constants. $s$ in Edelman et al. (2024) is the sparsity of the initialization that satisfies $s > k$. The activation function by Suzuki et al. (2023) is defined as $h_{\mathbf{w}}(\mathbf{x}) = \bar{R}[\tanh(\mathbf{x}^\top \mathbf{w}_1 + w_2) + 2\tanh(w_3)]/3$, where $\mathbf{w} = (\mathbf{w}_1, w_2, w_3)^\top \in \mathbb{R}^{d+2}$ and $\bar{R}$ is a hyper-parameter determining the network's scale. For the sample requirement and convergence iteration, we focus on the dependency of $d, \epsilon$ and omit another terms. Our method's sample requirement and convergence iteration are independent of the test error $\epsilon$, instead relying on a condition for $d$ that implicitly includes $\epsilon$. Specifically, we require $d \geq C \log^2(2m/\epsilon)$.

could attain a sample complexity of $O(d)$, provided the data distribution comprises a mix of sparse and dense inputs. The conditions outlined in this paper are compared to those from related work involving uniform Boolean data distribution, as presented in Table 2.

## 3 Problem Setup

In this section, we introduce the $k$-sparse parity problem and the neural network we consider in this paper.

**Definition 3.1** (***k*-parity**). *Let each data point $(\mathbf{x}, y)$ with $\mathbf{x} \in \mathbb{R}^d$ and $y \in \{-1, 1\}$ be generated from the following distribution $\mathcal{D}_A$, where $A$ is a non-empty set satisfying $A \subseteq [n]$:*

*1. $x_j \sim \{-1, 1\}$ as a uniform random bit for $j \in [d]$.*
*2. The label $y$ is generated as $\Pi_{j \in A} x_j$.*

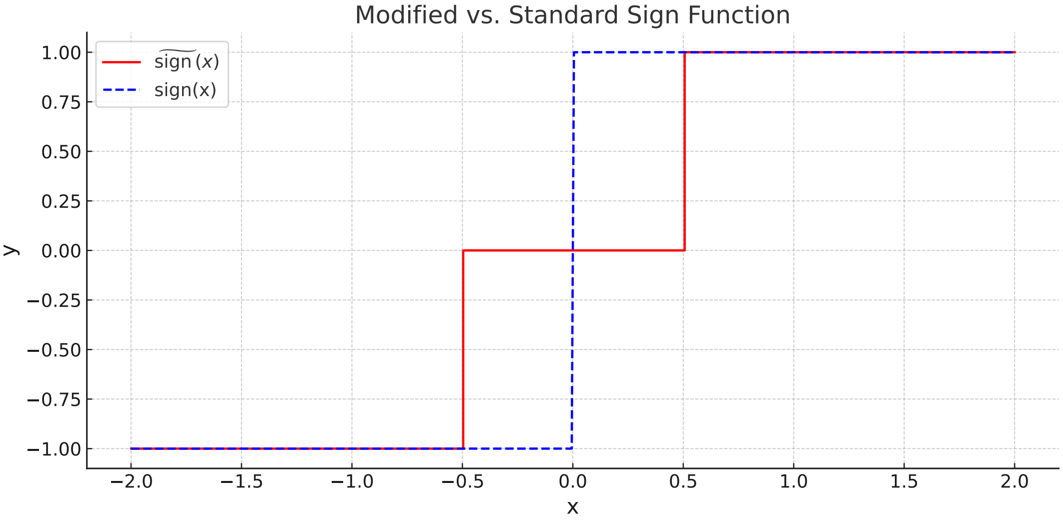

Figure 1: The plot above illustrates the comparison between the modified sign function $\widetilde{\text{sign}}(x)(\rho = 0.5)$ and the standard sign function $\text{sign}(x)$. The $\widetilde{\text{sign}}(x)$ function introduces a 'dead zone' between $-\rho$ and $\rho$ where the function value is zero, which is not present in the standard sign function. This modification effectively creates a threshold effect, only outputting non-zero values when the input $x$ exceeds the specified bounds of $\rho$ in either direction.

*The $k$-parity problem with dimension $d$ is defined as the task of recovering $A$, where $|A| = k$, using samples from $\mathcal{D}_A$.*

Without loss of generality, we assume that $A = \{1, \ldots, k\}$ if $|A| = k$. Under this assumption, we denote $\mathcal{D}_A$ by $\mathcal{D}$ for simplicity. This $k$-parity problem in Definition 3.1 is a classical one, which has been studied by Daniely and Malach (2020); Barak et al. (2022) using neural network learning. When restricted to the 2-parity function, the problem is reduced to the XOR problem (Wei et al., 2019).

**Two-layer Neural Networks.** We consider a two-layer fully-connected neural network, which is defined as follows:

$$f(\mathbf{W}, \mathbf{x}) = \sum_{r=1}^{m} a_r \sigma(\langle \mathbf{w}_r, \mathbf{x} \rangle), \tag{3.1}$$

where $m$ is the number of neurons. Here, we employ a polynomial activation function defined by $\sigma(z) = z^k$. The term $\mathbf{w}_r \in \mathbb{R}^d$ represents the weight vector for the $r$-th neuron, and $\mathbf{W}$ denotes the aggregate of all first-layer model weights. The second-layer weights $a_r$'s are sampled uniformly from the set $\{-1, 1\}$ and fixed during training.

**Algorithm.** We train the above neural network model by minimizing the correlation loss function:

$$L_{\mathcal{D}}(\mathbf{W}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \ell[y \cdot f(\mathbf{W}, \mathbf{x})],$$

where $\ell(z) = 1 - z$. We consider binary initialization with $\mathbf{w}_r^{(0)} \sim \text{Unif}(\{\pm 1\}^d)$, which is widely used for neural networks solving parity problem (Barak et al., 2022; Abbe and Boix-Adsera, 2022). We then employ the stochastic sign gradient descent with constant step size and weight decay (i.e., $\ell_2$ norm regularization on the first-layer weights) to minimize the correlation loss as follows:

$$\mathbf{W}^{(t+1)} = (1 - \lambda\eta)\mathbf{W}^{(t)} - \eta \cdot \widetilde{\text{sign}}\Big(\frac{\partial L^{(t)}}{\partial \mathbf{W}}\Big),$$

where $\lambda > 0$ is the weight decay parameter, $\eta > 0$ is the step size, and $\widetilde{\text{sign}}(x)$ is the modified sign function defined as:

$$\widetilde{\text{sign}}(x) = \text{sign}(x) \cdot \mathbb{1}_{\{|x| \geq \rho\}} = \begin{cases} 1, & \text{for } x \geq \rho, \\ 0, & \text{for } -\rho < x < \rho, \\ -1, & \text{for } x \leq -\rho. \end{cases}$$

Here, $\rho > 0$ is a threshold parameter. In this context, $L^{(t)}$ is computed using a randomly sampled online batch $S_t$ with batch size $|S_t| = B$:

$$L^{(t)} = \frac{1}{B} \sum_{(\mathbf{x},y)\in S_t} \ell[y \cdot f(\mathbf{W}^{(t)}, \mathbf{x})].$$

Consequently, the update rule for each $\mathbf{w}_r$ is given by:

$$\mathbf{w}_r^{(t+1)} = (1 - \lambda\eta)\mathbf{w}_r^{(t)} + \eta \cdot \widetilde{\mathrm{sign}}\bigg(\frac{1}{B} \sum_{(\mathbf{x},y)\in S_t} \sigma'(\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle) \cdot a_r y \mathbf{x}\bigg), \tag{3.2}$$

where $\widetilde{\mathrm{sign}}$ is applied on an element-wise basis.

**Remark 3.2.** *Sign SGD has been previously studied in Riedmiller and Braun (1993); Bernstein et al. (2018). Recently, it has become increasingly popular and has been utilized in adaptive optimizers for training large models (Chen et al., 2024; Liu et al., 2023). Previous studies (Balles and Hennig, 2018; Bernstein et al., 2018; Zou et al., 2021) have demonstrated that Sign SGD behaves similarly to Adam when using sufficiently small step sizes or small moving average parameters, $\beta_1$ and $\beta_2$. In our work, the choice of sign SGD over standard SGD stems primarily from our adoption of the polynomial activation function $\sigma(z) = z^k$. As later explained in Section 4, this specific activation function is pivotal in constructing a neural network that accurately tackles the $k$-parity problem. However, it introduces a trade-off: the gradient's dependency on the weights becomes polynomial rather than linear. Sign SGD addresses this issue by normalizing the gradient, ensuring that all neurons progress uniformly towards identifying the parity. Moreover, incorporating a threshold within the sign function plays a crucial role as it effectively nullifies the gradient of noisy coordinates. This, together with weight decay, aids in reducing noise, thereby enhancing the overall performance of the network.*

## 4 Main Results

In this section, we begin by demonstrating the capability of the two-layer fully connected neural network (3.1) to classify all examples correctly. Specifically, we construct the following *good* network:

$$f(\mathbf{W}^*, \mathbf{x}) = \sum_{r=1}^{2^k} a_r^* \sigma(\langle \mathbf{w}_r^*, \mathbf{x}\rangle), \tag{4.1}$$

where $\big\{\mathbf{w}^*_{r,[1:k]} \big| r \in [2^k]\big\} = \{\pm 1\}^k$, $a_r^* = \prod_{j=1}^k \mathrm{sign}(w^*_{r,j})$ and $\mathbf{w}^*_{r,[k+1:d]} = \mathbf{0}_{d-k}$ for any $r \in [2^k]$. Notably, leveraging the inherent symmetry within our neural network model, we can formally assert the following proposition: $yf(\mathbf{W}^*, \mathbf{x}) = y' f(\mathbf{W}^*, \mathbf{x}')$ for any $(y, \mathbf{x})$ and $(y', \mathbf{x}')$ generated from $\mathcal{D}_A$. The subsequent proposition demonstrates the precise value of the margin.

**Proposition 4.1.** *For any data point $(\mathbf{x}, y)$ generated from the distribution $\mathcal{D}_A$, it holds that*

$$yf(\mathbf{W}^*, \mathbf{x}) = k! \cdot 2^k. \tag{4.2}$$

*Proof.* Given a $(y, \mathbf{x}) \in \mathcal{D}_A$, we have that $y = \Pi_{i=1}^k x_i$. We divide the neurons into $(k+1)$ groups $\Omega_i, i \in \{0, \dots, k\}$. A neuron $r \in \Omega_i$ if and only if $\sum_{s=j}^k \mathbb{1}(\mathbf{w}_r^* = x_j) = i$. Then we have that

$$\begin{aligned}
yf(\mathbf{W}^*, \mathbf{x}) &= \sum_{r=1}^{2^k} (y \cdot a_r^*) \cdot \sigma(\langle \mathbf{w}_r^*, \mathbf{x}\rangle) \\
&= \sum_{i=0}^{k} \sum_{r\in\Omega_i} (y \cdot a_r^*) \cdot \sigma(\langle \mathbf{w}_r^*, \mathbf{x}\rangle) \\
&= \sum_{i=0}^{k} \sum_{r\in\Omega_i} \bigg(\prod_{j=1}^{k} \mathrm{sign}(x_j) \cdot \prod_{j=1}^{k} \mathrm{sign}(w^*_{r,j})\bigg) \cdot \sigma(\langle \mathbf{w}_r^*, \mathbf{x}\rangle) \\
&= \sum_{i=0}^{k} \binom{k}{i} (-1)^i \sigma(k - 2i)
\end{aligned}$$

$$= k! \cdot 2^k,$$

where the third equality is due to the fact that $y = \prod_{j=1}^k \operatorname{sign}(x_j)$ and $a_r^* = \prod_{j=1}^k \operatorname{sign}(w_{r,j}^*)$, the fourth equality is due to the definition of $\Omega_i$, the last equality holds because $\sigma$ is $k$-th order polynomial activation function and Lemma F.2. □

Therefore, we can conclude that for any $(\mathbf{x}, y)$, we have $yf(\mathbf{W}^*, \mathbf{x}) = k! \cdot 2^k > 0$. We will demonstrate in the next section that training using large batch size online SGD, as long as Condition 4.2 is met, will lead to the trained neural network $f(\mathbf{W}^{(T)}, \mathbf{x})$ approximating $(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})$ effectively after $T = O(k \log(d))$ iterations. Our main theorem is based on the following conditions on the training strategy.

**Condition 4.2.** *Suppose there exists a sufficiently large constant $C$, such that the following conditions hold:*

- *Neural network width $m$ satisfies $m \geq C \cdot 5^k \log(1/\delta)$.*
- *Dimension is sufficiently large: $d \geq C \log^2(2m/\epsilon)$.*
- *Online SGD batch size $B \geq C 2^k ((k-1)!)^{-2} d^{k-1} \log^{k-1}(16mdBT/\delta) \log^2(8mdT/\delta)$.*
- *Learning rate $\eta$ satisfies $\eta \leq C^{-1}$.*
- *Regularization parameter $\lambda$ is taken as $\lambda = 1$.*
- *The threshold $\rho$ for the modified sign function satisfies $\rho = 0.1k!$.*

In the $k$-parity problem, the label $y$ is determined by a set of $k$ bits. Consequently, the total count of distinct features is $2^k$, reflecting all possible combinations of these bits. The condition of $m$ is established to guarantee a roughly equal number of neurons within the *good* neuron class, each correctly aligned with distinct features. The condition of $d$ ensures that the problem is in a sufficiently high-dimensional setting. The condition of $m, d$ implies that $d \geq \Omega(\log^2 m) \geq \Omega(k^2)$, which is a mild requirement for the sparsity $k$. In comparison, Barak et al. (2022) requires $d \geq \Omega(k^4)$ for neural networks solving $k$-parity problem. By Stirling's approximation, the condition of $B$ can be simplified to $B \geq \widetilde{\Omega}\big(k(2e\log(16mdBT/\delta)d/k^2)^{k-1}\big)$. Therefore, the conditional batch size $B$ will exponentially increase as parity $k \leq O(\sqrt{d})$ goes up, which ensures that the stochastic gradient can sufficiently approximate the population gradient. Finally, the conditions of $\eta, \lambda$ ensure that gradient descent with weight decay can effectively learn the relevant features while simultaneously denoising the data. Finally, the threshold condition $\rho$ increases as sparsity $k$ increases to accommodate the increase of the population gradient. Based on these conditions, we give our main result on solving the parity problem in the following theorem.

**Theorem 4.3.** *Under Condition 4.2, we run online SGD for iteration $T = \Theta\big(k\eta^{-1}\lambda^{-1}\log d\big)$ iterations. Then with probability at least $1 - \delta$ we can find $\mathbf{W}^{(T)}$ such that*

$$\mathbb{P}\big(yf(\mathbf{W}^{(T)}, \mathbf{x}) \geq \gamma m\big) \geq 1 - \epsilon,$$

*where $\gamma = 0.25k!$ is a constant.*

Theorem 4.3 establishes that, under certain conditions, a neural network is capable of learning to solve the $k$-parity problem within $\Theta(k\eta^{-1}\lambda^{-1}\log d)$ iterations, achieving a population error of at most $\epsilon$. According to Condition 4.2, the total number of examples utilized amounts to $BT = \widetilde{O}(d^{k-1})$ given a polynomial logarithmic width requirement of $m = O(1)$ with respect to $d$.

**Remark 4.4.** *While Theorem 4.3 works for fixed second-layer training, we demonstrate that comparable results can be obtained when the second layer of the network is simultaneously trained with a lower learning rate. Detailed results and further elaboration of this aspect are provided in Appendix E. Our findings present a sample complexity of $\widetilde{O}(d^{k-1})$, aligning with Conjecture 2 posited by Abbe et al. (2023a), which suggests a sample complexity lower bound of $\widetilde{\Omega}(d^{k-1})$. Besides, our results for the uniform Boolean distribution match the complexity achieved by Abbe et al. (2023a) under the isotropic Gaussian scenario. Despite these similarities in outcomes, our methodology diverges significantly: we employ online sign SGD utilizing a large batch size of $\widetilde{O}(d^{k-1})$ and conduct training over merely $\widetilde{O}(1)$ iterations. In contrast, Abbe et al. (2023a) implement projected online SGD with a minimal batch size of $1$, extending training over $\widetilde{O}(d^{k-1})$ iterations. Abbe et al. (2023a) also requires a two-phase training process for the first and second layer weights, requiring them to be trained separately.*

## 5 Overview of Proof Technique

In this section, we discuss the main ideas used in the proof. Based on these main ideas, the proof of our main Theorem 4.3 will follow naturally. The complete proofs of all the results are given in the appendix. Section 5.1 serves as a warmup by examining population sign gradient descent. Here, three pivotal ideas crucial to the proof of stochastic sign gradient descent are introduced:

1. The impact of the initialization's positivity or negativity on the trajectory of neuron weights.
2. The divergence between feature coordinates and noise coordinates of different neurons.
3. How a trained neural network can effectively approximate the *good* neural network (4.1).

Moving on to Section 5.2, we delve into the analysis of sign SGD. Contrasting with population GD, the addition in SGD analysis involves accounting for the approximation error between the population gradient and the stochastic gradient. This consideration leads to the stipulation of the batch size $B$ outlined in Condition 4.2.

### 5.1 Warmup: Population Gradient Descent

For population gradient, we perform the following updates:

$$\mathbf{w}_r^{(t+1)} = (1-\eta\lambda)\cdot \mathbf{w}_r^{(t)} - \eta \cdot \widetilde{\mathrm{sign}}\big(\nabla_{\mathbf{w}_r} L_{\mathcal{D}}(\mathbf{W}^{(t)})\big),$$

where $L_{\mathcal{D}}(\mathbf{W}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(y, f(\mathbf{W},\mathbf{x}))] = 1 - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[yf(\mathbf{W},\mathbf{x})]$. Then, the following coordinate-wise population gradient update rules hold:

$$w_{r,j}^{(t+1)} = (1-\eta\lambda)w_{r,j}^{(t)} + \eta\cdot\widetilde{\mathrm{sign}}\left(k!a_r\frac{w_{r,1}^{(t)}w_{r,2}^{(t)}\cdots w_{r,k}^{(t)}}{w_{r,j}^{(t)}}\right), \qquad j\in[k], \tag{5.1}$$

$$w_{r,j}^{(t+1)} = (1-\eta\lambda)w_{r,j}^{(t)}, \qquad j\notin[k]. \tag{5.2}$$

In the preceding discussion of the update rule, we have identified that the noise coordinates ($j\notin[k]$) exhibit exponential decay, characterized by a decay constant of $1-\eta\lambda$. To further dissect the dynamics of this system, we turn our attention to the behavior of feature coordinates. We categorize neurons into two distinct types based on their initial alignment: a neuron is classified as a *good* neuron if $a_r = \prod_{j=1}^k \mathrm{sign}(w_{r,j}^{(0)})$, and conversely, as a *bad* neuron if $a_r = -\prod_{j=1}^k \mathrm{sign}(w_{r,j}^{(0)})$. This distinction is pivotal, as it divides the neuron population into two distinct classes: *good* and *bad*. Neurons in the *good* class are integral to the functionality of the final trained neural network, playing a significant role in its test accuracy. Conversely, neurons classified as *bad* tend to diminish in influence over the course of training, ultimately contributing minimally to the network's overall performance. For *good* neurons, the update rules for feature coordinates can be reformulated to

$$\begin{aligned}\mathrm{sign}(w_{r,j}^{(0)})w_{r,j}^{(t+1)} &= (1-\eta\lambda)\,\mathrm{sign}(w_{r,j}^{(0)})w_{r,j}^{(t)}\\ &\quad + \eta\cdot\frac{\mathrm{sign}(w_{r,1}^{(0)}w_{r,1}^{(t)})\,\mathrm{sign}(w_{r,2}^{(0)}w_{r,2}^{(t)})\cdots\mathrm{sign}(w_{r,k}^{(0)}w_{r,k}^{(t)})}{\mathrm{sign}(w_{r,j}^{(0)}w_{r,j}^{(t)})}.\end{aligned} \tag{5.3}$$

For neurons classified as *bad*, the update rules for feature coordinates can be rewritten as:

$$\begin{aligned}\mathrm{sign}(w_{r,j}^{(0)})w_{r,j}^{(t+1)} &= (1-\eta\lambda)\,\mathrm{sign}(w_{r,j}^{(0)})w_{r,j}^{(t)}\\ &\quad - \eta\cdot\frac{\mathrm{sign}(w_{r,1}^{(0)}w_{r,1}^{(t)})\,\mathrm{sign}(w_{r,2}^{(0)}w_{r,2}^{(t)})\cdots\mathrm{sign}(w_{r,k}^{(0)}w_{r,k}^{(t)})}{\mathrm{sign}(w_{r,j}^{(0)}w_{r,j}^{(t)})}.\end{aligned} \tag{5.4}$$

Comparing equations (5.3) and (5.4), it becomes apparent that the feature coordinates of *good* and *bad* neurons exhibit divergent behaviors. Consequently, the feature coordinates of *good* neurons will significantly outweigh those of *bad* neurons in the long term. With the regularization parameter $\lambda$ set as $1$ in Condition 4.2, we derive the following lemma illustrating the divergent trajectories of population gradient descent for both neuron types:

**Lemma 5.1.** *Under Condition 4.2, for good neurons* $r\in\Omega_{\mathrm{g}} := \{r\in[m] : a_r = \prod_{j=1}^k \mathrm{sign}(w_{r,j}^{(0)})\}$, *the feature coordinates will remain the same as initialization throughout the training:*

$$w_{r,j}^{(t)} = w_{r,j}^{(0)}, \qquad \forall j\in[k], t\geq 0.$$

*For bad neurons $r \in \Omega_{\mathrm{b}} := \{r \in [m] : a_r = -\prod_{j=1}^{k} \operatorname{sign}(w_{r,j}^{(0)})\}$, the feature coordinates will decay faster than noise coordiantes:*

$$0 < \operatorname{sign}(w_{r,j}^{(0)}) w_{r,j}^{(t+1)} \leq (1 - \eta\lambda) \operatorname{sign}(w_{r,j}^{(0)}) w_{r,j}^{(t)}, \qquad \forall j \in [k], t \geq 0.$$

According to (5.2) and Lemma 5.1, after training $T = \Theta(k\eta^{-1}\lambda^{-1}\log(d))$ iterations, *bad* neurons and noise coordinates in *good* neurons diminish to a magnitude of $\Theta(1/\mathrm{poly}(d))$, as shown in Lemma 5.2. In contrast, the feature coordinates of *good* neurons remain unchanged.

**Lemma 5.2.** *Under Condition 4.2, for $T \geq (k+1)\eta^{-1}\lambda^{-1}\log(d)$, it holds that*

$$|w_{r,j}^{(T)}| \leq d^{-(k+1)}, \qquad \forall r \in \Omega_{\mathrm{g}}, j \in [d] \setminus [k],$$
$$|w_{r,j}^{(T)}| \leq d^{-(k+1)}, \qquad \forall r \in \Omega_{\mathrm{b}}, j \in [d].$$

This leads to the following approximation for the trained neural network:

$$f(\mathbf{W}^{(T)}, \mathbf{x}) = \sum_{r=1}^{m} a_r \sigma(\langle \mathbf{w}_r^{(T)}, \mathbf{x} \rangle) \approx \sum_{r \in \Omega_{\mathrm{g}}} a_r \sigma(\langle \mathbf{w}_r^{(T)}, \mathbf{x} \rangle)$$
$$= \sum_{r \in \Omega_{\mathrm{g}}} \bigg( \prod_{j=1}^{k} \operatorname{sign}(w_{r,j}^{(0)}) \bigg) \sigma(\langle \mathbf{w}_r^{(T)}, \mathbf{x} \rangle) \approx \sum_{r \in \Omega_{\mathrm{g}}} \bigg( \prod_{j=1}^{k} \operatorname{sign}(w_{r,j}^{(0)}) \bigg) \sigma(\langle \mathbf{w}_{r,[1:k]}^{(T)}, \mathbf{x}_{[1:k]} \rangle).$$

Under Condition 4.2, the condition on $m$ ensures a balanced distribution of neurons across different initializations, approximately $m/2^{k+1}$, given $2^{k+1}$ kinds of possible initializations. This results in the trained neural network $f(\mathbf{W}^{(T)}, \mathbf{x})$ closely approximating $(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})$.

### 5.2 Stochastic Sign Gradient Descent

Transitioning from the trajectory trained by population gradient descent, this section delves into the dynamics under sign stochastic gradient descent (Sign SGD). We commence by presenting a lemma that estimates the approximation error between the population gradient and the stochastic gradient.

**Lemma 5.3.** *Under Condition 4.2, with probability at least $1 - \delta$ with respect to the online data generation, the stochastic gradient approximates the population gradient well:*

$$\left| \frac{\partial L_{\mathcal{D}}(\mathbf{W}^{(t)})}{\partial w_{r,j}} - \frac{\partial L^{(t)}}{\partial w_{r,j}} \right| \leq \epsilon_1 \cdot \|\mathbf{w}_r^{(t)}\|_2^{k-1}, \qquad \forall t \in [0, T], r \in [m], j \in [d],$$

*where $L^{(t)}$ is the loss of randomly sampled online batch $S_t$ and*

$$\epsilon_1 = \widetilde{O}(B^{-1/2} + d^{(k-3)/2} B^{-1}).$$

The choice of batch size $B = O(d^{k-1})$ in our algorithm is crucial for gradient concentration. According to Lemma 5.3, the gap between stochastic gradient and population gradient is bounded by $\epsilon_1 \cdot \|\mathbf{w}_r\|_2^{k-1}$. At initialization, the absolute value of the population gradient on the signal coordinate is approximately $\widetilde{O}(d^{-(k-1)/2}) \cdot \|\mathbf{w}_r\|_2^{k-1}$. To ensure the stochastic sign gradient matches the population sign gradient, the approximation error $\epsilon_1$ must be smaller than this value, which requires $\epsilon_1 = \widetilde{O}(d^{-(k-1)/2})$. Solving $B^{-1/2} + d^{(k-3)/2} B^{-1} = d^{-(k-1)/2}$ yields our sufficient batch size.

Building upon this approximation guarantee from Lemma 5.3, and considering the established order of $\rho, \epsilon_1$ and $\|\mathbf{w}_r^{(t)}\|_2$, we arrive at an important corollary.

**Corollary 5.4.** *Under Condition 4.2, given the same initialization, with probability at least $1 - \delta$, the stochastic sign gradient is the same as the population sign gradient:*

$$\widetilde{\operatorname{sign}}\bigg( \frac{\partial L_{\mathcal{D}}(\mathbf{W}^{(t)})}{\partial w_{r,j}} \bigg) = \widetilde{\operatorname{sign}}\bigg( \frac{\partial L^{(t)}}{\partial w_{r,j}} \bigg), \qquad \forall t \in [0, T], r \in [m], j \in [d].$$

This corollary suggests that, under identical initialization, the trajectory of a model trained using population gradient descent will, with high probability, align with the trajectory of a model trained using stochastic gradient descent.

## 6 Conclusion and Future Work

In our study, we have conducted a detailed analysis of the $k$-parity problem, investigating how sign Stochastic Gradient Descent (sign SGD) can effectively learn intricate features from binary datasets. Our findings reveal that sign SGD, when employed in two-layer fully-connected neural networks solving $k$-sparse parity problem, is capable of achieving a sample complexity $\widetilde{O}(d^{k-1})$. Remarkably, this result matches the theoretical expectations set by the Statistical Query (SQ) model, underscoring the efficiency and adaptability of sign SGD.

Looking ahead, an intriguing direction for future research is to explore the possibility of learning $k$-parity using SGD with even smaller queries that surpass the SQ lower bond, and understand whether more standard neural network architectures allow such improvement. This potential advancement could pave the way for developing more efficient algorithms capable of tackling complex problems with weaker data requirements. Another promising direction is to extend our results to non-isotropic data settings (Nitanda et al., 2024), where sign gradient descent with momentum could be effective in handling the transformed feature space.

## Acknowledgements

We thank the anonymous reviewers for their helpful comments. YK, ZC, and QG are supported in part by the National Science Foundation CAREER Award 1906169, IIS-2008981, and the Sloan Research Fellowship. SK acknowledges: this work has been made possible in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence; support from the Office of Naval Research under award N00014-22-1-2377, and the National Science Foundation Grant under award #IIS 2229881.

## References

ABBE, E., ADSERA, E. B. and MISIAKIEWICZ, T. (2022). 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*. PMLR.

ABBE, E., ADSERA, E. B. and MISIAKIEWICZ, T. (2023a). Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *The Thirty Sixth Annual Conference on Learning Theory*. PMLR.

ABBE, E. and BOIX-ADSERA, E. (2022). On the non-universality of deep learning: quantifying the cost of symmetry. *Advances in Neural Information Processing Systems* **35** 17188–17201.

ABBE, E., CORNACCHIA, E. and LOTFI, A. (2023b). Provable advantage of curriculum learning on parity targets with mixed inputs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

BALLES, L. and HENNIG, P. (2018). Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*. PMLR.

BARAK, B., EDELMAN, B. L., GOEL, S., KAKADE, S., MALACH, E. and ZHANG, C. (2022). Hidden progress in deep learning: Sgd learns parities near the computational limit. *arXiv preprint arXiv:2207.08799* .

BERNSTEIN, J., WANG, Y.-X., AZIZZADENESHELI, K. and ANANDKUMAR, A. (2018). signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*. PMLR.

BLUM, A. (2005). On-line algorithms in machine learning. *Online algorithms: the state of the art* 306–325.

BSHOUTY, N. H. and FELDMAN, V. (2002). On using extended statistical queries to avoid membership queries. *Journal of Machine Learning Research* **2** 359–395.

CHEN, X., LIANG, C., HUANG, D., REAL, E., WANG, K., PHAM, H., DONG, X., LUONG, T., HSIEH, C.-J., LU, Y. ET AL. (2024). Symbolic discovery of optimization algorithms. *Advances in Neural Information Processing Systems* **36**.

Chizat, L. and Bach, F. (2020). Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*. PMLR.

Daniely, A. and Malach, E. (2020). Learning parities with neural networks. *Advances in Neural Information Processing Systems* **33** 20356–20365.

Downey, R. G., Fellows, M. R., Vardy, A. and Whittle, G. (1999). The parametrized complexity of some fundamental problems in coding theory. *SIAM Journal on Computing* **29** 545–570.

Dumer, I., Micciancio, D. and Sudan, M. (2003). Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory* **49** 22–37.

Dutta, C., Kanoria, Y., Manjunath, D. and Radhakrishnan, J. (2008). A tight lower bound for parity in noisy communication networks. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete Algorithms*.

Edelman, B., Goel, S., Kakade, S., Malach, E. and Zhang, C. (2024). Pareto frontiers in deep feature learning: Data, compute, width, and luck. *Advances in Neural Information Processing Systems* **36**.

Farhi, E., Goldstone, J., Gutmann, S. and Sipser, M. (1998). Limit on the speed of quantum computation in determining parity. *Physical Review Letters* **81** 5442.

Frei, S., Chatterji, N. S. and Bartlett, P. L. (2022). Random feature amplification: Feature learning and generalization in neural networks. *arXiv preprint arXiv:2202.07626* .

Glasgow, M. (2023). 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* .

Hu, K., Ren, Z., Siska, D. and Szpruch, L. (2019). Mean-field langevin dynamics and energy landscape of neural networks. *arXiv preprint arXiv:1905.07769* .

Jacot, A., Gabriel, F. and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*.

Ji, Z. and Telgarsky, M. (2020). Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. In *International Conference on Learning Representations*.

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

Klivans, A. R., Servedio, R. A. and Ron, D. (2006). Toward attribute efficient learning of decision lists and parities. *Journal of Machine Learning Research* **7**.

Liu, H., Li, Z., Hall, D., Liang, P. and Ma, T. (2023). Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342* .

Mei, S., Montanari, A. and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences* **115** E7665–E7671.

Meng, X., Zou, D. and Cao, Y. (2023). Benign overfitting in two-layer relu convolutional neural networks for xor data.

Milne-Thomson, L. M. (2000). *The calculus of finite differences*. American Mathematical Soc.

Nitanda, A., Oko, K., Suzuki, T. and Wu, D. (2024). Improved statistical and computational complexity of the mean-field langevin dynamics under structured data. In *The Twelfth International Conference on Learning Representations*.

Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *IEEE international conference on neural networks*. IEEE.

SUZUKI, T., WU, D., OKO, K. and NITANDA, A. (2023). Feature learning via mean-field langevin dynamics: classifying sparse parities and beyond. In *Thirty-seventh Conference on Neural Information Processing Systems*.

TELGARSKY, M. (2022). Feature selection with gradient descent on two-layer networks in low-rotation regimes. *arXiv preprint arXiv:2208.02789* .

VARDY, A. (1997). The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory* **43** 1757–1766.

WEI, C., LEE, J. D., LIU, Q. and MA, T. (2019). Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel. *Advances in Neural Information Processing Systems* .

XU, Z., WANG, Y., FREI, S., VARDI, G. and HU, W. (2023). Benign overfitting and grokking in relu networks for xor cluster data.

ZOU, D., CAO, Y., LI, Y. and GU, Q. (2021). Understanding the generalization of adam in learning neural networks with proper regularization. *arXiv preprint arXiv:2108.11371* .

## Limitations

While our study provides valuable insights into the effectiveness of SGD in learning the $k$-parity problem, there are some limitations:

- Our study focuses on sign gradient descent (Sign SGD). This approach normalizes the gradient, ensures uniform neuron updates toward identifying parity, and effectively nullifies noisy coordinate gradients. However, it's worth noting that data may be presented in non-standard or unknown coordinate systems, which could limit Sign SGD's effectiveness. To address this limitation, future work could explore alternatives such as normalized gradient descent with an adaptive learning rate or incorporating momentum into Sign SGD.
- Our analysis is primarily based on polynomial activation functions. While effective for the standard k-parity problem, extending our approach to other activation functions like sigmoid or ReLU presents challenges. This extension could potentially be achieved through polynomial function approximation. However, the main challenge lies in identifying an appropriate functional decomposition and accurately characterizing the approximation error during training.

By acknowledging these limitations, we aim to provide a transparent assessment of our work's scope and potential areas for future exploration.

## A Experiments

In this section, we present a series of experiments designed to empirically validate the theoretical results established in our main theorem. The primary objectives of these experiments are to (1) assess the test accuracy of the trained neural network, thereby corroborating the theorem's results, and (2) verify the key lemmas concerning the behavior of *good* and *bad* neurons. Specifically, we aim to demonstrate that for *good* neurons, the feature coordinates remain largely unchanged from initialization while the noise coordinates decay exponentially. Conversely, for *bad* neurons, we expect both feature and noise coordinates to exhibit exponential decay.

**Model.** We generated synthetic $k$-parity data based on Definition 3.1. Each data point $(\mathbf{x}, y)$ with $\mathbf{x} \in \mathbb{R}^d$ and $y \in \{\pm 1\}$ is produced from distribution $\mathcal{D}_A$, where $A$ specifically is taken as $[k]$. We utilized two-layer fully-connected neural networks with $m$ number of neurons. The network employs a polynomial activation function $\sigma(z) = z^k$. The first layer weights for the $r$-th neuron, $\mathbf{w}_r^{(0)} \in \mathbb{R}^d$, were initialized following a binary scheme, where $\mathbf{w}_r^{(0)} \sim \text{Unif}(\{\pm 1\}^d)$. The second layer weights for the $r$-th neuron, $a_r$, is randomly initialized as $1$ or $-1$ with equal probability.

**Computation Resources** The experiments are conducted on an A6000 server. As these are synthetic experiments, the requirement for computational resources is minimal.

**Training.** Our model was trained to minimize the empirical correlation loss function, incorporating $L_2$ regularization with a regularization parameter set at $\lambda = 1$. The training process utilized online stochastic sign gradient descent (Sign SGD) with a fixed step size $\eta$ and a predetermined batch size $B$. The principal metric for assessment was test accuracy. Our experiments were conducted considering parity $k \in \{2, 3, 4\}$.

For $k = 2$, the model configuration included a data dimension of $d = 8$, a hidden layer width of $m = 12$, a total of $T = 25$ epochs, a learning rate $\eta = 0.1$, an online batch size of $B = 64$, and a threshold for $\widetilde{\text{sign}}$ set at $\rho = 0.3$. In the case of $k = 3$, we employed a data dimension of $d = 16$, increased the hidden layer width to $m = 48$, extended the training to $T = 50$ epochs, adjusted the learning rate to $\eta = 0.05$, used an online batch size of $B = 256$, and set the threshold for $\widetilde{\text{sign}}$ at $\rho = 1$. For $k = 4$, the model was further scaled up with a data dimension of $d = 20$, a hidden layer width of $m = 128$, a training epoch of $T = 100$, a smaller learning rate of $\eta = 0.02$, an online batch size of $B = 2048$, and a threshold for $\widetilde{\text{sign}}$ established at $\rho = 3$.

**Experimental Results.** The evaluation of test accuracy across various configurations is presented in Table 3. Using neural networks with merely $2^{\Theta(k)}$ neurons, we observed high test accuracy for $k$-parity problem with $k \in \{2, 3, 4\}$, confirming the results of our main theorem (Theorem 4.3).

These empirical results validate the efficacy of our studied model architecture (3.1) and training methodology in tackling the $k$-sparse parity problem.

To further validate our theoretical findings, we examined the change of feature and noise coordinates for the first neuron $\mathbf{w}_1^{(t)}$ over multiple iterations, focusing specifically on the setting $k \in \{2, 3, 4\}$. Figures 2, 3, 4, 5, 6, and 7 visually represent these trajectories. Our empirical findings reveal a consistent pattern: the feature coordinates $(w_{1,1}^{(t)}, \ldots, w_{1,k}^{(t)})$ of the neurons identified as *good* (with initialization satisfying $a_r = \prod_{j=1}^{k} w_{r,j}^{(0)}$) exhibit relative stability, while their noise coordinates $(w_{1,k+1}^{(t)}, \ldots)$ show a decreasing trend over time. In contrast, the trajectories for neurons classified as *bad* (with initialization satisfying $a_r = -\prod_{j=1}^{k} w_{r,j}^{(0)}$) indicate a general reduction in all coordinate values.

These empirical observations support our theoretical analyses, as outlined in Lemma 5.1 and Lemma 5.2, showcasing the consistency between the theoretical foundations of our model and its practical performance. The disparities between good and bad neurons, as evidenced by their feature and noise coordinate behaviors, underscore the nuanced dynamics inherent in the learning process of $k$-parity problems.

| $\boldsymbol{k}$ | $2$ | $3$ | $4$ |
|---|---|---|---|
| **Test Accuracy (%)** | $99.69\% \pm 0.29\%$ | $97.75\% \pm 1.37\%$ | $96.89\% \pm 0.44\%$ |

Table 3: Test accuracy for solving $k$-sparse parity problem with $k \in \{2, 3, 4\}$, averaged over 10 runs.

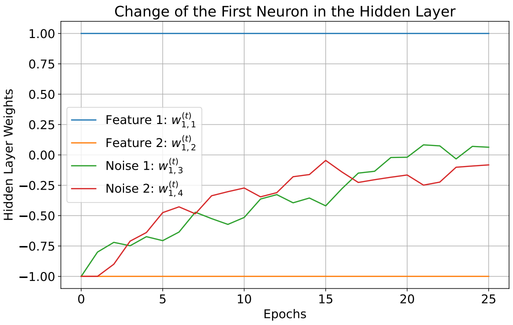

Figure 2: Illustration of a 2-parity *good* neuron with initial weights $w_{1,1}^{(0)} = 1$, $w_{1,2}^{(0)} = -1$, and $a_1 = -1$.

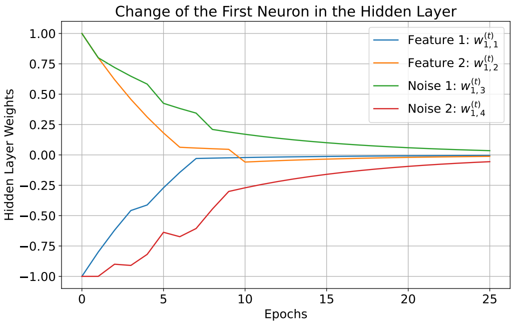

Figure 3: Illustration of a 2-parity *bad* neuron with initial weights $w_{1,1}^{(0)} = -1$, $w_{1,2}^{(0)} = 1$, and $a_1 = 1$.

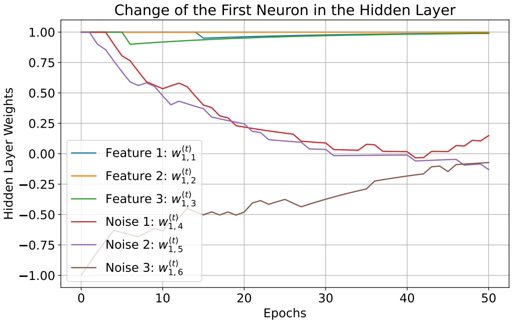

Figure 4: Illustration of a 3-parity *good* neuron with initial weights $w_{1,1}^{(0)} = 1$, $w_{1,2}^{(0)} = 1$, $w_{1,3}^{(0)} = 1$, and $a_1 = 1$.

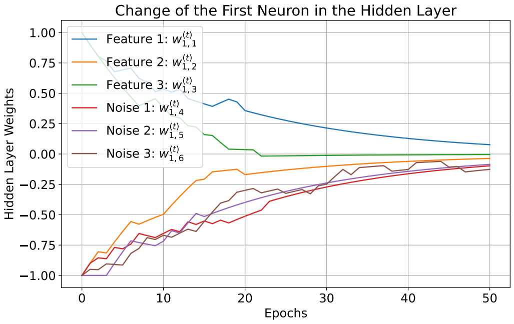

Figure 5: Illustration of a 3-parity *bad* neuron with initial weights $w_{1,1}^{(0)} = 1$, $w_{1,2}^{(0)} = -1$, $w_{1,3}^{(0)} = 1$, and $a_1 = 1$.

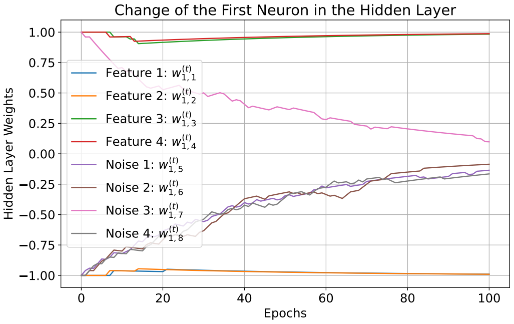

Figure 6: Illustration of a $4$-parity *good* neuron with initial weights $w_{1,1}^{(0)} = -1$, $w_{1,2}^{(0)} = -1$, $w_{1,3}^{(0)} = 1$, $w_{1,4}^{(0)} = 1$, and $a_1 = 1$.

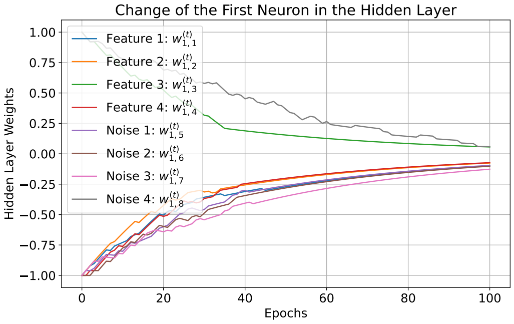

Figure 7: Illustration of a $4$-parity *bad* neuron with initial weights $w_{1,1}^{(0)} = -1$, $w_{1,2}^{(0)} = -1$, $w_{1,3}^{(0)} = 1$, $w_{1,4}^{(0)} = -1$, and $a_1 = -1$.

## B Preliminary Lemmas

During the initialization phase of a neural network, neurons can be categorized into $2^k$ distinct groups. This classification is based on whether each feature coordinate is positive or negative. We define these groups as follows:

$$\Omega_{b_1 b_2 \cdots b_k} = \{r \in [m] \mid \operatorname{sign}(w_{r,j}^{(0)}) = b_j, \forall j \in [k]\},$$

where $b_1, b_2, \cdots, b_k \in \{\pm 1\}$. To illustrate with specific examples, consider the following special cases:

$$\Omega_{11\cdots 1} = \{r \in [m] \,|\, w_{r,j}^{(0)} > 0, \forall j \in [k]\},$$
$$\Omega_{-1-1\cdots -1} = \{r \in [m] \,|\, w_{r,j}^{(0)} < 0, \forall j \in [k]\}.$$

In these cases, $\Omega_{11\dots 1}$ represents the group of neurons where all initial weights are positive across the $k$ features, while $\Omega_{-1-1\dots -1}$ consists of neurons with all initial weights being negative. Within each group of neurons, we can further subdivide them into two subgroups based on the value of $a_r$. Let's denote

$$\Omega_{\mathrm{g}} = \left\{ r \in [m] \,\middle|\, a_r = \prod_{j=1}^{k} \operatorname{sign}(w_{r,j}^{(0)}) \right\}, \Omega_{\mathrm{b}} = \left\{ r \in [m] \,\middle|\, a_r = -\prod_{j=1}^{k} \operatorname{sign}(w_{r,j}^{(0)}) \right\},$$

where $\Omega_g$ denotes the *good* neuron set and $\Omega_b$ denotes the *bad* neuron set. We will later demonstrate in the proof and experiments that neurons in $\Omega_{\mathrm{g}}$ and neurons in $\Omega_{\mathrm{b}}$ exhibit distinct behaviors during the training process. Specifically, for neurons in $\Omega_{\mathrm{g}}$, the feature coordinates will remain largely unchanged from their initial values throughout training, while the noise coordinates will decrease to a lower order compared to the feature coordinates. On the other hand, for neurons in $\Omega_{\mathrm{b}}$, both feature coordinates and noise coordinates will decrease to a lower order compared to their initial values.

In order to establish the test error result, it is essential to impose a condition on the initialization. Specifically, the number of neurons for each type of initialization should be approximately equal.

**Lemma B.1.** *With probability at least $1 - \delta$ for the randomness in the neural network's initialization, the sizes of the sets $\Omega_{\mathrm{g}}$ and $\Omega_{\mathrm{b}}$ are bounded as follows:*

$$|\Omega_{\mathrm{g}}|, |\Omega_{\mathrm{b}}| \in [(1 - \alpha)m/2, (1 + \alpha)m/2].$$

*Besides, the intersections of $\Omega_{b_1 b_2 \cdots b_k}$ with both $\Omega_{\mathrm{g}}$ and $\Omega_{\mathrm{b}}$ are also bounded within a specified range:*

$$|\Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{g}}|, |\Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{b}}| \in [(1 - \alpha)m/2^{k+1}, (1 + \alpha)m/2^{k+1}],$$

*where*

$$\alpha = \sqrt{\frac{3 \cdot 2^{k+1} \log(2^{k+2}/\delta)}{m}}.$$

*Proof.* Let $X_r = \mathbb{1}\left[w_{r,1}^{(0)} > 0, \cdots, w_{r,k}^{(0)} > 0, a_r = 1\right]$. Then, by Chernoff bound, we have

$$\mathbb{P}\left(\left|\sum_{r=1}^{m} X_r - \frac{m}{2^{k+1}}\right| \geq \alpha \cdot \frac{m}{2^{k+1}}\right) \leq 2\exp\Big(-\frac{\alpha^2 m}{3 \cdot 2^{k+1}}\Big).$$

Then, with probability at least $1-\delta$, we have

$$(1-\alpha) \cdot \frac{m}{2^{k+1}} \leq |\Omega_{11\cdots 1} \cap \Omega_{\mathrm{g}}| \leq (1+\alpha) \cdot \frac{m}{2^{k+1}},$$

where

$$\alpha = \sqrt{\frac{3 \cdot 2^{k+1}\log(2/\delta)}{m}}.$$

By applying union bound to all $2^{k+1}$ kinds of initialization, with probability at least $1-\delta$ it holds that for any $\Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{g}}$ and $\Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{b}}\big((b_1, \cdots, b_k) \in \{\pm 1\}^k\big)$

$$\begin{aligned}(1-\alpha) \cdot \frac{m}{2^{k+1}} &\leq |\Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{g}}| \leq (1+\alpha) \cdot \frac{m}{2^{k+1}}, \\ (1-\alpha) \cdot \frac{m}{2^{k+1}} &\leq |\Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{b}}| \leq (1+\alpha) \cdot \frac{m}{2^{k+1}},\end{aligned}$$

where

$$\alpha = \sqrt{\frac{3 \cdot 2^{k+1}\log(2^{k+2}/\delta)}{m}}.$$

□

# C Warmup: Population Sign GD

In this section, we train a neural network with gradient descent on the distribution $\mathcal{D}$. Here we use correlation loss function $\ell(y, \widehat{y}) = 1 - y\widehat{y}$. Then, the loss on this attribution is $L_{\mathcal{D}}(\mathbf{W}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(y, f(\mathbf{W}, \mathbf{x}))]$, and we perform the following updates:

$$\mathbf{w}_r^{(t+1)} = (1-\eta\lambda)\mathbf{w}_r^{(t)} - \eta \cdot \widetilde{\mathrm{sign}}\big(\nabla_{\mathbf{w}_r} L_{\mathcal{D}}(\mathbf{W}^{(t)})\big).$$

We assume the network is initialized with a symmetric initialization: for every $r \in [m]$, initialize $\mathbf{w}_r^{(0)} \sim \mathrm{Unif}(\{-1,1\}^d)$ and initialize $a_r \sim \mathrm{Unif}(\{-1,1\})$.

**Lemma C.1.** *The following coordinate-wise population sign gradient update rules hold:*

$$\begin{aligned} w_{r,j}^{(t+1)} &= (1-\eta\lambda)w_{r,j}^{(t)} + \eta \cdot \widetilde{\mathrm{sign}}\left(k! a_r \cdot \frac{w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}}{w_{r,j}^{(t)}}\right), && j \in [k], \\ w_{r,j}^{(t+1)} &= (1-\eta\lambda)w_{r,j}^{(t)}, && j \notin [k]. \end{aligned}$$

*Proof.* For population gradient descent, we have

$$\begin{aligned}\mathbf{w}_r^{(t+1)} &= (1-\eta\lambda) \cdot \mathbf{w}_r^{(t)} - \eta \cdot \widetilde{\mathrm{sign}}\big(\nabla_{\mathbf{w}_r} L_{\mathcal{D}}(\mathbf{W})\big) \\ &= (1-\eta\lambda) \cdot \mathbf{w}_r^{(t)} - \eta \cdot \widetilde{\mathrm{sign}}\big(a_r \cdot \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\nabla_{\mathbf{w}_r}\sigma(\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle)]\big) \\ &= (1-\eta\lambda) \cdot \mathbf{w}_r^{(t)} - \eta \cdot \widetilde{\mathrm{sign}}\big(k a_r \cdot \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[y\mathbf{x}(\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle)^{k-1}]\big).\end{aligned}$$

Notice that for $j \notin [k]$, we have

$$\begin{aligned}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[y x_j(\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle)^{k-1}] &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[x_1 x_2 \cdots x_k x_j\left(\sum_{j_1,\cdots,j_{k-1}} w_{r,j_1}^{(t)} \cdots w_{r,j_{k-1}}^{(t)} x_{j_1} \cdots x_{j_{k-1}}\right)\right] \\ &= \sum_{j_1,\cdots,j_{k-1}} w_{r,j_1}^{(t)} \cdots w_{r,j_{k-1}}^{(t)} \cdot \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\big[x_1 x_2 \cdots x_k x_j x_{j_1} \cdots x_{j_{k-1}}\big] \\ &= 0,\end{aligned}$$

where the last equality is because $\{j_1, \cdots, j_{k-1}\} \subsetneq \{1, \cdots, k, j\}$. This implies that

$$w_{r,j}^{(t+1)} = (1 - \eta\lambda) \cdot w_{r,j}^{(t)}.$$

For $j \in [k]$, we have

$$\begin{aligned}
\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[y x_j (\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle)^{k-1}] &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[x_1 x_2 \cdots x_k x_j \left( \sum_{j_1, \cdots, j_{k-1}} w_{r,j_1}^{(t)} \cdots w_{r,j_{k-1}}^{(t)} x_{j_1} \cdots x_{j_{k-1}} \right)\right] \\
&= \sum_{j_1, \cdots, j_{k-1}} w_{r,j_1}^{(t)} \cdots w_{r,j_{k-1}}^{(t)} \cdot \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[x_1 x_2 \cdots x_k x_j x_{j_1} \cdots x_{j_{k-1}}] \\
&= (k-1)! \cdot \frac{w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}}{w_{r,j}^{(t)}},
\end{aligned}$$

where the last inequality is because $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[x_1 x_2 \cdots x_k x_j x_{j_1} \cdots x_{j_{k-1}}] \neq 0$ if and only if $\{j, j_1, \cdots, j_{k-1}\} = \{1, 2, \cdots, k\}$. It follows that

$$w_{r,j}^{(t+1)} = (1 - \eta\lambda) \cdot w_{r,j}^{(t)} + \eta \cdot \widetilde{\mathrm{sign}}\left( k! a_r \cdot \frac{w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}}{w_{r,j}^{(t)}} \right).$$

□

Given the update rules in Lemma C.1, we observe distinct behaviors for good neurons ($\Omega_{\mathrm{g}}$) and bad neurons ($\Omega_{\mathrm{b}}$). The following corollary illustrates these differences:

**Corollary C.2.** *For any neuron $r \in \Omega_{b_1 \cdots b_k} \cap \Omega_{\mathrm{g}}$, the update rule for feature coordinates $(j \in [k])$ is given by:*

$$\begin{aligned}
b_j w_{r,j}^{(t+1)} &= (1 - \eta\lambda) b_j w_{r,j}^{(t)} \\
&\quad + \eta \cdot \frac{\mathrm{sign}(b_1 w_{r,1}^{(t)}) \,\mathrm{sign}(b_2 w_{r,2}^{(t)}) \cdots \mathrm{sign}(b_k w_{r,k}^{(t)})}{\mathrm{sign}(b_j w_{r,j}^{(t)})} \cdot \mathbb{1}\left[ \frac{|w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}|}{|w_{r,j}^{(t)}|} \geq \frac{\rho}{k!} \right].
\end{aligned}$$

*For any neuron $r \in \Omega_{b_1 \cdots b_k} \cap \Omega_{\mathrm{b}}$, the update rule for feature coordinates $(j \in [k])$ is given by:*

$$\begin{aligned}
b_j w_{r,j}^{(t+1)} &= (1 - \eta\lambda) b_j w_{r,j}^{(t)} \\
&\quad - \eta \cdot \frac{\mathrm{sign}(b_1 w_{r,1}^{(t)}) \,\mathrm{sign}(b_2 w_{r,2}^{(t)}) \cdots \mathrm{sign}(b_k w_{r,k}^{(t)})}{\mathrm{sign}(b_j w_{r,j}^{(t)})} \cdot \mathbb{1}\left[ \frac{|w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}|}{|w_{r,j}^{(t)}|} \geq \frac{\rho}{k!} \right].
\end{aligned}$$

By setting the regularization parameter $\lambda$ to 1, we observe a noteworthy property of the weight of feature coordinates in good neurons. This is formalized in the following lemma:

**Lemma C.3.** *Assume $\lambda = 1$ and $\rho < k!$. For a neuron $r \in \Omega_{b_1 \cdots b_k} \cap \Omega_{\mathrm{g}}$, the weight associated with any feature coordinate remains constant across all time steps $t \geq 0$. Specifically, it holds that:*

$$b_j w_{r,j}^{(t)} = 1, \quad \forall j \in [k],\, t \geq 0.$$

*Proof.* We prove this by using induction. The result is obvious at $t = 0$. Suppose the result holds when $t = \widetilde{t}$. Then, according to Corollary C.2, we have

$$\begin{aligned}
b_j w_{r,j}^{(\widetilde{t}+1)} &= (1 - \eta\lambda) b_j w_{r,j}^{(\widetilde{t})} \\
&\quad + \eta \cdot \frac{\mathrm{sign}(b_1 w_{r,1}^{(\widetilde{t})}) \,\mathrm{sign}(b_2 w_{r,2}^{(\widetilde{t})}) \cdots \mathrm{sign}(b_k w_{r,k}^{(\widetilde{t})})}{\mathrm{sign}(b_j w_{r,j}^{(\widetilde{t})})} \cdot \mathbb{1}\left[ \frac{|w_{r,1}^{(\widetilde{t})} w_{r,2}^{(\widetilde{t})} \cdots w_{r,k}^{(\widetilde{t})}|}{|w_{r,j}^{(t)}|} \geq \frac{\rho}{k!} \right] \\
&= (1 - \eta\lambda) + \eta \cdot \mathbb{1}[k! \geq \rho] \\
&= 1.
\end{aligned}$$

□

The following lemma demonstrates that for *bad* neurons, the weights of feature coordinates tend to shrink over time. The dynamics of this shrinking are characterized as follows:

**Lemma C.4.** *Assume $\lambda = 1$ and $\eta/(1-\eta\lambda) < (\rho/k!)^{\frac{1}{k-1}}$. For a neuron $r \in \Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{b}}$, the weights of any two feature coordinates $j$ and $j'$ (where $j, j' \in [k]$) are equal at any time step, that is, $b_j w_{r,j}^{(t)} = b_{j'} w_{r,j'}^{(t)}$. Furthermore, the weight of any feature coordinate $j \in [k]$ evolves according to the following inequality for any $t \geq 0$:*

$$0 < b_j w_{r,j}^{(t+1)} \leq (1-\eta\lambda) b_j w_{r,j}^{(t)}.$$

*Proof.* We prove this by using induction. We prove the following three hypotheses:

$$b_j w_{r,j}^{(t)} = b_{j'} w_{r,j'}^{(t)}, \qquad \forall j, j' \in [k]. \tag{H$_1$}$$

$$b_j w_{r,j}^{(t)} > 0, \qquad \forall j \in [k]. \tag{H$_2$}$$

$$b_j w_{r,j}^{(t+1)} \leq (1-\eta\lambda) b_j w_{r,j}^{(t)}, \qquad \forall j \in [k]. \tag{H$_3$}$$

We will show that $\mathrm{H}_1(0)$ and $\mathrm{H}_2(0)$ are true and that for any $t \geq 0$ we have

- $\mathrm{H}_2(t) \Longrightarrow \mathrm{H}_3(t)$,
- $\mathrm{H}_1(t), \mathrm{H}_2(t) \Longrightarrow \mathrm{H}_1(t+1)$,
- $\mathrm{H}_1(t), \mathrm{H}_2(t) \Longrightarrow \mathrm{H}_2(t+1)$.

$\mathrm{H}_1(0)$ and $\mathrm{H}_2(0)$ are obviously true since $b_j w_{r,j}^{(0)} = 1$ for any $j \in [k]$. Next, we prove that $\mathrm{H}_2(t) \Longrightarrow \mathrm{H}_3(t)$ and $\mathrm{H}_1(t), \mathrm{H}_2(t) \Longrightarrow \mathrm{H}_1(t+1)$. According to Corollary C.2, we have

$$\begin{aligned} b_j w_{r,j}^{(t+1)} &= (1-\eta\lambda) b_j w_{r,j}^{(t)} \\ &\quad - \eta \cdot \frac{\operatorname{sign}(b_1 w_{r,1}^{(t)}) \operatorname{sign}(b_2 w_{r,2}^{(t)}) \cdots \operatorname{sign}(b_k w_{r,k}^{(t)})}{\operatorname{sign}(b_j w_{r,j}^{(t)})} \cdot \mathbb{1}\left[\frac{|w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}|}{|w_{r,j}^{(t)}|} \geq \frac{\rho}{k!}\right] \\ &= (1-\eta\lambda) b_j w_{r,j}^{(t)} - \eta \cdot \mathbb{1}\left[\frac{|w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}|}{|w_{r,j}^{(t)}|} \geq \frac{\rho}{k!}\right] \qquad \text{(C.1)} \\ &\leq (1-\eta\lambda) b_j w_{r,j}^{(t)}, \end{aligned}$$

where the second equality is by $\mathrm{H}_2(t)$. This verifies $\mathrm{H}_3(t)$. Besides, given $\mathrm{H}_1(t)$, we have

$$\frac{|w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}|}{|w_{r,j}^{(t)}|} = \frac{|w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}|}{|w_{r,j'}^{(t)}|}, \qquad \forall j, j' \in [k].$$

Plugging this into (C.1), we can get

$$b_j w_{r,j}^{(t+1)} = b_{j'} w_{r,j'}^{(t+1)}, \qquad \forall j, j' \in [k],$$

which verifies $\mathrm{H}_1(t+1)$. Finally, we prove that $\mathrm{H}_1(t), \mathrm{H}_2(t) \Longrightarrow \mathrm{H}_2(t+1)$. By (C.1) and $\mathrm{H}_1(t)$, we have

$$b_j w_{r,j}^{(t+1)} = (1-\eta\lambda) b_j w_{r,j}^{(t)} - \eta \cdot \mathbb{1}\left[|w_{r,j}^{(t)}|^{k-1} \geq \frac{\rho}{k!}\right].$$

If $|w_{r,j}^{(t)}| < (\rho/k!)^{\frac{1}{k-1}}$, we can get

$$b_j w_{r,j}^{(t+1)} = (1-\eta\lambda) b_j w_{r,j}^{(t)} > 0.$$

If $|w_{r,j}^{(t)}| \geq (\rho/k!)^{\frac{1}{k-1}}$, given that

$$\frac{\eta}{1-\eta\lambda} < (\rho/k!)^{\frac{1}{k-1}},$$

we can get

$$b_j w_{r,j}^{(t+1)} = (1-\eta\lambda) b_j w_{r,j}^{(t)} - \eta > 0.$$

□

Given Lemma C.3 and Lemma C.4, we can directly get the change of neurons of all kinds of initialization.

**Corollary C.5.** *Assume $\lambda = 1$, $\rho < k!$ and $\eta/(1-\eta\lambda) < (\rho/k!)^{\frac{1}{k-1}}$. For any fixed $(b_1, b_2, \cdots, b_k) \in \{\pm 1\}^k$, considering $r \in \Omega_{b_1 b_2 \cdots b_k}$, we have the following statements hold.*

- *For any neuron $r \in \Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{g}}$, the weights of feature coordinates remain the same as initialization: $w_{r,j}^{(t)} = b_j$ for any $t \geq 0$ and $j \in [k]$.*
- *For any neuron $r \in \Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{b}}$, the weights of feature coordinates will shrink simultaneously over time: $b_j w_{r,j}^{(t)} = b_{j'} w_{r,j'}^{(t)}$ for any $t \geq 0$ and $j, j' \in [k]$ and*
$$0 < b_j w_{r,j}^{(t+1)} \leq (1 - \eta\lambda) \cdot b_j w_{r,j}^{(t)},$$
*for any $t \geq 0$ and $j \in [k]$.*

Building on Corollary C.5, we can now characterize the trajectory of all neurons over time. Specifically, after a time period $T = \Theta(k\eta^{-1}\lambda^{-1}\log(d))$, the following observations about neuron weights hold:

**Lemma C.6.** *Assume $\lambda = 1$, $\rho < k!$ and $\eta/(1-\eta\lambda) < (\rho/k!)^{\frac{1}{k-1}}$. For $T \geq (k+1)\eta^{-1}\lambda^{-1}\log d$, it holds that*
$$\begin{aligned} w_{r,j}^{(T)} &= b_j, && \forall r \in \Omega_{b_1 b_1 \cdots b_k} \cap \Omega_{\mathrm{g}}, j \in [k], \\ |w_{r,j}^{(T)}| &\leq d^{-(k+1)}, && \forall r \in \Omega_{\mathrm{g}}, j \in [d] \setminus [k], \\ |w_{r,j}^{(T)}| &\leq d^{-(k+1)}, && \forall r \in \Omega_{\mathrm{b}}, j \in [d]. \end{aligned}$$

*Proof.* The first equality is obvious according to Corollary C.5. We only need to prove the inequalities. According to Lemma C.1, we have for any $r \in [m]$ and $j \in [d] \setminus [k]$ that
$$|w_{r,j}^{(T)}| = (1-\eta\lambda)^T |w_{r,j}^{(0)}| = \big((1-\eta\lambda)^{(\eta\lambda)^{-1}}\big)^{T\eta\lambda} \leq \exp(-T\eta\lambda) \leq d^{-(k+1)},$$
where the last inequality is by $T \geq k\eta^{-1}\lambda^{-1}\log(d)$. According to Corollary C.5, for any $r \in \Omega_{\mathrm{b}}$ and $j \in [k]$, we have that
$$|w_{r,j}^{(T)}| \leq (1-\eta\lambda)^T |w_{r,j}^{(0)}| = \big((1-\eta\lambda)^{(\eta\lambda)^{-1}}\big)^{T\eta\lambda} \leq \exp(-T\eta\lambda) \leq d^{-(k+1)}.$$
□

**Lemma C.7.** *Under Condition 4.2, with a probability of at least $1 - \delta$ with respect to the randomness in the neural network's initialization, trained neural network $f(\mathbf{W}^{(T)}, \mathbf{x})$ approximates accurate classifier $(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})$ well:*
$$\mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\mathbf{x}}}\bigg(\frac{f(\mathbf{W}^{(T)}, \mathbf{x})}{(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})} \in [0.5, 1.5]\bigg) \geq 1 - \epsilon.$$

*Proof.* First, we can rewrite (4.1) as follows:
$$f(\mathbf{W}^*, \mathbf{x}) = \sum_{(b_1, \cdots, b_k) \in \{\pm 1\}^k} (b_1 \cdots b_k) \cdot \sigma(\langle \mathbf{w}_{b_1 \cdots b_k}^*, \mathbf{x} \rangle),$$
where $\mathbf{w}_{b_1 \cdots b_k}^* = [b_1, b_2, \cdots, b_k, 0, \cdots, 0]^\top$. To prove this lemma, we need to estimate the noise part of the inner product $\langle \mathbf{w}_r^{(T)}, \mathbf{x} \rangle$. By Hoeffding's inequality, we have the following upper bound for the noise part $\sum_{j=k+1}^d w_{r,j}^{(T)} x_j$:
$$\begin{aligned} \mathbb{P}_{\mathbf{x}}\Bigg(\Bigg|\sum_{j=k+1}^d w_{r,j}^{(T)} x_j - \mathbb{E}\Bigg[\sum_{j=k+1}^d w_{r,j}^{(T)} x_j\Bigg]\Bigg| \geq x\Bigg) &= \mathbb{P}_{\mathbf{x}}\Bigg(\Bigg|\sum_{j=k+1}^d w_{r,j}^{(t)} x_j\Bigg| \geq x\Bigg) \\ &\leq 2\exp\bigg(-\frac{x^2}{2\sum_{j=k+1}^d [w_{r,j}^{(t)}]^2}\bigg). \end{aligned}$$

Then with probability at least $1-\epsilon/m$ we have that for fixed $r \in [m]$

$$\begin{aligned}\bigg|\sum_{j=k+1}^{d} w_{r,j}^{(T)} x_j\bigg| &\le \sqrt{2}\log(2m/\epsilon)\sqrt{\sum_{j=k+1}^{d}[w_{r,j}^{(T)}]^2} \\ &= \sqrt{2}\log(2m/\epsilon)\|\mathbf{w}_{r,[k+1:d]}^{(T)}\|_2 \\ &\le \sqrt{2}\log(2m/\epsilon)d^{-(k+1)}(d-k)^{1/2} \\ &\le d^{-k},\end{aligned}$$

where the last inequality is by Condition 4.2. By applying union bound to all $m$ neurons, with probability at least $1-\epsilon$ we have that for any $r \in [m]$

$$\bigg|\sum_{j=k+1}^{d} w_{r,j}^{(T)} x_j\bigg| \le d^{-k}. \tag{C.2}$$

By Lemma C.6, we have

$$\begin{aligned}&\bigg|f(\mathbf{W}^{(T)},\mathbf{x}) - \frac{m}{2^{k+1}}\cdot f(\mathbf{W}^*,\mathbf{x})\bigg| \\ &\le \bigg|\sum_{r\in\Omega_{\mathrm{g}}} a_r\sigma(\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle) - \frac{m}{2^{k+1}}\cdot\sum_{r=1}^{2^k} a_r^*\sigma(\langle \mathbf{w}_r^*,\mathbf{x}\rangle)\bigg| + \bigg|\sum_{r\in\Omega_{\mathrm{b}}} a_r\sigma(\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle)\bigg| \\ &\le \sum_{(b_1,\cdots,b_k)\in\{\pm1\}^k}\bigg|\sum_{\Omega_{b_1\cdots b_k}\cap\Omega_{\mathrm{g}}}\sigma(\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle) - \frac{m}{2^{k+1}}\cdot\sigma(\langle \mathbf{w}_{b_1\cdots b_k}^*,\mathbf{x}\rangle)\bigg| + \bigg|\sum_{r\in\Omega_{\mathrm{b}}} a_r\sigma(\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle)\bigg| \\ &\le \sum_{(b_1,\cdots,b_k)\in\{\pm1\}^k}\Bigg(\Big(\frac{\alpha m}{2^{k+1}}\Big)\bigg|\sum_{j=1}^{k} b_j x_j\bigg|^k + \sum_{\Omega_{b_1\cdots b_k}\cap\Omega_{\mathrm{g}}}\big|\sigma(\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle) - \sigma(\langle \mathbf{w}_{b_1\cdots b_k}^*,\mathbf{x}\rangle)\big|\Bigg) \\ &\qquad + \bigg|\sum_{r\in\Omega_{\mathrm{b}}} a_r\sigma(\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle)\bigg| \\ &\le \sum_{(b_1,\cdots,b_k)\in\{\pm1\}^k}\Bigg(\Big(\frac{\alpha m}{2^{k+1}}\Big)\bigg|\sum_{j=1}^{k} b_j x_j\bigg|^k + \sum_{\Omega_{b_1\cdots b_k}\cap\Omega_{\mathrm{g}}} kd^{-k}\big(k+d^{-k}\big)^{k-1}\Bigg) + |\Omega_{\mathrm{b}}|\big(2d^{-k}\big)^k \\ &\le \Big(\frac{\alpha m}{2^{k+1}}\Big)2k^k(1+e^{-2})^k + |\Omega_{\mathrm{g}}|kd^{-k}\big(k+d^{-k}\big)^{k-1} + |\Omega_{\mathrm{b}}|\big(2d^{-k}\big)^k \\ &\le \Big(\alpha k^k 2^{-k}(1+e^{-2})^k + 0.5(1+\alpha)kd^{-k}\big(k+d^{-k}\big)^{k-1} + 0.5(1+\alpha)\big(2d^{-k}\big)^k\Big)\cdot m,\end{aligned}$$

where the first three inequalities are by triangle inequality and Lemma B.1; the fourth inequality is due to

$$\begin{aligned}&\big|\sigma(\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle) - \sigma(\langle \mathbf{w}_{b_1\cdots b_k}^*,\mathbf{x}\rangle)\big| \\ &\le \big|(\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle)^k - (\langle \mathbf{w}_{b_1\cdots b_k}^*,\mathbf{x}\rangle)^k\big| \\ &\le |\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle - \langle \mathbf{w}_{b_1\cdots b_k}^*,\mathbf{x}\rangle|\cdot k(\max\{|\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle|,|\langle \mathbf{w}_{b_1\cdots b_k}^*,\mathbf{x}\rangle|\})^{k-1} \\ &= \bigg|\sum_{j=k+1}^{d} w_{r,j}^{(T)}x_j\bigg|\cdot k\Bigg(\max\bigg\{\bigg|\sum_{j=1}^{k} b_j x_j + \sum_{j=k+1}^{d} w_{r,j}^{(T)}x_j\bigg|,\bigg|\sum_{j=1}^{k} b_j x_j\bigg|\bigg\}\Bigg)^{k-1} \\ &\le k\bigg|\sum_{j=k+1}^{d} w_{r,j}^{(T)}x_j\bigg|\cdot\Bigg(\bigg|\sum_{j=1}^{k} b_j x_j\bigg| + \bigg|\sum_{j=k+1}^{d} w_{r,j}^{(T)}x_j\bigg|\Bigg)^{k-1} \\ &\le kd^{-k}\big(k+d^{-k}\big)^{k-1}\end{aligned}$$

by (C.2), mean value theorem and Lemma C.6 and for $r \in \Omega_{\mathrm{b}}$

$$|\sigma(\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle)| \le |\langle \mathbf{w}_r^{(T)},\mathbf{x}\rangle|^k \le \big(kd^{-(k+1)} + d^{-k}\big)^k \le \big(2d^{-k}\big)^k. \tag{C.3}$$

Since $(m/2^{k+1}) \cdot |f(\mathbf{W}^*, \mathbf{x})| = 0.5k! \cdot m$, then as long as

$$\alpha \leq \frac{\sqrt{2\pi k}}{8} \cdot \Big(\frac{e + e^{-1}}{2}\Big)^{-k}, \text{ and } \alpha \leq 1,$$

we have

$$\begin{aligned}
&\frac{|f(\mathbf{W}^{(T)}, \mathbf{x}) - (m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})|}{(m/2^{k+1}) \cdot |f(\mathbf{W}^*, \mathbf{x})|} \\
&\leq \frac{\alpha k^k 2^{-k}(1 + e^{-2})^k + 0.5(1+\alpha)kd^{-k}\big(k + d^{-k}\big)^{k-1} + 0.5(1+\alpha)\big(2d^{-k}\big)^k}{0.5k!} \\
&\leq \frac{2\alpha k^k 2^{-k}(1 + e^{-2})^k + (1+\alpha)kd^{-k}\big(k + d^{-k}\big)^{k-1} + (1+\alpha)\big(2d^{-k}\big)^k}{\sqrt{2\pi k}(k/e)^k} \\
&= \sqrt{\frac{2}{\pi k}}\alpha\Big(\frac{e + e^{-1}}{2}\Big)^k + \sqrt{\frac{2}{\pi k}} \cdot \Big(1 + \frac{d^{-k}}{k}\Big)^{k-1} \cdot \Big(\frac{e}{d}\Big)^k + \sqrt{\frac{2}{\pi k}} \cdot \Big(\frac{2e}{kd^k}\Big)^k \\
&\leq 0.5,
\end{aligned}$$

where the second inequality is by Stirling's approximation. Then it follows that

$$\frac{f(\mathbf{W}^{(T)}, \mathbf{x})}{(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})} \in [0.5, 1.5].$$

□

# D Stochastic Sign GD

In this section, we consider stochastic sign gradient descent for learning $k$-parity function. The primary aim of this section is to demonstrate that the trajectory produced by SGD closely resembles that of population GD. To begin, let's recall the update rule of SGD:

$$\mathbf{w}_r^{(t+1)} = (1 - \lambda\eta)\mathbf{w}_r^{(t)} + \eta \cdot \widetilde{\operatorname{sign}}\bigg(\frac{1}{|S_t|}\sum_{(\mathbf{x},y)\in S_t} \sigma'(\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle) \cdot a_r y \mathbf{x}\bigg),$$

where $|S_t| = B$. Our initial step involves estimating the approximation error between the stochastic gradient and the population gradient, detailed within the lemma that follows.

**Lemma D.1.** *With probability at least $1 - \delta$ with respect to the randomness of online data selection, for all $t \leq T$, the following bound holds true for each neuron $r \in [m]$ and for each coordinate $j \in [d]$:*

$$\bigg|\frac{1}{|S_t|}\sum_{(\mathbf{x},y)\in S_t} \sigma'(\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle) \cdot a_r y \mathbf{x} - \mathbb{E}_{(\mathbf{x},y)}\big[\sigma'(\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle) \cdot a_r y x_j\big]\bigg| \leq \epsilon_1 \cdot \|\mathbf{w}_r^{(t)}\|_2^{k-1}, \qquad \text{(D.1)}$$

*where $\epsilon_1$ is defined as*

$$\epsilon_1 = \frac{2^{k/2}k(\log(16mdBT/\delta))^{(k-1)/2}\log(8mdT/\delta)}{\sqrt{B}} + \frac{kd^{(k-3)/2}\delta}{8mBT}.$$

*Proof.* To prove (D.1), let us introduce the following notations:

$$\begin{aligned}
g_{r,j}(\mathbf{x}, y) &= \sigma'(\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle) \cdot a_r y x_j = k(\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle)^{k-1} \cdot a_r y x_j, \\
h_{r,j}(\mathbf{x}, y) &= g_{r,j}(\mathbf{x}, y) \cdot \mathbb{1}\big[|\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle| \leq \gamma\big],
\end{aligned}$$

where $g_{r,j}(\mathbf{x}, y)$ represents the gradient at the point $(\mathbf{x}, y)$, and $h_{r,j}(\mathbf{x}, y)$ denotes the truncated version of $g_{r,j}(\mathbf{x}, y)$, which is employed for the convenience of applying the Hoeffding's inequality. Firstly, utilizing Hoeffding's inequality, we can assert the following:

$$\mathbb{P}\Bigg(\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t} h_{r,j}(\mathbf{x}, y) - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} h_{r,j}(\mathbf{x}, y)\bigg| \geq x\Bigg) \leq 2\exp\Bigg(-\frac{Bx^2}{2(k\gamma^{k-1})^2}\Bigg). \qquad \text{(D.2)}$$

Furthermore, we can establish an upper bound for the difference between the expectations of $h_{r,j}(\mathbf{x},y)$ and $g_{r,j}(\mathbf{x},y)$:

$$\begin{aligned}\Big|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}h_{r,j}(\mathbf{x},y)-\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}g_{r,j}(\mathbf{x},y)\Big| &= \Big|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}g_{r,j}(\mathbf{x},y)\cdot\mathbb{1}\left[|\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle|>\gamma\right]\Big| \\ &\le kd^{(k-1)/2}\|\mathbf{w}_r^{(t)}\|_2^{k-1}\cdot\mathbb{P}\big(|\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle|>\gamma\big) \\ &\le kd^{(k-1)/2}\|\mathbf{w}_r^{(t)}\|_2^{k-1}\cdot 2\exp\bigg(-\frac{\gamma^2}{2\|\mathbf{w}_r^{(t)}\|_2^2}\bigg),\end{aligned} \tag{D.3}$$

where the first inequality is by Cauchy inequality and the second inequality is by Hoeffding's inequality. Additionally, with high probability, the gradient and the truncated gradient are identical:

$$\begin{aligned}&\mathbb{P}\bigg(\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t}h_{r,j}(\mathbf{x},y)-\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t}g_{r,j}(\mathbf{x},y)\bigg|=0\bigg) \\ &=\mathbb{P}\bigg(\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t}g_{r,j}(\mathbf{x},y)\,\mathbb{1}\left[|\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle|>\gamma\right]\bigg|=0\bigg) \\ &\ge\mathbb{P}\Big(|\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle|\le\gamma,\forall(\mathbf{x},y)\in S_t\Big) \\ &\ge\bigg(1-2\exp\bigg(-\frac{\gamma^2}{2\|\mathbf{w}_r^{(t)}\|_2^2}\bigg)\bigg)^B \\ &\ge 1-2B\exp\bigg(-\frac{\gamma^2}{2\|\mathbf{w}_r^{(t)}\|_2^2}\bigg),\end{aligned} \tag{D.4}$$

where the second inequality applies Hoeffding's inequality. Combining inequalities (D.2), (D.3), and (D.4), we can assert with probability at least

$$1-2\exp\bigg(-\frac{Bx^2}{2(k\gamma^{k-1})^2}\bigg)-2B\exp\bigg(-\frac{\gamma^2}{2\|\mathbf{w}_r^{(t)}\|_2^2}\bigg) \tag{D.5}$$

that the following inequality holds:

$$\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t}g_{r,j}(\mathbf{x},y)-\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}g_{r,j}(\mathbf{x},y)\bigg|\le x+kd^{(k-1)/2}\|\mathbf{w}_r^{(t)}\|_2^{k-1}\cdot 2\exp\bigg(-\frac{\gamma^2}{2\|\mathbf{w}_r^{(t)}\|_2^2}\bigg). \tag{D.6}$$

By setting $\gamma=\sqrt{2\log(16B/\delta)}\|\mathbf{w}_r^{(t)}\|_2$ and $x=\sqrt{2}k\gamma^{k-1}\log(8/\delta)/\sqrt{B}$, we establish that with probability at least $1-\delta$, the following bound is true:

$$\begin{aligned}&\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t}g_{r,j}(\mathbf{x},y)-\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}g_{r,j}(\mathbf{x},y)\bigg| \\ &\le\frac{\sqrt{2}k\gamma^{k-1}\log(8/\delta)}{\sqrt{B}}+\frac{kd^{(k-1)/2}\|\mathbf{w}_r^{(t)}\|_2^{k-1}\delta}{8B} \\ &=\frac{2^{k/2}k(\log(16B/\delta))^{(k-1)/2}\log(8/\delta)\|\mathbf{w}_r^{(t)}\|_2^{k-1}}{\sqrt{B}}+\frac{kd^{(k-1)/2}\|\mathbf{w}_r^{(t)}\|_2^{k-1}\delta}{8B}.\end{aligned}$$

Applying a union bound over all indices $r\in[m], j\in[d]$, and iterations $t\in[0,T-1]$, we conclude with probability at least $1-\delta$ that

$$\begin{aligned}&\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t}g_{r,j}(\mathbf{x},y)-\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}g_{r,j}(\mathbf{x},y)\bigg| \\ &=\frac{2^{k/2}k(\log(16mdBT/\delta))^{(k-1)/2}\log(8mdT/\delta)\|\mathbf{w}_r^{(t)}\|_2^{k-1}}{\sqrt{B}}+\frac{kd^{(k-1)/2}\|\mathbf{w}_r^{(t)}\|_2^{k-1}\delta}{8mdBT}\end{aligned}$$

$$= \left( \frac{2^{k/2} k (\log(16mdBT/\delta))^{(k-1)/2} \log(8mdT/\delta)}{\sqrt{B}} + \frac{kd^{(k-3)/2}\delta}{8mBT} \right) \cdot \|\mathbf{w}_r^{(t)}\|_2^{k-1}.$$

□

Based on Lemma D.1, we can get the following lemma showing that with high probability, the stochastic sign gradient follows the same update rule as the population sign gradient.

**Lemma D.2.** *Under Condition 4.2, with probability at least* $1 - \delta$ *with respect to the randomness of online data selection, the following sign SGD update rule holds:*

$$\begin{aligned} w_{r,j}^{(t+1)} &= (1 - \eta\lambda) w_{r,j}^{(t)} + \eta \cdot \widetilde{\mathrm{sign}}\left( k! a_r \cdot \frac{w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}}{w_{r,j}^{(t)}} \right), && j \in [k], \\ w_{r,j}^{(t+1)} &= (1 - \eta\lambda) w_{r,j}^{(t)}, && j \notin [k]. \end{aligned}$$

*Proof.* We prove this by using induction. We prove the following hypotheses:

$$\|\mathbf{w}_r^{(t+1)}\|_2 \leq \|\mathbf{w}_r^{(t)}\|_2, \qquad \forall r \in [m]. \tag{H$_1$}$$

$$w_{r,j}^{(t+1)} = (1 - \eta\lambda) w_{r,j}^{(t)} + \eta \cdot \widetilde{\mathrm{sign}}\left( k! a_r \cdot \frac{w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}}{w_{r,j}^{(t)}} \right), \quad \forall r \in [m], j \in [k]. \tag{H$_2$}$$

$$w_{r,j}^{(t+1)} = (1 - \eta\lambda) w_{r,j}^{(t)}, \qquad \forall r \in [m], j \notin [k]. \tag{H$_3$}$$

$$b_j w_{r,j}^{(t)} = 1, \qquad \forall r \in \Omega_{b_1 \cdots b_k} \cap \Omega_{\mathrm{g}}, \forall j \in [k]. \tag{H$_4$}$$

$$b_j w_{r,j}^{(t)} = b_{j'} w_{r,j'}^{(t)}, \qquad \forall r \in \Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{g}}, \forall j, j' \in [k]. \tag{H$_5$}$$

$$0 < b_j w_{r,j}^{(t+1)} \leq (1 - \eta\lambda) b_j w_{r,j}^{(t)}, \qquad \forall r \in \Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{b}}, \forall j \in [k]. \tag{H$_6$}$$

We will show that $\mathrm{H}_2(0)$, $\mathrm{H}_3(0)$, $\mathrm{H}_4(0)$ and $\mathrm{H}_5(0)$ are true and for any $t \geq 0$ we have

- $\mathrm{H}_2(t), \mathrm{H}_4(t) \Longrightarrow \mathrm{H}_4(t+1)$. (This can be established by adapting the proof of Lemma C.3; hence, we omit the proof details here.)
- $\mathrm{H}_2(t), \mathrm{H}_5(t) \Longrightarrow \mathrm{H}_5(t+1), \mathrm{H}_6(t)$. (This can be shown by following the proof of Lemma C.4, so the proof details are omitted here.)
- $\mathrm{H}_3(t), \mathrm{H}_4(t), \mathrm{H}_4(t+1), \mathrm{H}_6(t) \Longrightarrow \mathrm{H}_1(t)$.
- $\{\mathrm{H}_1(s)\}_{s=0}^{t} \Longrightarrow \mathrm{H}_2(t+1), \mathrm{H}_3(t+1)$.

$\mathrm{H}_4(0)$ and $\mathrm{H}_5(0)$ are obviously true since $w_{r,j}^{(0)} = b_j$ for any $r \in \Omega_{b_1 b_2 \cdots b_k}$ and $j \in [k]$. To prove that $\mathrm{H}_2(0)$ and $\mathrm{H}_3(0)$ are true, we only need to verify that

$$\widetilde{\mathrm{sign}}\left( \frac{1}{|S_0|} \sum_{(\mathbf{x},y) \in S_0} \sigma'(\langle \mathbf{w}_r^{(0)}, \mathbf{x} \rangle) \cdot a_r y x_j \right) = \widetilde{\mathrm{sign}}\left( k! a_r \cdot \frac{w_{r,1}^{(0)} w_{r,2}^{(0)} \cdots w_{r,k}^{(0)}}{w_{r,j}^{(0)}} \right), \quad \forall j \in [k], \tag{D.7}$$

$$\widetilde{\mathrm{sign}}\left( \frac{1}{|S_0|} \sum_{(\mathbf{x},y) \in S_0} \sigma'(\langle \mathbf{w}_r^{(0)}, \mathbf{x} \rangle) \cdot a_r y x_j \right) = 0, \qquad \forall j \notin [k]. \tag{D.8}$$

By Lemma D.1, we have for $j \notin [k]$

$$\left| \frac{1}{|S_0|} \sum_{(\mathbf{x},y) \in S_t} \sigma'(\langle \mathbf{w}_r^{(0)}, \mathbf{x} \rangle) \cdot a_r y x_j \right| \leq \epsilon_1 \cdot \|\mathbf{w}_r^{(0)}\|_2^{k-1} = \epsilon_1 \cdot d^{\frac{k-1}{2}} < \rho,$$

leading to (D.8). For $j \in [k]$, we have

$$\left| \frac{1}{|S_0|} \sum_{(\mathbf{x},y)\in S_0} \sigma'(\langle \mathbf{w}_r^{(0)}, \mathbf{x}\rangle) \cdot a_r y x_j - k! a_r \cdot \frac{w_{r,1}^{(0)} w_{r,2}^{(0)} \cdots w_{r,k}^{(0)}}{w_{r,j}^{(0)}} \right| \leq \epsilon_1 \cdot \|\mathbf{w}_r^{(0)}\|_2^{k-1} = \epsilon_1 \cdot d^{\frac{k-1}{2}},$$

Since $k! - \epsilon_1 \cdot d^{\frac{k-1}{2}} \geq \rho$, we can get

$$\widetilde{\operatorname{sign}}\left( \frac{1}{|S_0|} \sum_{(\mathbf{x},y)\in S_0} \sigma'(\langle \mathbf{w}_r^{(0)}, \mathbf{x}\rangle) \cdot a_r y x_j \right) = \widetilde{\operatorname{sign}}\left( k! a_r \cdot \frac{w_{r,1}^{(0)} w_{r,2}^{(0)} \cdots w_{r,k}^{(0)}}{w_{r,j}^{(0)}} \right),$$

which verifies (D.7). Next, we verify that $\mathrm{H}_3(t), \mathrm{H}_4(t), \mathrm{H}_4(t+1), \mathrm{H}_6(t) \Longrightarrow \mathrm{H}_1(t)$. For $r \in \Omega_{\mathrm{g}}$, given $\mathrm{H}_3(t)$, $\mathrm{H}_4(t)$ and $\mathrm{H}_4(t+1)$, we can get

$$\|\mathbf{w}_r^{(t+1)}\|_2 = \left( \sum_{j=1}^{d} \big(w_{r,j}^{(t+1)}\big)^2 \right)^{\frac{1}{2}} = \left( \sum_{j=1}^{k} \big(w_{r,j}^{(t)}\big)^2 + \sum_{j=k+1}^{d} \big((1-\eta\lambda) w_{r,j}^{(t)}\big)^2 \right)^{\frac{1}{2}} \leq \|\mathbf{w}_r^{(t)}\|_2.$$

For $r \in \Omega_{\mathrm{b}}$, given $\mathrm{H}_3(t)$ and $\mathrm{H}_6(t)$, we can get

$$\|\mathbf{w}_r^{(t+1)}\|_2 = \left( \sum_{j=1}^{d} \big(w_{r,j}^{(t+1)}\big)^2 \right)^{\frac{1}{2}} \leq \left( \sum_{j=1}^{d} \big((1-\eta\lambda) w_{r,j}^{(t)}\big)^2 \right)^{\frac{1}{2}} \leq \|\mathbf{w}_r^{(t)}\|_2.$$

Finally, we verify that $\{\mathrm{H}_1(s)\}_{s=0}^{t} \Longrightarrow \mathrm{H}_2(t+1), \mathrm{H}_3(t+1)$. Notice that $\|\mathbf{w}_r^{(t+1)}\|_2 \leq \|\mathbf{w}_r^{(0)}\|_2$ given $\{\mathrm{H}_1(s)\}_{s=0}^{t}$, we can prove $\mathrm{H}_3(t+1)$ and $\mathrm{H}_2(t+1)$ by following the prove of (D.7) and (D.8) given Lemma D.1. □

Based on Lemma D.2 and the proof of Lemma C.6, Lemma C.7, we can get the following lemmas and theorems aligning with the result of population sign GD.

**Lemma D.3.** *Under Condition 4.2, for $T = \Theta(k\eta^{-1}\lambda^{-1}\log(d))$,with a probability of at least $1-\delta$ with respect to the randomness of the online data selection, it holds that*

$$\begin{aligned} w_{r,j}^{(T)} &= b_j, && \forall r \in \Omega_{b_1 b_1 \cdots b_k} \cap \Omega_{\mathrm{g}}, j \in [k], \\ |w_{r,j}^{(T)}| &\leq d^{-(k+1)}, && \forall r \in \Omega_{\mathrm{g}}, j \in [d] \setminus [k], \\ |w_{r,j}^{(T)}| &\leq d^{-(k+1)}, && \forall r \in \Omega_{\mathrm{b}}, j \in [d]. \end{aligned}$$

**Lemma D.4.** *Under Condition 4.2, with a probability of at least $1-2\delta$ with respect to the randomness in the neural network's initialization and the online data selection, trained neural network $f(\mathbf{W}^{(T)}, \mathbf{x})$ approximates accurate classifier $(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})$ well:*

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}\left( \frac{f(\mathbf{W}^{(T)}, \mathbf{x})}{(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})} \in [0.5, 1.5] \right) \geq 1 - \epsilon.$$

Based on Lemma D.4, we are now ready to prove our main theorem.

**Theorem D.5.** *Under Condition 4.2, we run mini-batch SGD for $T = \Theta(k\eta^{-1}\lambda^{-1}\log(d))$ iterations. Then with probability at least $1-2\delta$ with respect to the randomness of neural network initialization and the online data selection, it holds that*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}(y f(\mathbf{W}^{(T)}, \mathbf{x}) \geq \gamma m) \geq 1 - \epsilon.$$

*where $\gamma = 0.25k!$ is a constant.*

*Proof.* Given Lemma D.4, we have

$$\begin{aligned} &\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}\left( \frac{f(\mathbf{W}^{(T)}, \mathbf{x})}{(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})} \geq 0.5 \right) \\ &\geq \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}\left( \frac{f(\mathbf{W}^{(T)}, \mathbf{x})}{(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})} \in [0.5, 1.5] \right) \end{aligned}$$

$$\geq 1 - \epsilon.$$

According to Proposition 4.1, we can get

$$\begin{aligned}
&\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}\bigg(\frac{f(\mathbf{W}^{(T)},\mathbf{x})}{(m/2^{k+1})\cdot f(\mathbf{W}^*,\mathbf{x})} \geq 0.5\bigg)\\
&= \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}\bigg(\frac{yf(\mathbf{W}^{(T)},\mathbf{x})}{(m/2^{k+1})\cdot k!\cdot 2^k} \geq 0.5\bigg)\\
&= \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathbf{x}}}\big(yf(\mathbf{W}^{(T)},\mathbf{x}) \geq 0.25k!\cdot m\big),
\end{aligned}$$

which completes the proof. □

# E Trainable Second Layer

In this section, we consider sign SGD for training the first and second layers together. In this scenario, we have the following sign SGD update rule:

$$\mathbf{w}_r^{(t+1)} = (1-\lambda\eta)\mathbf{w}_r^{(t)} + \eta\cdot\widetilde{\mathrm{sign}}\bigg(\frac{1}{|S_t|}\sum_{(\mathbf{x},y)\in S_t}\sigma'(\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle)\cdot a_r^{(t)}y\mathbf{x}\bigg), \tag{E.1}$$

$$a_r^{(t+1)} = a_r^{(t)} + \eta_2\cdot\widetilde{\mathrm{sign}}\bigg(\frac{1}{|S_t|}\sum_{(\mathbf{x},y)\in S_t}\sigma(\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle)\bigg), \tag{E.2}$$

where $|S_t| = B$. For training the neural network over $T = \Theta(k(\eta\lambda)^{-1}\log(d))$ iterations, we adopt a small learning rate for the second layer, adhering to the condition:

$$\eta_2 \leq \frac{1}{4T}\sqrt{\frac{\pi k}{8}}\Big(\frac{e+e^{-1}}{2}\Big)^{-k}.$$

The network is initialized symmetrically: for every $r\in[m]$, initialize $\mathbf{w}_r^{(0)}\sim\mathrm{Unif}(\{-1,1\}^d)$ and initialize $a_r^{(0)}\sim\mathrm{Unif}(\{-1,1\})$. Under this setting, denote

$$\Omega_{\mathrm{g}} = \bigg\{r\in[m]\,\bigg|\,a_r = \prod_{j=1}^k\mathrm{sign}(w_{r,j}^{(0)})\bigg\}, \Omega_{\mathrm{b}} = \bigg\{r\in[m]\,\bigg|\,a_r = -\prod_{j=1}^k\mathrm{sign}(w_{r,j}^{(0)})\bigg\}.$$

Similar to the fix-second-layer case, our initial step involves estimating the approximation error between the SGD gradient and the population gradient, detailed within the lemma that follows.

**Lemma E.1.** *With probability at least $1-\delta$ with respect to the randomness of online data selection, for all $t\leq T$, we have for any $r\in[m]$ and $j\in[d]$ that*

$$\bigg|\frac{1}{|S_t|}\sum_{(\mathbf{x},y)\in S_t}\sigma'(\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle)\cdot a_r^{(t)}yx_j - \mathbb{E}\big[\sigma'(\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle)\cdot a_r^{(t)}yx_j\big]\bigg| \leq \epsilon_1\cdot|a_r^{(t)}|\|\mathbf{w}_r^{(t)}\|_2^{k-1}, \tag{E.3}$$

*where*

$$\epsilon_1 = \frac{2^{k/2}k(\log(16mdBT/\delta))^{(k-1)/2}\log(8mdT/\delta)}{\sqrt{B}} + \frac{kd^{(k-3)/2}\delta}{8mBT}.$$

*Proof.* To prove (E.3), we denote

$$\begin{aligned}
g_{r,j}(\mathbf{x},y) &= \sigma'(\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle)\cdot a_r^{(t)}yx_j = k(\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle)^{k-1}\cdot a_r^{(t)}yx_j,\\
h_{r,j}(\mathbf{x},y) &= g_{r,j}(\mathbf{x},y)\cdot\mathbb{1}\left[|\langle\mathbf{w}_r^{(t)},\mathbf{x}\rangle|\leq\gamma\right].
\end{aligned}$$

Initially, by invoking Hoeffding's inequality, we have the following probability bound:

$$\mathbb{P}\Bigg(\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t}h_{r,j}(\mathbf{x},y) - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}h_{r,j}(\mathbf{x},y)\bigg| \geq x\Bigg) \leq 2\exp\Bigg(-\frac{Bx^2}{2(k\gamma^{k-1}a_r^{(t)})^2}\Bigg). \tag{E.4}$$

Next, we establish an upper bound for the difference between the expected values of $h_{r,j}(\mathbf{x}, y)$ and $g_{r,j}(\mathbf{x}, y)$:

$$
\begin{aligned}
\Big|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} h_{r,j}(\mathbf{x}, y) - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} g_{r,j}(\mathbf{x}, y)\Big| &= \Big|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} g_{r,j}(\mathbf{x}, y) \cdot \mathbb{1}\left[|\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle| > \gamma\right]\Big| \\
&\leq k d^{(k-1)/2} \|\mathbf{w}_r^{(t)}\|_2^{k-1} |a_r^{(t)}| \cdot \mathbb{P}\big(|\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle| > \gamma\big) \\
&\leq k d^{(k-1)/2} \|\mathbf{w}_r^{(t)}\|_2^{k-1} |a_r^{(t)}| \cdot 2\exp\bigg(-\frac{\gamma^2}{2\|\mathbf{w}_r^{(t)}\|_2^2}\bigg),
\end{aligned}
\tag{E.5}
$$

where the first inequality follows from the Cauchy-Schwarz inequality and the second from Hoeffding's inequality. With high probability, the gradient and the truncated gradient coincide:

$$
\begin{aligned}
&\mathbb{P}\Bigg(\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t} h_{r,j}(\mathbf{x}, y) - \frac{1}{B}\sum_{(\mathbf{x},y)\in S_t} g_{r,j}(\mathbf{x}, y)\bigg| = 0\Bigg) \\
&= \mathbb{P}\Bigg(\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t} g_{r,j}(\mathbf{x}, y)\,\mathbb{1}\left[|\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle| > \gamma\right]\bigg| = 0\Bigg) \\
&\geq \mathbb{P}\Big(|\langle \mathbf{w}_r^{(t)}, \mathbf{x}\rangle| \leq \gamma, \forall (\mathbf{x}, y) \in S_t\Big) \\
&\geq \Bigg(1 - 2\exp\bigg(-\frac{\gamma^2}{2\|\mathbf{w}_r^{(t)}\|_2^2}\bigg)\Bigg)^B \\
&\geq 1 - 2B\exp\bigg(-\frac{\gamma^2}{2\|\mathbf{w}_r^{(t)}\|_2^2}\bigg).
\end{aligned}
\tag{E.6}
$$

Combing (E.4), (E.5) and (E.6), it holds with probability at least

$$
1 - 2\exp\bigg(-\frac{Bx^2}{2(k\gamma^{k-1}a_r^{(t)})^2}\bigg) - 2B\exp\bigg(-\frac{\gamma^2}{2\|\mathbf{w}_r^{(t)}\|_2^2}\bigg)
$$

that

$$
\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t} g_{r,j}(\mathbf{x}, y) - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} g_{r,j}(\mathbf{x}, y)\bigg| \leq x + k d^{(k-1)/2} \|\mathbf{w}_r^{(t)}\|_2^{k-1} |a_r^{(t)}| \cdot 2\exp\bigg(-\frac{\gamma^2}{2\|\mathbf{w}_r^{(t)}\|_2^2}\bigg).
$$

By taking $\gamma = \sqrt{2\log(16B/\delta)}\|\mathbf{w}_r^{(t)}\|_2$ and $x = \sqrt{2}k\gamma^{k-1}|a_r^{(t)}|\log(8/\delta)/\sqrt{B}$, then with probability at least $1-\delta$ it holds that

$$
\begin{aligned}
&\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t} g_{r,j}(\mathbf{x}, y) - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} g_{r,j}(\mathbf{x}, y)\bigg| \\
&\leq \frac{\sqrt{2}k\gamma^{k-1}|a_r^{(t)}|\log(8/\delta)}{\sqrt{B}} + \frac{k d^{(k-1)/2}\|\mathbf{w}_r^{(t)}\|_2^{k-1}|a_r^{(t)}|\delta}{8B} \\
&= \frac{2^{k/2}k(\log(16B/\delta))^{(k-1)/2}\log(8/\delta)|a_r^{(t)}|\|\mathbf{w}_r^{(t)}\|_2^{k-1}}{\sqrt{B}} + \frac{k d^{(k-1)/2}|a_r^{(t)}|\|\mathbf{w}_r^{(t)}\|_2^{k-1}\delta}{8B}.
\end{aligned}
$$

Then, by applying a union bound to all $r \in [m], j \in [d]$ and iterations $t \in [0, T-1]$, it holds with probability at least $1-\delta$ that

$$
\begin{aligned}
&\bigg|\frac{1}{B}\sum_{(\mathbf{x},y)\in S_t} g_{r,j}(\mathbf{x}, y) - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} g_{r,j}(\mathbf{x}, y)\bigg| \\
&= \frac{2^{k/2}k(\log(16mdBT/\delta))^{(k-1)/2}\log(8mdT/\delta)\|\mathbf{w}_r^{(t)}\|_2^{k-1}|a_r^{(t)}|}{\sqrt{B}} + \frac{k d^{(k-1)/2}\|\mathbf{w}_r^{(t)}\|_2^{k-1}|a_r^{(t)}|\delta}{8mdBT} \\
&= \Bigg(\frac{2^{k/2}k(\log(16mdBT/\delta))^{(k-1)/2}\log(8mdT/\delta)}{\sqrt{B}} + \frac{k d^{(k-3)/2}\delta}{8mBT}\Bigg) \cdot |a_r^{(t)}|\|\mathbf{w}_r^{(t)}\|_2^{k-1}.
\end{aligned}
$$

□

**Lemma E.2** (Stability of Second Layer Weights). *For $t \leq T = \Theta(k(\eta\lambda)^{-1}\log(d))$, the magnitude of change in the second layer weights from their initial values is bounded as follows:*

$$|a_r^{(t)} - a_r^{(0)}| \leq c,$$

*where*

$$c = \frac{1}{4}\sqrt{\frac{\pi k}{8}}\Big(\frac{e+e^{-1}}{2}\Big)^{-k}.$$

*Consequently, the sign of each weight remains consistent over time:*

$$\mathrm{sign}(a_r^{(t)}) = \mathrm{sign}(a_r^{(0)}).$$

*Proof.* Notice that by (E.2) and $\widetilde{\mathrm{sign}}(\cdot) \in \{-1, 0, 1\}$, we can get for any $t \geq 0$ that

$$a_r^{(t)} - \eta_2 \leq a_r^{(t+1)} \leq a_r^{(t)} + \eta_2,$$

which implies that

$$|a_r^{(t)} - a_r^{(0)}| \leq \eta_2 t \leq \eta_2 T \leq c,$$

where the second inequality is by $t \leq T$, and the last inequality is by the condition $\eta_2 \leq \frac{1}{4T}\sqrt{\frac{\pi k}{8}}\left(\frac{e+e^{-1}}{2}\right)^{-k}$. □

**Lemma E.3.** *With probability at least $1 - \delta$ with respect to the randomness of online data selection, the following sign SGD update rule holds:*

$$w_{r,j}^{(t+1)} = (1-\eta\lambda)w_{r,j}^{(t)} + \eta \cdot \widetilde{\mathrm{sign}}\Bigg(k!a_r^{(t)} \cdot \frac{w_{r,1}^{(t)}w_{r,2}^{(t)}\cdots w_{r,k}^{(t)}}{w_{r,j}^{(t)}}\Bigg), \qquad j \in [k],$$
$$w_{r,j}^{(t+1)} = (1-\eta\lambda)w_{r,j}^{(t)}, \qquad j \notin [k].$$

*Proof.* We prove this by using induction. We prove the following hypotheses:

$$\|\mathbf{w}_r^{(t+1)}\|_2 \leq \|\mathbf{w}_r^{(t)}\|_2, \qquad \forall r \in [m]. \tag{$\mathrm{H}_1$}$$

$$w_{r,j}^{(t+1)} = (1-\eta\lambda)w_{r,j}^{(t)} + \eta_1 \cdot \widetilde{\mathrm{sign}}\Bigg(k!a_r^{(t)} \cdot \frac{w_{r,1}^{(t)}w_{r,2}^{(t)}\cdots w_{r,k}^{(t)}}{w_{r,j}^{(t)}}\Bigg), \quad \forall r \in [m], j \in [k]. \tag{$\mathrm{H}_2$}$$

$$w_{r,j}^{(t+1)} = (1-\eta\lambda)w_{r,j}^{(t)}, \qquad \forall r \in [m], j \notin [k]. \tag{$\mathrm{H}_3$}$$

$$b_j w_{r,j}^{(t)} = 1, \qquad \forall r \in \Omega_{b_1\cdots b_k} \cap \Omega_{\mathrm{g}}, \forall j \in [k]. \tag{$\mathrm{H}_4$}$$

$$b_j w_{r,j}^{(t)} = b_{j'} w_{r,j'}^{(t)}, \qquad \forall r \in \Omega_{b_1 b_2\cdots b_k} \cap \Omega_{\mathrm{g}}, \forall j, j' \in [k]. \tag{$\mathrm{H}_5$}$$

$$0 < b_j w_{r,j}^{(t+1)} \leq (1-\eta\lambda) b_j w_{r,j}^{(t)}, \qquad \forall r \in \Omega_{b_1 b_2\cdots b_k} \cap \Omega_{\mathrm{b}}, \forall j \in [k]. \tag{$\mathrm{H}_6$}$$

We will show that $\mathrm{H}_2(0)$, $\mathrm{H}_3(0)$, $\mathrm{H}_4(0)$ and $\mathrm{H}_5(0)$ are true and for any $t \geq 0$ we have

- $\mathrm{H}_2(t), \mathrm{H}_4(t) \Longrightarrow \mathrm{H}_4(t+1)$.
- $\mathrm{H}_2(t), \mathrm{H}_5(t) \Longrightarrow \mathrm{H}_5(t+1), \mathrm{H}_6(t)$.
- $\mathrm{H}_3(t), \mathrm{H}_4(t), \mathrm{H}_4(t+1), \mathrm{H}_6(t) \Longrightarrow \mathrm{H}_1(t)$.
- $\{\mathrm{H}_1(s)\}_{s=0}^t \Longrightarrow \mathrm{H}_2(t+1), \mathrm{H}_3(t+1)$.

$\mathrm{H}_4(0)$ and $\mathrm{H}_5(0)$ are obviously true since $w_{r,j}^{(0)} = b_j$ for any $r \in \Omega_{b_1 b_2 \cdots b_k}$ and $j \in [k]$. To prove that $\mathrm{H}_2(0)$ and $\mathrm{H}_3(0)$ are true, we can follow the proof of Lemma D.2 by noticing that $|a_r^{(0)}| = 1$. Now, we verify that $\mathrm{H}_2(t), \mathrm{H}_4(t) \Longrightarrow \mathrm{H}_4(t+1)$. By $\mathrm{H}_2(t)$ and $\operatorname{sign}(a_r^{(t)}) = \operatorname{sign}(a_r^{(0)})$ according to Lemma E.2, we have for any neuron $r \in \Omega_{b_1 b_2 \cdots b_k} \cap \Omega_{\mathrm{g}}$ that

$$\begin{aligned}
b_j w_{r,j}^{(t+1)} &= (1 - \eta\lambda) b_j w_{r,j}^{(t)} \\
&\qquad + \eta \cdot \frac{\operatorname{sign}(b_1 w_{r,1}^{(t)}) \operatorname{sign}(b_2 w_{r,2}^{(t)}) \cdots \operatorname{sign}(b_k w_{r,k}^{(t)})}{\operatorname{sign}(b_j w_{r,j}^{(t)})} \cdot \mathbb{1}\left[ \frac{|a_r^{(0)} w_{r,1}^{(t)} w_{r,2}^{(t)} \cdots w_{r,k}^{(t)}|}{|w_{r,j}^{(t)}|} \geq \frac{\rho}{k!} \right] \\
&= (1 - \eta\lambda) + \eta \cdot \mathbb{1}[k! \geq \rho] \\
&= 1,
\end{aligned}$$

where the last equality is by $\rho \leq k!(1-c) \leq k! \cdot |a_r^{(t)}|$. $\mathrm{H}_2(t), \mathrm{H}_5(t) \Longrightarrow \mathrm{H}_5(t+1), \mathrm{H}_6(t)$ can be verified in the same way as Lemma C.4 by noticing that $\rho \leq k!(1-c) \leq k! \cdot |a_r^{(t)}|$. $\mathrm{H}_3(t)$, $\mathrm{H}_4(t), \mathrm{H}_4(t+1), \mathrm{H}_6(t) \Longrightarrow \mathrm{H}_1(t)$ can be proved by following exactly the same proof as Lemma C.4. $\{\mathrm{H}_1(s)\}_{s=0}^t \Longrightarrow \mathrm{H}_2(t+1), \mathrm{H}_3(t+1)$ be verified in the same way as Lemma C.4 by noticing that $\epsilon_1 \cdot (1+c) \cdot d^{\frac{k-1}{2}} < \rho$ and $k! - \epsilon_1 \cdot (1+c) \cdot d^{\frac{k-1}{2}} > \rho$. □

Based on Lemma E.2 and Lemma E.3, we can get the following lemmas and theorems aligning with the result of the fixed second-layer case.

**Lemma E.4.** *For $T = \Theta(k\eta^{-1}\lambda^{-1}\log(d))$,with a probability of at least $1-\delta$ with respect to the randomness of the online data selection, it holds that*

$$\begin{aligned}
w_{r,j}^{(T)} &= b_j, &&\forall r \in \Omega_{b_1 b_1 \cdots b_k} \cap \Omega_{\mathrm{g}}, j \in [k], \\
|w_{r,j}^{(T)}| &\leq d^{-(k+1)}, &&\forall r \in \Omega_{\mathrm{g}}, j \in [d] \setminus [k], \\
|w_{r,j}^{(T)}| &\leq d^{-(k+1)}, &&\forall r \in \Omega_{\mathrm{b}}, j \in [d].
\end{aligned}$$

**Lemma E.5.** *With a probability of at least $1 - 2\delta$ with respect to the randomness in the neural network's initialization and the online data selection, trained neural network $f(\mathbf{W}^{(T)}, \mathbf{x})$ approximates accurate classifier $(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})$ well:*

$$\mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\mathbf{x}}}\left( \frac{f(\mathbf{W}^{(T)}, \mathbf{x})}{(m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})} \in [0.25, 1.75] \right) \geq 1 - \epsilon.$$

*Proof.* Let

$$\widetilde{f}(\mathbf{W}^{(T)}, \mathbf{x}) = \sum_{r=1}^m a_r^{(0)} \cdot \sigma(\langle \mathbf{w}_r^{(T)}, \mathbf{x} \rangle).$$

By the proof of Lemma C.7, we can get

$$\frac{|\widetilde{f}(\mathbf{W}^{(T)}, \mathbf{x}) - (m/2^{k+1}) \cdot f(\mathbf{W}^*, \mathbf{x})|}{(m/2^{k+1}) \cdot |f(\mathbf{W}^*, \mathbf{x})|} \leq 0.5.$$

To prove the result, we need to estimate the difference between $\widetilde{f}(\mathbf{W}^{(T)}, \mathbf{x})$ and $f(\mathbf{W}^{(T)}, \mathbf{x})$:

$$\begin{aligned}
\big| f(\mathbf{W}^{(T)}, \mathbf{x}) - \widetilde{f}(\mathbf{W}^{(T)}, \mathbf{x}) \big| &= \left| \sum_{r=1}^m (a_r^{(T)} - a_r^{(0)}) \cdot \sigma(\langle \mathbf{w}_r^{(T)}, \mathbf{x} \rangle) \right| \\
&\leq \sum_{r=1}^m |a_r^{(T)} - a_r^{(0)}| \cdot |\langle \mathbf{w}_r^{(T)}, \mathbf{x} \rangle|^k \\
&\leq c \sum_{r=1}^m |\langle \mathbf{w}_r^{(T)}, \mathbf{x} \rangle|^k \\
&= \underbrace{c \sum_{r \in \Omega_{\mathrm{g}}} |\langle \mathbf{w}_r^{(T)}, \mathbf{x} \rangle|^k}_{I_1} + \underbrace{c \sum_{r \in \Omega_{\mathrm{b}}} |\langle \mathbf{w}_r^{(T)}, \mathbf{x} \rangle|^k}_{I_2}
\end{aligned} \tag{E.7}$$

where the first inequality is by triangle inequality; the second inequality is by Lemma E.2. Then, we provide upper bounds for terms $I_1$ and $I_2$ respectively. For $I_1$, we have the following upper bound:

$$\begin{aligned}
I_1 &= \sum_{r\in\Omega_{\mathrm{g}}} |\langle \mathbf{w}_{r,[1:k]}^{(T)}, \mathbf{x}_{[1:k]}\rangle + \langle \mathbf{w}_{r,[k+1:d]}^{(T)}, \mathbf{x}_{[k+1:d]}\rangle|^k \\
&\le \sum_{r\in\Omega_{\mathrm{g}}} |\langle \mathbf{w}_{r,[1:k]}^{(T)}, \mathbf{x}_{[1:k]}\rangle|^k + \sum_{r\in\Omega_{\mathrm{g}}} |\langle \mathbf{w}_{r,[k+1:d]}^{(T)}, \mathbf{x}_{[k+1:d]}\rangle| \cdot k\big(|\langle \mathbf{w}_{r,[1:k]}^{(T)}, \mathbf{x}_{[1:k]}\rangle| + |\langle \mathbf{w}_{r,[k+1:d]}^{(T)}, \mathbf{x}_{[k+1:d]}\rangle|\big)^k \\
&\le \sum_{(b_1,\cdots,b_k)\in\{\pm 1\}^k} \sum_{r\in\Omega_{b_1b_2\cdots b_k}\cap\Omega_{\mathrm{g}}} \bigg|\sum_{j=1}^k b_j x_j\bigg|^k + \sum_{r\in\Omega_{\mathrm{g}}} d^{-k}\cdot k\big(k+d^{-k}\big)^k \\
&\le \sum_{(b_1,\cdots,b_k)\in\{\pm 1\}^k} \Big(\frac{1+\alpha}{2^{k+1}}\Big)m\cdot\bigg|\sum_{j=1}^k b_j x_j\bigg|^k + \Big(\frac{1+\alpha}{2}\Big)m\cdot d^{-k}\cdot k\big(k+d^{-k}\big)^k \\
&\le \Big(\frac{1+\alpha}{2^{k+1}}\Big)m\cdot 2k^k(1+e^{-2})^k + \Big(\frac{1+\alpha}{2}\Big)m\cdot d^{-k}\cdot k\big(k+d^{-k}\big)^k, \qquad \text{(E.8)}
\end{aligned}$$

where the first inequality is by mean value theorem; the second inequality is by (C.2) and Lemma E.4; the third inequality is by Lemma B.1; the last inequality is by Lemma F.3. For $I_2$, we have the following upper bound

$$I_2 \le |\Omega_{\mathrm{b}}|\cdot(2d^{-k})^k \le \Big(\frac{1+\alpha}{2}\Big)m\cdot(2d^{-k})^k. \qquad \text{(E.9)}$$

By plugging (E.8) and (E.9) into (E.7), we can get:

$$\begin{aligned}
&\big|f(\mathbf{W}^{(T)},\mathbf{x}) - \widetilde{f}(\mathbf{W}^{(T)},\mathbf{x})\big| \\
&\le c\big((1+\alpha)2^{-k}m\cdot k^k(1+e^{-2})^k + 0.5(1+\alpha)m\cdot d^{-k}\cdot k\big(k+d^{-k}\big)^k + 0.5(1+\alpha)m\cdot(2d^{-k})^k\big).
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\frac{\big|f(\mathbf{W}^{(T)},\mathbf{x}) - \widetilde{f}(\mathbf{W}^{(T)},\mathbf{x})\big|}{(m/2^{k+1})\cdot|f(\mathbf{W}^*,\mathbf{x})|} \\
&\le c\cdot\frac{(1+\alpha)2^{-k}m\cdot k^k(1+e^{-2})^k + 0.5(1+\alpha)m\cdot d^{-k}\cdot k\big(k+d^{-k}\big)^k + 0.5(1+\alpha)m\cdot(2d^{-k})^k}{0.5k!\cdot m} \\
&\le c\cdot\frac{2(1+\alpha)2^{-k}\cdot k^k(1+e^{-2})^k + (1+\alpha)\cdot d^{-k}\cdot k\big(k+d^{-k}\big)^k + (1+\alpha)\cdot(2d^{-k})^k}{\sqrt{2\pi k}(k/e)^k} \\
&= c\cdot\left(\sqrt{\frac{2}{\pi k}}(1+\alpha)\Big(\frac{e+e^{-1}}{2}\Big)^k + \sqrt{\frac{2}{\pi k}}\cdot\left(1+\frac{d^{-k}}{k}\right)^{k-1}\cdot\Big(\frac{e}{d}\Big)^k + \sqrt{\frac{2}{\pi k}}\cdot\Big(\frac{2e}{kd^k}\Big)^k\right) \\
&\le c\cdot\sqrt{\frac{8}{\pi k}}\Big(\frac{e+e^{-1}}{2}\Big)^k \\
&= \frac{1}{4},
\end{aligned}$$

where the second inequality is by Stirling's approximation, and the last equality is due to $c = \frac{1}{4}\sqrt{\frac{\pi k}{8}}\big(\frac{e+e^{-1}}{2}\big)^{-k}$. □

## F Auxiliary Lemmas

We first introduce a finite difference operator with step $h$ and order $n$ as follows,

$$\Delta_h^n[f](x) = \sum_{i=0}^n \binom{k}{i}(-1)^{n-i}f(x+ih).$$

The following Lemma calculates the value finite difference operating on the polynomial function.

**Lemma F.1.** *(Milne-Thomson, 2000) For $f(x) = x^n$, we have that $\Delta_h^n[f](x) = h^n n!$.*

Based on Lemma F.1, we have the following Lemma, which calculates the margin of *good* NNs defined in (3.1).

**Lemma F.2.** *For any integer $k$, we have*

$$\sum_{i=0}^{k} \binom{k}{i} (-1)^i (k-2i)^k = 2^k k!. \tag{F.1}$$

*Proof.* Applying Lemma F.1 with $f(x) = x^k$, $n = k$ in Lemma F.1 gives,

$$\sum_{i=0}^{k} \binom{k}{i} (-1)^i (k-2i)^k = (-1)^k \Delta_{-2}^k [f](k) = (-1)^k (-2)^k k! = 2^k k!,$$

where the first equality is due to the definition of finite difference operator and the last equality is due to Lemma F.1. □

**Lemma F.3.** *For any positive integer $k$, it holds that*

$$\sum_{i=0}^{k} \binom{k}{i} |k-2i|^k \le 2k^k (1+e^{-2})^k.$$

*Proof.* We can establish the following inequality:

$$\begin{aligned}
\sum_{i=0}^{k} \binom{k}{i} |k-2i|^k &\le 2 \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} \binom{k}{i} |k-2i|^k \\
&= 2k^k \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} \binom{k}{i} \Big(1 - \frac{2i}{k}\Big)^k \\
&\le 2k^k \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} \binom{k}{i} \exp(-2i) \\
&\le 2k^k \sum_{i=0}^{k} \binom{k}{i} \exp(-2i) \\
&= 2k^k (1+e^{-2})^k,
\end{aligned}$$

where the first inequality is by $\binom{k}{i} = \binom{k}{k-i}$, the second inequality is by $1 - t \le \exp(-t), \forall t \in \mathbb{R}$, the last inequality is by $\binom{k}{i} \exp(-2i) > 0$. □

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in abstract and Section 1 are supported by results in Section 4 and also the additional expeiremnts in the Appendix.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitation section is presented in page 12.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Yes, the full set of assumptions are provided in Section 3 and the complete (and correct) proof are provided in the Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper provides detailed experiment settings in Section A, which should be sufficient for reproducing the main experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The experiments in this paper are based on synthetic data generated to support the theoretical findings. As such, there is no real-world dataset or code to be made openly accessible. The question of open access to data and code is not applicable in this case, as the main contributions are purely theoretical and do not rely on empirical results from specific datasets.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper provides detailed experiment settings in Section A, which should be sufficient for reproducing the main experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: : This paper provides detailed experiment settings in Section A, which include important details such as the error bars reported in Table 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The requirement of Compute Resources are stated in Section A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: As a theoretical paper, the research presented in this paper has been conducted with the highest ethical standards and is fully compliant with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work provides the theoretical understanding of generic optimization algorithm (SGD). Although there might be some potential social impacts on applications, according to the guidelines, we believe our result does not have a direct connection with these issues.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: As a theoretical paper, this work does not involve the development or release of any models or datasets, particularly those that might be considered high-risk for misuse, such as pretrained language models, image generators, or scraped datasets. Therefore, the question of safeguards for responsible release does not apply.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: This paper is a theoretical work that does not utilize or rely on any existing assets such as code, data, or models from other sources. Therefore, the question of crediting creators or detailing licenses and terms of use for such assets does not apply.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: As this is a theoretical paper, it does not introduce or release any new assets such as datasets, code, or models. Thus, there is no need for documentation related to new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This theoretical paper does not conduct any experiments involving crowdsourcing or research with human subjects, thus there are no participants, instructions, or compensation details to report.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: As this paper is purely theoretical and does not involve any research with human subjects, there are no study participants, thus no associated risks or requirements for Institutional Review Board (IRB) approvals.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.