Implicit Bias of Adam versus Gradient Descent in One-Hidden-Layer Neural Networks

Bhavya Vasudeva **Vatsal Sharan** **Mahdi Soltanolkotabi**
University of Southern California
{bvasudev, vsharan, soltanol}@usc.edu

Abstract

Adam is the de facto optimization algorithm for training deep neural networks, but understanding its implicit bias and how it differs from other algorithms, particularly standard gradient descent (GD), remains limited. We investigate the differences in the implicit biases of Adam and GD when training one-hidden-layer ReLU neural networks on a binary classification task using a synthetic data setting with diverse features. We find that GD exhibits a simplicity bias, resulting in a linear decision boundary, whereas Adam leverages diverse features, producing a nonlinear boundary that is closer to the Bayes optimal predictor. We theoretically prove this for a simple data setting in the infinite width regime by analyzing the population gradients. Our results offer important insights towards improving the understanding of Adam, which can aid the design of optimization algorithms with superior generalization.

1 Introduction

Adaptive optimization algorithms, particularly Adam [Kingma and Ba, 2015], have become ubiquitous in training deep neural networks due to their faster convergence rates and better performance, particularly on large language models (LLMs), as compared to (stochastic) gradient descent (SGD) [Zhang et al., 2019]. Despite its widespread use, the theoretical understanding of how Adam works and why it often outperforms (S)GD remains limited.

A large body of work on similar aspects of GD analyzes the training dynamics and parameter convergence of simple models. For instance, Soudry et al. [2017] show that when optimizing the logistic loss of a linear model for binary classification on linearly separable data, GD updates converge, in direction, to the max-margin or minimum ℓ_2-norm solution. This *implicit* preference of an algorithm towards a particular solution in the presence of multiple solutions which attain zero training error and/or loss is known as its implicit bias. The implicit bias of GD and its variants is well-studied in the literature [Soudry et al., 2017, Gunasekar et al., 2018, Wu et al., 2021, Ji and Telgarsky, 2019], for both linear models and other architectures like NNs and attention models (see Section 5 for a detailed discussion). However, there is limited work investigating the implicit bias of Adam. Recently, Zhang et al. [2024] showed that in a similar setting as Soudry et al. [2017], Adam iterates converge in direction, to the minimum ℓ_∞-norm solution. This difference in the implicit bias of the two algorithms for linear models motivates a similar investigation for NNs.

In this work, we aim to characterize the implicit bias of Adam and investigate how it differs from GD when training one-hidden-layer neural networks (NNs) with ReLU activation on a binary classification task using synthetic data. Our main contributions are:

- We identify a simple yet informative setting with Gaussian features where GD and Adam exhibit different implicit biases. The Bayes optimal predictor in this setting has a nonlinear decision

Mathematics of Modern Machine Learning Workshop at NeurIPS 2024.

boundary, and we observe that while GD exhibits simplicity bias, resulting in a linear predictor, Adam encourages reliance on diverse features, leading to a nonlinear decision boundary.

- We theoretically prove this difference in the implicit bias in the infinite width limit, analyzing the population gradients and updates. Specifically, one-hidden layer NNs trained with GD asymptotically converge to a linear predictor, while those trained with Adam with momentum parameters $\beta_1 = \beta_2 = 0$ (also known as signGD) converge to piece-wise linear predictors.
- We also analyze a simpler setting with variance $\to 0$, where we show that the decision boundaries learned with Adam with $\beta_1 = \beta_2 \approx 0$ or $\beta_1 = \beta_2 \approx 1$ are more nonlinear than the one learned with GD.
- Empirically, we verify that Adam and GD exhibit different implicit biases across various settings, with Adam outperforming GD in terms of test accuracy or generalization.

2 Setup

We consider a binary classification task using a one-hidden layer neural network with fixed final layer and ReLU activation, defined as:

$$f(\boldsymbol{W}; \boldsymbol{x}) := \boldsymbol{a}^\top \sigma(\boldsymbol{W}\boldsymbol{x}),$$

where $\boldsymbol{x} \in \mathbb{R}^d$ denotes the input, $\boldsymbol{W} \in \mathbb{R}^{m \times d}$ denotes the trainable parameters, $\boldsymbol{a} \in \{\pm 1\}^m$, and $\sigma(\cdot) = \max(0, \cdot)$. Let $S := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ denote the set of train samples. The model is trained to minimize the empirical risk written as:

$$\widehat{L}(\boldsymbol{W}) := \frac{1}{n} \sum_{i=1}^{n} \ell(-y_i f(\boldsymbol{W}; \boldsymbol{x}_i)),$$

where ℓ denotes a decreasing loss function. We consider two loss functions, namely logistic loss, where $\ell(z) := \log(1 + \exp(z))$, and correlation or linear loss, where $\ell(z) := z$, for $z \in \mathbb{R}$. We focus on the following two update rules.

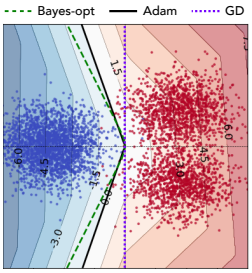

Figure 1: Illustration of the synthetic dataset considered in this work, and comparison of the Bayes optimal predictor with the decision boundaries learned by one-hidden-layer NNs trained with Adam and GD.

Gradient Descent. The updates for GD with step-size $\eta > 0$ at iteration $t \geq 0$ are written as

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \boldsymbol{G}_t, \text{ where } \boldsymbol{G}_t := \nabla_{\boldsymbol{W}} \widehat{L}(\boldsymbol{W}_t),$$

and each row of which is written as:

$$\boldsymbol{g}_{j,t} = -\frac{1}{n} \sum_{i=1}^{n} \ell'_{i,t} y_i \nabla_{\boldsymbol{w}_j} f(\boldsymbol{W}_t; \boldsymbol{x}_i) = -\frac{1}{n} \sum_{i=1}^{n} \ell'_{i,t} a_j \sigma'(\boldsymbol{w}_{j,t}^\top \boldsymbol{x}_i)(y_i \boldsymbol{x}_i),$$

where $\ell'_{i,t}$ denotes $\ell'(-y_i f(\boldsymbol{W}_t, \boldsymbol{x}_i))$ for convenience, and $\sigma'(\cdot) := \mathbb{1}[\cdot \geq 0]$.

Adam. The update rule for the Adam optimizer [Kingma and Ba [2015]] is as follows:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \hat{\boldsymbol{M}}_t \odot \hat{\boldsymbol{V}}_t^{\circ -1/2},$$

where $\hat{\boldsymbol{M}}_t = \frac{\boldsymbol{M}_{t+1}}{1-\beta_1^{t+1}} = \frac{1}{1-\beta_1^{t+1}} \Big(\beta_1 \boldsymbol{M}_t + (1-\beta_1) \boldsymbol{G}_t \Big)$ is the bias-corrected first moment estimate, and $\hat{\boldsymbol{V}}_t = \frac{\boldsymbol{V}_{t+1}}{1-\beta_2^{t+1}} = \frac{1}{1-\beta_2^{t+1}} \Big(\beta_2 \boldsymbol{V}_t + (1-\beta_2) \boldsymbol{G}_t \odot \boldsymbol{G}_t \Big)$ is the bias-corrected second (raw) moment estimate. Also, we set the stability constant $\epsilon = 0$, $\odot$ and $(\cdot)^\circ$ denote the Hadamard product and power, respectively, and $\boldsymbol{M}_0$ and $\boldsymbol{V}_0$ are initialized as zeroes. Note that we can write

$$\hat{\boldsymbol{M}}_t = \frac{\sum_{\tau=0}^{t} \beta_1^\tau \boldsymbol{G}_{t-\tau}}{\sum_{\tau=0}^{t} \beta_1^\tau} \quad \text{and} \quad \hat{\boldsymbol{V}}_t = \frac{\sum_{\tau=0}^{t} \beta_2^\tau \boldsymbol{G}_{t-\tau} \odot \boldsymbol{G}_{t-\tau}}{\sum_{\tau=0}^{t} \beta_2^\tau}.$$

At each optimization step, the descent direction is different from the gradient direction because of the second (raw) moment in the denominator. Further, the first update step exactly matches the update of signGD, since $\hat{\boldsymbol{M}}_0 = \boldsymbol{G}_0$ and $\hat{\boldsymbol{V}}_0 = \boldsymbol{G}_0 \odot \boldsymbol{G}_0$, and hence $(\hat{\boldsymbol{M}}_0 \odot \hat{\boldsymbol{V}}_0^{\circ -1/2})_{i,j} = \frac{(G_0)_{i,j}}{|(G_0)_{i,j}|} = \texttt{sign}((G_0)_{i,j})$. Similarly, when the parameters β_1 and β_2 are set as 0, the Adam updates are the same as signGD for every $t \geq 0$.

Dataset. Our synthetic dataset is designed to investigate the impact of feature diversity on the implicit biases of optimization algorithms in NN training. It models two classes with differing feature distributions to emulate real-world scenarios where feature complexity may vary between classes. See Fig. 1 for an illustration of the dataset. Concretely, each sample $(\boldsymbol{x}, y)$ is generated as follows:

$$y \sim \mathrm{Unif}(\{\pm 1\}), \quad \epsilon \sim \mathrm{Unif}(\{\pm 1\}) \tag{1}$$
$$x_1 \sim \mathcal{N}\left(\tfrac{\mu_1-\mu_3}{2} + y\tfrac{\mu_1+\mu_3}{2}, \sigma_x^2\right), \quad x_2 \sim \mathcal{N}\left(\epsilon\left(\tfrac{y+1}{2}\right)\mu_2, \sigma_y^2\right), \quad x_j \sim \mathcal{N}(0, \sigma_z^2), \forall j \in \{3, \ldots, d\}.$$

Our dataset construction is inspired by the synthetic "slabs" dataset introduced by Shah et al. [2020]. While their approach utilizes slab features to represent non-linearly separable components, we consider Gaussian features instead. This modification enhances the realism of the synthetic data and facilitates a more nuanced analysis of the NN training dynamics.

We first write the Bayes optimal predictor for this dataset as follows.

Proposition 1 (Bayes Optimal Predictor)**.** *The optimal predictor for the data in Eq. (1) with $d=2$ is:*

$$(\mu_1+\mu_3)x_1 + \tfrac{\sigma_x^2}{\sigma_y^2}\mu_2 x_2 = \tfrac{\mu_1^2-\mu_3^2}{2} + \tfrac{\mu_2^2\sigma_x^2}{2\sigma_y^2} - \sigma_x^2 \log\left(0.5\left(1+\exp\left(-\tfrac{2\mu_2 x_2}{\sigma_y^2}\right)\right)\right).$$

Since the NN we consider does not have a bias parameter, we make the following assumption on the data generating process to make the setting realizable, *i.e.*, ensure that the Bayes optimal predictor passes through the origin.

Assumption 1 (Realizability)**.** *Let $\mu := \mu_2$, $\kappa := \frac{\sigma_x^2}{\sigma_y^2}$ $\omega := \frac{\mu_1+\mu_3}{\kappa\mu} \geq 1$. For realizability, $\mu_1 = \frac{\mu}{2}\left(\kappa\omega - \frac{1}{\omega}\right)$ and $\mu_3 = \frac{\mu}{2}\left(\kappa\omega + \frac{1}{\omega}\right)$.*

3 Theoretical Analyses

In this section, we aim to theoretically analyze GD and Adam and the differences in the learned solution arising from the update rules. We study a simple setting to keep the analysis as clean as possible. Specifically, in this section, we consider the infinite sample and infinite width limit, with $d = 2$, fixed outer layer weights, and training with correlation or linear loss. As we will see later in Section 4, these algorithms learn different solutions even when these assumptions are relaxed.

3.1 Gaussian Data

For correlation loss, we can write the closed form of the population gradient for Gaussian data as follows.

Proposition 2 (Population Gradient)**.** *Consider the data in Eq. (1) with $d = 2$ and $\sigma_x = \sigma_y = \sigma$. The population gradient for a certain $\boldsymbol{w}$ is written as:*

$$\nabla_{\boldsymbol{w}}\widehat{L}(\boldsymbol{W}) = -a\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathbb{1}[\boldsymbol{w}^\top\boldsymbol{x} \geq 0]y\boldsymbol{x}]$$
$$= -\tfrac{a\sigma}{4}\Big(\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}})\lambda\bar{\boldsymbol{\mu}}_+ + \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}})\lambda\bar{\boldsymbol{\mu}}_- - 2\Phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}})\lambda\bar{\boldsymbol{\mu}}_0 + \left(\phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}) + \phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}) - 2\phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}})\right)\bar{\boldsymbol{w}}\Big),$$

where $\lambda := \frac{\mu}{\sigma}\frac{\omega^2+1}{2\omega}$ $\bar{\boldsymbol{\mu}}_+ := \left[\frac{\omega^2-1}{\omega^2+1}, \frac{2\omega}{\omega^2+1}\right]^\top$, $\bar{\boldsymbol{\mu}}_- := \left[\frac{\omega^2-1}{\omega^2+1}, -\frac{2\omega}{\omega^2+1}\right]^\top$, $\bar{\boldsymbol{\mu}}_0 := [-1, 0]^\top$, and ϕ and Φ denote the normal PDF and CDF, respectively.

The proof is included in the Appendix. We first use the above gradient expression to analyze GD iterates showing that they exhibit simplicity bias and learn a linear predictor.

Theorem 1. *(Informal) Consider the data in Eq. (1), neurons initialized such that $a_k = \pm 1$ with probability 0.5, small learning rate, and $\omega > c$, $\frac{\mu}{\sigma} \geq c_1$, where c, c_1 are constants. Let $\boldsymbol{w}_{k,\infty} := \lim_{t\to\infty}\frac{\boldsymbol{w}_{k,t}}{t}$ and $\bar{\boldsymbol{w}}_{k,\infty} := \frac{\boldsymbol{w}_{k,\infty}}{\|\boldsymbol{w}_{k,\infty}\|}$, for $k \in [m]$. Then, the solution learned by GD is:*

$$\bar{\boldsymbol{w}}_{k,\infty} = a_k[1, 0]^\top.$$

The proof is included in the Appendix. Next, we analyze Adam with $\beta_1 = \beta_2 = 0$ (signGD), and show that it learns both features resulting in a nonlinear predictor.

Theorem 2. *(Informal) Consider the data in Eq. (1), neurons initialized such that $a_k = \pm 1$ with probability 0.5, small learning rate, $\omega > c$ and $c_1 \le \frac{\mu}{\sigma} \le c_2$, where c, c_1, c_2 are constants. Let $\boldsymbol{w}_{k,\infty} := \lim_{t\to\infty} \frac{\boldsymbol{w}_{k,t}}{t}$ and $\bar{\boldsymbol{w}}_{k,\infty} := \frac{\boldsymbol{w}_{k,\infty}}{\|\boldsymbol{w}_{k,\infty}\|}$, for $k \in [m]$. Let θ_0 denote the direction of $\boldsymbol{w}_{k,0}$. Then, the solution learned by signGD is:*

$$\bar{\boldsymbol{w}}_{k,\infty} = \begin{cases} [1,1]^\top & a_k > 0, \sin\theta_{k,0} > 0, \\ [1,-1]^\top & a_k > 0, \sin\theta_{k,0} < 0, \\ [1,0]^\top & a_k > 0, \sin\theta_{k,0} = 0, \\ [-1,0]^\top & a_k < 0. \end{cases}$$

These results characterize the direction in which each neuron converges asymptotically. For GD, all neurons are in the same direction, with exactly half the neurons in $[1,0]^\top$ and $[-1,0]^\top$ directions, which leads to a linear predictor. In contrast, for Adam (no momentum), there is a fraction of neurons aligned in the directions $\frac{1}{\sqrt{2}}[1,1]^\top$ and $\frac{1}{\sqrt{2}}[1,-1]^\top$ which leads to a piece-wise linear decision boundary. This can also be seen in Fig. 1, where we consider $\beta_1 = \beta_2 \approx 1$ in the finite sample, finite width setting, with non-zero variance.

Analyzing this setting allows us to conceptually understand how Adam (without momentum) operates and leads to diverse feature learning, while GD exhibits simplicity bias. Importantly, we make no assumptions regarding the initialization direction of the neural network parameters, ensuring that any differences observed between Adam and GD arise solely from the inherent characteristics of the optimization algorithms themselves.

Next, we show that under some conditions, the piece-wise linear predictor obtains a strictly lower test error than the linear predictor learned by GD. The proof is included in the Appendix.

Theorem 3. *(Informal) Consider the data in Eq. (1) with $d = 2$ and $\sigma_x = \sigma_y = \sigma$, $\omega = \Theta(1)$ and $c_1\omega \le \frac{\mu}{\sigma} \le c_2$, where c_1, c_2 are constants. Consider two predictors,*

$$\textit{Linear: } \hat{y} = \texttt{sign}(x_1), \textit{ Piece-wise Linear: } \hat{y}' = \begin{cases} \texttt{sign}(x_1 + x_2) & x_2 \ge 0, \\ \texttt{sign}(x_1 - x_2) & x_2 < 0. \end{cases}$$

Then, it holds that $\mathbb{E}(\hat{y}' \ne y) - \mathbb{E}(\hat{y} \ne y) < 0$.

In the next section, we consider a simplified setting to investigate the effect of setting $\beta_1, \beta_2 \approx 1$ for Adam.

3.2 Toy Data Setting

We consider an extremely simple yet informative setting where $\sigma_x = \sigma_y = 0$, which we refer to as the toy data setting. Specifically, the samples are generated as follows:

$$y \sim \mathrm{Unif}(\{\pm 1\}), \quad \epsilon \sim \mathrm{Unif}(\{\pm 1\}) \quad x_1 = \frac{\mu}{2}\left(y\omega - \frac{1}{\omega}\right), \quad x_2 = \epsilon\frac{y+1}{2}\mu. \tag{2}$$

This setting allows us to characterize the full trajectory of each neuron for the three algorithms. We now state our main result.

Theorem 4. *(Informal) Consider the toy data in Eq. (2), neurons initialized at a small scale, and $c_1 < \omega < c_2$, where c_1, c_2 are constants. Let $\boldsymbol{w}_{k,\infty} := \lim_{t\to\infty} \frac{\boldsymbol{w}_{k,t}}{t}$ and $\bar{\boldsymbol{w}}_{k,\infty} := \frac{\boldsymbol{w}_{k,\infty}}{\|\boldsymbol{w}_{k,\infty}\|}$, for $k \in [m]$ and $p := \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi}$. Then, for $m \to \infty$, the solutions learned by GD, signGD, and Adam are:*

	GD		Adam ($\beta_1 = \beta_2 = 0$) or signGD		Adam ($\beta_1 = \beta_2 \approx 1$)	
$\bar{\mathbf{w}}_{k,\infty} =$	$[1,0]^\top$	w.p. $\frac{1}{4} + \frac{p}{2}$	$[1,0]^\top$	w.p. p	$[1,0]^\top$	w.p. p
	$[-1,0]^\top$	w.p. $\frac{1}{2}$	$[-1,0]^\top$	w.p. $\frac{1}{2}$	$[-1,0]^\top$	w.p. $\frac{1}{2}$
	$\frac{1}{\omega^2+1}[\omega^2-1, 2\omega]^\top$	w.p. $\frac{1}{8} - \frac{p}{4}$	$\frac{1}{\sqrt{2}}[1,1]^\top$	w.p. $\frac{1}{4} - \frac{p}{2}$	$\frac{1}{\sqrt{2}}[1,1]^\top$	w.p. $\frac{1}{8} - \frac{p}{4}$
	$\frac{1}{\omega^2+1}[\omega^2-1, -2\omega]^\top$	w.p. $\frac{1}{8} - \frac{p}{4}$	$\frac{1}{\sqrt{2}}[1,-1]^\top$	w.p. $\frac{1}{4} - \frac{p}{2}$	$\frac{1}{\sqrt{2}}[1,-1]^\top$	w.p. $\frac{1}{8} - \frac{p}{4}$
					$\frac{1}{\sqrt{s^2+1}}[s,1]^\top$	w.p. $\frac{1}{8} - \frac{p}{4}$
					$\frac{1}{\sqrt{s^2+1}}[s,-1]^\top$	w.p. $\frac{1}{8} - \frac{p}{4}$

where s is a constant $\in [0.72, 1]$, the probabilities are over the neurons, and the sign of the first element of $\boldsymbol{w}_{k,\infty}$ is the same as $\texttt{sign}(a_k)$.

The proof mainly relies on analyzing the updates of each algorithm, so we defer it to the Appendix.

We note that there is a larger difference in the learned predictors in the Gaussian settings compared to the toy dataset. The main reason is that in the toy dataset, there is a larger region where the gradients are in the $[1, 0]^\top$ direction, which makes the decision boundary for signGD more linear, as well as a larger region where the neurons are only active for one of $\boldsymbol{\mu}_+$ or $\boldsymbol{\mu}_-$, which makes the decision boundary for GD less linear.

4 Experimental Results

In this section, we present experimental results showing that GD and Adam exhibit different implicit biases. We consider Adam with momentum parameters $\beta_1 = \beta_2 = 0.9999$ in this section, although the results generalize to other values as well. Throughout, we consider a small initialization scale and fix the outer layer weights. Specifically, $\boldsymbol{w}_k \sim \mathcal{N}(0, \frac{\alpha}{\sqrt{d}})$, and $a_k = \pm\frac{1}{\sqrt{m}}$ for $k \in [m]$, where α is a small constant.

We consider the Gaussian data in Eq. (1) in this section, focusing on two main settings. First, we consider the *population setting* which is closer to the setting considered for the theoretical analysis. Specifically, we use correlation loss and population gradients for the training updates. The second setting is the *finite sample setting*, which is closer to practice as we use finite samples and train with the binary cross-entropy loss.

Population Setting. We consider $d = 2$ in this setting and use the closed form of the population gradient in Proposition 2. Fig. 2 shows the evolution of the decision boundary for one-hidden-layer NNs trained with GD and Adam as a function of the training epochs. It also shows the trajectory of the neurons as training progresses. The observed results align with the theoretical analysis: for GD, the neurons are aligned in a single direction and lead to a linear decision boundary, while for Adam, the presence of neurons in the $[1, 1]^\top$ and $[1, -1]^\top$ directions leads to a non-linear decision boundary, which is closer to the Bayes optimal predictor. Further, the test accuracy of Adam is 0.55% more than that of GD in this case.

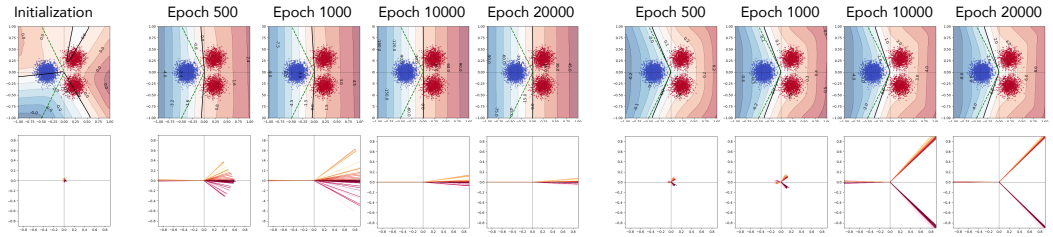

Figure 2: Evolution of the decision boundary and the neurons over time, for GD (left) and Adam (right) with learning rates 0.1 and 10^{-4} over $20\,000$ epochs of training a width 100 NN with population gradients (the samples are plotted for illustration purposes) on the Gaussian data setting (Eq. (1)) with $\mu = 0.3, \omega = 2, \sigma = 0.1$, and $\alpha = 0.02$. GD leads to a linear decision boundary, with neurons mostly aligned with the direction $[1, 0]^\top$, while Adam leads to a non-linear decision boundary, with neurons aligned with three main directions $[-1, 0]^\top, [1, 1]^\top, [1, -1]^\top$.

Finite Sample Setting. Fig. 1 compares the decision boundaries learned by Adam and GD in the finite sample setting, with the Bayes optimal predictor (for the population version of this setting). We set $n = 5000$, $m = 1000$, $\mu = 0.3, \omega = 2, \sigma_x = 0.2, \sigma_y = 0.15$, $\alpha = 0.001$, and use learning rates 0.1 and 10^{-4} for GD and Adam, respectively. These results are similar to the population setting and show that the difference in the implicit bias of Adam and GD is quite robust to the choice of the training setting. For comparable train loss, the test accuracy of Adam is 0.32% more than that of GD in this case. We also find that reducing μ increases the accuracy gap: repeating the same experiment with $\mu = 0.25$ leads to a gap of 0.595%. These results also generalize to settings where $d > 2$: with $m = 500$, $d = 20$ and $\mu = 0.25$, the gap is 0.203%.

5 Related Work

Implicit Bias of GD. Since the pioneering studies that identified the implicit bias of linear classifiers on separable datasets [Soudry et al., 2018], extensive research has been conducted on the implicit bias of gradient-based methods for linear models, NNs, and even self-attention models. Wang et al. [2024] shows that GD with momentum exhibits the same implicit bias for linear models trained on separable data as vanilla GD. Nacson et al. [2019], Ji and Telgarsky [2021], Ji et al. [2021] demonstrate fast convergence (in direction) of GD-based approaches with adaptive step-sizes to the ℓ_2 max-margin predictor. It has also been shown that multilayer perceptrons (MLPs) trained with exponentially tailed loss functions on classification tasks, GD or gradient flow converge in direction to the KKT points of the max-margin problem in both finite [Ji and Telgarsky, 2020, Lyu and Li, 2020] and infinite-width [Chizat and Bach, 2020] networks. Additionally, Phuong and Lampert [2021], Frei et al. [2022], Kou et al. [2023] analyze the implicit bias of ReLU and Leaky-ReLU networks trained with GD on orthogonal data. Other studies focus on the implicit bias to minimize rank in regression tasks using squared loss [Vardi and Shamir, 2021, Arora et al., 2019, Li et al., 2021]. The recent survey Vardi [2022] includes a comprehensive review of related work. More recently, Tarzanagh et al. [2023b,a] studied single-head prompt and self-attention models with fixed linear decoder and characterize the implicit bias of attention weights trained with GD to asymptotically converge to the solution that separates the token with the largest similarity with the linear decoder from the rest, for each sample. Vasudeva et al. [2024] study the self-attention model trained with GD with adaptive step-sizes and show fast, global convergence under some conditions.

Simplicity Bias of NNs Trained with GD. Kalimeris et al. [2019] conduct a set of experiments to demonstrate that NNs trained with SGD first learn to make predictions that are highly correlated with those of the best possible linear predictor for the task, and only later start to use more complex features to achieve further performance improvement. Shah et al. [2020] created synthetic datasets and show that in the presence of 'simple' and 'complex' features (linearly separable vs non-linearly separable), (two-layer) NNs trained with SGD rely heavily on 'simple' features even when they have equal or even slightly worse predictive power than the 'complex' features. They also show that using SGD leads to learning small-margin and feature-impoverished classifiers, instead of large-margin and feature-dense classifiers, even on convergence, which contrasts with Kalimeris et al. [2019].

Implicit Bias of Adam and Other Adaptive Algorithms. Wang et al. [2021] shows that homogeneous NNs trained with RMSprop or Adam without momentum (signGD) converge to a KKT point of the ℓ_2 max-margin problem, similar to GD, while AdaGrad has a different implicit bias. Zhang et al. [2024] shows that linear models trained on separable data with Adam converge to the ℓ_∞ max-margin solution. Xie and Li [2024] analyze loss minimization with AdamW and show that under some conditions, it converges to a KKT point of the ℓ_∞-norm constrained loss minimization.

Adam vs. (S)GD. There is also some recent work on understanding when Adam generalizes better or worse than (S)GD. Particularly, SGD is known to have better generalization for image datasets, while Adam is known to perform better on language datasets. Zhou et al. [2020] shows that SGD converges to flatter minima while Adam converges to sharper minima. Zou et al. [2023] study an image-inspired dataset and show that CNNs trained with GD can generalize better than Adam. Ma et al. [2023] show that adding noise to lower or higher frequency components of the data can lead to lower or higher robustness of Adam compared to GD. For language data, Kunstner et al. [2024] show that the performance of SGD deteriorates under imbalanced classes especially when they constitute a significant part of the data, whereas Adam is less sensitive and performs better.

6 Conclusion and Future Work

In this work, we investigate the implicit bias of GD and Adam when training one-hidden-layer ReLU NNs on a binary classification task. The synthetic dataset models settings where different classes may have different feature distributions, and we find that GD exhibits simplicity bias while Adam leads to more diverse feature learning. Through theoretical and empirical results, our work adds to the conceptual understanding of how Adam works. It also poses important directions for future work. For instance, it would be interesting to extend the theoretical analysis to $d > 2$ and other loss functions like cross-entropy.

References

S. Arora, N. Cohen, W. Hu, and Y. Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 6

J. Burkardt. The truncated normal distribution. https://people.sc.fsu.edu/~jburkardt/presentations/truncated_normal.pdf, 2023. Department of Scientific Computing, Florida State University. 9

L. Chizat and F. Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020. 6

S. Frei, G. Vardi, P. L. Bartlett, N. Srebro, and W. Hu. Implicit bias in leaky relu networks trained on high-dimensional data. *arXiv preprint arXiv:2210.07082*, 2022. 6

S. Gunasekar, J. Lee, D. Soudry, and N. Srebro. Characterizing implicit bias in terms of optimization geometry. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/gunasekar18a.html. 1

Z. Ji and M. Telgarsky. The implicit bias of gradient descent on nonseparable data. In A. Beygelzimer and D. Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1772–1798. PMLR, 25–28 Jun 2019. URL https://proceedings.mlr.press/v99/ji19a.html. 1

Z. Ji and M. Telgarsky. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33:17176–17186, 2020. 6

Z. Ji and M. Telgarsky. Characterizing the implicit bias via a primal-dual analysis. In *Algorithmic Learning Theory*, pages 772–804. PMLR, 2021. 6

Z. Ji, N. Srebro, and M. Telgarsky. Fast margin maximization via dual acceleration. In *International Conference on Machine Learning*, pages 4860–4869. PMLR, 2021. 6

D. Kalimeris, G. Kaplun, P. Nakkiran, B. Edelman, T. Yang, B. Barak, and H. Zhang. Sgd on neural networks learns functions of increasing complexity. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 6

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, abs/1412.6980, 2015. 1, 2

Y. Kou, Z. Chen, and Q. Gu. Implicit bias of gradient descent for two-layer relu and leaky relu networks on nearly-orthogonal data, 2023. 6

F. Kunstner, R. Yadav, A. Milligan, M. Schmidt, and A. Bietti. Heavy-tailed class imbalance and why adam outperforms gradient descent on language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=T56j6aV8Oc. 6

Z. Li, Y. Luo, and K. Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning, 2021. 6

K. Lyu and J. Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020. 6

A. Ma, Y. Pan, and A. massoud Farahmand. Understanding the robustness difference between stochastic gradient descent and adaptive gradient methods. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=ed8SkMdYFT. Featured Certification. 6

M. S. Nacson, J. Lee, S. Gunasekar, P. H. P. Savarese, N. Srebro, and D. Soudry. Convergence of gradient descent on separable data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3420–3428. PMLR, 2019. 6

M. Phuong and C. H. Lampert. The inductive bias of ReLU networks on orthogonally separable data. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=krz7T0xU9Z_. 6

H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9573–9585. Curran Associates, Inc., 2020. 3, 6

D. Soudry, E. Hoffer, and N. Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19, 10 2017. 1

D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018. 6

D. A. Tarzanagh, Y. Li, C. Thrampoulidis, and S. Oymak. Transformers as support vector machines, 2023a. 6

D. A. Tarzanagh, Y. Li, X. Zhang, and S. Oymak. Max-margin token selection in attention mechanism, 2023b. 6

G. Vardi. On the implicit bias in deep-learning algorithms, 2022. 6

G. Vardi and O. Shamir. Implicit regularization in relu networks with the square loss. In *Conference on Learning Theory*, pages 4224–4258. PMLR, 2021. 6

B. Vasudeva, P. Deora, and C. Thrampoulidis. Implicit bias and fast convergence rates for self-attention, 2024. URL https://arxiv.org/abs/2402.05738. 6

B. Wang, Q. Meng, W. Chen, and T.-Y. Liu. The implicit bias for adaptive optimization algorithms on homogeneous neural networks. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10849–10858. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wang21q.html. 6

B. Wang, Q. Meng, H. Zhang, R. Sun, W. Chen, Z.-M. Ma, and T.-Y. Liu. Does momentum change the implicit regularization on separable data? In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088. 6

J. Wu, D. Zou, V. Braverman, and Q. Gu. Direction matters: On the implicit bias of stochastic gradient descent with moderate learning rate. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3X64RLgzY6O. 1

S. Xie and Z. Li. Implicit bias of AdamW: ℓ_∞-norm constrained optimization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=CmXkdlO6JJ. 6

C. Zhang, D. Zou, and Y. Cao. The implicit bias of adam on separable data, 2024. URL https://arxiv.org/abs/2406.10650. 1, 6

J. Zhang, S. P. Karimireddy, A. Veit, S. Kim, S. J. Reddi, S. Kumar, and S. Sra. Why adam beats sgd for attention models. *ArXiv*, abs/1912.03194, 2019. URL https://api.semanticscholar.org/CorpusID:208858389. 1

P. Zhou, J. Feng, C. Ma, C. Xiong, S. Hoi, and E. Weinan. Towards theoretically understanding why sgd generalizes better than adam in deep learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546. 6

D. Zou, Y. Cao, Y. Li, and Q. Gu. Understanding the generalization of adam in learning neural networks with proper regularization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=iUYpN14qjTF. 6

Appendix

A Omitted Proofs

The proof for Proposition 1 is as follows.

Proof. The optimal predictor can be found by solving for the following:

$$\frac{1}{2}\exp\left(-\frac{(x_1-\mu_1)^2}{2\sigma_x^2}-\frac{(x_2-\mu_2)^2}{2\sigma_y^2}\right)+\frac{1}{2}\exp\left(-\frac{(x_1-\mu_1)^2}{2\sigma_x^2}-\frac{(x_2+\mu_2)^2}{2\sigma_y^2}\right)=\exp\left(-\frac{(x_1+\mu_3)^2}{2\sigma_x^2}-\frac{x_2^2}{2\sigma_y^2}\right).$$

Simplification yields

$$0.5\left(1+\exp\left(-\frac{2\mu_2x_2}{\sigma_y^2}\right)\right)=\exp\left(-\frac{(\mu_1+\mu_3)x_1}{\sigma_x^2}-\frac{\mu_2x_2}{\sigma_y^2}+\frac{\mu_1^2-\mu_3^2}{2\sigma_x^2}+\frac{\mu_2^2}{2\sigma_y^2}\right).$$

Taking $\log$ on both sides and rearranging, we get:

$$\frac{(\mu_1+\mu_3)x_1}{\sigma_x^2}+\frac{\mu_2x_2}{\sigma_y^2}=\frac{\mu_1^2-\mu_3^2}{2\sigma_x^2}+\frac{\mu_2^2}{2\sigma_y^2}-\log\left(0.5\left(1+\exp\left(-\frac{2\mu_2x_2}{\sigma_y^2}\right)\right)\right).$$

For isotropic Gaussians, it simplifies to

$$(\mu_1+\mu_3)x_1+\mu_2x_2=\frac{\mu_1^2+\mu_2^2-\mu_3^2}{2}-\sigma^2\log\left(0.5\left(1+\exp\left(-\frac{2\mu_2x_2}{\sigma^2}\right)\right)\right).$$

Under realizability, we get

$$\omega x_1+x_2=-\frac{\sigma^2}{\mu}\log\left(0.5\left(1+\exp\left(-\frac{2\mu x_2}{\sigma^2}\right)\right)\right).$$

□

A.1 Gaussian Data

We can prove Proposition 2 as follows.

Proof. The population gradient can be simplified as follows.

$$\begin{aligned}
\mathbb{E}[\mathbb{1}[\boldsymbol{w}_j^\top\boldsymbol{x}\geq 0]y\boldsymbol{x}] &= \mathbb{E}(\boldsymbol{x}|\boldsymbol{w}^\top\boldsymbol{x}\geq 0, y=1,\epsilon=1)\Pr[y=1]\Pr[\epsilon=1|y=1]\Pr[\boldsymbol{w}^\top\boldsymbol{x}\geq 0|y=1,\epsilon=1]\\
&\quad+\mathbb{E}(\boldsymbol{x}|\boldsymbol{w}_j^\top\boldsymbol{x}\geq 0, y=1,\epsilon=-1)\Pr[y=1]\Pr[\epsilon=-1|y=1]\Pr[\boldsymbol{w}^\top\boldsymbol{x}\geq 0|y=1,\epsilon=-1]\\
&\quad+\mathbb{E}(-\boldsymbol{x}|\boldsymbol{w}_j^\top\boldsymbol{x}\geq 0, y=-1)\Pr[y=-1]\Pr[\boldsymbol{w}_j^\top\boldsymbol{x}\geq 0|y=-1]\\
&= \frac{1}{4}(\Pr[\boldsymbol{w}_j^\top\boldsymbol{x}\geq 0|y=1,\epsilon=1]\mathbb{E}(\boldsymbol{x}|\boldsymbol{w}_j^\top\boldsymbol{x}\geq 0, y=1,\epsilon=1)+\Pr[\boldsymbol{w}_j^\top\boldsymbol{x}\geq 0|y=1,\epsilon=-1]\\
&\quad\mathbb{E}(\boldsymbol{x}|\boldsymbol{w}_j^\top\boldsymbol{x}\geq 0, y=1,\epsilon=-1)-2\Pr[\boldsymbol{w}_j^\top\boldsymbol{x}\geq 0|y=-1]\mathbb{E}(\boldsymbol{x}|\boldsymbol{w}_j^\top\boldsymbol{x}\geq 0, y=-1)).
\end{aligned}$$

The conditional expectation $\mathbb{E}(\boldsymbol{x}'|\boldsymbol{w}^\top\boldsymbol{x}'\geq 0)$ can be simplified as follows. Let $\boldsymbol{\mu}' := \mathbb{E}(\boldsymbol{x}')$. Since we can write $\boldsymbol{x}' = \bar{\boldsymbol{w}}^\top\boldsymbol{x}'\bar{\boldsymbol{w}}+\bar{\boldsymbol{w}}_\perp^\top\boldsymbol{x}'\bar{\boldsymbol{w}}_\perp =: \boldsymbol{x}'_\parallel+\boldsymbol{x}'_\perp$, we have

$$\begin{aligned}
\mathbb{E}(\boldsymbol{x}'|\boldsymbol{w}^\top\boldsymbol{x}'\geq 0) &= \mathbb{E}(\boldsymbol{x}'_\parallel|\boldsymbol{w}^\top\boldsymbol{x}'\geq 0)+\mathbb{E}(\boldsymbol{x}'_\perp|\boldsymbol{w}^\top\boldsymbol{x}'\geq 0)\\
&= \mathbb{E}(\bar{\boldsymbol{w}}^\top\boldsymbol{x}'\bar{\boldsymbol{w}}|\boldsymbol{w}^\top\boldsymbol{x}'\geq 0)+\mathbb{E}(\boldsymbol{x}'_\perp)\\
&= \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^2}\mathbb{E}(\boldsymbol{w}^\top\boldsymbol{x}'|\boldsymbol{w}^\top\boldsymbol{x}'\geq 0)+\mathbb{E}(\boldsymbol{x}')-\mathbb{E}(\boldsymbol{x}'_\parallel)\\
&= \boldsymbol{\mu}'-\bar{\boldsymbol{w}}^\top\boldsymbol{\mu}'\bar{\boldsymbol{w}}+\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^2}\mathbb{E}(\boldsymbol{w}^\top\boldsymbol{x}'|\boldsymbol{w}^\top\boldsymbol{x}'\geq 0).
\end{aligned}$$

Using a result on the mean of truncated normal distribution from Burkardt [2023], and that for a given $\boldsymbol{w}$, $\boldsymbol{w}^\top\boldsymbol{x}'$ is a Gaussian random variable, we have,

$$\mathbb{E}(\boldsymbol{w}^\top\boldsymbol{x}'|\boldsymbol{w}^\top\boldsymbol{x}'\geq 0)=\mu_{\boldsymbol{w}}+\sigma_{\boldsymbol{w}}\frac{\phi(-\frac{\mu_{\boldsymbol{w}}}{\sigma_{\boldsymbol{w}}})}{1-\Phi(-\frac{\mu_{\boldsymbol{w}}}{\sigma_{\boldsymbol{w}}})},$$

where $\mu_{\boldsymbol{w}} := \boldsymbol{w}^\top \boldsymbol{\mu}'$, $\sigma_{\boldsymbol{w}} := \sigma\|\boldsymbol{w}\|$. Then, we have

$$\mathbb{E}(\boldsymbol{x}'|\boldsymbol{w}^\top \boldsymbol{x}' \geq 0) = \boldsymbol{\mu}' + \sigma \frac{\phi(-\frac{\mu_{\boldsymbol{w}}}{\sigma_{\boldsymbol{w}}})}{1-\Phi(-\frac{\mu_{\boldsymbol{w}}}{\sigma_{\boldsymbol{w}}})}\bar{\boldsymbol{w}}.$$

Since $d = 2$, using the above, we can write the population gradient $-a_j\mathbb{E}[\mathbb{1}[\boldsymbol{w}_j^\top \boldsymbol{x} \geq 0]y\boldsymbol{x}]$ as:

$$-0.25a_j(p_+(\boldsymbol{\mu}_+ + \sigma\Gamma(\boldsymbol{\mu}_+, \bar{\boldsymbol{w}}_j)\bar{\boldsymbol{w}}_j) + p_-(\boldsymbol{\mu}_- + \sigma\Gamma(\boldsymbol{\mu}_-, \bar{\boldsymbol{w}}_j)\bar{\boldsymbol{w}}_j) - 2p_0(\boldsymbol{\mu}_0 + \sigma\Gamma(\boldsymbol{\mu}_0, \bar{\boldsymbol{w}}_j)\bar{\boldsymbol{w}}_j)),$$

where $p_+ := \Pr[\boldsymbol{w}_j^\top \boldsymbol{x} \geq 0|y = 1, \epsilon = 1] = \Phi(\frac{\boldsymbol{\mu}_+^\top \bar{\boldsymbol{w}}}{\sigma})$, $p_- := \Pr[\boldsymbol{w}_j^\top \boldsymbol{x} \geq 0|y = 1, \epsilon = -1] = \Phi(\frac{\boldsymbol{\mu}_-^\top \bar{\boldsymbol{w}}}{\sigma})$, $p_0 := \Pr[\boldsymbol{w}_j^\top \boldsymbol{x} \geq 0|y = -1] = \Phi(\frac{\boldsymbol{\mu}_0^\top \bar{\boldsymbol{w}}}{\sigma})$, and $\Gamma(\boldsymbol{\mu}, \bar{\boldsymbol{w}}) := \frac{\phi(-\frac{\boldsymbol{\mu}^\top \bar{\boldsymbol{w}}}{\sigma})}{1-\Phi(-\frac{\boldsymbol{\mu}^\top \bar{\boldsymbol{w}}}{\sigma})} = \frac{\phi(\frac{\boldsymbol{\mu}^\top \bar{\boldsymbol{w}}}{\sigma})}{\Phi(\frac{\boldsymbol{\mu}^\top \bar{\boldsymbol{w}}}{\sigma})}$ (using the facts that for any z, $\phi(-z) = \phi(z)$ and $1 - \Phi(-z) = \Phi(z)$). Simplifying the expression then finishes the proof. □

Next, we can prove Theorem 1 as follows. Let $\omega \geq 2$ and $\lambda_0 := \frac{\mu}{\sigma} \geq 0.8$.

Proof. For neuron $j \in [m]$, let $\theta_{j,t}$ denote the angle between $\boldsymbol{w}_{j,t}$ and the x-axis at iteration $t \geq 0$. Let $\bar{\boldsymbol{w}}_{GD}^* := [1, 0]^\top$. Then, $\cos\theta_t = \bar{\boldsymbol{w}}_t^\top \bar{\boldsymbol{w}}_{GD}^*$. We want to see if θ_t tends to 0 with time. Specifically, given $\theta \in [-\pi, \pi]$, we want to show that $a\frac{d\cos\theta_t}{dt} > 0$. We have:

$$\begin{aligned}
&a\tfrac{d\cos\theta_t}{dt} = a\dot{\bar{\boldsymbol{w}}}_t^\top \bar{\boldsymbol{w}}_{GD}^* = a\tfrac{\dot{\boldsymbol{w}}_t^\top}{\|\boldsymbol{w}_t\|}(I - \bar{\boldsymbol{w}}_t\bar{\boldsymbol{w}}_t^\top)\bar{\boldsymbol{w}}_{GD}^* \\
&= \tfrac{a^2\sigma}{4\|\boldsymbol{w}_t\|}\Big(\tfrac{\lambda(\omega^2-1)}{\omega^2+1}\left(\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}_t) + \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}_t)\right) + 2\lambda\Phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}}_t) + \tfrac{w_{t,1}}{\|\boldsymbol{w}_t\|}\left(\phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}_t) + \phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}_t) - 2\phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}}_t)\right) \\
&- \tfrac{w_{t,1}}{\|\boldsymbol{w}_t\|}\left(\lambda(\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}_t)\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}_t + \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}_t)\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}_t - 2\Phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}}_t)\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}}_t) + \left(\phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}_t) + \phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}_t) - 2\phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}}_t)\right)\right)\Big) \\
&= \tfrac{a^2\sigma\lambda}{4\|\boldsymbol{w}_t\|}\Big(\tfrac{(\omega^2-1)}{\omega^2+1}\tfrac{w_{t,2}^2}{\|\boldsymbol{w}_t\|^2}\left(\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}_t) + \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}_t)\right) - \tfrac{2\omega}{\omega^2+1}\tfrac{w_{t,1}w_{t,2}}{\|\boldsymbol{w}_t\|^2}\left(\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}_t) - \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}_t)\right) + 2\tfrac{w_{t,2}^2}{\|\boldsymbol{w}_t\|^2}\Phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}}_t)\Big) \\
&= \tfrac{a^2\sigma\lambda}{4\|\boldsymbol{w}_t\|}\Big(\tfrac{(\omega^2-1)}{\omega^2+1}\sin^2\theta_t\left(\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}_t) + \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}_t)\right) - \tfrac{2\omega}{\omega^2+1}\cos\theta_t\sin\theta_t\left(\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}_t) - \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}_t)\right) + 2\sin^2\theta_t\Phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}}_t)\Big).
\end{aligned}$$

The first and third terms are always positive, so the sign depends on the second term. We note that the derivative is 0 when $\boldsymbol{w}_{t,2} = 0$, *i.e.*, $\theta_t = 0$. This indicates that once $\boldsymbol{w}_t$ reaches this point, it stays there. Also, using MVT, we can write:

$$\Phi\left(\tfrac{\lambda((\omega^2-1)w_{t,1}+2\omega w_{t,2})}{\|\boldsymbol{w}_t\|(\omega^2+1)}\right) - \Phi\left(\tfrac{\lambda((\omega^2-1)w_{t,1}-2\omega w_{t,2})}{\|\boldsymbol{w}_t\|(\omega^2+1)}\right) = \phi(c)\tfrac{4\lambda\omega\sin\theta_t}{(\omega^2+1)},$$

for some $c \in \left[\frac{\lambda((\omega^2-1)w_{t,1}-2\omega w_{t,2})}{\|\boldsymbol{w}_t\|(\omega^2+1)}, \frac{\lambda((\omega^2-1)w_{t,1}+2\omega w_{t,2})}{\|\boldsymbol{w}_t\|(\omega^2+1)}\right]$. Clearly, $\phi(c) \leq \phi(\lambda)$. The second term is lower bounded by $-\frac{8\lambda\omega^2}{(\omega^2+1)^2}\phi(\lambda)\sin^2\theta_t\cos\theta_t$. We now consider two cases:

Case 1: $\theta_t \in [-\pi, -\pi/2]$ or $\theta_t \in [\pi/2, \pi]$: In this case, $\cos\theta_t < 0$, so the second term, and hence the derivative, is positive.

Case 2: $\theta_t \in [-\pi/2, \pi/2]$: In this case, $\cos\theta_t > 0$, so the second term is negative, and we have to compare its magnitude to the other terms. Using $\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}_t) + \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}_t) \geq 1$ and $\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}}_t \geq -1$, we have:

$$a\tfrac{d\cos\theta_t}{dt} \geq \tfrac{a^2\sigma\lambda\sin^2\theta_t}{4\|\boldsymbol{w}_t\|}\Big(\tfrac{(\omega^2-1)}{\omega^2+1} - \tfrac{8\omega^2\lambda}{(\omega^2+1)^2}\cos\theta_t\phi(\lambda) + 2\Phi(-\lambda)\Big).$$

Since $\cos\theta_t \leq 1$ and $\Phi(-\lambda) > 0$, the RHS is positive when $\frac{\omega^2-1}{4\omega} \geq \frac{\mu}{\sigma}\phi(\lambda)$.

Let $E(\lambda_0, \omega) := \frac{\omega^2-1}{4\omega} - \lambda_0\phi(\lambda_0\frac{\omega^2+1}{2\omega})$.

$$\tfrac{dE}{d\lambda_0} = -\phi(\lambda_0\tfrac{\omega^2+1}{2\omega}) + \left(\lambda_0\tfrac{\omega^2+1}{2\omega}\right)^2\phi(\lambda_0\tfrac{\omega^2+1}{2\omega}) \geq 0,$$

when $\lambda_0 \geq \frac{2\omega}{\omega^2+1}$. The RHS is a decreasing function of ω for $\omega \geq 2$. The condition becomes $\lambda_0 \geq 0.8$.

$$\tfrac{dE}{d\omega} = \tfrac{1}{4} + \tfrac{1}{4\omega^2} + \lambda_0^3\tfrac{\omega^2+1}{2\omega}\left(\tfrac{1}{2} - \tfrac{1}{2\omega^2}\right)\phi(\lambda_0\tfrac{\omega^2+1}{2\omega}) = \tfrac{\omega^2+1}{4\omega^2}\left(1 + \lambda_0^3\tfrac{\omega^4-1}{\omega}\phi(\lambda_0\tfrac{\omega^2+1}{2\omega})\right) \geq 0.$$

Since E is an increasing function of both ω and λ, and we can numerically verify that $E(0.8, 2) > 0$, the result is true for all $\omega \geq 2$ and $\lambda_0 \geq 0.8$.

□

Next, we prove Theorem 2 as follows. Let $\omega \geq 2$ and $0.8 \leq \lambda_0 \leq 1.5$.

Proof. For signGD, we can analyze the gradient expression for any $\boldsymbol{w}$:

$$\nabla_{\boldsymbol{w}} \widehat{L}(\boldsymbol{W}) = -\frac{a\sigma}{4}\Big(\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}})\lambda\bar{\boldsymbol{\mu}}_+ + \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}})\lambda\bar{\boldsymbol{\mu}}_- - 2\Phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}})\lambda\bar{\boldsymbol{\mu}}_0 + \big(\phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}) + \phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}) - 2\phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}})\big)\,\bar{\boldsymbol{w}}\Big).$$

Specifically, the gradient is in the direction $[\pm 1, 0]^\top$ only when $[0,1]^\top\nabla_{\boldsymbol{w}}\widehat{L}(\boldsymbol{W}) = 0$. We have:

$$\begin{aligned}[0,1]^\top\nabla_{\boldsymbol{w}}\widehat{L}(\boldsymbol{W}) &= -\frac{a\sigma}{4}\Big(\tfrac{2\omega\lambda}{\omega^2+1}\big(\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}) - \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}})\big) + \sin\theta\big(\phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}) + \phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}) - 2\phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}})\big)\Big)\\ &= -\frac{a\sigma\sin\theta}{4}\Big(2\big(\tfrac{2\omega\lambda}{\omega^2+1}\big)^2\phi(c) + \phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}) + \phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}) - 2\phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}})\Big),\end{aligned}$$

where $c \in [\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}, \lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}]$. Consider the expression in the parenthesis. Assuming $\frac{\omega^2-1}{2\omega} \geq 1$, we have:

$$\begin{aligned}2\big(\tfrac{2\omega\lambda}{\omega^2+1}\big)^2\phi(c) + \phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}) + \phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}) - 2\phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}}) &\geq 2\Big(\Big(\big(\tfrac{2\omega\lambda}{\omega^2+1}\big)^2+1\Big)\phi(\tfrac{2\omega\lambda}{\omega^2+1}) - \phi(0)\Big)\\ &= 2\Big(\Big(\big(\tfrac{\mu}{\sigma}\big)^2+1\Big)\phi(\tfrac{\mu}{\sigma}) - \phi(0)\Big) > 0,\end{aligned}$$

whenever $\frac{\mu}{\sigma} \leq 1.5$ (we can check this numerically, and use the fact that $\phi(z)$ is a decreasing function of $z \geq 0$).

Thus, the gradient is only in the $[\pm 1, 0]^\top$ direction when $\sin\theta = 0$, *i.e.*, when $\boldsymbol{w}$ is in that direction.

Next, we can check if there are neurons in the $[0, \pm 1]^\top$ direction. We have:

$$[1,0]\nabla_{\boldsymbol{w}}\widehat{L}(\boldsymbol{W}) = -\frac{a\sigma}{4}\Big(2\lambda\tfrac{\omega^2-1}{\omega^2+1}\big(\Phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}) + \Phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}})\big) + 2\lambda\Phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}}) + \cos\theta\big(\phi(\lambda\bar{\boldsymbol{\mu}}_+^\top\bar{\boldsymbol{w}}) + \phi(\lambda\bar{\boldsymbol{\mu}}_-^\top\bar{\boldsymbol{w}}) - 2\phi(\lambda\bar{\boldsymbol{\mu}}_0^\top\bar{\boldsymbol{w}})\big)\Big).$$

The expression in the parenthesis is positive as long as $\lambda\frac{\omega^2-1}{\omega^2+1} + 0.5\lambda \geq 0.4$, or $\frac{3\omega^2-1}{1.6\omega} \geq \frac{\sigma}{\mu}$. Let $E(\lambda_0, \omega) = \lambda_0 - \frac{1.6\omega}{3\omega^2-1}$. We can show that it is an increasing function of both λ_0 and ω. Since $E(0.8, 2) > 0$, the result holds for all $\omega \geq 2$ and $\lambda_0 \geq 0.8$.

Based on these calculations, the updates are along $[\pm 1, \pm 1]^\top$ directions, depending on the sign of a and $\sin\theta$. Specifically, we have the following cases for θ_t and θ_{t+1} at any t:

$\texttt{sign}(\sin\theta_t)$	$\texttt{sign}(a)$	$\texttt{sign}([0,1]^\top\nabla_{\boldsymbol{w}}\widehat{L}(\boldsymbol{W}))$	$\texttt{sign}(\sin\theta_{t+1})$
+ve	+ve	+ve	+ve
+ve	-ve	-ve	-ve
-ve	+ve	-ve	-ve
-ve	-ve	+ve	+ve

This shows that whenever $a > 0$, the updates for neurons in the first/second or third/fourth quadrant are along $[1,1]^\top$ or $[1,-1]^\top$, respectively. However, when $a < 0$, the updates for neurons in the first/second or third/fourth quadrants alternate between $[-1, \pm 1]^\top$ and $[-1, \mp 1]^\top$. As a result, at even iterations, these neurons are close to the $[-1, 0]^\top$ direction (but may not be exactly aligned due to the initialization). However, in the limit $t \to \infty$, these neurons converge in this direction. □

Next, we prove Theorem 3 as follows. Let $d = 2$ for simplicity, $\omega \in [2, 4]$ and $\lambda_0 \in [2\omega/3, 5]$.

Linear: $\hat{y} = \texttt{sign}(ax_1 + bx_2)$. Piece-wise Linear: $\hat{y}' = \begin{cases}\texttt{sign}(ax_1 + bx_2) & x_2 \geq 0,\\ \texttt{sign}(ax_1 - bx_2) & x_2 < 0.\end{cases}$

Proof.

$$\mathbb{E}(\hat{y} \neq y) = \frac{1}{4}\left[\Phi\left(-\frac{\mu\left(\frac{a}{2}\left(\kappa\omega - \frac{1}{\omega}\right)+b\right)}{\sigma_y\sqrt{a^2\kappa+b^2}}\right) + \Phi\left(-\frac{\mu\left(\frac{a}{2}\left(\kappa\omega - \frac{1}{\omega}\right)-b\right)}{\sigma_y\sqrt{a^2\kappa+b^2}}\right)\right] + \frac{1}{2}\,\Phi\left(-\frac{\mu\frac{a}{2}\left(\kappa\omega + \frac{1}{\omega}\right)}{\sqrt{\kappa}\,\sigma_y}\right).$$

When $a = 1, b = 0$, we get:

$$\mathbb{E}(\hat{y} \neq y) = \frac{1}{2}\Phi\left(-\frac{\mu\left(\kappa\omega - \frac{1}{\omega}\right)}{2\sigma_y\sqrt{\kappa}}\right) + \frac{1}{2}\,\Phi\left(-\frac{\mu\left(\kappa\omega + \frac{1}{\omega}\right)}{2\sigma_y\sqrt{\kappa}}\right).$$

For isotropic Gaussians, we get:

$$\mathbb{E}(\hat{y} \neq y) = \frac{1}{2}\Phi\left(-\frac{\mu}{\sigma}\frac{\omega^2-1}{2\omega}\right) + \frac{1}{2}\,\Phi\left(-\frac{\mu}{\sigma}\frac{\omega^2+1}{2\omega}\right).$$

Considering the non-isotropic case, we have:

$$\begin{aligned}
&\mathbb{E}(\hat{y}' \neq y) < \frac{1}{2}\,\Phi\left(-\frac{\mu}{\sqrt{\kappa+1}\sigma_y}\frac{\kappa\omega^2-1+2\omega}{2\omega}\right) + \Phi\left(-\frac{\mu}{\sqrt{\kappa}\sigma_y}\frac{\kappa\omega^2+1}{2\omega}\right) \\
\implies &\mathbb{E}(\hat{y}' \neq y) - \mathbb{E}(\hat{y} \neq y) < \frac{1}{2}\,\Phi\left(-\frac{\mu}{\sqrt{\kappa+1}\sigma_y}\frac{\kappa\omega^2-1+2\omega}{2\omega}\right) + \frac{1}{2}\,\Phi\left(-\frac{\mu}{\sqrt{\kappa}\sigma_y}\frac{\kappa\omega^2+1}{2\omega}\right) - \frac{1}{2}\,\Phi\left(-\frac{\mu}{\sqrt{\kappa}\sigma_y}\frac{\kappa\omega^2-1}{2\omega}\right) \\
&= \frac{1}{2}\left(\Phi\left(-\frac{\mu}{\sqrt{\kappa+1}\sigma_y}\frac{\kappa\omega^2-1+2\omega}{2\omega}\right) + \Phi\left(\frac{\mu}{\sqrt{\kappa}\sigma_y}\frac{\kappa\omega^2-1}{2\omega}\right) - \Phi\left(\frac{\mu}{\sqrt{\kappa}\sigma_y}\frac{\kappa\omega^2+1}{2\omega}\right)\right)
\end{aligned}$$

For the isotropic case, define $E(\lambda_0, \omega) := \Phi(-\lambda_0\frac{\omega^2-1+2\omega}{2\sqrt{2}\omega}) + \Phi(\lambda_0\frac{\omega^2-1}{2\omega}) - \Phi(\lambda_0\frac{\omega^2+1}{2\omega})$. We have:

$$E(\lambda_0, \omega) \leq \frac{\phi(\lambda_0\frac{\omega^2-1+2\omega}{2\sqrt{2}\omega})}{\lambda_0\frac{\omega^2-1+2\omega}{2\sqrt{2}\omega}} - \frac{\lambda_0}{\omega}\phi(\lambda_0\frac{\omega^2+1}{2\omega})$$

We can analyze the first derivatives:

$$\begin{aligned}
\frac{dE}{d\omega} &= \frac{\lambda_0}{2\omega^2}\left(-\frac{\omega^2+1}{\sqrt{2}}\phi(\lambda_0\frac{\omega^2-1+2\omega}{2\sqrt{2}\omega}) + (\omega^2+1)\phi(\lambda_0\frac{\omega^2-1}{2\omega}) - (\omega^2-1)\phi(\lambda_0\frac{\omega^2+1}{2\omega})\right) \\
&= \frac{\lambda_0(\omega^2+1)}{2\omega^2}\phi(\lambda_0\frac{\omega^2+1}{2\omega})\left(1 - \frac{1}{\sqrt{2}}\exp\left(-\frac{\lambda_0^2}{8\omega^2}\left(\frac{(\omega^2-1+2\omega)^2}{2} - (\omega^2-1)^2\right)\right) - \frac{\omega^2-1}{\omega^2+1}\exp\left(-\frac{\lambda_0^2((\omega^2+1)^2-(\omega^2-1)^2)}{8\omega^2}\right)\right) \\
&= \frac{\lambda_0(\omega^2+1)}{2\omega^2}\phi(\lambda_0\frac{\omega^2-1}{2\omega})\left(1 - \frac{1}{\sqrt{2}}\exp\left(-\frac{\lambda_0^2}{16\omega^2}(-\omega^4+4\omega^3+6\omega^2-4\omega-1)\right) - \frac{\omega^2-1}{\omega^2+1}\exp\left(-\frac{\lambda_0^2}{2}\right)\right)
\end{aligned}$$

The sign of the derivative depends on the expression inside the parenthesis. Second and third term have similar behaviour when $|\omega^2 - 1| < |\frac{\omega^2-1+2\omega}{\sqrt{2}}|$, which is true for $1 \leq \omega \leq 5$. They are a decreasing function of λ_0. Since $\omega \in [2, 4]$ and $\lambda_0 \geq 2\omega/3$, then the derivative is positive because

$$\begin{aligned}
&1 - \frac{1}{\sqrt{2}}\exp\left(-\frac{1}{36}(-\omega^4+4\omega^3+6\omega^2-4\omega-1)\right) - \frac{\omega^2-1}{\omega^2+1}\exp\left(-\frac{2\omega^2}{9}\right) \\
&> 1 - \frac{1}{\sqrt{2}}\exp\left(-\frac{1}{36}(3.732)\right) - 0.8824\exp\left(-\frac{8}{9}\right) > 0.
\end{aligned}$$

Next we compute the derivative wrt λ_0. $\frac{dE}{d\lambda_0} = -\frac{17}{5\sqrt{2}}\phi(\lambda_0\frac{17}{5\sqrt{2}}) + \frac{12}{5}\phi(\lambda_0\frac{12}{5}) - \frac{13}{5}\phi(\lambda_0\frac{13}{5})$

$$\begin{aligned}
\frac{dE}{d\lambda_0} &= -\frac{\omega^2-1+2\omega}{2\sqrt{2}\omega}\phi(\lambda_0\frac{\omega^2-1+2\omega}{2\sqrt{2}\omega}) + \frac{\omega^2-1}{2\omega}\phi(\lambda_0\frac{\omega^2-1}{2\omega}) - \frac{\omega^2+1}{2\omega}\phi(\lambda_0\frac{\omega^2+1}{2\omega}) \\
&= \frac{\omega^2-1}{2\omega}\phi(\lambda_0\frac{\omega^2-1}{2\omega})\left(1 - \frac{\omega^2-1+2\omega}{\sqrt{2}(\omega^2-1)}\exp\left(-\frac{\lambda_0^2}{16\omega^2}(-\omega^4+4\omega^3+6\omega^2-4\omega-1)\right) - \frac{\omega^2+1}{\omega^2-1}\exp\left(-\frac{\lambda_0^2}{2}\right)\right).
\end{aligned}$$

For a fixed $\omega \in [2, 4]$, this is an increasing function of λ_0. This implies that E first decreases upto some $\lambda_0 = \bar{\lambda}_0$, and then starts increasing.

Numerically, $E(5, 4) \leq 0$, implying $E(\lambda_0, \omega) \leq 0$ for all $\omega \in [2, 4]$, $\lambda_0 \in [2\omega/3, 5]$.

□

A.2 Toy Data

We can write the Bayes' optimal predictor in the toy setting as follows. We consider three datapoints $(\boldsymbol{x}, y)$: $([-\mu_3, 0]^\top, -1)$, $([\mu_1, \mu_2]^\top, 1)$ and $([\mu_1, -\mu_2]^\top, 1)$, where $\mu_1, \mu_2, \mu_3 > 0$. The optimal predictor can be found by solving the following:

$$\min(\sqrt{(x_1-\mu_1)^2 + (x_2-\mu_2)^2}, \sqrt{(x_1-\mu_1)^2 + (x_2+\mu_2)^2}) = \sqrt{(x_1+\mu_3)^2 + x_2^2}.$$

Solving this gives a piecewise linear function:

$$(\mu_1+\mu_3)x_1+\mu_2 x_2 = \tfrac{\mu_1^2+\mu_2^2-\mu_3^2}{2}, \quad x_2>0$$
$$(\mu_1+\mu_3)x_1-\mu_2 x_2 = \tfrac{\mu_1^2+\mu_2^2-\mu_3^2}{2}, \quad x_2\le 0.$$

In the realizable setting, this is:

$$\omega x_1 + x_2 = 0, \quad x_2>0$$
$$\omega x_1 - x_2 = 0, \quad x_2\le 0.$$

We now restate Theorem 4 for convenience, followed by the proof.

Theorem 5. *Consider* $\|\boldsymbol{w}_k\| \le \frac{\eta\mu}{8\omega(\omega^2-1)}(((3\omega^2+1)(\omega^2-1)-4\omega^2)\wedge(4\omega^2-(\omega^2-1)^2)\wedge\frac{8\omega}{\mu}(2\omega+1-\omega^2))$ *and* $1+\frac{2}{\sqrt{3}}<\omega^2<3+2\sqrt{2}$. *Let* $\bar{\boldsymbol{w}}_{k,\infty} := \frac{\lim_{t\to\infty}\frac{\boldsymbol{w}_{k,t}}{t}}{\|\lim_{t\to\infty}\frac{\boldsymbol{w}_{k,t}}{t}\|}$, *for neuron* $k\in[m]$ *and* $p := \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi}$. *Then, for* $m\to\infty$, *the solutions learned by GD, signGD, and Adam are:*

	GD	Adam ($\beta_1=\beta_2=0$) or signGD	Adam ($\beta_1=\beta_2\approx 1$)
$\bar{\mathbf{w}}_{k,\infty} =$	$\begin{cases}[1,0]^\top & \text{w.p. } \frac{1}{4}+\frac{p}{2}\\ [-1,0]^\top & \text{w.p. } \frac{1}{2}\\ \frac{1}{\omega^2+1}[\omega^2-1,2\omega]^\top & \text{w.p. } \frac{1}{8}-\frac{p}{4}\\ \frac{1}{\omega^2+1}[\omega^2-1,-2\omega]^\top & \text{w.p. } \frac{1}{8}-\frac{p}{4}\end{cases}$	$\begin{cases}[1,0]^\top & \text{w.p. } p\\ [-1,0]^\top & \text{w.p. } \frac{1}{2}\\ \frac{1}{\sqrt{2}}[1,1]^\top & \text{w.p. } \frac{1}{4}-\frac{p}{2}\\ \frac{1}{\sqrt{2}}[1,-1]^\top & \text{w.p. } \frac{1}{4}-\frac{p}{2}\end{cases}$	$\begin{cases}[1,0]^\top & \text{w.p. } p\\ [-1,0]^\top & \text{w.p. } \frac{1}{2}\\ \frac{1}{\sqrt{2}}[1,1]^\top & \text{w.p. } \frac{1}{8}-\frac{p}{4}\\ \frac{1}{\sqrt{2}}[1,-1]^\top & \text{w.p. } \frac{1}{8}-\frac{p}{4}\\ \frac{1}{\sqrt{s^2+1}}[s,1]^\top & \text{w.p. } \frac{1}{8}-\frac{p}{4}\\ \frac{1}{\sqrt{s^2+1}}[s,-1]^\top & \text{w.p. } \frac{1}{8}-\frac{p}{4}\end{cases}$

where s *is a constant between* 0.72 *and* 1. *In each case, the sign of the first element of* $\boldsymbol{w}_{k,\infty}$ *is the same as* $\mathtt{sign}(a_k)$.

Proof. Let $z := (\boldsymbol{x}, y)$, and $\overline{\boldsymbol{z}}_1 := -\frac{\mu}{2}[\omega+\frac{1}{\omega},0]$, $\overline{\boldsymbol{z}}_2 := \frac{\mu}{2}[\omega-\frac{1}{\omega},2]$, $\overline{\boldsymbol{z}}_3 := \frac{\mu}{2}[\omega-\frac{1}{\omega},-2]$. Define three sets S_1, S_2, S_3 as $S_1 := \{z\in S : x_1<0\}$, $S_2 := \{z\in S : x_2>0\}$, $S_3 := \{z\in S : x_2<0\}$.

First iteration. We first analyze the gradients at the first iteration. Consider different cases where $\boldsymbol{w}_{k,0}^\top\boldsymbol{x}\ge 0$ depending on different samples $\boldsymbol{x}$. Table 1 lists the population gradients depending on which samples contribute to the gradient. See Fig. 3 for an illustration. Note that $\theta = \tan^{-1}\frac{\mu_2}{\mu_1} = \tan^{-1}\frac{2\omega}{\omega^2-1}$, and $\frac{\pi}{2}-\theta = \tan^{-1}\frac{\omega^2-1}{2\omega}$.

Set S s.t. $\boldsymbol{w}_{k,0}^\top\boldsymbol{x}>0$	Pop. Gradient $\mathbb{E}_{z\sim\mathcal{D}}[y\boldsymbol{x}\mid\boldsymbol{x}\in S]$	Prob. of such $\boldsymbol{w}_k$
$S_2\cup S_3$	$\frac{1}{2}[\mu_1,0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi}$
S_2	$\frac{1}{4}[\mu_1,\mu_2]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$
$S_1\cup S_2$	$\frac{1}{4}[\mu_1+2\mu_3,\mu_2]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$
S_1	$\frac{1}{2}[\mu_3,0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi}$
$S_3\cup S_1$	$\frac{1}{4}[\mu_1+2\mu_3,-\mu_2]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$
S_3	$\frac{1}{4}[\mu_1,-\mu_2]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$

Table 1: Population gradients and corresponding probabilities depending on the region of initialization of the neurons.

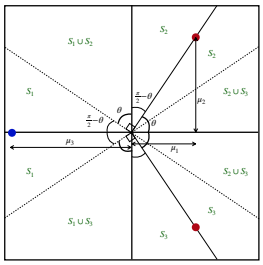

Figure 3: An illustration of the toy dataset and the set S such that $\boldsymbol{w}_{k,0}^\top\boldsymbol{x}>0$ depending on the region of initialization.

Using the population gradients in Table 1, the updates for the different algorithms, are written as:

$$\text{GD:} \quad \boldsymbol{w}_{k,1} = \boldsymbol{w}_{k,0} + \frac{a_k \eta \mu}{4} \begin{cases} [\frac{\omega^2-1}{\omega}, 0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi} \\ [\frac{\omega^2+1}{\omega}, 0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi} \\ \left[\frac{\omega^2-1}{2\omega}, 1\right] & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ \left[\frac{\omega^2-1}{2\omega}, -1\right] & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ [\frac{3\omega^2+1}{2\omega}, 1]^\top & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ [\frac{3\omega^2+1}{2\omega}, -1]^\top & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \end{cases},$$

$$\text{signGD/Adam:} \quad \boldsymbol{w}_{k,1} = \boldsymbol{w}_{k,0} + a_k \eta \begin{cases} [1, 0]^\top & \text{w.p. } 2\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi} \\ [1, 1]^\top & \text{w.p. } \frac{1}{2} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi} \\ [1, -1]^\top & \text{w.p. } \frac{1}{2} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi} \end{cases}.$$

Second iteration. Next, we use these updates to analyze the second iteration. Tables 2 to 4 include the updates at the second iteration for GD, signGD and Adam, respectively, where we use the conditions on ω and the (small) initialization scale. Specifically, the small initialization scale helps ensure that the gradient and the corresponding updated neuron are in the same region (in terms of which samples contribute to the gradient for the next iteration). Using the condition on ω, the updates in rows 5 and 7 of Table 2 remain in the direction of the points $\overline{\boldsymbol{z}}_2$ and $\overline{\boldsymbol{z}}_3$, respectively, whereas those in rows 9 and 11 get along the direction of $[1, 0]^\top$.

$4(\boldsymbol{w}_{k,1} - \boldsymbol{w}_{k,0})/(\eta\mu)$	Prob.	Set S s.t. $\boldsymbol{w}_{k,1}^\top \boldsymbol{x} > 0$	$4\texttt{sign}(a_k)\mathbb{E}_{z\sim\mathcal{D}}[\mathbb{1}[\boldsymbol{w}_{k,1}^\top \boldsymbol{x} \geq 0] y\boldsymbol{x}]/\mu$
$[\frac{\omega^2-1}{\omega}, 0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	$S_2 \cup S_3$	$[\frac{\omega^2-1}{\omega}, 0]^\top$
$-[\frac{\omega^2-1}{\omega}, 0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	S_1	$-[\frac{\omega^2+1}{\omega}, 0]^\top$
$[\frac{\omega^2+1}{\omega}, 0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	$S_2 \cup S_3$	$[\frac{\omega^2-1}{\omega}, 0]^\top$
$-[\frac{\omega^2+1}{\omega}, 0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	S_1	$-[\frac{\omega^2+1}{\omega}, 0]^\top$
$\left[\frac{\omega^2-1}{2\omega}, 1\right]$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	S_2	$[\frac{\omega^2-1}{2\omega}, 1]^\top$
$-\left[\frac{\omega^2-1}{2\omega}, 1\right]$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	S_1	$-[\frac{\omega^2+1}{\omega}, 0]^\top$
$\left[\frac{\omega^2-1}{2\omega}, -1\right]$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	S_3	$[\frac{\omega^2-1}{2\omega}, -1]^\top$
$-\left[\frac{\omega^2-1}{2\omega}, -1\right]$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	S_1	$-[\frac{\omega^2+1}{\omega}, 0]^\top$
$[\frac{3\omega^2+1}{2\omega}, 1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	$S_2 \cup S_3$	$[\frac{\omega^2-1}{\omega}, 0]^\top$
$-[\frac{3\omega^2+1}{2\omega}, 1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	S_1	$-[\frac{\omega^2+1}{\omega}, 0]^\top$
$[\frac{3\omega^2+1}{2\omega}, -1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	$S_2 \cup S_3$	$[\frac{\omega^2-1}{\omega}, 0]^\top$
$-[\frac{3\omega^2+1}{2\omega}, -1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	S_1	$-[\frac{\omega^2+1}{\omega}, 0]^\top$

Table 2: Population gradients at the second iteration for GD.

Based on the updates in Table 2, we can write the GD iterate at any time $t > 1$ as:

$$\boldsymbol{w}_{k,t} = \boldsymbol{w}_{k,1} + \frac{\eta\mu(t-1)}{4}\begin{cases}\left[\frac{\omega^2-1}{\omega}, 0\right]^\top & \text{w.p. } \frac{1}{4} + \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ -\left[\frac{\omega^2-1}{\omega}, 0\right]^\top & \text{w.p. } \frac{1}{2} \\ \left[\frac{\omega^2-1}{2\omega}, 1\right]^\top & \text{w.p. } \frac{1}{8} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi} \\ \left[\frac{\omega^2-1}{2\omega}, -1\right]^\top & \text{w.p. } \frac{1}{8} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}\end{cases}.$$

$\boldsymbol{w}_{k,1} - \boldsymbol{w}_{k,0}$	Prob.	$4\mathbb{E}_{z\sim\mathcal{D}}[\mathbb{1}[\boldsymbol{w}_{k,1}^\top \boldsymbol{x} \geq 0] y\boldsymbol{x}]/\mu$	$\boldsymbol{w}_{k,2} - \boldsymbol{w}_{k,1}$
$\eta[1,0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi}$	$[\frac{\omega^2-1}{\omega}, 0]^\top$	$\eta[1,0]^\top$
$-\eta[1,0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi}$	$[\frac{\omega^2+1}{\omega}, 0]^\top$	$-\eta[1,0]^\top$
$\eta[1,1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	$[\frac{\omega^2-1}{2\omega}, 1]^\top$	$\eta[1,1]^\top$
$-\eta[1,1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	$[\frac{\omega^2+1}{\omega}, 0]^\top$	$-\eta[1,0]^\top$
$\eta[1,-1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	$[\frac{\omega^2-1}{2\omega}, -1]^\top$	$\eta[1,-1]^\top$
$-\eta[1,-1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	$[\frac{\omega^2+1}{\omega}, 0]^\top$	$-\eta[1,0]^\top$

Table 3: Population gradients at the second iteration for SignGD (Adam, $\beta_1 = \beta_2 = 0$).

Based on the updates in Table 3, we can write the signGD iterate at any time t as:

$$\boldsymbol{w}_{k,t} = \boldsymbol{w}_{k,0} + \eta t\begin{cases}[1,0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi} \\ -[1,0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi} \\ [1,1]^\top & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ [1,-1]^\top & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ -[1,1/t]^\top & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ -[1,-1/t]^\top & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}\end{cases}.$$

$\boldsymbol{w}_{k,1}-\boldsymbol{w}_{k,0}$	$4\boldsymbol{g}_{k,0}/(\eta\mu)$	Prob.	$4\mathbb{E}_{z\sim\mathcal{D}}[\mathbb{1}[\boldsymbol{w}_{k,1}^\top\boldsymbol{x}\geq 0]y\boldsymbol{x}]/\mu$	$\boldsymbol{w}_{k,2}-\boldsymbol{w}_{k,1}$
$\eta[1,0]^\top$	$[\frac{\omega^2-1}{\omega},0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	$[\frac{\omega^2-1}{\omega},0]^\top$	$\eta[1,0]^\top$
$-\eta[1,0]^\top$	$-[\frac{\omega^2-1}{\omega},0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	$[\frac{\omega^2+1}{\omega},0]^\top$	$-\eta[\frac{1}{\sqrt{2}}\frac{2\omega^2}{\sqrt{(\omega^2-1)^2+(\omega^2+1)^2}},0]^\top$
$\eta[1,0]^\top$	$[\frac{\omega^2+1}{\omega},0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	$[\frac{\omega^2-1}{\omega},0]^\top$	$\eta[\frac{1}{\sqrt{2}}\frac{2\omega^2}{\sqrt{(\omega^2+1)^2+(\omega^2-1)^2}},0]^\top$
$-\eta[1,0]^\top$	$-[\frac{\omega^2+1}{\omega},0]^\top$	$\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}$	$[\frac{\omega^2+1}{\omega},0]^\top$	$-\eta[1,0]^\top$
$\eta[1,1]^\top$	$\left[\frac{\omega^2-1}{2\omega},1\right]$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	$[\frac{\omega^2-1}{2\omega},1]^\top$	$\eta[1,1]^\top$
$-\eta[1,1]^\top$	$-\left[\frac{\omega^2-1}{2\omega},1\right]$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	$[\frac{\omega^2+1}{\omega},0]^\top$	$-\frac{\eta}{\sqrt{2}}[\frac{(\omega^2-1)/2+(\omega^2+1)}{\sqrt{((\omega^2-1)/2)^2+(\omega^2+1)^2}},1]^\top$
$\eta[1,-1]^\top$	$\left[\frac{\omega^2-1}{2\omega},-1\right]$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	$[\frac{\omega^2-1}{2\omega},-1]^\top$	$\eta[1,-1]^\top$
$-\eta[1,-1]^\top$	$-\left[\frac{\omega^2-1}{2\omega},-1\right]$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	$[\frac{\omega^2+1}{\omega},0]^\top$	$-\frac{\eta}{\sqrt{2}}[\frac{(\omega^2-1)/2+(\omega^2+1)}{\sqrt{((\omega^2-1)/2)^2+(\omega^2+1)^2}},-1]^\top$
$\eta[1,1]^\top$	$[\frac{3\omega^2+1}{2\omega},1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	$[\frac{\omega^2-1}{2\omega},1]^\top$	$\eta[\frac{1}{\sqrt{2}}\frac{4\omega^2}{\sqrt{(3\omega^2+1)^2+(\omega^2-1)^2}},1]^\top$
$-\eta[1,1]^\top$	$-[\frac{3\omega^2+1}{2\omega},1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	$[\frac{\omega^2+1}{\omega},0]^\top$	$-\frac{\eta}{\sqrt{2}}[1\frac{2.5\omega^2+1.5}{\sqrt{((3\omega^2+1)/2)^2+(\omega^2+1)^2}},1]^\top$
$\eta[1,-1]^\top$	$[\frac{3\omega^2+1}{2\omega},-1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	$[\frac{\omega^2-1}{2\omega},-1]^\top$	$\eta[\frac{1}{\sqrt{2}}\frac{4\omega^2}{\sqrt{(3\omega^2+1)^2+(\omega^2-1)^2}},-1]^\top$
$-\eta[1,-1]^\top$	$-[\frac{3\omega^2+1}{2\omega},-1]^\top$	$\frac{\frac{\pi}{2}-\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}$	$[\frac{\omega^2+1}{\omega},0]^\top$	$-\frac{\eta}{\sqrt{2}}[\frac{2.5\omega^2+1.5}{\sqrt{((3\omega^2+1)/2)^2+(\omega^2+1)^2}},-1]^\top$

Table 4: Population gradients at the second iteration for Adam, $\beta_1=\beta_2\approx 1$.

Based on the updates in Table 4, we can write the Adam iterate at any time t as follows:

$$\boldsymbol{w}_{k,t}=\boldsymbol{w}_{k,0}+\eta\sum_{\tau=1}^{t}\begin{cases}[1,0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}\\ \frac{1}{\sqrt{\tau}}\left[-\frac{\omega^2-1+(\tau-1)(\omega^2+1)}{\sqrt{(\omega^2-1)^2+(\tau-1)(\omega^2+1)^2}},0\right]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}\\ \frac{1}{\sqrt{\tau}}\left[\frac{\omega^2+1+(\tau-1)(\omega^2-1)}{\sqrt{(\omega^2+1)^2+(\tau-1)(\omega^2-1)^2}},0\right]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}\\ [-1,0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi}\\ [1,1]^\top & \text{w.p. } \frac{1}{8}-\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}\\ \frac{-1}{\sqrt{\tau}}\left[\frac{(\omega^2-1)/2+(\tau-1)(\omega^2+1)}{\sqrt{((\omega^2-1)/2)^2+(\tau-1)(\omega^2+1)^2}},1\right]^\top & \text{w.p. } \frac{1}{8}-\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}\\ [1,-1]^\top & \text{w.p. } \frac{1}{8}-\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}\\ \frac{-1}{\sqrt{\tau}}\left[\frac{(\omega^2-1)/2+(\tau-1)(\omega^2+1)}{\sqrt{((\omega^2-1)/2)^2+(\tau-1)(\omega^2+1)^2}},-1\right]^\top & \text{w.p. } \frac{1}{8}-\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}\\ \left[\frac{1}{\sqrt{\tau}}\frac{3\omega^2+1+(\tau-1)(\omega^2-1)}{\sqrt{(3\omega^2+1)^2+(\tau-1)(\omega^2-1)^2}},1\right]^\top & \text{w.p. } \frac{1}{8}-\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}\\ \frac{-1}{\sqrt{\tau}}\left[\frac{(3\omega^2+1)/2+(\tau-1)(\omega^2+1)}{\sqrt{((3\omega^2+1)/2)^2+(\tau-1)(\omega^2+1)^2}},1\right]^\top & \text{w.p. } \frac{1}{8}-\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}\\ \left[\frac{1}{\sqrt{\tau}}\frac{3\omega^2+1+(\tau-1)(\omega^2-1)}{\sqrt{(3\omega^2+1)^2+(\tau-1)(\omega^2-1)^2}},-1\right]^\top & \text{w.p. } \frac{1}{8}-\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}\\ \frac{-1}{\sqrt{\tau}}\left[\frac{(3\omega^2+1)/2+(\tau-1)(\omega^2+1)}{\sqrt{((3\omega^2+1)/2)^2+(\tau-1)(\omega^2+1)^2}},-1\right]^\top & \text{w.p. } \frac{1}{8}-\frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi}\end{cases}$$

$t\to\infty$ iterations. Based on the analysis above, we can compute $\lim_{t\to\infty}\frac{\boldsymbol{w}_{k,t}}{t}$ for each algorithm. For GD, we have:

$$\lim_{t\to\infty}\frac{\boldsymbol{w}_{k,t}}{t}=\frac{a_k\eta\mu}{2}\left(\omega-\frac{\texttt{sign}(a_k)}{\omega}\right)[1,0]^\top.$$

For Adam with $\beta = 0$ or signGD, we have:

$$\lim_{t\to\infty} \frac{\boldsymbol{w}_{k,t}}{t} = \eta \begin{cases} [1,0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{\pi} \\ [-1,0]^\top & \text{w.p. } \frac{1}{2} \\ [1,1]^\top & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ [1,-1]^\top & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \end{cases}.$$

For Adam with $\beta \approx 1$, using the results in Appendix B, we have:

$$\lim_{t\to\infty} \frac{\boldsymbol{w}_{k,t}}{t} = \eta \begin{cases} [1,0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ [-m_1,0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ [m_2,0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ [-1,0]^\top & \text{w.p. } \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ [1,1]^\top & \text{w.p. } \frac{1}{8} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi} \\ [1,-1]^\top & \text{w.p. } \frac{1}{8} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi} \\ [-m_3,0]^\top & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \\ [m_4,1]^\top & \text{w.p. } \frac{1}{8} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi} \\ [m_4,-1]^\top & \text{w.p. } \frac{1}{8} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{4\pi} \\ [-m_5,0]^\top & \text{w.p. } \frac{1}{4} - \frac{\tan^{-1}\frac{\omega^2-1}{2\omega}}{2\pi} \end{cases},$$

where $m_1, \ldots, m_5$ are constants that satisfy $0.935 \le m_1 \le 1$, $0.923 \le m_2 \le 1$, $0.84 \le m_3 \le 1$, $0.72 \le m_4 \le 1$, $0.98 \le m_5 \le 1$. Taking $s = m_4$ and normalizing each direction then finishes the proof. □

B Auxiliary Results

Lemma 1. *Given a constant $r > 0$ and function $f_r(x) = \frac{x-1+r}{x(x-1+r^2)}$, where $x \ge 1$, it holds that $f_r'(x) \ge 0$ when $x \ge 1+r$. Further, when $x \in \mathbb{N}$, the minima occurs at either $x = 1+\lfloor r \rfloor$ or $x = 2+\lfloor r \rfloor$, and it holds that:*

$$\min\left(\frac{\lfloor r\rfloor+r}{\sqrt{(1+\lfloor r\rfloor)(\lfloor r\rfloor+r^2)}}, \frac{1+\lfloor r\rfloor+r}{\sqrt{(2+\lfloor r\rfloor)(1+\lfloor r\rfloor+r^2)}}\right) \le f_r(x) \le 1.$$

The result can be obtained by examining the derivative of $f_r(x)$ with respect to x, so we omit the proof.

Further, given $r_1 := \frac{\omega^2-1}{\omega^2+1}$, $r_2 := \frac{\omega^2+1}{\omega^2-1} = 1/r_1$, $r_3 := 0.5r_1$, $r_4 := \frac{3\omega^2+1}{\omega^2-1}$, $r_5 := 0.5\frac{3\omega^2+1}{\omega^2+1}$, and $\omega \ge \frac{1+\sqrt{5}}{2}$, it holds that:

$$0.4472 \le r_1 < 1, \quad 1 \le r_2 < 2.236, \quad 0.2236 \le r_3 < 0.5, \quad 3 \le r_4 \le 5.4721, \quad 1.2236 \le r_5 \le 1.5.$$

Alternately, for a specific value of ω, we can compute these exactly. For instance, when $\omega = 2$, $r_1 = 0.6$, $r_2 \approx 1.6667$, $r_3 = 0.3$, $r_4 \approx 4.3333$, $r_5 = 1.3$.

Also, we can simplify the lower bound on $f_r(x)$ as follows:

$$c(r) := \min\left(\frac{\lfloor r\rfloor+r}{\sqrt{(1+\lfloor r\rfloor)(\lceil r\rceil+r^2)}}, \frac{1+\lfloor r\rfloor+r}{\sqrt{(2+\lfloor r\rfloor)(1+\lceil r\rceil+r^2)}}\right) = \begin{cases} \frac{1+r}{\sqrt{2(1+r^2)}}, & 0 < r < 1 \\ \frac{2+r}{\sqrt{3(2+r^2)}}, & 1 < r < 2 \\ \frac{3+r}{\sqrt{4(3+r^2)}}, & 2 < r < 2\sqrt{3} \\ \frac{4+r}{\sqrt{5(4+r^2)}}, & 2\sqrt{3} < r < 4 \\ \frac{5+r}{\sqrt{6(5+r^2)}}, & 4 < r < 5 \\ \frac{6+r}{\sqrt{7(6+r^2)}}, & 5 < r < 6 \end{cases}.$$

We can use this to obtain the exact lower bounds for the aforementioned intervals:

$$\min_r c(r) \approx \begin{cases} 0.9354, & 0.4472 \le r < 1, \\ 0.9238, & 1 \le r < 2.236, \\ 0.8443, & 0.2236 \le r < 0.5, \\ 0.7240, & 3 \le r \le 5.4721, \\ 0.9802, & 1.2236 \le r \le 1.5. \end{cases}$$

When $\omega = 2$, $c(r_1) \approx 0.9713$, $c(r_2) \approx 0.9681$, $c(r_3) \approx 0.8803$, $c(r_4) \approx 0.7808$, $c(r_5) \approx 0.9936$.

Lemma 2. *The sum* $f(x) := \sum_{\tau=1}^{x} \frac{1}{\sqrt{\tau}}$ *satisfies* $2\sqrt{x} - 2 \le f(x) \le 2\sqrt{x} - 1$.

Proof. To establish the bounds for $f(x)$, we can compare the sum to the corresponding integral. We have:

$$f(x) = \sum_{\tau=1}^{x} \frac{1}{\sqrt{\tau}} \ge \int_1^{x+1} \frac{1}{\sqrt{n}}\, dn = 2\sqrt{x+1} - 2,$$

$$f(x) = \sum_{\tau=1}^{x} \frac{1}{\sqrt{\tau}} \le 1 + \int_1^{x} \frac{1}{\sqrt{n}}\, dn = 1 + 2\sqrt{x} - 2 = 2\sqrt{x} - 1.$$

Combining both inequalities and using Using the fact that $\sqrt{x+1} \ge \sqrt{x}$ finishes the proof. □