Effects of Momentum in Implicit Bias of Gradient Flow for Diagonal Linear Networks

Bochen Lyu[1, 2], **He Wang**[3], **Zheng Wang**[4], **Zhanxing Zhu**[*2]
[1]DataCanvas Lab, DataCanvas, Beijing, China
[2]University of Southampton, UK
[3]UCL Centre for Artificial Intelligence, Department of Computer Science, UK
[4]University of Leeds, UK
bochen.lv@gmail.com, he_wang@ucl.ac.uk, z.wang5@leeds.ac.uk, z.zhu@soton.ac.uk

Abstract

This paper targets on the regularization effect of momentum-based methods in regression settings and analyzes the popular diagonal linear networks to precisely characterize the implicit bias of continuous versions of heavy-ball (HB) and Nesterov's method of accelerated gradients (NAG). We show that, HB and NAG exhibit different implicit bias compared to GD for diagonal linear networks, which is different from the one for classic linear regression problem where momentum-based methods share the same implicit bias with GD. Specifically, the role of momentum in the implicit bias of GD is twofold: (a) HB and NAG induce extra initialization mitigation effects similar to SGD that are beneficial for generalization of sparse regression; (b) the implicit regularization effects of HB and NAG also depend on the initialization of gradients explicitly, which may not be benign for generalization. As a result, whether HB and NAG have better generalization properties than GD jointly depends on the aforementioned twofold effects determined by various parameters such as learning rate, momentum factor, and integral of gradients. Our findings highlight the potential beneficial role of momentum and can help understand its advantages in practice such as when it will lead to better generalization performance.

1 Introduction

Extensive deep learning tasks aim to solve the optimization problem

$$\arg\min_{\beta} L(\beta) \tag{1}$$

where L is the loss function and β is the parameter. Gradient descent (GD) and its variants underpin such optimization of parameters for deep learning, thus understanding these simple yet highly effective algorithms is crucial to unveil the thrilling generalization performance of deep neural networks (DNNs). Recently, Soudry et al. (2018); Ji and Telgarsky (2019); Lyu and Li (2020); Pesme, Pillaud-Vivien, and Flammarion (2021); Azulay et al. (2021); Nacson et al. (2019) have made significant efforts in this direction to understand GD-based methods through the lens of *implicit bias*, which states that *GD and its variants are implicitly biased towards selecting particular solutions among all global minimum.*

In particular, Soudry et al. (2018) pioneered the study of implicit bias of GD and showed that GD selects the max-margin solution for logistic regression on separable dataset. For regression problems, the simplest setting is the linear regression problem, where GD and its stochastic variant, SGD, are biased towards the interpolation solution that is closest to the initialization measured by the Euclidean distance (Ali, Dobriban, and Tibshirani 2020). In order to investigate the implicit bias for DNNs, diagonal linear network, a simplified version of deep learning models, has been proposed. For this model, Woodworth et al. (2020); Azulay et al. (2021); Yun, Krishnan, and Mobahi (2021) showed that the solution selected by GD is equivalent to that of a constrained norm minimization problem interpolating between ℓ_1 and ℓ_2 norms up to the magnitude of the initialization scale. Pesme, Pillaud-Vivien, and Flammarion (2021) further characterized that adding stochastic noise to GD additionally induces a regularization effect equivalent to reducing the initialization magnitude.

Besides these fruitful progresses, Gunasekar et al. (2018); Wang et al. (2022) studied the implicit bias of momentum-based methods for one-layer linear models and showed that they have the same implicit bias as GD. Jelassi and Li (2022), on the other hand, revealed that momentum-based methods have better generalization performance than GD for a special linear CNN in classification problems. Ghosh et al. (2023) conducted a model-agnostic analysis of $\mathcal{O}(\eta^2)$ approximate continuous version of HB from the perspective of IGR (Barrett and Dherin 2021). Momentum methods adopt a two-step manner and can induce different dynamics compared to vanilla GD: from the perspective of their $\mathcal{O}(\eta)$-continuous approximation modelling, the approximation for GD is a first-order ODE (gradient flow, GF): $d\beta/dt = -\nabla L(\beta)$, while, as a comparison, the approximation for momentum-based methods can be seen as a damped second-order Hamiltonian dynamic with potential $L(\beta)$:

$$m\frac{d^2\beta}{dt^2} + \lambda\frac{d\beta}{dt} + \nabla L(\beta) = 0,$$

which was first observed by Polyak (1964). Due to the clear difference in their dynamics, it is natural and intriguing to ask from the theoretical point of view:

(Q): *Will adding momentum to GD change its implicit bias for DNNs?*

*Corresponding author

For the least squares problem (single layer linear network), Gunasekar et al. (2018) argued that momentum does not change the implicit bias of GD, while the case for deep learning models is more complex. From the empirical point of view, momentum typically leads to a better generalization performance and is crucial in the training of modern DNNs. This suggests that they might enjoy a different implicit bias compared to GD. Therefore, its theoretical characterization is meaningful and necessary.

Hence, our goal in this work is to precisely characterize the implicit bias of momentum-based methods to take a step towards answering the above fundamental question. To explore the case for deep neural works, we consider the popular deep linear models: *diagonal linear networks*. Although the structures of diagonal linear networks are simple, they already capture many insightful properties of DNNs, including the dependence on the initialization, the over-parameterization of the parameters, the non-convexity, and the transition from lazy regime to rich regime (Woodworth et al. 2020; Pesme, Pillaud-Vivien, and Flammarion 2021; Azulay et al. 2021) which is an intriguing phenomenon observed in many complex neural networks.

Heavy-Ball algorithms (Polyak 1964) (HB) and Nesterov's method of accelerated gradients (Nesterov 1983) (NAG) are the most widely adopted momentum-based methods. These algorithms are generally implemented with a fixed momentum factor in deep learning libraries such as PyTorch (Paszke et al. 2017). To be consistent with such practice, we focus on HB and NAG with a fixed momentum factor that is independent of learning rate or iteration count. For the purpose of conducting a tractable theoretical analysis, we rely on the tools of continuous time approximations of momentum-based methods (with fixed momentum factor), HB and NAG flow, which were recently interpreted by Kovachki and Stuart (2021) as *modified equations* in the numerical analysis literature and by Shi et al. (2018) as high resolution ODE approximation.

Our findings are summarized as follows. We show that, *unlike* the case for single layer linear networks where momentum-based methods HB and NAG share similar implicit bias with GF, they exhibit different implicit bias for diagonal linear networks compared to GF in two main aspects:

1. Compared to GF, although HB and NAG flow also converge to solutions that minimize a norm interpolating between ℓ_2-norm and ℓ_1-norm up to initialization scales of parameters, they induce an *extra effect that is equivalent to mitigating the influence of initialization of the model parameters*, which is beneficial for generalization properties of sparse regression (Theorem 3). Note that stochastic gradient flow (SGF) could also yield an initialization mitigation effects (Pesme, Pillaud-Vivien, and Flammarion 2021), although momentum-based methods and SGF modify GF differently.
2. The solutions of HB and NAG flow also *depend on the initialization of both parameters and gradients explicitly and simultaneously*, which may not be benign for the generalization performances for sparse regression.

Therefore, HB and NAG are not always better than GD from the perspective of generalization for sparse regression. Whether HB and NAG have the advantages of generalization over GD is up to the overall effects of the above two distinct effects determined by various hyper-parameters. In particular, when mitigation effects of initialization of parameters brought by HB and NAG outperform their dependence on the initialization of gradients, HB and NAG will have better generalization performances than GD, e.g., when the initialization is highly biased (Fig. 2(a)), otherwise, they will not show such advantages over GD (Fig. 1(a)).

Organization. This paper is organized as follows. In Section 2, we summarize notations, setup, and continuous time approximation modelling details for HB and NAG. Section 3 concentrates on our main results of the implicit bias of momentum-based methods for diagonal linear networks with corresponding numerical experiments to support our theoretical findings. We conclude this work in Section 4. All the proof details and additional experiments are presented in Appendix.

1.1 Related works

To characterize the properties of momentum-based methods, continuous time approximations of them are introduced in several recent works. Su, Boyd, and Candes (2014) provided a second-order ODE to precisely describe the NAG with momentum factor depending on the iteration count. Wilson, Recht, and Jordan (2016) derived a limiting equation for both HB and NAG when the momentum factor depends on learning rate or iteration count. Shi et al. (2018) further developed high-resolution limiting equations for HB and NAG, and Wibisono, Wilson, and Jordan (2016) designed a general framework from the perspective of Bregman Lagrangian. When the momentum is fixed and does not depend on learning rate or iteration count, Kovachki and Stuart (2021) developed the continuous time approximation, the modified equation in the numerical analysis literature, for HB and NAG.

Compared to these works, we develop the continuous time approximations of HB and NAG for deep learning models and regression problems, and we focus on the implicit bias of HB and NAG flow rather than GF. Furthermore, we also take into account the effects of other sources of implicit bias such as initialization and model architecture, which is different from the model-agnostic analysis in Ghosh et al. (2023).

The recent work Papazov, Pesme, and Flammarion (2024) also studied HB for diagonal linear networks using continuous approximation where the initialization of speed of parameters are all zero. In particular, they characterized a crucial quantity $\eta(1-\mu)^{-2}$ that can induce acceleration of the optimization of HB and can make the solution of HB generalize better when $\eta(1-\mu)^{-2}$ is sufficiently small. As a comparison, we do not impose restriction to initialization and characterize the implicit bias for both HB and NAG flow. In addition, we reveal the role of initialization of gradients and show when HB and NAG flow can generalize better/worse than GF.

2 Preliminaries

Notations. We let $\{1, \dots, L\}$ be all integers between 1 and L. The dataset with n samples is denoted by $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ is the d-dimensional input and $y_i \in \mathbb{R}$ is the scalar output. The data matrix is represented by $X \in \mathbb{R}^{n \times d}$ where each row is a feature x_i and $y = (y_1, \dots, y_n)^T \in \mathbb{R}^n$ is the collection of y_i. For a vector $a \in \mathbb{R}^d$, a_j denotes its j-th component and its ℓ_p-norm is $\|a\|_p$. For a vector $a(t)$ depending on time, we use $\dot{a}(t) = da/dt$ to denote the first time derivative and $\ddot{a}(t) = d^2a/dt^2$ for the second time derivative. The element-wise product is denoted by $\odot$ such that $(a \odot b)_j = a_j b_j$. We let $\mathbf{e}_d = (1, \dots, 1)^T \in \mathbb{R}^d$. For a square matrix $W \in \mathbb{R}^{d \times d}$, we use $\mathrm{diag}(W)$ to denote the corresponding vector $(W_{11}, \dots, W_{dd})^T \in \mathbb{R}^d$.

Heavy-Ball and Nesterov's method of accelerated gradients. Heavy-Ball (HB) and Nesterov's method of accelerated gradients (NAG) are perhaps the most widely adopted momentum-based methods. Different from GD, HB and NAG apply a two-step scheme (Sutskever et al. 2013). In particular, for Eq. (1) let k be the iteration number, μ be the momentum factor, η be the learning rate, and $p \in \mathbb{R}^d$ be the momentum of parameter β, then HB updates β as follows:

$$p_{k+1} = \mu p_k - \eta \nabla L(\beta_k),\ \beta_{k+1} = \beta_k + p_{k+1} \qquad (2)$$

where $p_0 = 0$. Similarly, NAG can also be written as a two-step manner

$$p_{k+1} = \mu p_k - \eta \nabla L(\beta_k + \mu p_k),\ \beta_{k+1} = \beta_k + p_{k+1} \quad (3)$$

with $p_0 = 0$. Note that although previous works (Su, Boyd, and Candes 2014; Nesterov 2014; Wilson, Recht, and Jordan 2016; Shi et al. 2018) considered HB and NAG with momentum factor depending on the learning rate η or iteration count k, HB and NAG are generally implemented with constant momentum factor such as in PyTorch (Paszke et al. 2017). Therefore a constant momentum factor μ is assumed in this work as in Kovachki and Stuart (2021) to be consistent with such practice.

HB and NAG flow: continuous time approximations of HB and NAG. In this work, we analyze the implicit bias of HB and NAG through their continuous time approximations summarized as follows, which provide insights to the corresponding discrete algorithms and enable us to take the great advantages of the convenience of theoretical analysis at the same time. We start with the definition of order of convergence of continuous approximation for discrete HB and NAG.

Definition 1 (Order of convergence of HB and NAG flow)**.** *An ODE whose solution is $\beta(t)$ is the order $\mathcal{O}(\eta^\gamma)$ continuous approximate version of the discrete HB Eq.* (2) *and NAG Eq.* (3) *if for $k = 0, 1, 2, \dots$, let $\bar{\beta}_k$ be the sequence given by Eq.* (2) *or Eq.* (3) *and let $\beta_k = \beta(t = k\eta)$, then for any $T \geq 0$, there exists a constant $C > 0$ such that $\sup_{0 \leq k\eta \leq T} |\beta_k - \bar{\beta}_k| \leq C\eta^\gamma$.*

Proposition 1 (HB and NAG flow: $\mathcal{O}(\eta)$ continuous approximate version of HB and NAG)**.** *For the model $f(x; \beta)$ with empirical loss function $L(\beta)$, let $\mu \in (0, 1)$ be the fixed momentum factor and η be the learning rate, the $\mathcal{O}(\eta)$ continuous approximate versions of the discrete HB (Eq.* (2)*) and NAG (Eq.* (3)*) are of the form*

$$\alpha \ddot{\beta} + \dot{\beta} + \frac{\nabla L(\beta)}{1 - \mu} = 0, \qquad (4)$$

where $\alpha = \frac{\eta(1+\mu)}{2(1-\mu)}$ for HB, and $\alpha = \frac{\eta(1-\mu+2\mu^2)}{2(1-\mu)}$ for NAG.

Eq. (4) follows from Kovachki and Stuart (2021) and we present an alternative proof in Appendix. Note that since the learning rate η is small, Proposition 1 indicates that, for the model parameter β, modifying GD with fixed momentum is equivalent to perturb the re-scaled gradient flow equation $d\beta/dt = \nabla L(\beta)/(1 - \mu)$ by a small term proportional to η. More importantly, this modification term offers us considerably more qualitative understanding regarding the dynamics of momentum-based methods than the re-scaled gradient flow, which will become more significant for large learning rate—a preferable choice in practice.

Over-parameterized regression. We consider the regression problem for the n-sample dataset $\{(x_i, y_i)\}_{i=1}^n$ where $n < d$ and assume the existence of the perfect solution, i.e., there exist interpolation solutions $\beta^* \in \mathbb{R}^d$ such that $\forall i \in \{1, \dots, n\} : x_i^T \beta^* = y_i$. For the parametric model $f(x; \beta) = \beta^T x$, we use the quadratic loss $\ell_i = (f(x_i; \beta) - y_i)^2$ and the empirical loss $L(\beta)$ is

$$L(\beta) = \frac{1}{2n} \sum_{i=1}^n \ell_i(\beta) = \frac{1}{2n} \sum_{i=1}^n (f(x_i; \beta) - y_i)^2. \qquad (5)$$

Diagonal linear networks. The diagonal linear network (Woodworth et al. 2020) is a popular proxy model for DNNs. It corresponds to an equivalent linear predictor $f(x; \beta) = \theta^T x$, where $\theta = \theta(\beta)$ is parameterized by the model parameters β. For the diagonal linear networks considered in this paper, we study the 2-layer diagonal linear network, which corresponds to the parameterization[1] of $\theta = u \odot u - v \odot v$ in the sense that

$$f(x; \beta) := f(x; u, v) = (u \odot u - v \odot v)^T x, \qquad (6)$$

and the model parameters are $\beta = (u, v)$, where $u, v \in \mathbb{R}^d$. We slightly abuse the notation of $L(\theta) = L(\beta)$. Our goal in this paper is to characterize the implicit bias of HB and NAG by precisely *capturing the property of the limit point of θ and its dependence on various parameters* such as the learning rate and the initialization of parameters for diagonal linear networks $f(x; \beta)$ trained with HB and NAG.

The use of $\mathcal{O}(\eta)$ order of convergence. The main reason why we use the $\mathcal{O}(\eta)$ continuous approximation for HB and NAG is that we aim to precisely characterize the the role of momentum in the implicit bias of the widely-studied GF, which is the $\mathcal{O}(\eta)$ approximate continuous version of GD, for diagonal linear networks. The same order of approximations ($\mathcal{O}(\eta)$ in our case) for both momentum-based methods and GD should be used to make a "fair" comparison on their implicit bias.

[1] A standard diagonal linear network is $\theta = u \odot v$, which is shown in (Woodworth et al. 2020) to be equivalent to our parameterization here. We further discuss this in Appendix.

3 Implicit Bias of HB and NAG Flow for Diagonal Linear Nets

To clearly reveal the difference between the implicit bias of (S)GD and momentum-based methods, we start with discussing existing results under the unbiased initialization assumption, and our main result is summarized in Theorem 3. We then discuss the dynamics of θ for diagonal linear networks trained with HB and NAG flow, which is necessary for the proof of Theorem 3 and may be of independent interest.

For convenience, given a diagonal linear network Eq. (6), let $\xi = (\xi_1, \dots, \xi^d) \in \mathbb{R}^d$ where $\forall i \in \{1, \dots, d\} : \xi_j = |u_j(0)||v_j(0)|$ measures the scale of the initialization, we first present the definition of the unbiased initialization assumed frequently in previous works (Woodworth et al. 2020; Azulay et al. 2021; Pesme, Pillaud-Vivien, and Flammarion 2021).

Definition 2 (Unbiased initialization for diagonal linear networks)**.** *The initialization for the diagonal linear network Eq.* (6) *is unbiased if $u(0) = v(0)$, which implies that $\theta(0) = 0$ and $\xi = u(0) \odot v(0)$.*

Implicit bias of GF. Recently, Woodworth et al. (2020); Azulay et al. (2021) showed that, for diagonal linear network with parameterization Eq. (6), if the initialization is unbiased (Definition 2) and $\theta(t) = u(t) \odot u(t) - v(t) \odot v(t)$ converges to the interpolation solution, i.e., $\forall i \in \{1, \dots, n\} : \theta^T(\infty) x_i = y_i$, then under GF $\theta^{\mathrm{GF}}(\infty)$ implicitly solves the constrained optimization problem: $\theta^{\mathrm{GF}}(\infty) = \arg\min_\theta \ Q_\xi^{\mathrm{GF}}(\theta), \ s.t. \ X\theta = y$, where $Q_\xi^{\mathrm{GF}}(\theta) = \sum_{j=1}^d \left[\theta_j \operatorname{arcsinh}\left(\theta_j/(2\xi_j)\right) - \sqrt{4\xi_j^2 + \theta_j^2} + 2\xi_j\right]/4$. The form of $Q_\xi^{\mathrm{GF}}(\theta)$ highlights the transition from kernel regimes to rich regimes of diagonal linear networks under gradient flow up to different scales of the initialization: the initialization $\xi \to \infty$ corresponds to the *kernel regime* or *lazy* regime where $Q_\xi^{\mathrm{GF}}(\theta) \propto \|\theta\|_2^2$ and the parameters only move slowly during training, and $\xi \to 0$ corresponds to the *rich regime* where $Q_\xi^{\mathrm{GF}}(\theta) \to \|\theta\|_1$ and the corresponding solutions enjoy better generalization properties for sparse regression. For completeness, we characterize the implicit bias of GF without requiring the unbiased initialization (Definition 2) in the following proposition.

Proposition 2 (Implicit bias of GF for diagonal linear net with biased initialization)**.** *For diagonal linear network Eq.* (6) *with biased initialization ($u(0) \neq v(0)$), if $u(t)$ and $v(t)$ follow the gradient flow dynamics for $t > 0$, i.e., $\dot{u} = -\nabla_u L$ and $\dot{v} = -\nabla_v L$, and if the solution converges to the interpolation solution, then*

$$\theta(\infty) = \arg\min_\theta Q_\xi^{\mathrm{GF}}(\theta) + \theta^T \mathcal{R}^{\mathrm{GF}}, \ s.t. \ X\theta = y \quad (7)$$

where $\mathcal{R}^{\mathrm{GF}} = (\mathcal{R}_1^{\mathrm{GF}}, \dots, \mathcal{R}_d^{\mathrm{GF}})^T \in \mathbb{R}^d$, $\forall j \in \{1, \dots, d\} : \mathcal{R}_j^{\mathrm{GF}} = \operatorname{arcsinh}\left(\theta_j(0)/2\xi_j\right)/4$.

Compared to the unbiased initialization case, besides Q_ξ^{GF}, an additional term $\mathcal{R}^{\mathrm{GF}}$ that depends on $\theta(0)$ is required to capture the implicit bias when the initialization is biased. Note that $\mathcal{R}^{\mathrm{GF}}$ indicates that $\theta(\infty)$ also depends on the direction of the initialization and serves as a kind of self-regularization.

3.1 Implicit bias of HB and NAG flow

Gunasekar et al. (2018); Wang et al. (2022) argued that momentum does not change the implicit bias of GF for single layer model for both linear regression and classification. For DNNs, will modifying GF with the widely adopted momentum change the implicit bias? If it does, will momentum-based methods lead to solutions that have better generalization properties? In the following, we characterize the implicit bias of HB and NAG flow (Proposition 1) for diagonal linear networks to compare with that of GF and answer these questions. For completeness, we do not require the unbiased initialization $u(0) = v(0)$ condition and let $\exp(a) \in \mathbb{R}^d$ denote the vector $(e^{a_1}, \dots, e^{a_d})^T$ for a vector $a \in \mathbb{R}^d$. We now present our main theorem.

Theorem 3 (Implicit bias of HB and NAG flow for diagonal linear networks)**.** *For diagonal linear network Eq.* (6)*, let $\mathcal{R}^{\mathrm{M}} = (\mathcal{R}_1^{\mathrm{M}}, \dots, \mathcal{R}_d^{\mathrm{M}})^T \in \mathbb{R}^d$, if $u(t)$ and $v(t)$ follow the $\mathcal{O}(\eta)$ approximate continuous version of HB and NAG Eq.* (4) *for $t \geq 0$ and if the solution $\theta(\infty) = u(\infty) \odot u(\infty) - v(\infty) \odot v(\infty)$ converges to the interpolation solution, then, neglecting all terms of the order $\mathcal{O}(\eta^2)$,*

$$\theta(\infty) = \arg\min_\theta Q_{\bar{\xi}(\infty)}^{\mathrm{M}}(\theta) + \theta^T \mathcal{R}^{\mathrm{M}}, \ s.t. \ X\theta = y \quad (8)$$

where we define $\bar{\xi}(\infty) = \xi \odot \exp\left(-\alpha\phi(\infty)\right)$, and $\forall j \in \{1, \dots, d\}$:

$$\mathcal{R}_j^{\mathrm{M}} = \frac{1}{4} \operatorname{arcsinh}\left(\frac{\theta_j(0)}{2\xi_j} + \frac{4\alpha\partial_{\theta_j} L(\theta(0))}{1-\mu}\sqrt{1 + \frac{\theta_j^2(0)}{4\xi_j^2}}\right)$$

$$\phi(\infty) = \frac{8}{(1-\mu)^2} \int_0^\infty \nabla_\theta L\left(\theta(s)\right) \odot \nabla_\theta L\left(\theta(s)\right) ds,$$

$$Q_{\bar{\xi}(\infty)}^{\mathrm{M}}(\theta) = \frac{1}{4} \sum_{j=1}^d \Big[\theta_j \operatorname{arcsinh}\left(\frac{\theta_j}{2\bar{\xi}_j(\infty)}\right) - \sqrt{4\bar{\xi}_j^2(\infty) + \theta_j^2} + 2\bar{\xi}_j(\infty)\Big]. \quad (9)$$

Specifically, α is chosen as $\frac{\eta(1+\mu)}{2(1-\mu)}$ if we run HB and $\alpha = \frac{\eta(1-\mu+2\mu^2)}{2(1-\mu)}$ for NAG.

Remark. The $Q_{\bar{\xi}(\infty)}^{\mathrm{M}}$ part for HB and NAG flow has a formulation similar to Q_ξ^{GF} of GF: both of them are the *hyperbolic entropy* (Ghai, Hazan, and Singer 2020). The transition from kernel regime to rich regime by decreasing ξ from ∞ to 0 also exists for HB and NAG (see Appendix). The difference between $Q_{\bar{\xi}(\infty)}^{\mathrm{M}}$ and Q_ξ^{GF} lies in that HB and NAG flow induce an extra initialization mitigation effect: given ξ, $Q_{\bar{\xi}(\infty)}^{\mathrm{M}}$ for HB and NAG flow is equivalent to the hyperbolic entropy of GF with a smaller initialization scale since $\bar{\xi}(\infty)$ is strictly smaller than ξ due to the fact that $\phi(\infty)$ is a positive integral and finite. As a result, $Q_{\bar{\xi}(\infty)}^{\mathrm{M}}$ is closer to an ℓ_1-norm of θ than Q_ξ^{GF}. Furthermore, compared to the implicit

bias of GF when the initialization is biased (Proposition 2), an additional term in $\mathcal{R}^{\mathrm{M}}$ that depends on the initialization of gradient explicitly is required to capture the implicit bias of HB and NAG flow. Such dependence is as expected since the first step update of momentum methods simply assigns the initialization of gradient to the momentum, which is crucial for the following updates. Therefore, Theorem 3 takes an important step towards positively answering our fundamental question **(Q)** in the sense that *momentum changes the implicit bias of GD for diagonal linear networks*.

A natural question following the fact that HB and NAG flow induce different implicit bias compared to GF is: will this difference lead to better generalization properties of HB and NAG? The implicit bias of HB and NAG flow is captured by two distinct parts, the hyperbolic entropy $Q^{\mathrm{M}}_{\tilde{\xi}(\infty)}$ and $\mathcal{R}^{\mathrm{M}}$, where the effects of momentum on $Q^{\mathrm{M}}_{\tilde{\xi}(\infty)}$ is beneficial for generalization while the effects on $\mathcal{R}^{\mathrm{M}}$ may hinder the generalization performance and is affected by the biased initialization. Thus the answer highly depends on various conditions.

Due to the aforementioned initialization mitigation effects of HB and NAG, GD with a smaller initialization might achieve a similar regularization effect as HB and NAG. The main harm, however, of using GD with a smaller initialization is the saddle point escape issue: very small initialization scales lead to the issue that these scales correspond to the initialization highly close to a saddle point (here $u = v = 0$) that might be difficult to escape. This reveals the benefit of the extra effects brought by momentum, i.e., avoiding the saddle point escape problem by using a relatively large initialization to achieve good generalization.

In the following, we present a detailed analysis with corresponding numerical experimental results to compare the implicit bias of GF and that of HB and NAG flow for the case of both unbiased and biased initialization, respectively.

3.2 Comparison of HB/NAG flow and (S)GF for unbiased initialization

When the initialization is unbiased, it is worth to mention that a recent work (Pesme, Pillaud-Vivien, and Flammarion 2021) studied the stochastic version of gradient flow, the stochastic gradient flow (SGF), and revealed that the existence of sampling noise changes the implicit bias of GF in the sense that $\theta^{\mathrm{SGF}}(\infty) = \arg\min_\theta Q^{\mathrm{SGF}}_{\tilde{\xi}_\infty}(\theta)$ under the constraint $X\theta = y$, where

$$Q^{\mathrm{SGF}}_{\tilde{\xi}(\infty)}(\theta) = \sum_{j=1}^{d} \frac{1}{4}\Big[\theta_j \operatorname{arcsinh}\left(\frac{\theta_j}{2\tilde{\xi}_j(\infty)}\right) - \sqrt{4\tilde{\xi}_j^2(\infty) + \theta_j^2} + 2\tilde{\xi}_j(\infty)\Big] \tag{10}$$

with $\tilde{\xi}(\infty)$ being strictly smaller than ξ. The remarkable point appears when we compare $Q^{\mathrm{M}}_{\tilde{\xi}_\infty}$ with $Q^{\mathrm{SGF}}_{\tilde{\xi}(\infty)}$: although SGF and momentum-based methods modify GF differently, i.e., SGF adds stochastic sampling noise while momentum-based methods add momentum to GF, *both of them induce an effect equivalent to reducing the initialization scale!* The difference between them lies in the way how they control such initialization mitigation effect. For SGF this is controlled by the integral of loss function, while the effect depends on the integral of gradients for HB and NAG flow.

To show the difference between (S)GF and momentum-based methods HB and NAG flow, we note that $\mathcal{R}^{\mathrm{M}}_j$ in Theorem 3 becomes

$$\mathcal{R}^{\mathrm{M}}_j = \operatorname{arcsinh}\left(\frac{4\alpha(X^T y)_j}{n(1-\mu)}\right),$$

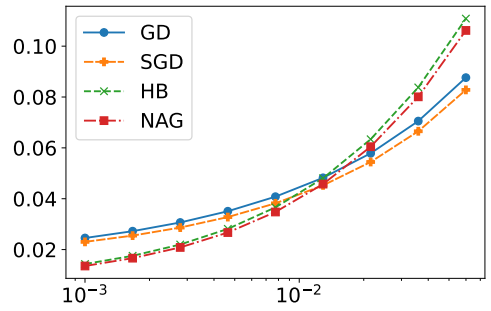
which is also determined by the dataset, and $\mathcal{R}^{\mathrm{GF}}$ in Proposition 2 is simply zero. Therefore, as long as the initialization of gradients $X^T y = o(\alpha^{-1} n(1-\mu))$, i.e., $\mathcal{R}^{\mathrm{M}}$ is small compared to $Q^{\mathrm{M}}_{\tilde{\xi}(\infty)}$ such that only $Q^{\mathrm{M}}_{\tilde{\xi}(\infty)}$ matters for characterizing the implicit bias, HB and NAG flow will exhibit better generalization properties for sparse regression due to the initialization mitigation effects of HB and NAG flow that lead $Q^{\mathrm{M}}_{\tilde{\xi}(\infty)}$ to be closer to the ℓ_1-norm of θ than Q^{GF}_{ξ}. On the other hand, when $\mathcal{R}^{\mathrm{M}}_j$ is not small compared to $Q^{\mathrm{M}}_{\tilde{\xi}(\infty)}$, the initialization mitigation effects of HB and NAG flow may not be significant, thus there may not be generalization benefit for HB and NAG flow.

To summarize, for unbiased initialization, HB and NAG outperform GD regarding the generalization when $\alpha X^T y$ is much smaller than $n(1-\mu)$ and they would have worse generalization performance than GD otherwise. In the following, we conduct numerical experiments to verify this claim.

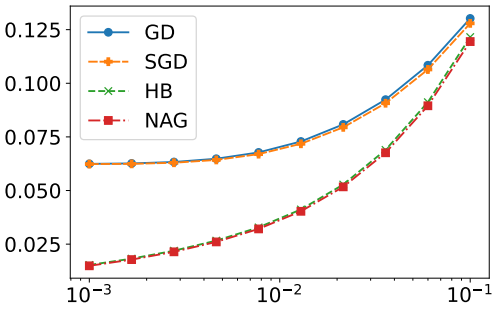
Numerical Experiments. We consider the over-parameterized sparse regression. For the dataset $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, we set $n = 40, d = 100$ and $x_i \sim \mathcal{N}(0, I)$. y_i is generated by $y_i = x_i^T \theta^*$ where $\theta^* \in \mathbb{R}^d$ is the ground truth solution. We let 5 components of θ^* be non-zero. Our models are 2-layer diagonal linear networks $f(x; \beta) = u \odot u - v \odot v$. We use $\|\xi\|_1$ to measure the scale of initialization. The initialization of parameters is unbiased by letting $u(0) = v(0) = c\mathbf{e}_d$ where c is a constant and $\|\xi\|_1 = c^2 d$. We consider training algorithms GD, SGD, HB, and NAG. And the generalization performance of the solution for each training algorithm is measured by the distance $D(\theta(\infty), \theta^*) = \|\theta(\infty) - \theta^*\|_2^2$. Since $\mathcal{R}^{\mathrm{M}}$ is determined by the dataset, to control its magnitude, we build two new datasets $\mathcal{D}_\varepsilon = \{(x_{i;\varepsilon}, y_{i;\varepsilon})\}_{i=1}^d$ where $\forall i \in \{1 \ldots, d\} : x_{i;\varepsilon} = \varepsilon x_i,\ y_{i;\varepsilon} = \varepsilon y_i$. We then train diagonal linear networks using GD and momentum-based methods HB and NAG on each dataset, respectively, and learning rate $\eta = 3 \times 10^{-2}$ and momentum factor $\mu = 0.9$. As shown in Fig. 1, as we decrease the value of ε which decreases the magnitude of $\mathcal{R}^{\mathrm{M}}$, the generalization benefit of HB and NAG becomes more significant since their initialization mitigation effects are getting more important. Note that Fig. 1 also reveals the transition to rich regime by decreasing the initialization scales.

3.3 Comparison of HB/NAG flow and GF for biased initialization

If the initialization is biased, i.e., $u(0) \neq v(0)$, both the implicit bias of GF and that of HB and NAG flow additionally

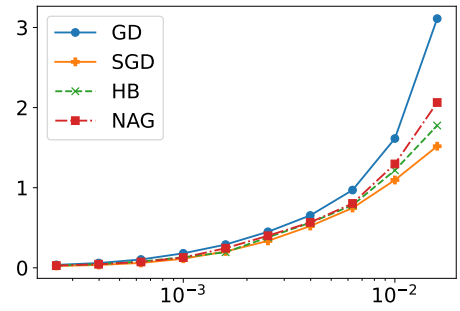


(a) $\varepsilon = 0.6$

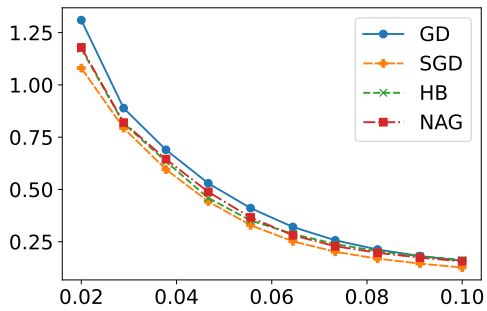


(b) $\varepsilon = 0.2$

Figure 1: $D(\theta(\infty), \theta^*)$ for diagonal linear networks with unbiased initialization trained with different algorithms and ε (smaller ε for smaller $\nabla L(\theta(0))$). $\mathbf{x}$-axis denotes $\|\xi\|_1$.



(a) $\mathbf{x}$-axis denotes $\|\xi\|_1$



(b) $\mathbf{x}$-axis denotes φ

Figure 2: $D(\theta(\infty), \theta^*)$ for diagonal linear networks with biased initialization trained with different algorithms and **(a).** different initialization scales with $\varphi = 0.03$; **(b).** different extents of the biased part of the initialization (smaller φ implies more biased initialization) and $\|\xi\|_1 = 0.0046$.

depend on $\theta(0)$ ($\mathcal{R}^{\mathrm{GF}}$ for GF in Proposition 2 and $\mathcal{R}^{\mathrm{M}}$ in Theorem 3 for HB and NAG flow) besides the hyperbolic entropy. Compared to $\mathcal{R}^{\mathrm{GF}}$, $\mathcal{R}^{\mathrm{M}}$ also includes the explicit dependence on the initialization of gradient that is proportional to $\alpha\nabla L(\theta(0))$. Therefore, recall that α is the order of η, if $\nabla L(\theta(0)) = o(\alpha^{-1}n(1-\mu))$ and $\alpha\nabla L(\theta(0))$ is small compared to $\theta(0)$, then $\mathcal{R}^{\mathrm{GF}}$ is close to $\mathcal{R}^{\mathrm{M}}$, leading to the fact that the difference between the implicit bias of GF and that of HB and NAG flow are mainly due to the initialization mitigation effects of HB and NAG. As a result, we can observe the generalization advantages of HB and NAG over GF (Fig. 2(a)). However, when the initialization is only slightly biased, i.e., $u(0) \neq v(0)$ and $u(0)$ is close to $v(0)$, the dependence on $\nabla L(\theta(0))$ of the solutions of HB and NAG is important and the generalization benefit of HB and NAG for sparse regression may disappear.

Numerical Experiments. We use the same dataset $\{(x_i, y_i)\}_{i=1}^d$ as in Section 3.2. We set $\eta = 10^{-1}$ and $\mu = 0.9$. To characterize the influence of the extent of the biased part of the initialization, we let $u(0) = \varphi c\mathbf{e}_d$ and $v(0) = \varphi^{-1}c\mathbf{e}_d$ where $\varphi \in (0, 1]$ is a constant measuring the extent of the unbiased part of the initialization. In this way, for any φ, we have $\xi_j = |u_j(0)||v_j(0)| = c^2$. In order to verify the above theoretical claims, we conduct two sets of experiments: (i). We train diagonal linear network with different algorithms for different scales of initialization $\|\xi\|_1$ and fixed φ. As shown in Fig. 2(a), as a result of the initialization mitigation effects, HB, NAG, and SGD exhibit better generalization performance than GD for sparse regression. (ii). We fix $\|\xi\|_1$ and train diagonal linear networks with different biased initialization (different values of φ). As shown in Fig. 2(b), as we increasing φ, the initialization becomes less biased and the extra dependence on the initialization of gradient of HB and NAG outperforms their initialization mitigation effects, and, as a result, the generalization benefits of momentum-based methods disappear.

3.4 Dynamics for θ of diagonal linear networks under HB and NAG Flow

For diagonal linear networks Eq. (6), dynamics for θ under HB and NAG flow is crucial to the proof of Theorem 3, and may be of independent interest. Interestingly, different from diagonal linear networks under gradient flow where θ follows a mirror flow or stochastic gradient flow where θ follows a stochastic mirror flow with time-varying potential, due to the second-order ODE nature of HB and NAG flow as formulated in Eq. (4), θ does not directly follow a mirror flow. Instead, HB and NAG flow is special—it is $\theta + \alpha\dot{\theta}$ that

follows a mirror flow form with time-varying potential, as shown below.

Proposition 4 (Dynamics of θ for diagonal nets trained with HB and NAG flow)**.** *For diagonal linear networks Eq.* (6) *trained with HB and NAG flow (Eq.* (4)*) and initialized as* $u(0) = v(0)$ *and* $u(0) \odot u(0) = \xi \in \mathbb{R}^d$*, let* $\bar{\theta}_\alpha := \theta + \alpha\dot{\theta} \in \mathbb{R}^d$ *and its* j*-th component be* $\bar{\theta}_{\alpha;j}$*, then* $\bar{\theta}_\alpha$ *follows a mirror flow form with time-varying potential (*$\mathcal{R}^{\mathrm{M}}$ *is defined in Theorem 3) in the sense that* $\forall j \in \{1, \ldots, d\}$:

$$\frac{d}{dt}\nabla\left[Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t) + \bar{\theta}_{\alpha;j}\mathcal{R}^{\mathrm{M}}_j\right] = -\frac{\partial_{\theta_j} L(\theta)}{1-\mu}, \tag{11}$$

where $Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t)$ *is given by*

$$Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t) = \frac{1}{4}\Big[\bar{\theta}_{\alpha;j}\,\mathrm{arcsinh}\left(\frac{\bar{\theta}_{\alpha;j}}{2\bar{\xi}_j(t)}\right) - \sqrt{4\bar{\xi}^2_j(t) + \bar{\theta}^2_{\alpha;j}} + 2\bar{\xi}_j(t)\Big]$$

and $\bar{\xi}_j(t) = \xi_j e^{-\alpha\phi_j(t)}$ *with*

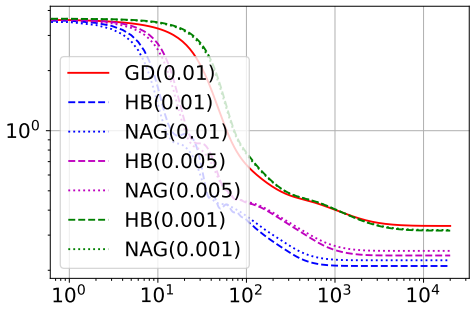
$$\phi_j(t) = \frac{8}{(1-\mu)^2}\int_0^t \partial_{\theta_j} L\left(\theta(s)\right)\partial_{\theta_j} L\left(\theta(s)\right) ds. \tag{12}$$

Compared to the mirror flow form of diagonal linear networks under gradient flow $d\nabla Q^{\mathrm{GF}}_\xi(\theta)/dt = -\nabla L(\theta)$, there are three main differences in Eq. (11): (i). It is a second-order ODE since Eq. (11) can be written as $\alpha\nabla^2 Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t)\ddot{\theta}_j + \nabla^2 Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t)\dot{\theta}_j + \frac{\partial \nabla Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t)}{\partial t} + \frac{\partial_{\theta_j} L(\theta)}{1-\mu} = 0$, while the dynamics of GF is a first-order ODE; (ii). It is $\bar{\theta}_\alpha$, not θ, appears in the mirror flow potential for diagonal linear networks under HB and NAG flow, and an extra term depending on the initialization of gradients is included; (iii). The hyperbolic entropy part of the mirror flow potential $Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t)$ under HB and NAG flow is a time-varying one, and the time-varying part mainly mitigates the influence of the initialization ξ ($\bar{\xi}_j(t) \le \xi$ for any $t \ge 0$).

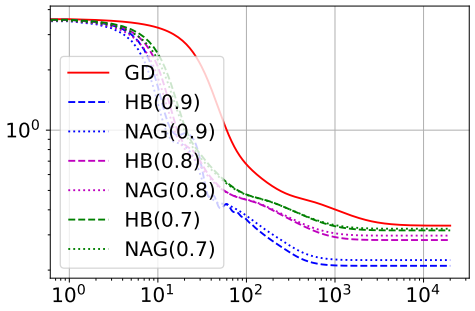
3.5 Effects of Hyper-parameters for Implicit Bias

As a result of the fact that momentum-based methods (HB and NAG) add a perturbation proportional to the learning rate η to re-scaled gradient flow (Proposition 1), the difference of their implicit bias depends on η: the limit $\eta \to 0$ leads to $\bar{\xi}(\infty) \to \xi$ and, as a consequence, $Q^{\mathrm{M}}_{\bar{\xi}(\infty)} \to Q^{\mathrm{GF}}_\xi$. Therefore, for small learning rate, the implicit bias of momentum-based methods and that of GD are almost the same. This observation coincides with the experience of Rumelhart, Hinton, and Williams (1986); Kovachki and Stuart (2021) that setting momentum factor as 0 returns the same solution as reducing the learning rate when momentum factor is non-zero. The discrepancy between the implicit bias of momentum-based methods and that of GD becomes significant for moderate learning rate and momentum factor.

To verify this, we set $\|\xi\|_1 = 0.1240$, and run: (i). GD with $\eta = 10^{-2}$; (ii). HB and NAG with $\mu = 0.9$ and different η; (iii). HB and NAG with $\eta = 10^{-2}$ and different μ. We present the generalization performance $D(\theta(t), \theta^*)$ during training for each algorithm with its corresponding training parameters in Fig. 3(a) and Fig. 3(b). These results clearly reveal that both decreasing the learning rate and the momentum factor make the difference between the implicit bias of momentum-based methods and that of GD not significant. Experimental details are in Appendix.



(a)



(b)

Figure 3: Diagonal nets trained with different algorithms and hyper-parameters: **(a).** $D(\theta(t), \theta^*)$ for different η (numbers in the brackets) and $\mu = 0.9$. **(b).** $D(\theta(t), \theta^*)$ for different μ (numbers in the brackets) and $\eta = 0.01$. $\mathbf{x}$-axis denotes iterations.

4 Conclusion

In this paper, we have targeted on the unexplored regularization effect of momentum-based methods and we have shown that, *unlike* the single layer linear network, momentum-based methods HB and NAG flow exhibit different implicit bias compared to GD for diagonal linear networks. In particular, we reveal that HB and NAG flow induce an extra initialization mitigation effect similar to SGD that is beneficial for generalization of sparse regression and controlled by the integral of the gradients, learning rate, data matrix, and the momentum factor. In addition, the implicit bias of HB and NAG flow also depends on the initialization of both parameters and gradients explicitly, which may also hinder the generalization, while GD and SGD only depend on the initialization of parameters.

References

Ali, A.; Dobriban, E.; and Tibshirani, R. 2020. The Implicit Regularization of Stochastic Gradient Flow for Least Squares. In *International Conference on Machine Learning*.

Azulay, S.; Moroshko, E.; Nacson, M. S.; Woodworth, B.; Srebro, N.; Globerson, A.; and Soudry, D. 2021. On the Implicit Bias of Initialization Shape: Beyond Infinitesimal Mirror Descent. arXiv:2102.09769.

Barrett, D.; and Dherin, B. 2021. Implicit Gradient Regularization. In *International Conference on Learning Representations*.

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*.

Chizat, L.; Oyallon, E.; and Bach, F. 2019. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. In *Journal of Machine Learning Research*.

Even, M.; Pesme, S.; Gunasekar, S.; and Flammarion, N. 2023. (S)GD over Diagonal Linear Networks: Implicit Regularisation, Large Stepsizes and Edge of Stability. In *arXiv:2302.08982*.

Ghai, U.; Hazan, E.; and Singer, Y. 2020. Exponentiated gradient meets gradient descent. In *International Conference on Algorithmic Learning Theory*.

Ghosh, A.; Lyu, H.; Zhang, X.; and Wang, R. 2023. Implicit regularization in Heavy-ball momentum accelerated stochastic gradient descent. arXiv:2302.00849.

Gunasekar, S.; Lee, J.; Soudry, D.; and Srebro, N. 2018. Characterizing Implicit Bias in Terms of Optimization Geometry. In *International Conference on Machine Learning*.

Jelassi, S.; and Li, Y. 2022. Towards understanding how momentum improves generalization in deep learning. arXiv:2207.05931.

Ji, Z.; and Telgarsky, M. 2019. Gradient Descent Aligns the Layers of Deep Linear Networks. In *International Conference on Learning Representations*.

Kingma, D. P.; and Ba, J. 2017. Adam: A method for stochastic optimization. In *arXiv:1412.6980*.

Kovachki, N. B.; and Stuart, A. M. 2021. Continuous Time Analysis of Momentum Methods. In *Journal of Machine Learning Research*.

Li, Z.; Luo, Y.; and Lyu, K. 2021. Towards Resolving the Implicit Bias of Gradient Descent for Matrix Factorization: Greedy Low-Rank Learning. In *International Conference on Learning Representations*.

Lyu, B.; and Zhu, Z. 2022. Implicit Bias of Adversarial Training for Deep Neural Networks. In *International Conference on Learning Representations*.

Lyu, B.; and Zhu, Z. 2023. Implicit Bias of (Stochastic) Gradient Descent for Rank-1 Linear Neural Network. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 58166–58201. Curran Associates, Inc.

Lyu, K.; and Li, J. 2020. Gradient Descent Maximizes the Margin of Homogeneous Neural Networks. In *International Conference on Learning Representations*.

Nacson, M. S.; Gunasekar, S.; Lee, J.; Srebro, N.; and Soudry, D. 2019. Lexicographic and Depth-Sensitive Margins in Homogeneous and Non-Homogeneous Deep Models. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 4683–4692. PMLR.

Nesterov, Y. 1983. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In *Soviet Mathematics Doklady*.

Nesterov, Y. 2014. Introductory Lectures on Convex Optimization: A Basic Course. In *Springer Publishing Company, Incorporated*.

Papazov, H.; Pesme, S.; and Flammarion, N. 2024. Leveraging Continuous Time to Understand Momentum When Training Diagonal Linear Networks. In Dasgupta, S.; Mandt, S.; and Li, Y., eds., *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, 3556–3564. PMLR.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems 2017 Workshop Autodiff*.

Pesme, S.; Pillaud-Vivien, L.; and Flammarion, N. 2021. Implicit Bias of SGD for Diagonal Linear Networks: a Provable Benefit of Stochasticity. In *Advances in Neural Information Processing Systems*.

Pillaud-Vivien, L.; Reygner, J.; and Flammarion, N. 2020. Label Noise (stochastic) Gradient Descent Implicitly Solves the Lasso for Quadratic Parametrisation. In *Conference on Learning Theory*.

Polyak, B. 1964. Some Methods of Speeding Up the Convergence of Iteration Methods. In *Ussr Computational Mathematics and Mathematical Physics*.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in Microstructures in Cognition*, volume 1: Foundations.

Shi, B.; Du, S. S.; Jordan, M. I.; and Su, W. J. 2018. Understanding the Acceleration Phenomenon via High-Resolution Differential Equations. In *arXiv: 1810.08907*.

Soudry, D.; Hoffer, E.; Nacson, M. S.; Gunasekar, S.; and Srebro, N. 2018. The Implicit Bias of Gradient Descent on Separable Data. *Journal of Machine Learning Research*, 19(70): 1–57.

Su, W.; Boyd, S.; and Candes, E. 2014. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*.

Sutskever, I.; Martens, J.; Dahl, G.; and Hinton, G. 2013. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*.

Wang, B.; Meng, Q.; Zhang, H.; Sun, R.; Chen, W.; Ma, Z.-M.; and Liu, T.-Y. 2022. Does Momentum Change the Implicit Regularization on Separable Data? In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 26764–26776. Curran Associates, Inc.

Wibisono, A.; Roelofs, R.; Stern, M.; Srebro, N.; and Recht, B. 2017. The Marginal Value of Adaptive Gradient Methods in Machine Learning. In *Advances in Neural Information Processing Systems*.

Wibisono, A.; Wilson, A. C.; and Jordan, M. I. 2016. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47): E7351–E7358.

Wilson, A. C.; Recht, B.; and Jordan, M. I. 2016. A lyapunov analysis of momentum methods in optimization. In *arXiv:1611.02635*.

Woodworth, B.; Gunasekar, S.; Lee, J. D.; Moroshko, E.; Savarese, P.; Golan, I.; Soudry, D.; and Srebro, N. 2020. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*.

Yun, C.; Krishnan, S.; and Mobahi, H. 2021. A Unifying View on Implicit Bias in Training Linear Neural Networks. In *International Conference on Learning Representations*.

We first add more related works and provide additional numerical experiments to support our theoretical results. We then present the detailed modelling techniques of momentum-based methods HB and NAG. Finally we discuss the proofs for main theorems.

A Additional Related Works and Detailed Comparisons

Besides the early applications of momentum-based methods in the convex optimization literature, Rumelhart, Hinton, and Williams (1986) firstly applied HB to the training of deep learning models. The recent work Sutskever et al. (2013) then summarized these momentum-based methods and illustrated their importance in the area of deep learning. Wibisono et al. (2017) demonstrated that SGD, HB and NAG generalize better than adaptive methods such as Adam (Kingma and Ba 2017) and AdaGrad (Duchi, Hazan, and Singer 2011) for deep networks by conducting experiments on classification problems.

Implicit bias of GD and its variants. The study of implicit bias started from Soudry et al. (2018) where GD has been shown to return the max-margin classifier for the logistic-regression problem. The analysis for classification problems was then generalized to linear networks (Ji and Telgarsky 2019), more general homogeneous networks (Lyu and Li 2020; Chizat and Bach 2020), and other training strategies (Lyu and Zhu 2022) for homogeneous networks. For regression problems, Li, Luo, and Lyu (2021) showed that gradient flow for matrix factorization implicitly prefers the low-rank solution. Azulay et al. (2021); Yun, Krishnan, and Mobahi (2021) studied the implicit bias of GF for standard linear networks. For the diagonal linear networks, Azulay et al. (2021); Yun, Krishnan, and Mobahi (2021); Woodworth et al. (2020) further revealed the transition from kernel regime (or lazy regime) (Chizat, Oyallon, and Bach 2019) to rich regime by decreasing the initialization scales from ∞ to 0. Besides the full-batch version of gradient descent, Pesme, Pillaud-Vivien, and Flammarion (2021); Lyu and Zhu (2023) studied the SGF and showed that the stochastic sampling noise implicitly induces an effect equivalent to reducing the initialization scale. Pillaud-Vivien, Reygner, and Flammarion (2020) then analyzed GF with label noise and (Even et al. 2023), removing the infinitesimal learning rate approximation, studied the implicit bias of discrete GD and SGD with moderate learning rate for diagonal linear networks. Gunasekar et al. (2018); Wang et al. (2022) showed that momentum-based methods converge to the same max-margin solution as GD for single layer model and linear classification problem. Jelassi and Li (2022) further revealed that momentum-based methods have better generalization performance than GD for classification problem. Ghosh et al. (2023) conducted a model-agnostic analysis of $\mathcal{O}(\eta^2)$ continuous approximate version of HB and also showed the generalization advantages of HB.

Comparison to Gunasekar et al. (2018); Wang et al. (2022). Gunasekar et al. (2022) revealed that there is no difference between the implicit bias of momentum-based methods and that of GD for linear regression problem. In addition, (Wang et al. 2022) studied the linear classification problem and showed that momentum-based methods converge to the same max-margin solution as GD for single-layer linear networks, i.e., they share the same implicit bias. These works confirmed that momentum-based methods does not enjoy possible better generalization performance than GD for single-layer models. Compared to these works, our results reveal that momentum-based methods will have different implicit bias when compared to GD for diagonal linear networks, a deep learning models, indicating the importance of the over-parameterization on the implicit bias of momentum-based methods.

Comparison to Jelassi and Li (2022). Jelassi and Li (2022) studied classification problem and also showed that momentum-based methods improve generalization of a linear CNN model partly due to the historical gradients. The setting of our work is different from that of Jelassi and Li (2022): our work focuses on regression problems and diagonal linear networks. In addition, there are also differences between the conclusion of our work and that of Jelassi and Li (2022), in the sense that we conclude that momentum-based methods does not always lead to solutions with better generalization performance than GD, which depends on whether the initialization mitigation effect of momentum-based methods (interestingly this effect can also be regarded as coming from the historical gradients as in Jelassi and Li (2022)) outperforms their extra dependence on initialization of gradients. Therefore, the momentum-based method is not always a better choice than GD.

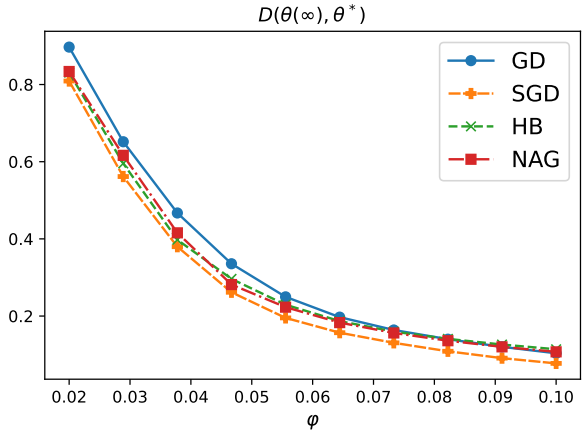
Comparison to Ghosh et al. (2023). The analysis in Ghosh et al. (2023) is model-agnostic, in the sense that it did not consider other sources that affect the implicit bias such as model architectures and initialization, while our work focuses on precisely characterizing the implicit bias of momentum-based methods and its explicit dependence on the architecture and the initialization of both parameters and gradients, which cannot be captured solely by the analysis in Ghosh et al. (2023).

B Additional Experiments and Missing Experimental Details

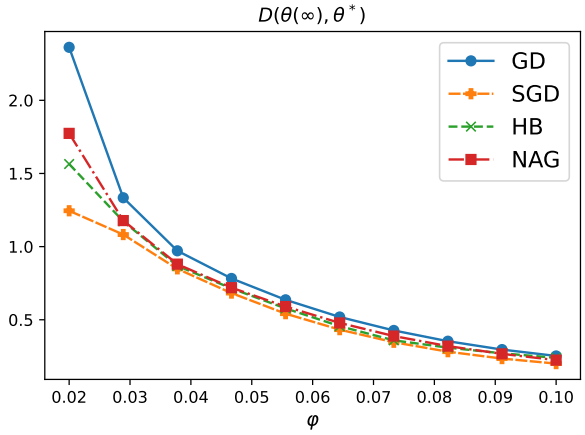
B.1 Additional Numerical Experiments for Biased Initialization.

To further characterize the influence of the extent of the biased part of the initialization, we run similar experiments with same hyper-parameters as in Fig. 2(b) except for the scale of the initialization $\|\xi\|_1$. The results are presented in Fig. 4. It can be seen that, for different $\|\xi\|_1$, the generalization benefits of HB and NAG are significant when φ is small, i.e., the initialization is highly biased.

For other biased initialization, we consider $u(0) = c\mathbf{e}_d$ for some constant $c \in \mathbb{R}$ and $v(0) = c\mathbf{e}_d + \rho$ for a random gaussian vector $\rho \in \mathbb{R}^d$. We use the same dataset as in Fig. 2(a). As shown in Fig. 5, the generalization performance of HB



(a)



(b)

Figure 4: $D(\theta(\infty), \theta^*)$ for diagonal linear networks trained with different algorithms and different values of φ. **(a).** $\|\xi\|_1 = 0.0022$. **(b).** $\|\xi\|_1 = 0.01$.

and NAG solutions become better as we decrease the scale of the initialization, indicating the transition from kernel regime to rich regime. Furthermore, as a result of the initialization mitigation effects, Fig. 5 shows that HB, NAG, and SGD exhibit better generalization performance than GD, which further verifies the benefit of momentum on the generalization when the initialization is biased.

B.2 Non-linear Networks

To explore whether the generalization benefit of momentum-based methods exists for non-linear networks, we conduct experiments for non-linear networks in this section to compare GD with HB and NAG.

Experiment details for non-linear networks. We train a four-layer non-linear network $f(x; W)$ with the architecture of 100×100 `Linear-ReLU-`100×100 `Linear-ReLU-`100×100 `Linear-ReLU-`100×1 `Linear`. The learning rate is fixed as $\eta = 10^{-2}$ and the momentum factor is fixed as $\mu = 0.9$ for HB and NAG. To measure the initialization scales, we vectorize all layer matrices and calculate the sum of ℓ_2-norm, i.e., we calculate $\sum_{k=1}^{4} \|W_k\|_F^2$ where W_k is the weight matrix of the k-th layer. Since non-linear networks are not equivalent to a linear predictor $\theta^T x$ as diagonal linear networks, we sample a newly test data with $\{(x_{i;\text{test}}, y_{i;\text{test}})\}_{i=1}^{40}$ using the ground truth solution θ^* and the training data distribution and let the test error

$$D = \frac{1}{2n} \sum_{i=1}^{40} (f(x_{i;\text{test}}; W) - y_{i;\text{test}})^2$$

measure the generalization performance.

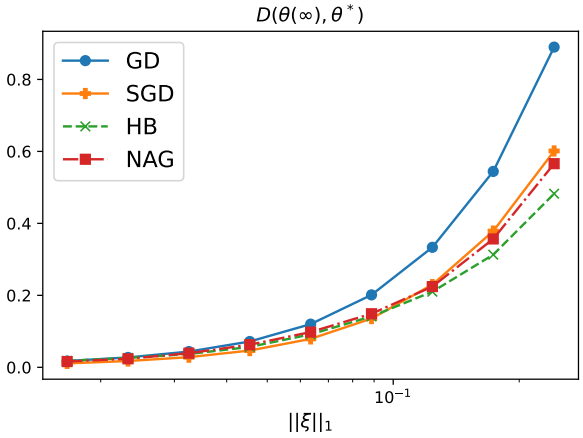


Figure 5: $D(\theta(\infty), \theta^*)$ for diagonal linear networks with biased initialization trained with different algorithms and different values of $\|\xi\|_1$.

We show the benefits of momentum for non-linear networks in the same data of Section 3.2 in Fig. 6, which reveals that the benefit of momentum also exists in the non-linear networks, and the test errors are getting lower for smaller initialization scales similar to the diagonal linear networks.

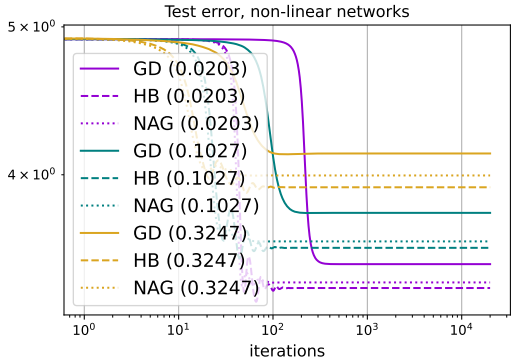


Figure 6: $D(\theta(\infty), \theta^*)$ for non-linear networks trained with different algorithms and different initialization scales (numbers in the bracket)

B.3 Experimental Details for Fig. 3

The dataset is the same as that in Section 3.2. To make the initialization biased, we consider $u(0) = c\mathbf{e}_d$ for some constant $c \in \mathbb{R}$ and $v(0) = c\mathbf{e}_d + \rho$ for a random gaussian vector $\rho \in \mathbb{R}^d$, where we fix ρ for the training of different algorithms with different hyper-parameters.

Furthermore, to verify the statement of Section 3.5, we also report the dependence of $D(\theta(\infty), \theta^*)$ on η in Fig. 7(a) and the effects of μ in Fig. 7(b).

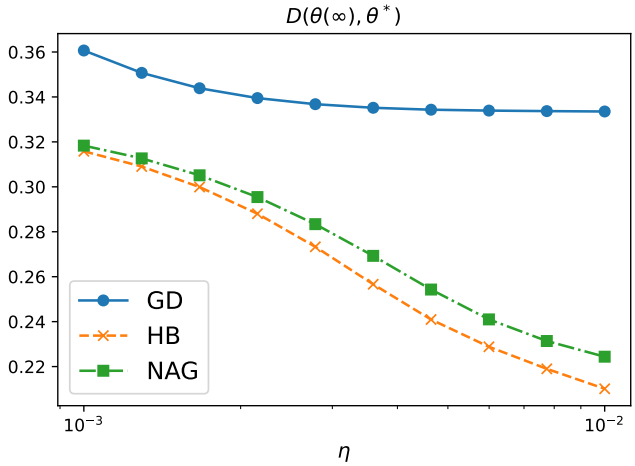
C Details of the Continuous Modelling

In Section C.1 we study HB and present the results for NAG in Section C.2.

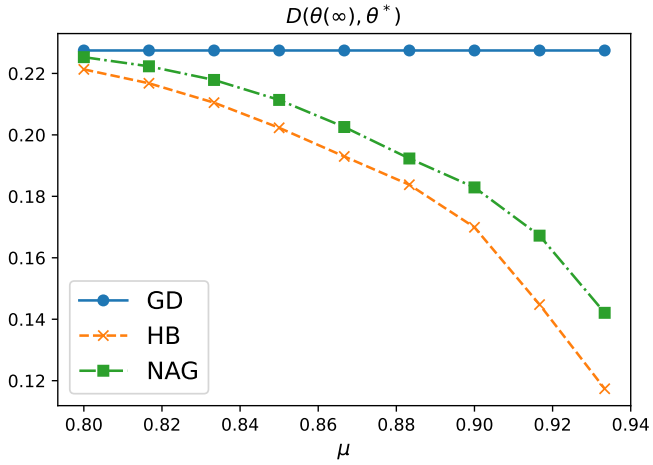
C.1 Continuous time approximation of HB

Recall that the discrete update rule for β is

$$p_{k+1} = \mu p_k - \nabla L(\beta_k),$$
$$\beta_{k+1} = \beta_k + \eta p_{k+1},$$



(a) $D(\theta(\infty), \theta^*)$ for different η and $\mu = 0.9$



(b) $D(\theta(\infty), \theta^*)$ for different μ and $\eta = 0.01$

Figure 7: Diagonal nets trained with different algorithms and hyper-parameters

which, noting that $\eta p_k = \beta_k - \beta_{k-1}$, can be further written as a single-step update

$$\beta_{k+1} = \beta_k + \mu(\beta_k - \beta_{k-1}) - \eta \nabla L(\beta). \tag{13}$$

We let time $t = k\eta$ and the continuous version of β_k be $\beta(t)$. Note that $\beta(t+\eta) \approx \beta_{k+1}$ and that

$$\beta(t+\eta) = \beta(t) + \eta\dot{\beta}(t) + \frac{\eta^2}{2}\ddot{\beta}(t) + \mathcal{O}(\eta^3),$$

by replacing all β_k and β_{k+1} with $\beta(t)$ and $\beta(t+\eta)$, respectively, Eq. (13) becomes

$$\begin{aligned} &\beta(t+\eta) - \beta(t) = \mu(\beta(t) - \beta(t-\eta)) - \eta\nabla L(\beta) \\ \implies &\eta\dot{\beta}(t) + \frac{\eta^2}{2}\ddot{\beta}(t) = \mu\left(\eta\dot{\beta}(t) - \frac{\eta^2}{2}\ddot{\beta}(t)\right) - \eta\nabla L(\beta), \end{aligned} \tag{14}$$

which gives us the continuous time approximation of HB:

$$\alpha\ddot{\beta}(t) + \dot{\beta}(t) + \frac{\nabla L(\beta)}{1-\mu} = 0.$$

with $\alpha = \frac{\eta(1+\mu)}{2(1-\mu)}$.

C.2 Continuous time approximations of NAG

The discrete update rule for β trained with NAG is

$$\begin{aligned} p_{k+1} &= \mu p_k - \eta\nabla L(\beta_k + \mu p_k), \\ \beta_{k+1} &= \beta_k + p_{k+1}, \end{aligned}$$

which can also be written as a single-step update

$$\beta_{k+1} = \beta_k + \mu(\beta_k - \beta_{k-1}) - \eta\nabla L(\beta)|_{\beta=\rho_k},$$

where we let

$$\rho_k = \beta_k + \mu p_k.$$

Following similar approach as in the case for HB, this discrete update rule implies that

$$\eta\dot{\beta}(t) + \frac{\eta^2}{2}\ddot{\beta}(t) = \mu\left(\eta\dot{\beta}(t) - \frac{\eta^2}{2}\ddot{\beta}(t)\right) - \eta\nabla L(\beta)|_{\beta=\rho(t)} \tag{15}$$

$$\implies \frac{\eta(1+\mu)}{2}\ddot{\beta}(t) + (1-\mu)\dot{\beta}(t) = -\nabla L(\beta)|_{\beta=\rho(t)}. \tag{16}$$

Since the gradient is evaluated at $\beta = \rho(t)$ rather than $\beta(t)$, to further simplify this equation, we note that

$$\begin{aligned} \rho(t) &= \beta(t) + \mu(\beta(t) - \beta(t-\eta)) \\ &= \beta(t) + \eta\mu\dot{\beta}(t) + \mathcal{O}(\eta^3), \end{aligned}$$

therefore $\eta\nabla L(\beta)|_{\beta=\rho(t)}$ can be expanded around $\beta(t)$:

$$\eta\nabla L(\beta)|_{\beta=\rho(t)} = \eta\nabla L(\beta)|_{\beta=\beta(t)} + \eta^2\mu\nabla^2 L(\beta)\cdot\dot{\beta}|_{\beta=\beta(t)} + \mathcal{O}(\eta^2).$$

Meanwhile, by differentiating both sides of Eq. (16) w.r.t t, we have

$$\begin{aligned} &\frac{\eta(1+\mu)}{2}\dddot{\beta}(t) + (1-\mu)\ddot{\beta}(t) = -\nabla^2 L(\beta)\cdot\dot{\beta}|_{\beta=\rho(t)} \\ \implies &\eta^2\mu\nabla^2 L(\beta)\cdot\dot{\beta}|_{\beta=\beta(t)} = -\eta^2(1-\mu)\ddot{\beta}(t) \end{aligned}$$

where we multiply η^2 to both sides of the last equality and omit the terms of the order $\mathcal{O}(\eta^3)$. In this way, Eq. (16) becomes

$$\begin{aligned} &\frac{\eta(1+\mu)}{2}\ddot{\beta}(t) + (1-\mu)\dot{\beta}(t) = -\nabla L(\beta) + \eta\mu(1-\mu)\ddot{\beta}(t) \\ \implies &\alpha\ddot{\beta} + \dot{\beta} + \frac{\nabla L(\beta)}{1-\mu} = 0 \end{aligned}$$

with

$$\alpha = \frac{2\mu^2 - \mu + 1}{2(1-\mu)}.$$

C.3 Generalization to $\mathcal{O}(\eta^2)$ approximate continuous version of HB

Note that if it is compared to the $\mathcal{O}(\eta^2)$ approximate continuous version for GD, e.g., the modified flow discussed in (Barrett and Dherin 2021), the $\mathcal{O}(\eta^2)$ approximate continuous version of HB presented in (Ghosh et al. 2023) is a justifiable choice to characterize the role of momentum on the implicit bias for diagonal linear networks. The important thing is that the same order of approximations for both momentum-based methods and GD should be used to make a "fair" comparison on their implicit bias for diagonal linear networks. This is similar to the spirit in (Ghosh et al. 2023) where the authors compared $\mathcal{O}(\eta^2)$ approximate continuous version of HB with the $\mathcal{O}(\eta^2)$ version of GD (Barrett and Dherin 2021), rather than the usual $\mathcal{O}(\eta)$ approximate continuous version of GD, i.e., gradient flow.

Therefore, the main reason why we use the $\mathcal{O}(\eta)$ continuous approximation for HB (and NAG) as discussed in Proposition 1 is that we aim to precisely characterize the the role of momentum in the implicit bias of the widely-studied gradient flow, which is $\mathcal{O}(\eta)$ approximate continuous version of GD, for diagonal linear networks. And one can naturally generalize the current work to the case following similar techniques: replacing the current $\mathcal{O}(\eta)$ approximate second-order ODE (Proposition 1) with the $\mathcal{O}(\eta^2)$ approximate version of HB in (Ghosh et al. 2023).

D Proofs for Theorems

In the following, we first discuss the proof sketch for Theorem 3. Then we prove Proposition 4 to show the θ dynamics and its relation to mirror flow and present the convergence result of the corresponding time-varying potential. Finally, we prove Theorem 3. The analysis of the effects of the initialization scale on the implicit bias is then presented. Finally, we discuss the proof of Proposition 2.

Proof sketch for Theorem 3. Since our main result is Theorem 3, we first present the proof sketch. The proof mainly consists of three steps. The first step is to derive the dynamics for $\theta = u \odot u - v \odot v$ of diagonal linear networks for HB and NAG. The second step is to construct the connection of the θ dynamics with "accelerated" mirror flow (Wilson, Recht, and Jordan 2016) with time-varying potential (Proposition 4). Since this time-varying potential converges along training, the third step is simply to apply the optimality condition and derive the implicit bias result.

For diagonal linear networks, the parameterization of θ is $u \odot u - v \odot v$ and the model parameters are $\beta = (u, v)$. According to Proposition 1, the continuous dynamics of u and v are thus

$$\alpha\ddot{u} + \dot{u} + \frac{\nabla_u L(u,v)}{1-\mu} = 0,\ \alpha\ddot{v} + \dot{v} + \frac{\nabla_v L(u,v)}{1-\mu} = 0, \tag{17}$$

where

$$\alpha = \begin{cases} \frac{\eta(1+\mu)}{2(1-\mu)} & \text{for HB} \\ \frac{\eta(1-\mu+2\mu^2)}{2(1-\mu)} & \text{for NAG} \end{cases}$$

and η is the learning rate. Note that since α is the order of η, we will omit all terms of the order $\mathcal{O}(\eta^2)$, which is eligible because the momentum ODE is established to the order of $\mathcal{O}(\eta)$, i.e., the momentum ODE in Proposition 1 is in fact

$$\alpha\ddot{\beta} + \dot{\beta} + \frac{\nabla_\beta}{1-\mu} = \mathcal{O}(\eta^2).$$

Thus all terms of order higher than η appearing in anywhere else should also be dropped. To be consistent with this convention, terms proportional to and higher than η^2 are ignored in the following, otherwise there would be contradictions if $\mathcal{O}(\eta^2)$ terms are neglected in the momentum ODE while they are maintained in other equations. For convenience, we first present several useful properties for the dynamics of HB and NAG for diagonal linear networks Eq. (6) for $\forall j \in \{1, \ldots, d\}$:

1. **Property 1**: $v_j \partial_{u_j} L + u_j \partial_{v_j} L = 0$.

   Proof. This is because

   $$\partial_{u_j} L = \frac{2}{n} u_j \sum_{i=1}^{n} r_i x_{i;j}, \partial_{v_j} L = -\frac{2}{n} v_j \sum_{i=1}^{n} r_i x_{i;j} \implies v_j \partial_{u_j} L + u_j \partial_{v_j} L = 0 \tag{18}$$

   where

   $$\frac{1}{n} \sum_{i=1}^{n} r_i x_{i;j} = \partial_{\theta_j} L(\theta) \tag{19}$$

   and $r_i = \theta^T x_i - y_i$ is the residual. □

2. **Property 2:** $\alpha(v_j \dot{u}_j + \dot{v}_j u_j) = \mathcal{O}(\eta^2)$.

Proof. This can be obtained from

$$\alpha\dot{u}_j = -\alpha\frac{\partial_{u_j}L}{1-\mu} + \mathcal{O}(\eta^2),\ \alpha\dot{v}_j = -\alpha\frac{\partial_{v_j}L}{1-\mu} + \mathcal{O}(\eta^2) \tag{20}$$

since α is the order of η, thus, according to Property 1,

$$\alpha(v_j\dot{u}_j + \dot{v}_j u_j) = -\alpha(v_j\partial_{u_j}L + u_j\partial_{v_j}L) + \mathcal{O}(\eta^2) = \mathcal{O}(\eta^2). \tag{21}$$

□

3. **Property 3:** $(u_j + \alpha\dot{u}_j)(v_j + \alpha\dot{v}_j) = u_j v_j + \mathcal{O}(\eta^2)$.

   Proof. This can be obtained from

   $$(u_j + \alpha\dot{u}_j)(v_j + \alpha\dot{v}_j) = u_j v_j + \alpha\dot{u}_j v_j + \alpha\dot{v}_j u_j + \alpha^2\dot{u}_j\dot{v}_j = u_j v_j + \mathcal{O}(\eta^2),$$

   where we use Property 2 in the last equality. □

4. **Property 4:** $du_j v_j/dt = 2\alpha\dot{u}_j\dot{v}_j + \mathcal{O}(\eta^2)$.

   Proof. To show this, we directly calculate

   $$\begin{aligned}\frac{d}{dt}u_j v_j &= \dot{u}_j v_j + \dot{v}_j u_j \\ &= \left[-\alpha\ddot{u}_i - \frac{\partial_{u_j}L}{1-\mu}\right]v_j + \left[-\alpha\ddot{v}_i - \frac{\partial_{v_j}L}{1-\mu}\right]u_j \\ &= -\alpha(\ddot{u}_i v_j + \dot{u}_j\dot{v}_j + \ddot{v}_i u_j + \dot{u}_j\dot{v}_j) + 2\alpha\dot{u}_j\dot{v}_j \\ &= -\frac{d}{dt}\left[\alpha\dot{u}_j v_j + \alpha\dot{v}_j u_j\right] + 2\alpha\dot{u}_j\dot{v}_j \\ &= 2\alpha\dot{u}_j\dot{v}_j + \mathcal{O}(\eta^2),\end{aligned} \tag{22}$$

   where the second line is due to the dynamics of u and v in Eq. (17), and the last equaility is due to Property 2. □

D.1 Proof of Proposition 4

For convenience, we first recall that, under the conditions of Proposition 4, $\bar{\theta}_{\alpha;j}$ follows a mirror flow form

$$\forall j \in \{1,\dots,d\}:\ \frac{d}{dt}\left[\nabla Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t) + \theta_j\mathcal{R}_j\right] = -\frac{\partial_{\theta_j}L(\theta)}{1-\mu},$$

where

$$Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t) = \frac{1}{4}\left[\bar{\theta}_{\alpha;j}\,\mathrm{arcsinh}\left(\frac{\bar{\theta}_{\alpha;j}}{2\bar{\xi}_j(t)}\right) - \sqrt{4\bar{\xi}_j^2(t) + \bar{\theta}_{\alpha;j}} + 2\bar{\xi}_j(t)\right],$$

$$\bar{\xi}_j(t) = \xi_j e^{-\alpha\phi_j(t)},\ \phi_j(t) = \frac{8}{(1-\mu)^2}\int_0^t \partial_{\theta_j}L\left(\theta(s)\right)\partial_{\theta_j}L\left(\theta(s)\right)ds.$$

Below we prove this result.

Proof. The proof consists of two steps: the first step is to derive the dynamics of θ and the second step is to derive the mirror flow form of the dynamics.

The dynamics of θ. We start with the first step. Recall that the parameterization of $\theta_j = u_j^2 - v_j^2$, we conclude that θ_j follows a second-order ODE different from that of u and v (Eq. (17)) by inspecting the exact expression of $\dot{\theta}_j$

$$\begin{aligned}\dot{\theta}_j &= 2u_j\dot{u}_j - 2v_j\dot{v}_j \\ &= 2u_j\left(-\alpha\ddot{u}_j - \frac{\partial_{u_j}L}{1-\mu}\right) - 2v_j\left(-\alpha\ddot{v}_j - \frac{\partial_{v_j}L}{1-\mu}\right) \\ &= -2\alpha\left[u_j\ddot{u}_j + \dot{u}_j\dot{u}_j - v_j\ddot{v}_j - \dot{v}_j\dot{v}_j\right] + 2\alpha\dot{u}_j\dot{u}_j - 2\alpha\dot{v}_j\dot{v}_j - \frac{2\left(u_j\partial_{u_j}L - v_j\partial_{v_j}L\right)}{1-\mu} \\ &= -\alpha\ddot{\theta}_j - \frac{2\left[(u_j + \alpha\dot{u}_j)\partial_{u_j}L - (v_j + \alpha\dot{v}_j)\partial_{v_j}L\right]}{1-\mu}\end{aligned}$$

where the second equality is because Eq. (17) and we use Eq. (20) in the last line. Note that if we let

$$G_j = 2\left[(u_j + \alpha\dot{u}_j)\partial_{u_j}L - (v_j + \alpha\dot{v}_j)\partial_{v_j}L\right],$$

then the dynamics of θ_j follows a second-order ODE

$$\alpha\ddot{\theta}_j + \dot{\theta}_j + \frac{G_j}{1-\mu} = 0, \tag{23}$$

note that although this is similar to the dynamics of u and v, they are not the same. To proceed, we need to express G_j with θ_j, which can be done by observing that

$$G_j = 4\left[u_j(u_j + \alpha\dot{u}_j) + v_j(v_j + \alpha\dot{v}_j)\right]\partial_{\theta_j}L = 4H_j\partial_{\theta_j}L$$

where use Eq. (19) in the first equality. Expressing H_j with θ_j will give us the desired results, which can be done as follows.

$$\begin{aligned}
H_j^2 &= \left[u_j(u_j + \alpha\dot{u}_j) + v_j(v_j + \alpha\dot{v}_j)\right]^2 \\
&= u_j^2(u_j + \alpha\dot{u}_j)^2 + v_j^2(v_j + \alpha\dot{v}_j)^2 + 2v_ju_j(u_j + \alpha\dot{u}_j)(v_j + \alpha\dot{v}_j) \\
&= u_j^4 + v_j^4 + 2\alpha u_j^3\dot{u}_j + 2\alpha v_j^3\dot{v}_j + 2u_j^2v_j^2 + 2\alpha u_jv_j(u_j\dot{v}_j + v_j\dot{u}_j) + \alpha^2u_j^2\dot{u}_j^2 \\
&\quad + \alpha^2v_j^2\dot{v}_j^2 + 2\alpha^2u_jv_j\dot{u}_j\dot{v}_j \\
&= u_j^4 + v_j^4 + 2u_j^2v_j^2 + 2\alpha u_j^3\dot{u}_j + 2\alpha v_j^3\dot{v}_j + \mathcal{O}(\eta^2)
\end{aligned} \tag{24}$$

where we use Eq. (21) and α is the order of η in the last equality. On the other hand, we observe that the quantity $(\theta_j + \alpha\dot{\theta}_j)^2$ is

$$\begin{aligned}
(\theta_j + \alpha\dot{\theta}_j)^2 &= \left[u_j^2 - v_j^2 + \alpha(2u_j\dot{u}_j - 2v_j\dot{v}_j)\right]^2 \\
&= u_j^2(u_j + 2\alpha\dot{u}_j)^2 + v_j^2(v_j + 2\alpha\dot{v}_j)^2 - 2u_jv_j(u_j + 2\alpha\dot{u}_j)(v_j + 2\alpha\dot{v}_j) \\
&= u_j^4 + v_j^4 + 4\alpha u_j^3\dot{u}_j + 4\alpha v_j^3\dot{v}_j - 2u_j^2v_j^2 + \mathcal{O}(\eta^2)
\end{aligned} \tag{25}$$

where we use Eq. (21) in the last equality. Combining Eq. (24) and Eq. (25), we have

$$H_j^2 - (\theta_j + \alpha\dot{\theta}_j)^2 = \underbrace{4u_j^2v_j^2}_{\clubsuit} - \underbrace{(2\alpha u_j^3\dot{u}_j + 2\alpha v_j^3\dot{v}_j)}_{\diamondsuit}, \tag{26}$$

which establishes the relation between G_j and θ. In the following, our goal is to find the relation between $\clubsuit$ and $\diamondsuit$ and θ to complete the dynamics of θ. Now let $\xi \in \mathbb{R}^d$ and $\xi_j = |u_j(0)||v_j(0)|$ at the initialization[2], then for the term $\clubsuit$, according to Property 4 (Eq. (22)),

$$\begin{aligned}
\frac{du_jv_j}{dt} &= 2\alpha\dot{u}_j\dot{v}_j && (27)\\
&= \frac{2\alpha}{(1-\mu)^2}\partial_{u_j}L\partial_{v_j}L + \mathcal{O}(\eta^2) \\
&= -\frac{8\alpha}{(1-\mu)^2n^2}u_jv_j\left(\sum_{i=1}^n r_ix_{i;j}\right)^2 + \mathcal{O}(\eta^2) \\
&= -\frac{8\alpha}{(1-\mu)^2}u_jv_j\partial_{\theta_j}L(\theta)\partial_{\theta_j}L(\theta) + \mathcal{O}(\eta^2) && (28)
\end{aligned}$$

where we use Eq. (21) in the second equality and Eq. (19) in the third equality. Dividing u_jv_j on both sides and integrating the above equation give us that

$$\ln(u_jv_j) = \ln(u_j(0)v_j(0)) - \frac{8\alpha}{(1-\mu)^2}\int_0^t \partial_{\theta_j}L(\theta(s))\partial_{\theta_j}L(\theta(s))ds \tag{29}$$

$$\implies u_j(t)v_j(t) = u_j(0)v_j(0)e^{-\frac{8\alpha}{(1-\mu)^2}\int_0^t \partial_{\theta_j}L(\theta(s))\partial_{\theta_j}L(\theta(s))ds}. \tag{30}$$

For ease of notation, we denote

$$\phi_j(t) = \frac{8}{(1-\mu)^2}\int_0^t \partial_{\theta_j}L(\theta(s))\partial_{\theta_j}L(\theta(s))ds \geq 0, \tag{31}$$

[2]Note that ξ measures the scale of the initialization and ξ becomes $u(0) \odot v(0)$ for unbiased initialization $u(0) = v(0)$. Here we consider the more general biased initialization case.

then ♣ becomes

$$\clubsuit = 4u_j^2(t)v_j^2(t) = 4\xi_j^2 e^{-2\alpha\phi_j(t)} \leq 4\xi_j^2. \tag{32}$$

For the $\diamondsuit$ term, we note that

$$\begin{aligned}\theta_j\dot{\theta}_j &= 2(u_j^2 - v_j^2)(u_j\dot{u}_j - v_j\dot{v}_j)\\ &= 2\left[u_j^3\dot{u}_j - u_j^2v_j\dot{v}_j - v_j^2u_j\dot{u}_j + v_j^3\dot{v}_j\right]\\ &= 2\left[u_j^3\dot{u}_j + v_j^3\dot{v}_j\right] - 2u_jv_j(u_j\dot{v}_j + v_j\dot{u}_j),\end{aligned}$$

which, considering Property 2, further gives us

$$\alpha\theta_j\dot{\theta}_j = 2\alpha\left[u_j^3\dot{u}_j + v_j^3\dot{v}_j\right] + \mathcal{O}(\eta^2).$$

Comparing with the form of $\diamondsuit$, we have

$$\diamondsuit = \alpha\theta_j\dot{\theta}_j + \mathcal{O}(\eta^2). \tag{33}$$

Combined with the expressions of ♣, H_j can be completely expressed by θ since Eq. (26) now becomes

$$\begin{aligned}H_j^2 &= (\theta_j + \alpha\dot{\theta}_j)^2 + 4\xi_j^2e^{-2\alpha\phi_j(t)} - \alpha\theta_j\dot{\theta}_j\\ \implies H_j &= \sqrt{(\theta_j + \alpha\dot{\theta}_j)^2 + 4\xi_j^2e^{-2\alpha\phi_j(t)} - \alpha\theta_j\dot{\theta}_j}.\end{aligned} \tag{34}$$

Thus the form of θ dynamics Eq. (23) is now

$$\frac{1}{4H_j}(\alpha\ddot{\theta}_j + \dot{\theta}_j) = -\frac{\partial_{\theta_j}L}{1-\mu}, \tag{35}$$

where $1/H_j$ can be expanded to the order of η:

$$\begin{aligned}\frac{1}{H_j} &= \frac{1}{\sqrt{(\theta_j + \alpha\dot{\theta}_j)^2 + 4\xi_j^2e^{-2\alpha\phi_j(t)}}\sqrt{1 - \frac{\alpha\theta\dot{\theta}_j}{(\theta_j+\alpha\dot{\theta}_j)^2+4\xi_j^2e^{-2\alpha\phi_j(t)}}}}\\ &= \frac{1}{\sqrt{(\theta_j + \alpha\dot{\theta}_j)^2 + 4\xi_j^2e^{-2\alpha\phi_j(t)}}}\left(1 + \frac{1}{2}\frac{\alpha\theta_j\dot{\theta}_j}{\theta_j^2 + 4\xi_j^2} + \mathcal{O}(\eta^2)\right)\\ &= \frac{1}{\sqrt{(\theta_j + \alpha\dot{\theta}_j)^2 + 4\xi_j^2e^{-2\alpha\phi_j(t)}}} + \frac{\alpha}{2}\frac{\theta_j\dot{\theta}_j}{(\theta_j^2 + 4\xi_j^2)^{\frac{3}{2}}} + \mathcal{O}(\eta^2).\end{aligned}$$

Deriving the mirror flow form. Now we present the second part of the proof. Given $1/H_j$ and its relation with θ, in the following, we are now ready to derive the mirror flow form of θ_j. Note that the L.H.S of Eq. (35) includes a time derivative of $\theta + \alpha\dot{\theta}$, thus we need to find a mirror flow potential as a function of $\theta + \alpha\dot{\theta}$, rather than θ. For this purpose, if we define

$$Q_{\xi,j}^{\mathrm{M}}(\theta + \alpha\dot{\theta}, t) = q_{\xi,j}(\theta + \alpha\dot{\theta}, t) + h_j(t)(\theta_j + \alpha\dot{\theta}_j) \tag{36}$$

such that $q_{\xi,j}$ and $h_j(t)$ satisfy that

$$\nabla^2 q_{\xi,j}(\theta + \alpha\dot{\theta}, t) = \frac{1}{4\sqrt{(\theta_j + \alpha\dot{\theta}_j)^2 + 4\xi_j^2e^{-2\alpha\phi_j(t)}}}, \tag{37}$$

$$\frac{\partial\nabla q_{\xi,j}(\theta + \alpha\dot{\theta}, t)}{\partial t} + \frac{dh_j(t)}{dt} = \frac{\alpha}{8}\frac{\theta_j\dot{\theta}_j\dot{\theta}_j}{(\theta_j^2 + 4\xi_j^2)^{\frac{3}{2}}}, \tag{38}$$

then we will have

$$\begin{aligned}\frac{d}{dt}\nabla Q_{\xi,j}^{\mathrm{M}}(\theta + \alpha\dot{\theta}, t) &= \frac{d}{dt}\nabla q_{\xi,j}(\theta + \alpha\dot{\theta}, t) + \frac{d}{dt}h_j(t)\\ &= \nabla^2 q_{\xi,j}(\theta + \alpha\dot{\theta}, t)(\alpha\ddot{\theta} + \dot{\theta}) + \frac{\partial\nabla q_{\xi,j}(\theta + \alpha\dot{\theta}, t)}{\partial t} + \frac{dh_j(t)}{dt},\end{aligned}$$

which is exactly the L.H.S of Eq. (35). And we will have the desired mirror flow form of Proposition 4

$$\frac{d}{dt}\nabla Q_{\xi,j}^{\mathrm{M}}(\theta + \alpha\dot{\theta}, t) = -\frac{\partial_{\theta_j}L}{1-\mu}.$$

Therefore, it is now left for us to find $q_{\xi,j}$ and $h_j(t)$ that satisfy Eq. (37) and Eq. (38).

- **Find $q_{\xi,j}(\theta+\alpha\dot{\theta},t)$.** Since $q_{\xi,j}(\theta+\alpha\dot{\theta},t)$ satisfies Eq. (37), let $\bar{\xi}_j(t)=\xi_j e^{-\alpha\phi_j(t)}$, we integrate both sides of Eq. (37) to obtain that

$$\begin{aligned}\nabla q_{\xi,j}(\theta+\alpha\dot{\theta},t) &= \int \frac{d(\theta_j+\alpha\dot{\theta}_j)}{4\sqrt{(\theta_j+\alpha\dot{\theta}_j)^2+4\bar{\xi}_j^2(t)}} \\ &= \frac{\ln\left(\sqrt{(\theta_j+\alpha\dot{\theta}_j)^2+4\bar{\xi}_j^2(t)}+(\theta_j+\alpha\dot{\theta}_j)\right)}{4}+C. \end{aligned} \tag{39}$$

To determine the constant C, we require that

$$\nabla Q^{\mathrm{M}}_{\xi,j}(\theta(0)+\alpha\dot{\theta}(0),0)=0, \tag{40}$$

which gives us $\nabla q_{\xi,j}(\theta(0)+\alpha\dot{\theta}(0),0)+h_j(0)=0$. Let $\Delta_j=\theta_j(0)+\alpha\dot{\theta}_j(0)$ and note that $\bar{\xi}_j(0)=\xi_j$, we can determine the constant C as

$$\begin{aligned} C &= -\frac{\ln\left(\sqrt{(\theta_j(0)+\alpha\dot{\theta}_j(0))^2+4\bar{\xi}_j^2(0)}+(\theta_j(0)+\alpha\dot{\theta}_j(0))\right)}{4}-h_j(0) \\ &= -\frac{\ln\left[2\xi_j\left(\sqrt{1+\frac{\Delta_j^2}{4\xi_j^2}}+\frac{\Delta_j}{2\xi_j}\right)\right]}{4}-h_j(0) \\ &= -\frac{\ln(2\xi_j)}{4}-D_{\xi_j,\Delta_j}-h_j(0) \end{aligned} \tag{41}$$

where

$$D_{\xi_j,\Delta_j}=\frac{\ln\left(\sqrt{1+\frac{\Delta_j^2}{4\xi_j^2}}+\frac{\Delta_j}{2\xi_j}\right)}{4}=\frac{1}{4}\operatorname{arcsinh}\left(\frac{\Delta_j}{2\xi_j}\right).$$

Therefore, $\nabla q_{\xi,j}(\theta+\alpha\dot{\theta},t)$ should satisfy that

$$\begin{aligned} &\nabla q_{\xi,j}(\theta+\alpha\dot{\theta},t) \\ &= \frac{\ln\left(\sqrt{(\theta_j+\alpha\dot{\theta}_j)^2+4\bar{\xi}_j^2(t)}+(\theta_j+\alpha\dot{\theta}_j)\right)-\ln(2\xi_j)}{4}-D_{\xi_j,\Delta_j}-h_j(0). \end{aligned} \tag{42}$$

The form of $q_{\xi,j}$ can be obtained by solving the above equation. For convenience, we replace all $\theta_j+\alpha\dot{\theta}_j$ with a variable x in the above equation and solve

$$\begin{aligned} \nabla q_{\xi,j}(x,t) &= \frac{1}{4}\ln\left(\frac{\sqrt{x^2+4\bar{\xi}_j^2(t)}+x}{2\xi_j}\right)-D_{\xi_j,\Delta_j}-h_j(0) \\ &= \frac{1}{4}\ln\left(\frac{\sqrt{x^2+4\bar{\xi}_j^2(t)}+x}{2\xi_j e^{-\alpha\phi_j(t)}}\right)+\frac{\ln(e^{-\alpha\phi_j(t)})}{4}-D_{\xi_j,\Delta_j}-h_j(0) \\ &= \frac{1}{4}\ln\left(\sqrt{\frac{x^2}{4\bar{\xi}_j^2(t)}+1}+\frac{x}{2\bar{\xi}_j(t)}\right)-\frac{\alpha\phi_j(t)}{4}-D_{\xi_j,\Delta_j}-h_j(0) \\ &= \frac{1}{4}\operatorname{arcsinh}\left(\frac{x}{2\bar{\xi}_j(t)}\right)-\frac{\alpha\phi_j(t)}{4}-D_{\xi_j,\Delta_j}-h_j(0). \end{aligned} \tag{43}$$

Integrating both sides of the above equation directly gives us that $q_{\xi,j}(x,t)$ has the form of

$$\begin{aligned}
&q_{\xi,j}(x,t)\\
&= \frac{1}{4}\int \operatorname{arcsinh}\left(\frac{x}{2\bar{\xi}_j(t)}\right)dx - \frac{\alpha\phi_j(t)x}{4} - D_{\xi_j,\Delta_j}x - h_j(0)x\\
&= \frac{2\bar{\xi}_j(t)}{4}\left[\frac{x}{2\bar{\xi}_j(t)}\operatorname{arcsinh}\left(\frac{x}{2\bar{\xi}_j(t)}\right) - \sqrt{1+\frac{x}{4\bar{\xi}_j^2(t)}} + C_1\right] - \frac{\alpha\phi_j(t)x}{4} - D_{\xi_j,\Delta_j}x - h_j(0)x\\
&= \frac{2\bar{\xi}_j(t)}{4}\left[\frac{x}{2\bar{\xi}_j(t)}\operatorname{arcsinh}\left(\frac{x}{2\bar{\xi}_j(t)}\right) - \sqrt{1+\frac{x^2}{4\bar{\xi}_j(t)^2}} + 1\right]\\
&\quad - \frac{\alpha x\phi_j(t)}{4} - D_{\xi_j,\Delta_j}x - h_j(0)x,
\end{aligned} \tag{44}$$

where we set $C_1 = 1$.

- **Find** $h_j(t)$**.** The form of $h_j(t)$ can be obtained by solving Eq. (38). According to the form of ∇q_j in Eq. (42) and the definition of $\phi_j(t)$ in Eq. (31), we need to first calculate $\partial_t\nabla q_{\xi,j}$:

$$\begin{aligned}
&\partial_t\nabla q_{\xi,j}(\theta+\alpha\dot{\theta},t)\\
&= \frac{1}{4}\frac{4\bar{\xi}_j(t)}{\sqrt{(\theta_j+\alpha\dot{\theta}_j)^2+4\bar{\xi}_j^2(t)}\left(\sqrt{(\theta_j+\alpha\dot{\theta}_j)^2+4\bar{\xi}_j^2(t)}+(\theta_j+\alpha\dot{\theta}_j)\right)}\frac{d\bar{\xi}_i}{dt}\\
&= -\frac{\alpha\xi_j\bar{\xi}_j}{\sqrt{(\theta_j+\alpha\dot{\theta}_j)^2+4\bar{\xi}_j^2(t)}\left(\sqrt{(\theta_j+\alpha\dot{\theta}_j)^2+4\bar{\xi}_j^2(t)}+(\theta_j+\alpha\dot{\theta}_j)\right)}\frac{d\phi_j(t)}{dt}\\
&= -\alpha\frac{\xi_j^2}{\sqrt{\theta_j^2+4\xi_j^2}\left(\sqrt{\theta_j^2+4\xi_j^2}+\theta_j\right)}\frac{8\left(\partial_{\theta_j}L\right)^2}{(1-\mu)^2}+\mathcal{O}(\eta^2).
\end{aligned} \tag{45}$$

Putting the above equation back to Eq. (38) immediately gives us that

$$\begin{aligned}
h_j(t) &= \alpha\int_0^t \frac{\theta_j\dot{\theta}_j\dot{\theta}_j}{8(\theta_j^2+4\xi_j^2)^{\frac{3}{2}}} + \frac{\xi_j^2}{\sqrt{\theta_j^2+4\xi_j^2}\left(\sqrt{\theta_j^2+4\xi_j^2}+\theta_j\right)}\frac{8\left(\partial_{\theta_j}L\right)^2}{(1-\mu)^2}ds + C_2\\
&= \frac{2\alpha}{(1-\mu)^2}\int_0^t \frac{\left(\partial_{\theta_j}L(s)\right)^2}{\sqrt{\theta_j^2(s)+4\xi_j^2}}\left[\frac{4\xi_j^2}{\sqrt{\theta_j^2(s)+4\xi_j^2}+\theta_j(s)}\right]ds + C_2 + \mathcal{O}(\eta^2)
\end{aligned} \tag{46}$$

where we use

$$\alpha\dot{\theta}_j = -4\alpha H_j\frac{\partial_{\theta_j}L}{1-\mu}+\mathcal{O}(\eta^2) = -4\alpha\sqrt{\theta_j^2+4\xi_j^2}\frac{\partial_{\theta_j}L}{1-\mu}+\mathcal{O}(\eta^2) \tag{47}$$

according to Eq. (35) in the second equality and $C_2 = h_j(0)$ is a constant.

We are now ready to find $Q^{\mathrm{M}}_{\xi,j}$ by combining the form of $q_{\xi,j}(\theta+\alpha\dot{\theta},t)$ in Eq. (44) and the form of $h_j(t)$ in Eq. (46), which gives us

$$\begin{aligned}
Q^{\mathrm{M}}_{\xi,j}(\theta+\alpha\dot{\theta},t) &= \frac{2\bar{\xi}_j(t)}{4}\left[\frac{\theta_j+\alpha\dot{\theta}_j}{2\bar{\xi}_j(t)}\operatorname{arcsinh}\left(\frac{\theta_j+\alpha\dot{\theta}_j}{2\bar{\xi}_j(t)}\right) - \sqrt{1+\frac{(\theta_j+\alpha\dot{\theta}_j)^2}{4\bar{\xi}_j(t)^2}} + 1\right]\\
&\quad - \frac{\alpha\theta_j\phi_j(t)}{4} + (\theta_j+\alpha\dot{\theta}_j)h_j(t) - (\theta_j+\alpha\dot{\theta}_j)D_{\xi_j,\Delta_j} - (\theta_j+\alpha\dot{\theta}_j)h_j(0),
\end{aligned}$$

where, interestingly,

$$
\begin{aligned}
&-\frac{\alpha\theta_j\phi_j(t)}{4}+(\theta_j+\alpha\dot{\theta}_j)h_j(t)-(\theta_j+\alpha\dot{\theta}_j)h_j(0)\\
=&-\frac{2\alpha\theta_j}{(1-\mu)^2}\int_0^t(\partial_{\theta_j}L)^2ds+\frac{2\alpha\theta_j}{(1-\mu)^2}\int_0^t\frac{(\partial_{\theta_j}L(s))^2}{\sqrt{\theta_j^2(s)+4\xi_j^2}}\left[\frac{4\xi_j^2}{\sqrt{\theta_j^2(s)+4\xi_j^2}+\theta_j(s)}\right]ds\\
=&\frac{2\alpha\theta_j}{(1-\mu)^2}\int_0^t\frac{(\partial_{\theta_j}L(s))^2}{\sqrt{\theta_j^2(s)+4\xi_j^2}}\left[\frac{4\xi_j^2}{\sqrt{\theta_j^2(s)+4\xi_j^2}+\theta_j(s)}+\theta_j(s)-\sqrt{\theta_j^2(s)+4\xi_j^2}\right]ds\\
=&\,0.
\end{aligned}
\tag{48}
$$

As a result, let $\bar{\theta}_\alpha=\theta+\alpha\dot{\theta}$ and recall that

$$
D_{\xi_j,\Delta_j}=\frac{1}{4}\operatorname{arcsinh}\left(\frac{\theta_j(0)+\alpha\dot{\theta}_j(0)}{2\xi_j}\right)
\tag{49}
$$

where, let $\delta_j=u_j^2(0)-v_j^2(0)$,

$$
\begin{aligned}
\theta_j(0)+\alpha\dot{\theta}_j(0)&=u_j^2(0)-v_j^2(0)+2\alpha(u_j(0)\dot{u}_j(0)-v_j(0)\dot{v}_j(0))\\
&=u_j^2(0)-v_j^2(0)+\frac{4\alpha}{1-\mu}\left[u_j^2(0)+v_j^2(0)\right]\partial_{\theta_j}L(\theta(0))\\
&=\delta_j+\frac{4\alpha\partial_{\theta_j}L(\theta(0))}{1-\mu}\sqrt{\delta_j^2+4\xi_j^2}
\end{aligned}
\tag{50}
$$

we have the final form of $Q_{\xi,j}^{\mathrm{M}}$:

$$
\begin{aligned}
Q_{\xi,j}^{\mathrm{M}}(\bar{\theta}_\alpha,t)=&\frac{1}{4}\left[\bar{\theta}_{\alpha;j}\operatorname{arcsinh}\left(\frac{\bar{\theta}_{\alpha;j}}{2\bar{\xi}_j(t)}\right)-\sqrt{4\bar{\xi}_j^2(t)+\bar{\theta}_{\alpha;j}^2}+2\bar{\xi}_j(t)\right]\\
&-\frac{1}{4}\bar{\theta}_{\alpha;j}\operatorname{arcsinh}\left(\frac{\delta_j+\frac{4\alpha\partial_{\theta_j}L(\theta(0))}{1-\mu}\sqrt{\delta_j^2+4\xi_j^2}}{2\xi_j}\right).
\end{aligned}
\tag{51}
$$

The simplest case is when $\delta_j=0$, i.e., the unbiased initialization with $u(0)=v(0)$ such that $\theta(0)=0$ and $\nabla_\theta L(\theta(0))=\frac{1}{n}X^T(X\theta(0)-y)=-\frac{X^Ty}{n}$, then $Q_{\xi,j}^{\mathrm{M}}(\bar{\theta}_\alpha,t)$ has the form of

$$
\frac{1}{4}\left[\bar{\theta}_{\alpha;j}\operatorname{arcsinh}\left(\frac{\bar{\theta}_{\alpha;j}}{2\bar{\xi}_j(t)}\right)-\sqrt{4\bar{\xi}_j^2(t)+\bar{\theta}_{\alpha;j}^2}+2\bar{\xi}_j(t)+\bar{\theta}_{\alpha;j}\operatorname{arcsinh}\left(\frac{4\alpha(X^Ty)_j}{n(1-\mu)}\right)\right].
\tag{52}
$$

Simply redefining

$$
\begin{aligned}
Q_{\xi,j}^{\mathrm{M}}(\bar{\theta}_\alpha,t)&=\frac{1}{4}\left[\bar{\theta}_{\alpha;j}\operatorname{arcsinh}\left(\frac{\bar{\theta}_{\alpha;j}}{2\bar{\xi}_j(t)}\right)-\sqrt{4\bar{\xi}_j^2(t)+\bar{\theta}_{\alpha;j}^2}+2\bar{\xi}_j(t)\right],\\
\mathcal{R}_j&=\operatorname{arcsinh}\left(\frac{4\alpha(X^Ty)_j}{n(1-\mu)}\right),
\end{aligned}
$$

we finish the proof of Proposition 4:

$$
\forall j\in\{1,\dots,d\}:\ \frac{d}{dt}\nabla\left[Q_{\xi,j}^{\mathrm{M}}(\bar{\theta}_\alpha,t)+\bar{\theta}_{\alpha;j}\mathcal{R}_j\right]=-\frac{\partial_{\theta_j}L(\theta)}{1-\mu}.
$$

□

D.2 Convergence results for θ dynamics of HB and NAG

Since $Q_{\bar{\xi}_\infty}^{\mathrm{M}}(\theta)$ in Theorem 3 involves an integral from $t=0$ to ∞, it is necessary to show the convergence of this integral to guarantee the implicit bias result. For this purpose, we establish the convergence result of $\phi(\infty)$ in Theorem 3, whose j-th component for $t<\infty$ is $\phi_j(t)$ in Proposition 4. Recall that $Q_{\xi,j}^{\mathrm{M}}$ is defined in Eq. (12), we have the following proposition.

Proposition 5 (Convergence of $\phi(\infty)$)**.** *Under the same setting of Theorem 3 and assuming that $\theta(\infty)$ is an interpolation solution, i.e., $X\theta(\infty) = y$, then the integral $\phi(\infty)$ converges and its j-th component $\phi_j(\infty)$ satisfies that*

$$\phi_j(\infty) = \frac{16\left[\operatorname{diag}\left(X^TX\right)\right]_j}{n(1-\mu)} Q^{\mathrm{M}}_{\xi,j}\left(\theta_j(\infty), 0\right) + C,$$

where $C = \frac{4\alpha}{n(1-\mu)}\left[(\sum_{i=1}^n x_{i;j}^2)\left(\sqrt{\theta_j^2(0)+4\xi_j^2} - 2\xi_j\right) + \sum_{i=1}^n \epsilon_{i;j}\operatorname{arcsinh}\left(\frac{\theta_j(0)}{2\xi_j}\right)\right]$ *is a constant,* $\epsilon_{i;j} = \left(\sum_{k=1,k\neq j}^d \theta_k(0)x_{i;k} - y_i\right)x_{i;j}$*, and $C = 0$ for unbiased initialization $\theta(0) = 0$.*

Typically, solving the integral needs the entire training trajectory of θ, which is hard and equivalent to being aware of the limiting point of θ. From this aspect, Proposition 5 is interesting due to the fact that $\phi(\infty)$ has a rather simple explicit form depending on the data matrix X^TX and $Q^{\mathrm{M}}(\theta(\infty), 0)$. Furthermore, since the value of $\phi_j(\infty)$ controls the initialization mitigation effects of HB and NAG according to Theorem 3, as an immediate consequence of Proposition 5, such effects depend on learning rate η (through the dependence on α), the data matrix X^TX, initialization ξ (through the dependence on $Q^{\mathrm{M}}_{\xi,j}$), and the momentum factor μ.

Proof. Recall that $\phi_j(t)$ is defined as

$$\phi_j(t) = \frac{8}{(1-\mu)^2}\int_0^t \partial_{\theta_j} L(\theta(s))\partial_{\theta_j} L(\theta(s))ds$$

and, according to the dynamics of θ Eq. (35),

$$\alpha\dot{\theta}_j = -4\alpha\sqrt{\theta_j^2 + 4\xi_j^2}\frac{\partial_{\theta_j} L}{1-\mu} + \mathcal{O}(\eta^2),$$

then we get that $\phi_j(t)$ satisfies

$$\begin{aligned}
\alpha\phi_j(t) &= -\frac{4\alpha}{1-\mu}\int_0^t \frac{\partial_{\theta_j} L(\theta(s))}{\sqrt{\theta_j^2(s)+4\xi_j^2}}\frac{d\theta_j(s)}{ds}ds \\
&= -\frac{4\alpha}{n(1-\mu)}\int_0^t \frac{\sum_{i=1}^n(\sum_{k=1}^d \theta_k(s)x_{i;k} - y_i)x_{i;j}}{\sqrt{\theta_j^2(s)+4\xi_i^2}}d\theta_j(s) \\
&= -\frac{4\alpha(\sum_{i=1}^n x_{i;j}^2)}{n(1-\mu)}\underbrace{\int_0^t \frac{\theta_j(s)}{\sqrt{\theta_j^2(s)+4\xi_j^2}}d\theta_j(s)}_{\heartsuit} \\
&\quad - \frac{4\alpha}{n(1-\mu)}\sum_{i=1}^n \underbrace{\int_0^t \frac{\left(\sum_{k=1,k\neq j}^d \theta_k(s)x_{i;k} - y_i\right)x_{i;j}}{\sqrt{\theta_j^2(s)+4\xi_j^2}}d\theta_j(s)}_{\clubsuit},
\end{aligned} \tag{53}$$

where we replace $\alpha\partial_{\theta_j} L$ with $\alpha\dot{\theta}_j$ in the first equality. For the two integral terms, we note that

$$\begin{aligned}
\heartsuit &= \frac{1}{2}\int_0^t \frac{1}{\sqrt{\theta_j^2(s)+4\xi_j^2}}d(\theta_j^2(s)+4\xi_j^2) \\
&= \sqrt{\theta_j^2(t)+4\xi_j^2} + C_1
\end{aligned} \tag{54}$$

and, let $\epsilon_{i;j}(t) = \left(\sum_{k=1,k\neq j}^d \theta_k(t)x_{i;k} - y_i\right)x_{i;j}$,

$$\clubsuit = \epsilon_{i;j}(t)\operatorname{arcsinh}\left(\frac{\theta_j(t)}{2\xi_j}\right) + C_2. \tag{55}$$

As a result, we obtain the form of $\phi_j(t)$:

$$\phi_j(t) = -\frac{4\alpha}{n(1-\mu)}\left[(\sum_{i=1}^n x_{i;j}^2)\left(\sqrt{\theta_j^2(t)+4\xi_j^2} - 2\xi_j\right) + \sum_{i=1}^n \epsilon_{i;j}(t)\operatorname{arcsinh}\left(\frac{\theta_j(t)}{2\xi_j}\right)\right] + C',$$

where C' is a constant to make $\phi_j(0) = 0$:

$$C' = \frac{4\alpha}{n(1-\mu)} \left[(\sum_{i=1}^{n} x_{i;j}^2) \left(\sqrt{\theta_j^2(0) + 4\xi_j^2} - 2\xi_j \right) + \sum_{i=1}^{n} \epsilon_{i;j}(0) \operatorname{arcsinh} \left(\frac{\theta_j(0)}{2\xi_j} \right) \right]. \tag{56}$$

Note that when the initialization is unbiased, then we simply have $C' = 0$. Since we assume θ converges to the interpolation solution, i.e.,

$$\theta^T(\infty) x_i = y_i, \ \forall i \in \{1, \dots n\},$$

which implies that

$$\begin{aligned} \sum_{k=1}^{d} \theta_k(\infty) x_{i;k} - y_i = 0, \ \forall i \in \{1, \dots n\} \implies -\theta_j(\infty) x_{i;j} &= \sum_{k=1, k\neq j}^{d} \theta_k(t) x_{i;k} - y_i \\ \implies \epsilon_{i;j}(\infty) &= -\theta_j(\infty) x_{i;j}^2, \end{aligned} \tag{57}$$

we obtain the form of $\phi_j(\infty)$:

$$\begin{aligned} \alpha\phi_j(\infty) &= \frac{4\alpha(\sum_{i=1}^{n} x_{i;j})^2}{n(1-\mu)} \left[\theta_j(\infty) \operatorname{arcsinh} \left(\frac{\theta_j(\infty)}{2\xi_j} \right) - \sqrt{\theta_j^2(\infty) + 4\xi_j^2} + 2\xi_j \right] + C' \\ &= \frac{16\alpha(\sum_{i=1}^{n} x_{i;j})^2}{n(1-\mu)} Q_{\xi,j}^{\mathrm{M}}(\theta(\infty), 0) + C'. \end{aligned} \tag{58}$$

□

D.3 Proof of Theorem 3

In this section, we prove Theorem 3.

Proof. If we define

$$Q_{\bar{\xi}(t)}^{\mathrm{M}}(\bar{\theta}_\alpha, t) = \sum_{j=1}^{d} Q_{\xi,j}^{\mathrm{M}}(\bar{\theta}_\alpha, t), \tag{59}$$

then its gradient w.r.t $\bar{\theta}_\alpha$ is

$$\nabla Q_{\bar{\xi}(t)}^{\mathrm{M}}(\bar{\theta}_\alpha, t) = \left(\nabla Q_{\xi,1}^{\mathrm{M}}(\bar{\theta}_\alpha, t), \dots, \nabla Q_{\xi,d}^{\mathrm{M}}(\bar{\theta}_\alpha, t) \right)^T,$$

which implies that

$$\begin{aligned} \frac{d}{dt} \nabla Q_{\bar{\xi}(t)}^{\mathrm{M}}(\bar{\theta}_\alpha, t) &= \begin{pmatrix} \frac{d}{dt} \nabla Q_{\xi,1}^{\mathrm{M}}(\bar{\theta}_\alpha, t) \\ \vdots \\ \frac{d}{dt} \nabla Q_{\xi,d}^{\mathrm{M}}(\bar{\theta}_\alpha, t) \end{pmatrix} \\ &= -\frac{\nabla L(\theta)}{1-\mu} \end{aligned} \tag{60}$$

where we apply Proposition 4 in the second equality. Integrating both sides of the above equation from 0 to ∞ gives us

$$\nabla Q_{\bar{\xi}(\infty)}^{\mathrm{M}}(\bar{\theta}_\alpha(\infty), \infty) - \nabla Q_{\bar{\xi}(0)}^{\mathrm{M}}(\bar{\theta}_\alpha(0), 0) = -\sum_{i=1}^{n} \frac{x_i}{n(1-\mu)} \int_0^\infty r_i(s) ds = \sum_{i=1} x_i \lambda_i. \tag{61}$$

On the other hand, as $t \to \infty$, since we assume $\theta(\infty)$ converges to the interpolation solution, we have, according to Eq. (35),

$$\alpha\dot{\theta}(\infty) \propto \alpha \nabla L(\theta(\infty)) = 0 \implies \bar{\theta}_\alpha(\infty) = \theta(\infty).$$

Considering that $\nabla Q_{\bar{\xi}(0)}^{\mathrm{M}}(\bar{\theta}_\alpha(0), 0) = 0$ when we derive the form of $Q_{\xi,j}^{\mathrm{M}}$ (Eq. (40)), Eq. (61) implies that

$$\nabla Q_{\bar{\xi}(\infty)}^{\mathrm{M}}(\theta(\infty), \infty) = \sum_{i=1}^{n} x_i \epsilon_i. \tag{62}$$

Note that the KKT condition for the optimization problem in Theorem 3 is

$$\nabla Q_{\bar{\xi}(\infty)}^{\mathrm{M}}(\theta(\infty), \infty) - \sum_{i=1}^{n} x_i \lambda_i = 0, \tag{63}$$

which is exactly Eq. (62). Thus we finish the proof. □

D.4 Equivalence between Eq. (6) and standard diagonal linear networks

A standard diagonal linear network is

$$f(x;u,v) = (u \odot v)^T x.$$

If u and v of this model follows the HB and NAG flow (Proposition 1) under the square loss, we note that there is no difference between the forms of their dynamics, since the model $f(x;u,v)$ is completely symmetrical regarding u and v, i.e., changing the places of u and v would induce exactly the same model and make no difference. More specifically, the dynamics of $u \odot u$ is

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}u \odot u = u \odot \dot{u} &= -u \odot \left(\frac{\nabla_u L}{1-\mu} + \alpha \ddot{u}\right) \\ &= -\frac{2}{1-\mu} u \odot (X^T r) \odot v - \left[\alpha \frac{d}{dt} u \odot \dot{u} - \alpha \dot{u} \odot \dot{u}\right] \\ &= -\frac{2}{1-\mu} u \odot (X^T r) \odot v + \alpha \left[\frac{2}{1-\mu}\frac{d}{dt} u \odot (X^T r) \odot v + \frac{4(X^T r) \odot v \odot (X^T r) \odot v}{(1-\mu)^2}\right] \end{aligned} \tag{64}$$

and the dynamics of $v \odot v$ can be obtained by simply replacing all u with v in Eq. (64). Thus if $u(0) \odot u(0) = v(0) \odot v(0)$, i.e., $|u(0)| = |v(0)|$, then we can immediately conclude that $|u(t)| = |v(t)|$ for any $t \geq 0$. Thus we can equivalently parameterize this model with $f(x;u) = (u \odot u)^T x$ further add the weight v such that $f(x;u,v) = (u \odot u - v \odot v)^T x$ can output negative values.

D.5 Analysis on the effects of the initialization

For simplicity, we consider the unbiased initialization, and the case for biased initialization is similar. Since the solutions of HB and NAG are composed of two parts, $Q^{\mathrm{M}}_{\bar{\xi}(\infty)} = \sum_{j=1}^d Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, \infty)$ where (note that $\bar{\theta}_\alpha(\infty) = \theta(\infty)$ according to the proof of Theorem 3)

$$Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t) = \frac{1}{4}\left[\bar{\theta}_{\alpha;j} \operatorname{arcsinh}\left(\frac{\bar{\theta}_{\alpha;j}}{2\bar{\xi}_j(t)}\right) - \sqrt{4\bar{\xi}_j^2(t) + \bar{\theta}^2_{\alpha;j}} + 2\bar{\xi}_j(t)\right]$$

and $\mathcal{R} = (\mathcal{R}_j, \ldots, \mathcal{R}_d)^T \in \mathbb{R}^d$ where

$$\forall j \in \{1, \ldots, d\} : \mathcal{R}_j = \frac{1}{4} \operatorname{arcsinh}\left(\frac{4\alpha (X^T y)_j}{n(1-\mu)}\right),$$

we need to analyze both parts to show the transition from the rich regime to kernel regime, which is different from the case for GD where one only needs to consider the hyperbolic entropy part.

Small initialization $\xi \to 0$. We first discuss $Q^{\mathrm{M}}_{\xi,j}$. When $\xi \to 0$, we have that

$$-\sqrt{4\bar{\xi}_j^2(t) + \bar{\theta}^2_{\alpha;j}} + 2\bar{\xi}_j(t) \to -|\bar{\theta}_{\alpha;j}|$$

and

$$\frac{\bar{\theta}_{\alpha;j}}{2\bar{\xi}_j(t)} \to \operatorname{sign}(\bar{\theta}_{\alpha;j})\infty \implies \bar{\theta}_{\alpha;j} \operatorname{arcsinh}\left(\frac{\bar{\theta}_{\alpha;j}}{2\bar{\xi}_j(t)}\right) \to \operatorname{sign}(\bar{\theta}_{\alpha;j})\bar{\theta}_{\alpha;j}\infty = |\bar{\theta}_{\alpha;j}|\infty,$$

thus $Q^{\mathrm{M}}_{\bar{\xi}(\infty)} \to \sum_{j=1}^d |\theta_j(\infty)|\infty = \|\theta(\infty)\|_1 \infty$. On the other hand, the $\theta^T \mathcal{R}^{\mathrm{M}}$ part is finite thus negligible compared to $Q^{\mathrm{M}}_{\bar{\xi}(\infty)}$. As a result, we conclude that $\xi \to 0$ corresponds to the rich regime.

Large initialization $\xi \to \infty$. For $Q^{\mathrm{M}}_{\xi,j}$, we note that as $\xi \to \infty$, similar to the case for GD,

$$-\sqrt{4\bar{\xi}_j^2(t) + \bar{\theta}^2_{\alpha;j}} + 2\bar{\xi}_j(t) \to 0$$

and

$$\bar{\theta}_{\alpha;j} \operatorname{arcsinh}\left(\frac{\bar{\theta}_{\alpha;j}}{2\bar{\xi}_j(t)}\right) \to \frac{\bar{\theta}^2_{\alpha;j}}{2\xi_j},$$

thus we obtain that, as $\xi \to \infty$

$$Q^{\mathrm{M}}_{\xi,j}(\bar{\theta}_\alpha, t) \propto \bar{\theta}^2_{\alpha;j} \implies Q^{\mathrm{M}}_{\bar{\xi}(\infty)} \propto \|\theta(\infty)\|_2^2. \tag{65}$$

On the other hand, $\theta^T \mathcal{R}^{\mathrm{M}}$ is simply a inner produce between θ and $\mathcal{R}^{\mathrm{M}}$. Thus $Q^{\mathrm{M}} + \theta^T \mathcal{R}^{\mathrm{M}}$ is captured by a kernel and $\xi \to \infty$ corresponds to the kernel regime.

D.6 Proof of Proposition 2

Proof. Since momentum-based methods is a second-order ODE by adding a perturbation proportional to η to the re-scaled GF ODE

$$\dot{u} = -\frac{\nabla_u L}{1-\mu},\ \dot{v} = -\frac{\nabla_v L}{1-\mu},$$

Proposition 2 can be proved by following similar steps as the proof of Theorem 3 in Appendix D.3. In particular, let all terms of the order of η be zero (thus we ignore the perturbation brought by momentum) and let $\mu = 0$ (thus we make the re-scaled GF a standard GF) in the proof of Theorem 3, we can directly conclude that

$$\theta(\infty) = \arg\min_{\theta} Q(\theta),\ s.t.\ X\theta = y \tag{66}$$

where

$$\begin{aligned} &Q(\theta) = Q_{\xi}^{\mathrm{GF}}(\theta) + \theta^T \mathcal{R}^{\mathrm{GF}}, \\ &\mathcal{R}^{\mathrm{GF}} = (\mathcal{R}_1^{\mathrm{GF}}, \ldots, \mathcal{R}_d^{\mathrm{GF}})^T \in \mathbb{R}^d,\ \forall j \in \{1, \ldots, d\} : \mathcal{R}_j^{\mathrm{GF}} = \frac{1}{4} \operatorname{arcsinh}\left(\frac{\theta_j(0)}{2\xi_j}\right). \end{aligned} \tag{67}$$

□