arXiv:2506.14806v2 [cs.LG] 22 Oct 2025

Heavy-Ball Momentum Method in Continuous Time and Discretization Error Analysis

Bochen Lyu[a,b,*] **Xiaojing Zhang**[b] **Fangyi Zheng**[c] **He Wang**[d] **Zheng Wang**[e]

Zhanxing Zhu[a*]

[a]University of Southampton [b]DataCanvas [c]Pony.ai [d]University College London
[e]University of Leeds
[*]{bochen.lyu, z.zhu}@soton.ac.uk

Abstract

This paper establishes a continuous time approximation, a piece-wise continuous differential equation, for the discrete Heavy-Ball (HB) momentum method with explicit discretization error. Investigating continuous differential equations has been a promising approach for studying the discrete optimization methods. Despite the crucial role of momentum in gradient-based optimization methods, the gap between the original discrete dynamics and the continuous time approximations due to the discretization error has not been comprehensively bridged yet. In this work, we study the HB momentum method in continuous time while putting more focus on the discretization error to provide additional theoretical tools to this area. In particular, we design a first-order piece-wise continuous differential equation, where we add a number of counter terms to account for the discretization error explicitly. As a result, we provide a continuous time model for the HB momentum method that allows the control of discretization error to *arbitrary order* of the step size. As an application, we leverage it to find a new implicit regularization of the directional smoothness and investigate the implicit bias of HB for diagonal linear networks, indicating how our results can be used in deep learning. Our theoretical findings are further supported by numerical experiments.

1 Introduction

Gradient descent (GD) and its variants momentum methods, such as Polyak's Heavy-Ball momentum (HB) (Polyak, 1964), Nesterov's method of accelerated gradients (NAG) (Nesterov, 1983), and Adam (Kingma and Ba, 2017), are at the core of the success of training deep neural networks. This leads to the importance of understanding the gradient-based optimization methods, whereas the direct analysis for their discrete learning dynamics is challenging. Hence an optional approach is generally applied in this area: leveraging tools from dynamical systems to study the learning dynamics in continuous time to shed light on the dynamical behaviors of the discrete updates, e.g., Barrett and Dherin (2022); Lyu and Li (2020); Ji and Telgarsky (2019); Li et al. (2022); Lyu and Zhu (2023); Miyagawa (2023); Rosca et al. (2023). However, one fundamental gap—the discrepancy between the discrete optimizations and their continuous time models named as the *discretization error*—becomes manifest when employing this approach.

While recent works have made significant efforts to fill the aforementioned gap for GD (Barrett and Dherin, 2022; Miyagawa, 2023; Rosca et al., 2023), the analysis for the discretization error of the important variants of GD—momentum methods—in continuous time is still far from comprehensive. Along this line of research, Kovachki and Stuart (2020) demonstrated that the continuous time approximation of HB and NAG, which solves $\min_\beta L(\beta)$, can be expressed as a rescaled gradient

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

flow (RGF):

$$\dot{\beta} = -\frac{\nabla L(\beta)}{1-\mu}, \tag{1}$$

where L is the objective function and μ is the momentum factor. Despite its convenience for studying discrete HB and NAG, this approximation has one drawback: the insufficient consideration for discretization error when approaching the continuous time limit. As a result, it is insufficient to differentiate between HB and GD in continuous time. For example, the solutions of Eq. (1) and that of the gradient flow (GF) $\dot{\beta} = -\nabla L(\beta)$, the simplest continuous time model of GD, are not very different: if $\beta^*(t)$ solves RGF and $\beta^\dagger(t)$ solves GF, then $\beta^\dagger(t) = \beta^*((1-\mu)t)$, which suggests that GD and HB will converge to similar points. However, this is inconsistent with the actual behavior, e.g., in Fig. 2(a) GD and HB converge to different points, which reveals that the RGF cannot fully capture the difference between HB and GD and motivates us to study continuous approximations for HB with smaller discretization error.

Recently, backward error analysis (Hairer et al., 2006) has been successfully applied to construct continuous time approximations for GD (Barrett and Dherin, 2022; Miyagawa, 2023; Rosca et al., 2023) with lower discretization error than GF. Inspired by these works, Ghosh et al. (2023) provided a continuous differential equation for HB, which achieves smaller discretization error compared to RGF (Eq. (1)). While promising, it still does not fully capture the actual discrete learning dynamics since the discretization error is only to the second order of the step size.

To this end, we also study the HB momentum method in continuous time while putting more focus on closing the gap due to the discretization error. In particular, we propose HB Flow (HBF), a pice-wise continuous differential equation as a novel continuous time approximation for the HB momentum method. We add a number of counter terms to the improved version of Eq. (1) to cancel the discretization error, obtaining a continuous time differential equation that can be *arbitrarily close* to HB, i.e., the discretization error can be controlled to arbitrary order of the step size. Our HBF provides a more reliable foundation for studying momentum methods in continuous time when the direct study of discrete learning dynamics is cumbersome. As a case study, we examine the implicit bias of HB, the preference for certain kind of solution without explicit regularization, by employing the HBF for the popular diagonal linear networks where there are already abundant results for GD (Azulay et al., 2021; Even et al., 2023; Pesme et al., 2021; Pillaud-Vivien et al., 2020; Woodworth et al., 2020; Yun et al., 2021).

Contributions. We study HB—one of the most important gradient-based optimization algorithms with momentum—in continuous time. In particular,

1. We propose a piece-wise continuous differential equation (HBF) by adding a counter term (Theorem 2.1) that can be expanded in a series formulation to different orders of the step size to the RGF (Eq. (1)), such that the HBF can precisely capture the learning dynamics of the discrete HB.
2. We explicitly show the leading order of the discretization error for the HBF and indicate the way how one can obtain HBF with a discretization error to any order of the step size (Section 2.2), revealing the explicit precision of HBF for approximating the discrete HB. These results are firstly revealed in this work to the best of our knowledge, bridging the gap between the discrete HB and the its precise characterization in continuous time.
3. We leverage our HBF to examine learning dynamics of HB. For example, we reveal that, as HBF implicitly has a regularization term for the directional smoothness, the learning dynamics of HB exhibits smaller directional smoothness compared to GD (Fig.1). In addition, for the application of HBF in deep learning, we investigate the implicit bias of HB for the diagonal linear network (Theorem 3.1) through the lens of HBF as a case study, revealing its difference compared to that of GF which cannot be obtained by the RGF.

1.1 Related Works

Continuous approximation Prior works (Shi et al., 2022; Su et al., 2016; Wilson et al., 2016) have constructed second-order ODEs to study the convergence properties of HB/NAG in continuous time, where the momentum factor typically depends on the step size and the iteration count. In this paper, we study HB with a constant momentum factor that is independent of step size and iteration

count, a setting that is more consistent with the practical case, e.g., PyTorch Paszke et al. (2019), and we focus more on the order of the discretization error of the continuous time model. Towards this direction, besides Kovachki and Stuart (2020); Ghosh et al. (2023), Cattaneo et al. (2023) focused on the continuous limit of a more general adaptive gradient-based methods with momentum, Adam, achieving a discretization error to the second order of the step size. As a comparison, we provide a continuous differential equation that can be arbitrarily close to the discrete HB.

Implicit bias of optimizers The implicit bias of GD for various deep neural networks has been widely studied in, e.g., Ji and Telgarsky (2019); Soudry et al. (2018); Yun et al. (2021) for linear networks and Chizat and Bach (2020); Lyu and Li (2020) for homogeneous networks. For diagonal linear networks studied in this paper, Azulay et al. (2021); Pesme et al. (2021); Pillaud-Vivien et al. (2020); Woodworth et al. (2020) revealed the interesting transition from kernel to rich regime by altering the scale of initialization. Papazov et al. (2024) studied HB using a second-order ordinary differential equation (ODE) with discretization error to the second order of step size. For linearly separable data and linear model, Wang et al. (2022) showed that momentum method converges to the ℓ_2-max-margin solution, which is the same as GD. As a comparison, Zhang et al. (2024) revealed that Adam with negligible stability constant exhibits the preference of ℓ_∞-max-margin solution. Furthermore, Wang et al. (2021) studied the implicit bias of adaptive optimization algorithms on homogeneous deep neural networks.

1.2 Preliminaries

Notations For a vector $\beta \in \mathbb{R}^d$ that depends on time t, we use $\dot{\beta}$ and $\ddot{\beta}$ to denote its first and second derivative with respect to time t, respectively. We use β_j to denote its j-th component and $\|\beta\|_p$ for its ℓ_p-norm. We use $\alpha \cdot \beta$ to denote the inner product and $\odot$ to denote elementwise product, e.g., $\alpha \cdot A \cdot \beta$ denotes $\alpha^T A \beta$ for $\alpha \in \mathbb{R}^{d_1}, A \in \mathbb{R}^{d_1 \times d_2}, \beta \in \mathbb{R}^{d_2}$. We use $[N]$ for integers between $[0, N]$.

Heavy-Ball momentum method HB (Polyak, 1964) employs a two-step updating scheme (Sutskever et al., 2013), rather than the single-step manner of GD. Particularly, HB first accumulates the history of past iterations before updating the model parameter $\beta \in \mathbb{R}^d$, i.e., $m_{k+1} = \mu m_k - \eta \nabla L(\beta_k),\ \beta_{k+1} = \beta_k + m_{k+1}$ where $\mu \in (0, 1)$ is the momentum factor, η is the step size, k is the iteration number, and $m \in \mathbb{R}^d$ is the momentum, which can be further written in a single equation

$$\textbf{Discrete HB:}\ \ \beta_{k+1} - \beta_k = -\eta \nabla L(\beta_k) + \mu \left(\beta_k - \beta_{k-1}\right). \tag{2}$$

2 HB Momentum Methods in Continuous Time and Discretization Error

In this section, we will propose an ODE

$$\dot{\beta}(t) = -\mathcal{G}(\beta) \tag{3}$$

that can be arbitrarily close to the discrete learning dynamics of HB Eq. (33). To characterize the gap between the discrete learning dynamics and the ODE Eq. (3), given $N > 0$, we define the *discretization error* of Eq. (3) as

$$\forall k \in [N]:\ \varepsilon_k = \beta(k\eta) - \beta_k, \tag{4}$$

where $\beta(k\eta)$ solves Eq. (3) and we will denote $t_k = k\eta$ for convenience in the rest of this paper. An investigation of Euler forward method will identify η as the step size. When $\varepsilon_k = \mathcal{O}\left(\eta^\alpha\right)$, we say that the discretization error of the continuous time model Eq. (3) is to the α-th order of the step size. We aim to find the formulation of $\mathcal{G}(\beta)$ such that Eq. (3) can be arbitrarily close to the discrete HB, i.e., $\varepsilon_k = \mathcal{O}\left(\eta^\alpha\right)$ for any given α.

Intuition of our approach Given the discrete HB Eq. (33), Kovachki and Stuart (2020) showed that $\varepsilon_k = \mathcal{O}\left(\eta\right)$ if $\mathcal{G}(\beta) = \nabla L/(1-\mu)$ as in Eq. (1). To coincide with such observation, our overall goal is to find the formulation of a modified ODE $\dot{\beta} = -G - \eta\gamma$ such that the discretization error can be controlled arbitrarily low, where its design should follow two basic principles: (i). it should coincide with RGF, the simplest continuous approximation of HB RGF, when η is very small, thus, G should degenerate to $\nabla L/(1-\mu)$; (ii). by adding the *counter term* γ, it should allow us to further

decrease the discretization error of RGF to get better continuous approximations for HB. To achieve this, we need to derive the exact forms of G and γ under the condition $\varepsilon_k = \mathcal{O}(\eta^\alpha)$. And there will be two key steps:

1. find the equations that G and γ must satisfy if we require the discretization error $\varepsilon_k = \mathcal{O}(\eta^\alpha)$ for any $\alpha > 0$;
2. solve these equations to give the formulations of G and γ.

Overview of our approach We now apply the above intuition to establish such a continuous time approximation of HB. Adding the counter term γ directly to the RGF is, however, problematic due to the fact that each iteration of momentum methods exploits the history of previous iterations. This renders the local error analysis unreliable since it ignores previous updates. To perform a global analysis, instead of directly utilizing the backward error analysis in Barrett and Dherin (2022), we propose a piece-wise continuous differential equation by decomposing $\mathcal{G}(\beta)$ into two parts, named as HB Flow (HBF),

$$t \in [t_k, t_{k+1}) : \ \dot{\beta} = -\mathcal{G}(\beta) := -G_k(\beta) - \eta\gamma_k(\beta). \tag{5}$$

This was previously discussed in Ghosh et al. (2023). In Eq. (5), k denotes the iteration count for the discrete updates, $t_k = k\eta$, and G_k depends on k and should degenerate to $\nabla L/(1-\mu)$ as in Eq. (1) for small step size. Additionally, the counter term γ_k is designed to cancel the further discretization error brought by $G_k(\beta)$ such that ε_k can be controlled to be arbitrarily small, hence the name counter term. As discussed in the above intuition, deriving the formulation of γ_k needs a set of equations for it to satisfy. We establish such equations as follows. When approximating t_k from $t > t_k$ and $t < t_k$, respectively, Taylor expansion for β in Eq. (5) provides us

$$\begin{aligned} \beta(t_{k+1}) - \beta(t_k) &= \eta\dot{\beta}(t_k^+) + \eta^2 I_k^+ = -\eta G_k - \eta^2\gamma_k + \eta^2 I_k^+ \\ \beta(t_k) - \beta(t_{k-1}) &= \eta\dot{\beta}(t_k^-) - \eta^2 I_k^- = -\eta G_{k-1} - \eta^2\gamma_{k-1} - \eta^2 I_k^- \end{aligned} \tag{6}$$

where t_k^+ and t_k^- mean that we approximate t_k from $t > t_k$ and $t < t_k$, respectively, $\eta^2 I_k^\pm = \int_{t_k}^{t_{k\pm1}} \ddot{\beta}(\tau)(t_{k\pm1} - \tau)d\tau$, and we apply Eq. (5) for $\dot{\beta}(t_k^\pm)$. A subtraction of the discrete HB update Eq. (2) to the first equality of Eq. (6) now allows us to construct the relation between ε_k and γ_k:

$$\varepsilon_{k+1} - \varepsilon_k = \mu(\varepsilon_k - \varepsilon_{k-1}) - \eta\left[G_k - \mu G_{k-1} - \nabla L(\beta_k)\right] + \eta^2\left[I_k^+ + \mu I_k^- - \gamma_k + \mu\gamma_{k-1}\right]. \tag{7}$$

As $I_k^\pm$ can be expressed by γ_k, requiring $\varepsilon_{k+1} - \varepsilon_k = \mathcal{O}(\eta^{\alpha+1})$ in Eq. (7) [1] immediately builds an equation for γ_k, which can be solved to give the form of γ_k. Hence, the discretization error of the continuous time model Eq. (5) for the discrete update Eq. (2) can be controlled to the α-th order of the step size η. The formulation of Eq. (5) is presented in Theorem 2.1, and the detailed technical proofs are deferred to Appendix A.

2.1 HBF with Discretization Error to Arbitrary Order of the Step Size

Following the intuition and overview of our approach, we now solve Eq. (7) to derive γ_k and G_k, and reveal that the obtained HBF by doing so is indeed an $\mathcal{O}(\eta^\alpha)$ (piece-wise) continuous time model of HB. As discussed earlier in Eq. (7), we need to find γ_k to ensure that $\varepsilon_{k+1} - \varepsilon_k = \mathcal{O}\left(\eta^{\alpha+1}\right)$, hence the following integral functional equation

$$\eta^2(I_k^+ + \mu I_k^- - \gamma_k + \mu\gamma_{k-1}) = \mathcal{O}\left(\eta^{\alpha+1}\right) \tag{8}$$

must be satisfied. We solve this equation following three steps as shown below: **(1).** write the solution of Eq. (8), γ_k, as a series form

$$\gamma_k = \sum_{\sigma=0}^{\infty} \eta^\sigma \gamma_k^{(\sigma)}; \tag{9}$$

(2). derive $I_k^\pm$ explicitly by employing Eq. (9) in the learning dynamics Eq. (5) in the series form

$$I_k^+ = \sum_{\sigma=0}^{\infty} \eta^\sigma (\mathscr{I}_k^+)^{(\sigma)}; \tag{10}$$

[1] We will show in Appendix that this condition is sufficient for proving $\varepsilon_k = \mathcal{O}\left(\eta^\alpha\right)$ by induction.

(3). match the terms of Eq. (9) with that of Eq. (10) to each order of the step size η, i.e., $\forall \sigma \in \mathbb{N}$: $\gamma_k^{(\sigma)} = (\mathscr{J}_k^+)^{(\sigma)}$. As a result, a simple truncation of γ_k will automatically lead Eq. (7) to give us $\varepsilon_{k+1} - \varepsilon_k = \mathcal{O}\left(\eta^{\alpha+1}\right)$ as desired. Below we formalize the aforementioned discussion.

Theorem 2.1 (HBF with the discretization error $\varepsilon_k = \mathcal{O}(\eta^\alpha)$)**.** *Let $k \in [N]$ be the iteration count, η be the step size, and $t_k = k\eta$, the piece-wise continuous time differential equation HB Flow (HBF) for the discrete update Eq.* (33)

$$\textbf{HBF:}\ \dot{\beta} = -G_k(\beta) - \eta\gamma_k(\beta), \quad t \in [t_k, t_{k+1}) \tag{11}$$

has a discretization error that satisfies

$$\varepsilon_{k+1} - \varepsilon_k = \mathcal{O}\left(\eta^{\alpha+1}\right) \tag{12}$$

and, as a result, $\varepsilon_k = \mathcal{O}(\eta^\alpha)$ for $\alpha \geq 1$, if G_k and γ_k has the following formulations:

$$G_k = \mu G_{k-1} + \nabla L \tag{13}$$

$$\gamma_k = \sum_{\sigma=0}^{\alpha-2} \eta^\sigma \gamma_k^{(\sigma)}. \tag{14}$$

In particular, for ease of notation, let $\mathbf{L}_\beta^{(k,\sigma)} = \gamma_k^{(\sigma-1)} \cdot \nabla$ be a differential operator and

$$\gamma_k^{(-1)} = G_k, \quad \mathcal{S}_{m,\sigma} = \{(\sigma_1, \ldots, \sigma_m) | \sum_{i=1}^m \sigma_i = \sigma - m + 2, \sigma_i \in \mathbb{N}\}$$

$$\chi_j^{(\sigma)} = \sum_{m=2}^{\sigma+2} \sum_{\mathcal{S}_{m,\sigma}} \frac{1}{m!} \Big[(-1)^m \mathbf{L}_\beta^{(j,\sigma_1)} \cdots \mathbf{L}_\beta^{(j,\sigma_{m-1})} \gamma_j^{(\sigma_m - 1)} + \mu \mathbf{L}_\beta^{(j-1,\sigma_1)} \cdots \mathbf{L}_\beta^{(j-1,\sigma_{m-1})} \gamma_{j-1}^{(\sigma_m-1)} \Big],$$

then each term of γ_k in Eq. (14) *can be simply written as*

$$\forall \sigma \in \mathbb{N} : \ \gamma_k^{(\sigma)} = \sum_{j=0}^{k} \mu^{k-j} \chi_j^{(\sigma)}. \tag{15}$$

Remark 1 Eq. (12) is an *equality* obtained from solving the integral functional equation Eq. (8), rather than an inequality bound. In addition, Eq. (12) is established to *arbitrary order of the step size* η for any given $\alpha \geq 1$. Our approach is different from the previous continuous time model for HB method (Ghosh et al., 2023) where the discretization error is specifically constructed to the second order of the step size.

Remark 2 In Eq. (13), the formulation of G_k intuitively resembles the update of momentum in the discrete learning dynamics of HB $m_k = \mu m_{k-1} - \nabla L$. Interestingly, G_k can be further simplified as

$$G_k = \frac{1 - \mu^{k+1}}{1 - \mu} \nabla L \xrightarrow{\text{large } k} \frac{\nabla L}{1 - \mu}, \tag{16}$$

which is exactly the R.H.S of the RGF (Eq. (1)). Moreover, Eq. (15) indicates that each iteration of HB depends on the history of previous iterations as $\gamma_k^{(\sigma)}$ incorporates information of all previous $\chi_j^{(\sigma)}$ with $j \leq k$, and such dependence decays very fast due to the coefficient $\mu^{k-j} \ll 1$ for small j. By letting $\mu = 0$, all the dependence on k will disappear and our results can recover those of GD. Interestingly, it is worth mentioning that the difference between HBF and the continuous approximations of GD is closely related to the powers of $\eta(1+\mu)/(1-\mu)^2$ as we will show in Section 2.2.

Remark 3 The formulations of $\gamma_k^{(\sigma)}$ are obtained in a recursive manner, ($\chi^{(\sigma)}$ is obtained from $\gamma_k^{(\sigma')}$'s with $\sigma' < \sigma$), hence Theorem 2.1 allows us to always build continuous time models for HB method with smaller discretization errors given those with larger discretization errors, instead of performing the whole set of analysis. For example, to build the HBF with $\varepsilon_k = \mathcal{O}\left(\eta^3\right)$ given HBF with $\varepsilon_k = \mathcal{O}\left(\eta^2\right)$, we only need to derive $\gamma_k^{(1)}$ (since γ_k is truncated to the order $\alpha - 2 = 1$), which can be obtained by applying $\gamma_k^{(0)}$ provided by the HBF with $\varepsilon_k = \mathcal{O}\left(\eta^2\right)$. In addition, the recursive manner in Theorem 2.1 provides a possibility to calculate the involved terms automatically by using software for symbolic mathematics such as SymPy, which could be an interesting future work.

$\varepsilon_k = \mathcal{O}(\eta^\alpha)$	GD	HB
$\alpha = 1$	$\dot{\beta} = -\nabla L$	$\dot{\beta} = -\frac{\nabla L}{(1-\mu)}$ (Kovachki and Stuart, 2020)
$\alpha = 2$	$\dot{\beta} = -\nabla L - \eta\frac{\nabla L \cdot \nabla^2 L}{2}$ (Barrett and Dherin, 2022)	$\dot{\beta} = -\frac{\nabla L}{1-\mu} - \eta\frac{1+\mu}{(1-\mu)^3}\frac{\nabla L \cdot \nabla^2 L}{2}$ Eq. (19) and (Ghosh et al., 2023)
$\alpha = 3$	$\dot{\beta} = -\nabla L - \eta\frac{\nabla L \cdot \nabla^2 L}{2}$ $-\eta^2\left[\frac{\omega_1}{4} + \frac{\omega_2}{12}\right]$ Miyagawa (2023); Rosca et al. (2023)	$\dot{\beta} = -\frac{\nabla L}{1-\mu} - \eta\frac{1+\mu}{(1-\mu)^3}\frac{\nabla L \cdot \nabla^2 L}{2}$ $-\frac{\eta^2(1+\mu)^2}{(1-\mu)^5}\left[\frac{\omega_1}{4} + \frac{(1+10\mu+\mu^2)\omega_2}{12(1+\mu)^2}\right]$ Eq. (22) of this work
Arbitrary α	Miyagawa (2023); Rosca et al. (2023)	Theorem 2.1 of this work
Discrete	$\beta_{k+1} = \beta_k - \nabla L(\beta_k)$	$\beta_{k+1} = \beta_k - \eta\nabla L(\beta_k) + \mu(\beta_k - \beta_{k-1})$

Table 1: Continuous approximations for GD and HB up to different orders of discretization error.

2.2 HBF with Discretization Error $\varepsilon_k = \mathcal{O}(\eta^2)$ and $\varepsilon_k = \mathcal{O}(\eta^3)$

In this section, we derive HBF with discretization error to the second and third order of the step size to indicate how our approach works. There are basically three steps for finding a HBF with $\varepsilon_k = \mathcal{O}(\eta^\alpha)$ for $\alpha \geq 1$:

1. truncate γ_k to the order $\alpha - 2$ in Eq. (14), i.e, $\gamma_k = \sum_{\sigma=0}^{\alpha-2}\gamma_k^{(\sigma)}$;
2. from the smallest $\sigma = 0$ to $\sigma = \alpha - 2$, find all $\chi_j^{(\sigma)}$ with $j \leq k$ by identifying the corresponding $\mathcal{S}_{m,\sigma}$ with $m = \{2, \ldots, \sigma + 2\}$ for each σ;
3. derive the expression of $\gamma_k^{(\sigma)}$ for all $\sigma \leq \alpha - 2$ in a recursive manner using the relation $\gamma_k^{(\sigma)} = \sum_{j=0}^{k}\mu^{k-j}\chi_j^{(\sigma)}$.

Below we discuss the cases for $\alpha = 2, 3$. We also summarize these results in Table 1. Note that the case for $\alpha = 1$ states that HBF is a RGF, i.e., $\dot{\beta} = -\nabla L/(1-\mu)$, which might not fully characterize the difference between momentum methods and vanilla GD.

2.2.1 HBF with $\varepsilon_k = \mathcal{O}(\eta^2)$

According to Theorem 2.1, there is only one term in the series of γ_k, i.e., $\gamma_k^{(0)}$. Recall that $\mathbf{L}_\beta^{(k,0)} = G_k \cdot \nabla$ and there is only one element in the set $\mathcal{S}_{m=2,\sigma=0}$, i.e., $\mathcal{S}_{m=2,\sigma=0} = \{(\sigma_1 = 0, \sigma_2 = 0)\}$, we obtain for $j \geq 1$:

$$\chi_j^{(0)} = \frac{1}{2}\left[\mathbf{L}_\beta^{(j,0)}\gamma_j^{(-1)} + \mu\mathbf{L}_\beta^{(j-1,0)}\gamma_{j-1}^{(-1)}\right]. \tag{17}$$

Thus, using the definition of $\mathbf{L}_\beta^{(j,0)}$ and $\gamma_j^{(1)}$ according to Eq. (15) in Theorem 2.1, we can immediately derive that

$$\gamma_k = \gamma_k^{(0)} = \frac{\nabla L \cdot \nabla^2 L}{2(1-\mu)^2}\sum_{j=0}^{k}\mu^{k-j}\left[(1-\mu^{j+1})^2 + \mu(1-\mu^j)^2\right]. \tag{18}$$

Interestingly, all the dependence on the iteration count k exists in the form of μ^k, then for large iteration count k, the form of γ_k can be largely simplified as $\gamma_k \approx \frac{1+\mu}{(1-\mu)^3}\frac{\nabla L \cdot \nabla^2 L}{2}$. This gives us HBF with $\varepsilon_k = \mathcal{O}(\eta^2)$:

$$\dot{\beta} = -\frac{\nabla L}{1-\mu} - \eta\frac{1+\mu}{(1-\mu)^3}\frac{\nabla L \cdot \nabla^2 L}{2}, \tag{19}$$

which is consistent with the $\mathcal{O}(\eta^2)$continuous approximation of HB in Ghosh et al. (2023) while our derivation of HBF is in a different approach that does not depend inequality bounds. It is worth to mention that when $\mu = 0$, HBF recovers the $\mathcal{O}(\eta^2)$ continuous approximation of GD as expected.

2.2.2 HBF with $\varepsilon_k = \mathcal{O}(\eta^3)$

As shown in Theorem 2.1, in this case, the series of γ_k should be truncated to the order $\alpha - 2 = 1$, hence $\gamma_k = \gamma_k^{(0)} + \eta\gamma_k^{(1)}$. Since we have already derived $\chi_j^{(0)}$ for HBF with $\varepsilon_k = \mathcal{O}(\eta^2)$ in

Section 2.2.1, we only need to find $\chi_k^{(1)}$, which will give us $\gamma_k^{(1)}$. We first find the collection of sets $\mathcal{S}_{m,\sigma}$, where $m = \{2, 3\}$ given $\sigma = 1$ as m can be taken as integers between $[2, \sigma + 2]$ according to Theorem 2.1. Specifically, we have

$$\begin{aligned} \mathcal{S}_{2,1} &= \{(\sigma_1 = 1, \sigma_2 = 0), (\sigma_1 = 0, \sigma_2 = 1)\}, \\ \mathcal{S}_{3,1} &= \{(\sigma_1 = 0, \sigma_2 = 0, \sigma_3 = 0)\}. \end{aligned} \tag{20}$$

We defer the rest of the detailed calculation to Appendix A and directly present the results for large k below:

$$\gamma_k^{(1)} = \frac{(1+\mu)^2}{4(1-\mu)^5}\left[\omega_1 + \frac{1+10\mu+\mu^2}{3(1+\mu)^2}\omega_2\right]. \tag{21}$$

where we let

$$\omega_1 = \left(\nabla L \cdot \nabla^2 L\right) \cdot \nabla^2 L, \quad \omega_2 = \nabla L \cdot \nabla\left(\nabla L \cdot \nabla^2 L\right).$$

As a result, the HBF with $\varepsilon_k = \mathcal{O}\left(\eta^3\right)$ is

$$\dot{\beta} = -\frac{\nabla L}{1-\mu} - \frac{\eta}{2}\frac{1+\mu}{(1-\mu)^3}\nabla L \cdot \nabla^2 L - \frac{\eta^2}{4}\frac{(1+\mu)^2}{(1-\mu)^5}\left[\omega_1 + \frac{1+10\mu+\mu^2}{3(1+\mu)^2}\omega_2\right]. \tag{22}$$

Implicit regularization of HBF According to Eq. (19), HBF with $\varepsilon_k = \mathcal{O}\left(\eta^2\right)$ in Section 2.2.1 indicates that momentum induces a stronger implicit gradient regularization (IGR, Barrett and Dherin (2022)), i.e., $\gamma_{\mathrm{HB}} = (1+\mu)/(1-\mu)^3\gamma_{\mathrm{GD}}$ where γ_{HB} is the implicit regularization of HB while γ_{GD} is that of GD. For HBF with $\varepsilon_k = \mathcal{O}\left(\eta^3\right)$, we can conclude that the difference between HB and GD is more complicated since HBF now relies more on ω_2 that primarily depends on $\nabla^3 L$. Interestingly, the formulation of HBF with $\alpha = 3$ suggests that HB will implicitly impose a regularization effect of *directional smoothness*, which is not the case for GD. In particular, for $\mu \approx 1$, the third term of Eq. (22) is close to

$$\omega_1 + \omega_2 = \nabla\left(\nabla L \cdot (\nabla^2 L \nabla L)\right), \tag{23}$$

which is an approximation for the directional smoothness (Ahn et al., 2022)

$$\mathscr{D} = \frac{\nabla L(\beta) \cdot (\nabla L(\beta) - \nabla L(\beta - \eta \nabla L(\beta)))}{\eta \|\nabla L(\beta)\|^2}$$

by expanding $\nabla L(\beta - \eta \nabla L(\beta))$ around β. The directional smoothness measures the extent of oscillating behavior of optimization algorithms, i.e., the discrepancy between two adjacent iterates. For GD, Ahn et al. (2022) revealed that it exhibits oscillatory behavior such that its directional smoothness, $\mathscr{D}^{\mathrm{GD}}$, would saturate around $2/\eta$. As a comparison, due to the implicit regularization for the directional smoothness of our HBF, HB prefers learning dynamics with smaller directional smoothness $\mathscr{D}^{\mathrm{HB}} < \mathscr{D}^{\mathrm{GD}}$, implying an oscillatory behavior to less extent compared to GD. This is verified in Fig. 1, where $\mathscr{D}^{\mathrm{GD}} \approx 2/\eta = 20$ while $\mathscr{D}^{\mathrm{HB}} < \mathscr{D}^{\mathrm{GD}}$ and keeps decreasing. The numerical experimental details can be found in Appendix C.

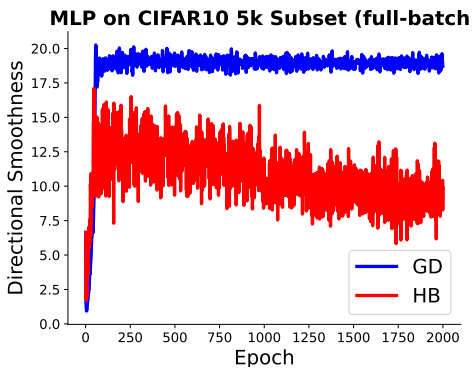


Figure 1: Comparison of directional smoothness for HB and GD for MLP on CIFAR-10 with full-batch GD and HB ($\mu = 0.9$) with $\eta = 0.1$.

3 Implicit Bias of Momentum Methods through HBF

The HBF proposed in Theorem 2.1 provides a reliable mathematical tool for analyzing a wide variety of properties of HB. One crucial aspect is its implicit bias in deep learning. To demonstrate the significance of HBF, we characterize the implicit bias of HBF specifically for the two layer diagonal linear network (Woodworth et al., 2020) as a case study.

The formulation of 2-layer diagonal linear networks A 2-layer diagonal linear network (DLN) with parameter $\mathbf{w} = (\mathbf{w}_+, \mathbf{w}_-)$ where $\mathbf{w}_\pm \in \mathbb{R}^d$ is equivalent to a linear predictor

$$f(x; \mathbf{w}) = x^T \mathbf{w} := x^T(\mathbf{w}_+ \odot \mathbf{w}_+ - \mathbf{w}_- \odot \mathbf{w}_-), \tag{24}$$

where we use the parameterization $\mathbf{w} = \mathbf{w}_+ \odot \mathbf{w}_+ - \mathbf{w}_- \odot \mathbf{w}_-$ (Woodworth et al., 2020). This model is a popular proxy model for deep neural networks as it shares many interesting phenomena with more complex architectures, e.g., the transition from kernel to rich regime. In this section, we focus on this model with $\mathbf{w}_{+;j}(0) = \mathbf{w}_{-;j}(0)$.

Leaning task For our task, given a dataset $\{(x_i, y_i)\}_{i=1}^n$ with n samples where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, we assume that $n < d$ and consider the regression problem with quadratic loss $L(\mathbf{w}_+, \mathbf{w}_-) = \sum_{i=1}^n (x_i^T \mathbf{w} - y_i)^2/(2n)$. We use $X \in \mathbb{R}^{n \times d}$ to denote the data matrix and $y = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$.

Implicit bias of GF for diagonal linear networks For GF, Azulay et al. (2021); Woodworth et al. (2020) showed that the limit point of $\mathbf{w}$ is equivalent to the solution of the constrained optimization problem $\mathbf{w}(\infty) = \arg\min_{\mathbf{w}} \Lambda^{\mathrm{GF}}(\mathbf{w}; \kappa(0)),\ s.t.\ X\mathbf{w} = y$ where $\kappa_j(0) = \mathbf{w}_{+;j}(0)\mathbf{w}_{-;j}(0)$, the potential function $\Lambda^{\mathrm{GF}}(\mathbf{w}; \kappa(0)) = \sum_{j=1}^d \Lambda_j^{\mathrm{GF}}(\mathbf{w}; \kappa(0))$, and

$$\Lambda_j^{\mathrm{GF}}(\mathbf{w}; \kappa(0)) = \frac{1}{4}\left[\mathbf{w}_j \operatorname{arcsinh}\left(\frac{\mathbf{w}_j}{2\kappa_j(0)}\right) - \sqrt{4(\kappa_j(0))^2 + \mathbf{w}_j^2} + 2\kappa_j(0)\right]. \tag{25}$$

Note that $\kappa(0)$ controls the transition from rich regime to kernel regime, i.e., $\Lambda^{\mathrm{GF}}(\mathbf{w}; \kappa(0)) \to \|\mathbf{w}\|_1$ for small $\kappa(0)$ while $\Lambda^{\mathrm{GF}}(\mathbf{w}; \kappa(0)) \to \|\mathbf{w}\|_2$ large $\kappa(0)$ (Woodworth et al., 2020).

3.1 Implicit Bias of HBF for Diagonal Linear Networks

According to Theorem 2.1, the learning dynamics of the diagonal linear networks $f(x; \mathbf{w})$ can be written as

$$\dot{\mathbf{w}}_+ = -\frac{\nabla_{\mathbf{w}_+} L}{1-\mu} - \eta\gamma^{\mathbf{w}_+},\ \dot{\mathbf{w}}_- = -\frac{\nabla_{\mathbf{w}_-} L}{1-\mu} - \eta\gamma^{\mathbf{w}_-} \tag{26}$$

where we use $\gamma^{\mathbf{w}_+} \in \mathbb{R}^d$ and $\gamma^{\mathbf{w}_-} \in \mathbb{R}^d$ for HBF of $\mathbf{w}_+$ and HBF of $\mathbf{w}_-$, respectively, and we use $\gamma_{;j}^{\mathbf{w}_\pm}$ to denote its j-th component. Compared to RGF (Eq. (1)), Eq. (26) has one extra term that accounts for the high-order discretization error. The implicit bias of $\mathbf{w}$ under the RGF is similar to that of GF, which, however, is not the case for Eq. (26).

Theorem 3.1 (Implicit bias of HBF for diagonal linear networks)**.** *If the dynamics of diagonal linear network $f(x; \mathbf{w}) = x^T\mathbf{w}$ where $\mathbf{w} = \mathbf{w}_+ \odot \mathbf{w}_+ - \mathbf{w}_- \odot \mathbf{w}_-$ follows HBF defined in Theorem 2.1 and if $\mathbf{w}(\infty)$ converges to an interpolation solution, let $\kappa_j(t) = \mathbf{w}_{+;j}(0)\mathbf{w}_{-;j}(0)\exp(-\eta\epsilon_j(t))$ where $\epsilon_j(t) = \int_0^t ds\,\big(\gamma_{;j}^{\mathbf{w}_+}(s)/\mathbf{w}_{+;j}(s) + \gamma_{;j}^{\mathbf{w}_-}(s)/\mathbf{w}_{-;j}(s)\big)$, then $\mathbf{w}(\infty)$ satisfies that*

$$\mathbf{w}(\infty) = \arg\min_{\mathbf{w}} \Lambda(\mathbf{w}; \kappa) \quad s.t.\ X\mathbf{w} = y, \tag{27}$$

where $\Lambda(\mathbf{w}; \kappa) = \sum_{j=1}^d \Lambda_j(\mathbf{w}, t = \infty; \kappa(\infty))$ with

$$\Lambda_j(\mathbf{w}, t; \kappa(t)) = \Lambda_j^{\mathrm{GF}}(\mathbf{w}; \kappa(t)) + \mathbf{w}_j\varphi_j(t),\ \varphi_j(t) = \frac{\eta}{4}\int_0^t ds\left(\frac{\gamma_{;j}^{\mathbf{w}_+}(s)}{\mathbf{w}_{+;j}(s)} - \frac{\gamma_{;j}^{\mathbf{w}_-}(s)}{\mathbf{w}_{-;j}(s)}\right). \tag{28}$$

Comparison with the implicit bias of GF Compared to the implicit bias of GF in Eq. (25), there are two differences brought by the high-order correction terms of HBF: **(1)**. the potential function $\Lambda_j^{\mathrm{GF}}(\mathbf{w}; \kappa(0))$ for GF becomes $\Lambda_j^{\mathrm{GF}}(\mathbf{w}; \kappa(\infty))$ for HBF where $\kappa(\infty)$ is different from $\kappa(0)$, meaning that HBF induces an effect equivalent to a rescaling of the initialization; **(2)**. $\Lambda_j(\mathbf{w}, \infty; \kappa)$ additionally depends on the product $\mathbf{w}_j\varphi_j(\infty)$. A similar term will appear in the potential function of GF Λ^{GF} if the initialization no longer satisfies $\mathbf{w}_+(0) = \mathbf{w}_-(0)$. In this sense, HBF also brings an effect that is equivalent to breaking the symmetry of the initialization. Theorem 3.1 also applies to the case for higher-order continuous approximation of GD by setting $\mu = 0$, suggesting an effect equivalent to the rescaling of the initialization that has been verified in GD (Even et al., 2023). This further reveals the reliability and usefulness of high-order continuous approximations.

The comprehensive characterization for the HBF requires a detailed investigation for the formulations of $\gamma^{\pm}$ specifically for the diagonal linear networks, which will be an open problem, while below we focus on $\varepsilon_k = \mathcal{O}\left(\eta^2\right)$ as an example.

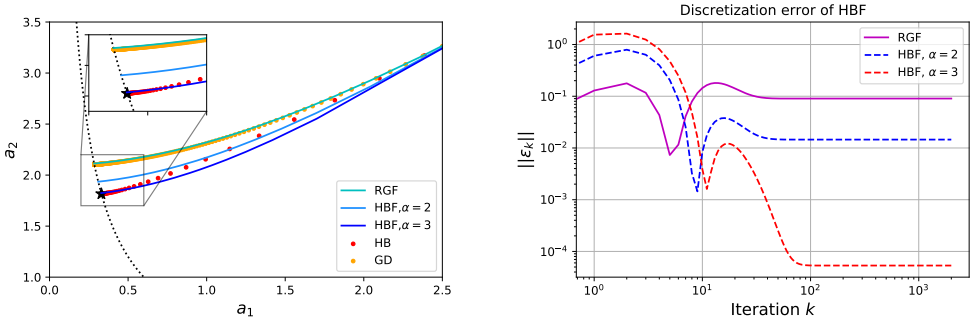
Corollary 3.2 (Implicit bias of HBF for diagonal linear networks with $\varepsilon_k = \mathcal{O}\left(\eta^2\right)$)**.** *Under conditions of Theorem 3.1, if we use HBF with $\varepsilon_k = \mathcal{O}\left(\eta^2\right)$, then*

$$\kappa_j(t) = \kappa_j(0) \exp\left[\frac{\eta(1+\mu)}{(1-\mu)^2}\left(-\frac{\Phi_j(t)}{1-\mu} + \frac{\left(X^T X \mathbf{q}\right)_j}{n}\right)\right] \tag{29}$$

where $\Phi_j(t) = 4\int_0^t ds(\partial_{\mathbf{w}_j} L)^2 > 0$, $\mathbf{q} \in \mathbb{R}^d$ *with* $\mathbf{q}_i = \sqrt{\mathbf{w}_i^2(\infty) + 4\kappa_i^2(0)} - 2\kappa_i(0)$.

For the exponent of $\kappa_j(\infty)$, when $\Phi_j(\infty) > 0$ dominates, e.g., $\kappa_i(0) \gg \mathbf{w}_i(\infty)$, we will conclude that $\kappa_j(\infty) < \kappa_j(0)$, hence the rescaling effect brought by HBF equivalently reduces the initialization. Therefore, compared to Λ^{GF}, $\Lambda(\mathbf{w}; \kappa)$ will closer to the ℓ_1-norm and the solution $\mathbf{w}(\infty)$ will enjoy better sparsity. This finding is consistent with parts of results in Papazov et al. (2024), which analyzed the implicit bias of HB also with a continuous time differential equation that is a second-order ODE, while our HBF is a first-order ODE and can also cover the case for GD simply by letting $\mu = 0$.

4 Numerical Experiments



(a) (b)

Figure 2: (a). Trajectories for learning dynamics of GD, HB, RGF, and HBF with discretization error $\mathcal{O}\left(\eta^2\right)$ and $\mathcal{O}\left(\eta^3\right)$ in a 2-d model. All dynamics start from the same point $(a_1 = 2.8, a_2 = 3.5)$. The convergence point of HB is denoted as a black star. The black dotted line denotes the set of all global minima. (b). Discretization errors for different continuous approximations of HB during training in (a).

In this section, we show numerical experiments on a simple 2-d model to verify our theoretical claims, and we present numerical experiments details and more experiments for diagonal linear networks in Appendix C.

Our simple 2-d model has the formulation $f(x; a_1, a_2) = a_1 a_2 x$, where $a_1, a_2 \in \mathbb{R}$ are the model parameters and $x, y \in \mathbb{R}$ is the training data. The loss function is $L = (f(x; a_1, a_2) - y)^2/2$. All parameters a_1, a_2 satisfying $a_1 a_2 x = y$ are global minima. To show that higher-order HBFs with discretization error $\mathcal{O}\left(\eta^\alpha\right)$ are better approximations for HB, we visualize trajectories for different learning dynamics, i.e., GD, HB, RGF, HBF with $\alpha = 2$, and HBF with $\alpha = 3$, in Fig. 2(a). The trajectory of HBF with $\alpha = 3$ is closer to that of HB than both RGF and HBF with $\alpha = 2$. Furthermore, Fig. 2(a) also reveals that RGF is more similar to GD and it cannot capture the discrete learning dynamics of HB well. We also plot the norm of discretization errors $\|\varepsilon_k\|^2$ for these continuous approximations during training in Fig. 2(b), where HBF with $\alpha = 3$ has the lowest discretization error after several steps. These results validate the reliability of HBF as a proxy of HB.

5 Conclusion

In this paper, we have established a new continuous time model for the discrete HB method (Eq. (2)), namely HBF, with an explicit discretization error that can be controlled to arbitrary order of the step size η. In particular, our approach constructs a relation, which is a functional integral equation, between discretization errors of adjacent iterates for any step and can be solved to arbitrary order of the step size. This is a different approach compared to prior works (Kovachki and Stuart, 2020; Ghosh et al., 2023). Our results provide a reliable foundation for analyzing the momentum methods in the continuous time limit. We leverage our HBF to shed lights on a newly observed implicit regularization effect of the HB method: the preference for small directional smoothness compared to GD. In addition, as another interesting application of our HBF in deep learning, we study the implicit bias of HBF for the popular proxy model diagonal linear networks, and we reveal the difference between the implicit bias of HB and that of GD which cannot be captured by RGF.

Limitation and future directions The framework in this paper does not consider optimization methods with adaptive learning rate, e.g., Adam (Kingma and Ba, 2017). A generalization of our framework to such case would be an interesting future direction. In addition, our analysis can be generalized to the stochastic case by replacing ∇L with $\tilde{\nabla} L$, the approximate stochastic gradient, by following Li et al. (2018); Latz (2021). Finally, we only consider the simple diagonal linear networks in this paper, and future works can explore more complex deep learning models with our HBF to study the implicit bias of HB.

Acknowledgments and Disclosure of Funding

B.L. is funded by a studentship provided by the School of Electronics and Computer Science, University of Southampton. The authors acknowledge DataCanvas AlayaNeW for providing computational resources. The authors thank the insightful and constructive feedback from the anonymous reviewers.

References

Kwangjun Ahn, Jingzhao Zhang, and Suvrit Sra. Understanding the unstable convergence of gradient descent, 2022. URL https://arxiv.org/abs/2204.01050.

Shahar Azulay, Edward Moroshko, Mor Shpigel Nacson, Blake E Woodworth, Nathan Srebro, Amir Globerson, and Daniel Soudry. On the implicit bias of initialization shape: Beyond infinitesimal mirror descent. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 468–477. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/azulay21a.html.

David G. T. Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Machine Learning*, 2022.

Matias D. Cattaneo, Jason M. Klusowski, and Boris Shigida. On the implicit bias of adam, 2023.

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

Mathieu Even, Scott Pesme, Suriya Gunasekar, and Nicolas Flammarion. (s)gd over diagonal linear networks: Implicit regularisation, large stepsizes and edge of stability, 2023.

Avrajit Ghosh, He Lyu, Xitong Zhang, and Rongrong Wang. Implicit regularization in heavy-ball momentum accelerated stochastic gradient descent, 2023.

Ernst Hairer, Gerhard Wanner, and Christian Lubich. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer Berlin, Heidelberg, 2006.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *arXiv:1412.6980*, 2017.

Nikola B. Kovachki and Andrew M. Stuart. Continuous time analysis of momentum methods. In *Journal of Machine Learning Research*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

Jonas Latz. Analysis of stochastic gradient descent in continuous time. *Statistics and Computing*, 31(4), May 2021. ISSN 1573-1375. doi: 10.1007/s11222-021-10016-8. URL http://dx.doi.org/10.1007/s11222-021-10016-8.

Guillaume Leclerc and Aleksander Madry. The two regimes of deep network training, 2020. URL https://arxiv.org/abs/2002.10376.

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations, 2018. URL https://arxiv.org/abs/1811.01558.

Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after SGD reaches zero loss? –a mathematical framework. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=siCt4xZn5Ve.

Bochen Lyu and Zhanxing Zhu. Implicit bias of (stochastic) gradient descent for rank-1 linear neural network. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

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

Taiki Miyagawa. Toward equation of motion for deep neural networks: Continuous-time gradient descent and discretization error analysis. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2023.

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

Hristo Papazov, Scott Pesme, and Nicolas Flammarion. Leveraging continuous time to understand momentum when training diagonal linear networks. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. URL http://arxiv.org/abs/1912.01703.

Scott Pesme, Loucas Pillaud-Vivien, and Nicolas Flammarion. Implicit bias of SGD for diagonal linear networks: a provable benefit of stochasticity. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=vvi7KqHQiA.

Loucas Pillaud-Vivien, Julien Reygner, and Nicolas Flammarion. Label noise (stochastic) gradient descent implicitly solves the lasso for quadratic parametrisation. In *Conference on Learning Theory*, 2020.

Boris Polyak. Some methods of speeding up the convergence of iteration methods. In *Ussr Computational Mathematics and Mathematical Physics*, 1964.

Mihaela Rosca, Yan Wu, Chongli Qin, and Benoit Dherin. On a continuous time model of gradient descent dynamics and instability in deep learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=EYrRzKPinA.

Bin Shi, Simon S. Du, Michael I. Jordan, and Weijie J. Su. Understanding the acceleration phenomenon via high-resolution differential equations. *Mathematical Programming*, 195, 2022.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on spearable data. In *Journal of Machine Learning Research*, 2018.

Weijie Su, Stephen Boyd, and Emmanuel J. Candès. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17 (153):1–43, 2016. URL http://jmlr.org/papers/v17/15-084.html.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/sutskever13.html.

Bohan Wang, Qi Meng, Wei Chen, and Tie-Yan Liu. The implicit bias for adaptive optimization algorithms on homogeneous neural networks, 2021. URL https://arxiv.org/abs/2012.06244.

Bohan Wang, Qi Meng, Huishuai Zhang, Ruoyu Sun, Wei Chen, Zhi-Ming Ma, and Tie-Yan Liu. Does momentum change the implicit regularization on separable data? In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=i-8uqlurj1f.

Ashia C. Wilson, Benjamin Recht, and Michael I. Jordan. A lyapunov analysis of momentum methods in optimization. In *arXiv:1611.02635*, 2016.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, 2020.

Chulhee Yun, Shankar Krishnan, and Hossein Mobahi. A unifying view on implicit bias in training linear neural networks. In *International Conference on Learning Representations*, 2021.

Chenyang Zhang, Difan Zou, and Yuan Cao. The implicit bias of adam on separable data, 2024. URL https://arxiv.org/abs/2406.10650.

Appendix

- In Appendix A, we provide proofs for Section 2.
- In Appendix B, we present proofs of Section 3.
- In Appendix C, we show details of numerical experiments in Section 2.2.2 and 4, and present more related numerical experiments to support our theoretical claims.

A Proofs for Section 2

We prove Theorem 2.1 in Appendix A.2 and give the details for the first several orders HBF in A.3. We first discuss the conditions that guarantee HBF as an effective approximation of HB.

A.1 Conditions of the Effectiveness of HBF

To make the HBF a valid continuous approximation of HB, there are two necessary conditions:

1. It is crucial to control the ratio between η and $1-\mu$ to avoid $\eta \gg 1-\mu$, which might lead the Taylor series to diverge. More interestingly, we conjecture that it is the magnitude of a special composite quantity
$$\psi := \frac{\eta}{(1-\mu)^2} \tag{30}$$
that matters for the effectiveness of the HBF. This quantity spontaneously appears in both HBF with $\alpha = 2$ and $\alpha = 3$ but not in the RGF, i.e., for HBF with $\alpha = 2$ the counter term is proportional to ψ while for HBF with $\alpha = 3$ a new counter term proportional to ψ^2 will appear. If ψ is too large, then our results would no longer hold.

   Hence, we need to fix the value of μ and treat only η as the variable to denote the higher-order terms as $\mathcal{O}(\eta^\alpha)$ while hide μ in the expansion. And it would be interesting for future works to study the case when both μ and η are treated as variables such that higher-order terms are denoted as $\mathcal{O}(\psi^\alpha)$ for $\alpha \geq 1$. In addition, in the regime of large μ and large η, the model might not be trained properly either: the update direction coming from the gradient and that from the momentum will jointly affect the training direction significantly, while these two directions can be very different due to the large value of μ and η hence cannot give a consistent updating direction.

   In addition, the dependence of HBF on the special composite quantity ψ is consistent with the empirical observation in Leclerc and Madry (2020), where the optimization curves for different momentum values can be recovered by a corresponding change in the learning rate. The dependence of HB on η and μ at the same time further indicates the advantage of HBF with $\alpha > 1$ and that RGF, which only depends on μ, is not sufficient to reflect the optimization properties of HB.

2. Given α, the continuous approximations include derivatives of L up to the α-th order, hence L should at least be α-times continuously differentiable and $||\nabla^\alpha L||$ should be upper bounded.

A.2 Proof of Theorem 2.1

Given the HBF for $k \in [N]$

$$t \in [t_k, t_{k+1}): \quad \dot{\beta} = -G_k(\beta) - \eta\gamma_k(\beta) \tag{31}$$

with unknown G_k and γ_k, we expect that the counter term γ_k could cancel higher-order discretization errors and G_k should degenerate to rescaled gradient, i.e., $\nabla L/(1-\mu)$. Hence, γ_k and G_k should be designed in such a way that $\beta(t_k)$ is close to β_k in the sense that the discretization error

$$\varepsilon_k = \beta(t_k) - \beta_k \tag{32}$$

is small. Inspired by Miyagawa (2023), we first present the outline of our three main steps for deriving their formulations below:

Step I: Unknown $G_k,\ \gamma_k \xrightarrow{\text{determine}}$ Taylor expansion residual $I_k^{\pm}$

Step II:
- Taylor Expansion of $\beta(t_{k\pm1})$ (35), (36)
  - $\downarrow$ constructs
- Expression of $\varepsilon_{k+1}-\varepsilon_k$ (41)
  - $\downarrow$ required to be $\mathcal{O}(\eta^{\alpha+1})$ for $\varepsilon_k=\mathcal{O}(\eta^{\alpha})$ (Lemma A.2)
- Equalities for $G_k,\ \gamma_k$ (44)
  - $\downarrow$ solved by matching to each order of η

Step III: Solution of G_k and γ_k

We now discuss the detailed proof following this outline.

Proof. We start with Step I, which deals with how $I_k^{\pm}$ is determined by G_k and γ_k.

Step I Recall that the discrete learning dynamics of HB is

$$\beta_{k+1}-\beta_k=-\eta\nabla L(\beta_k)+\mu(\beta_k-\beta_{k-1}), \tag{33}$$

where μ is the momentum factor and k is the iteration number. Based on our discussion in Section 2, the continuous differential equation for HB is

$$\dot{\beta}=-G_k(\beta)-\eta\gamma_k(\beta) \tag{34}$$

for $t\in[t_k,t_{k+1})$ where $t_k=k\eta$ and the solution is $\beta(t)$. For arbitrary unknown γ_k in Eq. (34), $I_k^{\pm}$ is determined, which is also unknown but depends on γ_k. Specifically, the first-order Taylor expansion with the remainder term in the integral form gives us

$$\beta(t_{k+1})-\beta(t_k)=\eta\dot{\beta}(t_k^+)+\eta^2 I_k^+=-\eta G_k-\eta^2\gamma_k+\eta^2 I_k^+ \tag{35}$$

where t_k^+ means we approximate t_k from $t>t_k$,

$$I_k^+=\int_0^1 \ddot{\beta}\left(\eta(k+s)\right)(1-s)ds,$$

and we use Eq. (34) in the second equality. Similarly, when approximating t_k from $t<t_k$, we obtain that

$$\beta(t_k)-\beta(t_{k-1})=-\eta G_{k-1}-\eta^2\gamma_{k-1}-\eta^2 I_k^-. \tag{36}$$

Now we construct the dependence of $I_k^{\pm}$ on γ_k in the series form, as shown in Lemma. A.1 (proof can be found in Appendix A.2.1.).

Lemma A.1. *Given the series form*

$$\gamma_k=\sum_{\sigma=0}^{\infty}\eta^{\sigma}\gamma_k^{(\sigma)} \tag{37}$$

and the continuous time differential equation Eq. (34)*, $I_k^{\pm}$ in Eq.* (35) *and Eq.* (36) *have the following series forms:*

$$I_k^+=\sum_{p=0}^{\infty}\eta^p(\mathscr{I}_k^+)^{(p)}:=\sum_{p=0}^{\infty}\sum_{q=2}^{p+2}\sum_{\sum_{j=1}^{q}\sigma_j=p-q+2}\frac{(-1)^q}{q!}\eta^p\mathbf{L}_{\beta}^{(k,\sigma_1)}\cdots\mathbf{L}_{\beta}^{(k,\sigma_{q-1})}\gamma_k^{(\sigma_q-1)}, \tag{38}$$

$$I_k^-=\sum_{p=0}^{\infty}\eta^p(\mathscr{I}_k^-)^{(p)}:=\sum_{p=0}^{\infty}\sum_{q=2}^{p+2}\sum_{\sum_{j=1}^{q}\sigma_j=p-q+2}\frac{1}{q!}\eta^p\mathbf{L}_{\beta}^{(k-1,\sigma_1)}\cdots\mathbf{L}_{\beta}^{(k-1,\sigma_{q-1})}\gamma_{k-1}^{(\sigma_q-1)}. \tag{39}$$

Step II Given the dependence of $I_k^{\pm}$ on γ_k, we are now able to write the Taylor expansion of $\beta(t_{k\pm1})$ explicitly. This allows us to construct a relation between ε_{k+1} and ε_k by subtracting Eq. (33) from both sides of Eq. (35):

$$\varepsilon_{k+1} - \varepsilon_k = -\eta \left[G_k(\beta(t_k)) - \nabla L(\beta_k) \right] - \eta^2 \gamma_k + \eta^2 I_k^+ - \mu(\beta_k - \beta_{k-1}). \tag{40}$$

Note that

$$\begin{aligned} \beta_k - \beta_{k-1} &= \beta(t_k) - \beta(t_{k-1}) - (\varepsilon_k - \varepsilon_{k-1}) \\ &= -\eta G_{k-1} - \eta^2 \gamma_{k-1} - \eta^2 I_k^- - (\varepsilon_k - \varepsilon_{k-1}), \end{aligned}$$

we obtain the expression of $\varepsilon_{k+1} - \varepsilon_k$ as in the Step II of our outline:

$$\begin{aligned} \varepsilon_{k+1} - \varepsilon_k = \mu(\varepsilon_k - \varepsilon_{k-1}) &- \eta \left[G_k - \mu G_{k-1} - \nabla L(\beta(t_k) - \varepsilon_k) \right] \\ &+ \eta^2 \left[I_k^+ + \mu I_k^- - \gamma_k + \mu \gamma_{k-1} \right]. \end{aligned} \tag{41}$$

Eq. (41) builds the connection between the discretization error ε_k and the counter term γ_k in the continuous time differential equation Eq. (5). We can now construct the equalities for G_k and γ_k under the constraint of low discretization error by following the lemma below (proof can be found in Appendix A.2.2).

Lemma A.2. *For the continuous differential equation Eq.* (34) *and the discrete sequence given by Eq.* (33)*, if*

$$G_k(\beta(t_k)) = \mu G_{k-1}(\beta(t_k)) + \nabla L(\beta(t_k)) \tag{42}$$

with $G_{-1} = 0$ *and*

$$I_k^+ + \mu I_k^- - \gamma_k + \mu \gamma_{k-1} = \mathcal{O}\left(\eta^{\alpha-1}\right), \tag{43}$$

as in Eq. (49)*, then we have*

$$\varepsilon_k - \varepsilon_{k-1} = \mathcal{O}\left(\eta^{\alpha+1}\right)$$

and, as a result,

$$\varepsilon_k = \mathcal{O}\left(\eta^{\alpha}\right).$$

As shown in Lemma. A.2, to ensure that the leading order of the L.H.S of Eq. (41) is to the order of $\alpha > 1$, we only need to require $\varepsilon_{k+1} - \varepsilon_k = \mathcal{O}\left(\eta^{\alpha}\right)$ which can be guaranteed by Eq. (42) and the functional integral equation Eq. (43). As Eq. (42) can be easily solved by induction, we only need to solve Eq. (43). To achieve this, we build a stronger functional equation below:

$$I_k^+ + \mu I_k^- = \gamma_k - \mu \gamma_{k-1}, \tag{44}$$

which is the final equation that we aim to solve to derive γ_k.

Step III The Step III of our outline is solving the functional integral equation Eq. (44). Our core idea is simple: as the series form of $I_k^{\pm}$ has already derived in Lemma. A.1, we make Eq. (44) satisfied by matching both sides of it for each order of η. In particular, given the series forms of $I_k^{\pm}$ in Lemma A.1 and that of γ_k (Eq. (37)), we require

$$\forall p \in \mathbb{N}: \ \eta^p(\gamma_k^{(p)} - \mu \gamma_{k-1}^{(p)}) = \eta^p \left((\mathscr{I}_k^+)^{(p)} + \mu (\mathscr{I}_k^-)^{(p)} \right), \tag{45}$$

which, let

$$\chi_k^{(\sigma)} = (\mathscr{I}_k^+)^{(\sigma)} + \mu (\mathscr{I}_k^-)^{(\sigma)}, \tag{46}$$

gives us the recursive relation of $\gamma_k^{(\sigma)}$ for $\sigma \in \mathbb{N}$

$$\gamma_k^{(\sigma)} = \mu \gamma_{k-1}^{(\sigma)} + \chi_k^{(\sigma)} \tag{47}$$

because $\chi_k^{(\sigma)}$ only depends on $\gamma_k^{(\sigma')}$ with $\sigma' < \sigma$ according to Lemma A.1. Now given $\alpha \in \mathbb{Z}^+$, we can truncate γ_k to the order $\alpha - 2$ such that

$$\gamma_k = \sum_{\sigma=0}^{\alpha-2} \eta^{\sigma} \gamma_k^{(\sigma)}, \tag{48}$$

then the functional integral equation is solved to the $(\alpha-2)$-th order of the step size η:

$$\begin{aligned} I_k^+ + \mu I_k^- - \gamma_k + \mu\gamma_{k-1} &= \sum_{\sigma=0}^{\infty} \eta^\sigma \chi_k^{(\sigma)} - \sum_{\sigma=0}^{\alpha-2} \eta^\sigma \left(\gamma_k^{(p)} - \mu\gamma_{k-1}^{(p)}\right) \\ &= \sum_{\sigma=\alpha-1}^{\infty} \eta^\sigma \chi_k^{(\sigma)} = \mathcal{O}\left(\eta^{\alpha-1}\right), \end{aligned} \tag{49}$$

which is exactly the condition in Lemma. A.2. Therefore, by constructing γ_k following Eq. (47) and truncating γ_k to preserver its first $\alpha-2$ terms, we can prove that the discretization error of the continuous time differential equation Eq. (34) is to the order $\mathcal{O}\left(\eta^\alpha\right)$. □

A.2.1 Proof for Lemma A.1

Proof. We first rewrite $I_k^\pm$ as follows :

$$\begin{aligned} I_k^\pm &= \frac{1}{\eta^2}\int_{k\eta}^{k\eta\pm\eta} \ddot{\beta}(\tau)(k\eta \pm \eta - \tau)d\tau \\ &\overset{\tau'\leftarrow\tau-k\eta}{=} \frac{1}{\eta^2}\int_0^{\pm\eta} \left[\sum_{n=0}^{\infty} \frac{1}{n!}\frac{d^n}{dt^n}\ddot{\beta}(k\eta)\tau'^n\right]^\pm (\pm\eta - \tau')d\tau' \\ &= \sum_{n=0}^{\infty} \frac{(\pm\eta)^n}{(n+2)!}\frac{d^n}{dt^n}\ddot{\beta}(t_k^\pm) \end{aligned} \tag{50}$$

where we use $\int_0^\eta \tau'^n(\eta-\tau')d\tau' = \frac{\eta^{n+2}}{n+1} - \frac{\eta^{n+2}}{n+2} = \frac{\eta^{n+2}}{(n+1)(n+2)}$ in the last equality. To continue, we need the expression of $d^n\beta/dt^n$ and we start with $t \to t_k^+$:

$$\begin{aligned} \frac{d^n}{dt^n}\beta(t_k^+) &= \frac{d}{dt}\left(\frac{d^{n-1}}{dt^n}\beta(t_k^+)\right) \\ &= \dot{\beta}(t_k^+)\cdot\nabla\left(\frac{d^{n-1}}{dt^n}\beta(t_k^+)\right) \\ &= -(G_k + \eta\gamma_k)\cdot\nabla\left(\frac{d^{n-1}}{dt^n}\beta(t_k^+)\right) \\ &= (-1)^n(\mathbf{L}_\beta^{(k)})^{n-1}(G_k + \eta\gamma_k) \end{aligned} \tag{51}$$

where we denote the differential operator $\mathbf{L}_\beta^{(k)} = (G_k + \eta\gamma_k)\cdot\nabla$ and use Eq. (34) in the third equality. Now suppose that γ_k can be written as a series

$$\gamma_k = \sum_{\sigma=0}^{\infty} \eta^\sigma \gamma_k^{(\sigma)},\ \gamma_k^{(-1)} = G_k,$$

then Eq. (51) becomes

$$\begin{aligned} \frac{d^n}{dt^n}\beta(t) &= (-1)^n \left(\sum_{\sigma_1=0}^{\infty} \eta^{\sigma_1}\gamma_k^{(\sigma_1-1)}\cdot\nabla\right)\cdots \\ &\qquad \cdots\left(\sum_{\sigma_{n-1}=0}^{\infty} \eta^{\sigma_{n-1}}\gamma_k^{(\sigma_{n-1}-1)}\cdot\nabla\right)\left(\sum_{\sigma_{n-1}=0}^{\infty} \eta^{\sigma_n}\gamma_k^{(\sigma_n-1)}\right) \\ &= (-1)^n \sum_{\sigma_1,\ldots,\sigma_n=0}^{\infty} \eta^{\sum_{j=1}^n \sigma_j}\mathbf{L}_\beta^{(k,\sigma_1)}\cdots\mathbf{L}_\beta^{(k,\sigma_{n-1})}\gamma_k^{(\sigma_n-1)} \end{aligned} \tag{52}$$

Combined with Eq. (50), we obtain the form of I_k^+ for $t \in (t_k, t_{k+1})$ as

$$\begin{aligned} I_k^+ &= \sum_{n=0}^{\infty} \frac{\eta^n}{(n+2)!} \frac{d^{n+2}}{dt^{n+2}} \beta(t) \\ &= \sum_{n=0}^{\infty} \sum_{\sigma_1,\ldots,\sigma_{n+2}=0}^{\infty} \frac{(-1)^{n+2}}{(n+2)!} \eta^{n+\sum_{j=1}^{n+2}\sigma_j} \mathbf{L}_\beta^{(k,\sigma_1)} \cdots \mathbf{L}_\beta^{(k,\sigma_{n+1})} \gamma_k^{(\sigma_{n+2}-1)} \\ &= \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \sum_{\sum_{j=1}^{n+2}\sigma_j = m} \frac{(-1)^{n+2}}{(n+2)!} \eta^{n+m} \mathbf{L}_\beta^{(k,\sigma_1)} \cdots \mathbf{L}_\beta^{(k,\sigma_{n+1})} \gamma_k^{(\sigma_{n+2}-1)} \\ &= \sum_{p=0}^{\infty} \sum_{q=2}^{p+2} \sum_{\sum_{j=1}^{q}\sigma_j = p-q+2} \frac{(-1)^q}{q!} \eta^p \mathbf{L}_\beta^{(k,\sigma_1)} \cdots \mathbf{L}_\beta^{(k,\sigma_{q-1})} \gamma_k^{(\sigma_q-1)} \end{aligned} \tag{53}$$

where we let $p \leftarrow n + \sum_{j=1}^{n+2} \sigma_j, q \leftarrow n+2$, in the last equality. Similarly, when $t \to t_k^-$, we have

$$\frac{d^n}{dt^n}\beta(t_k^-) = (-1)^n (\mathbf{L}_\beta^{(k-1)})^{n-1} \left(G_{k-1} + \eta\gamma_{k-1}\right)$$

which implies that

$$I_k^- = \sum_{p=0}^{\infty} \sum_{q=2}^{p+2} \sum_{\sum_{j=1}^{q}\sigma_j = p-q+2} \frac{1}{q!} \eta^p \mathbf{L}_\beta^{(k-1,\sigma_1)} \cdots \mathbf{L}_\beta^{(k-1,\sigma_{q-1})} \gamma_{k-1}^{(\sigma_q-1)}. \tag{54}$$

□

A.2.2 Proof for Lemma A.2

Proof. We first present several useful relations. As Eq. (49) is established by solving the functional equation for any iteration count k, we can write the relation between ε_{k+1} and ε_k Eq. (41) as

$$\varepsilon_{k+1} - \varepsilon_k = \mu(\varepsilon_k - \varepsilon_{k-1}) - \eta \left[\nabla L\left(\beta(t_k)\right) - \nabla L\left(\beta(t_k) - \varepsilon_k\right)\right] + \mathcal{O}\left(\eta^{\alpha+1}\right). \tag{55}$$

If $\varepsilon_k = \mathcal{O}(\eta^\alpha)$, then Eq. (55) implies

$$\begin{aligned} \|\varepsilon_{k+1} - \varepsilon_k\| &\le \mu\|\varepsilon_k - \varepsilon_{k-1}\| + \eta\|\nabla L\left(\beta(t_k)\right) - \nabla L\left(\beta(t_k) - \varepsilon_k\right)\| + c_1\eta^{\alpha+1} \\ &\le \mu\|\varepsilon_k - \varepsilon_{k-1}\| + \eta\lambda\|\varepsilon_k\| + c_1\eta^{\alpha+1} \end{aligned} \tag{56}$$

for some constant c_1 where we use $G_k - \mu G_{k-1} = \nabla L$ and let $\lambda = \max_\beta \|\nabla^2 L(\beta)\|$.

Denoting

$$\forall k : c_2(k) = \frac{c_1}{\lambda} e^{\frac{2\lambda\eta}{1-\mu}k},\ c_3(k) = \frac{2c_1}{1-\mu} e^{\frac{2\lambda\eta}{1-\mu}k},$$

we now prove by induction.

1. For the first step ($k = 0$), by definition we have

   $$\varepsilon_0 = 0 \le c_2(0)\eta^\alpha.$$

   Note that $G_{-1} = 0$ and $\gamma_{-1} = 0$ by definition, then we have

   $$\|\varepsilon_1 - \varepsilon_0\| \le c_1\eta^{\alpha+1} \le \frac{2c_1}{1-\mu} = c_3(0)\eta^{\alpha+1}$$

   since $\mu < 1$.

2. Suppose that for the k-th step the following relations hold:

   $$\begin{aligned} \|\varepsilon_k\| &\le c_2(k)\eta^\alpha, \\ \|\varepsilon_{k+1} - \varepsilon_k\| &\le c_3(k)\eta^{\alpha+1}. \end{aligned} \tag{57}$$

Then for the $(k+1)$-th step, we have

$$\begin{aligned}\|\varepsilon_{k+1}\| &= \|\varepsilon_{k+1} - \varepsilon_k + \varepsilon_k\| \\ &\leq \|\varepsilon_{k+1} - \varepsilon_k\| + \|\varepsilon_k\| \\ &\leq c_2(k)\left(1 + \eta\frac{c_3(k)}{c_2(k)}\right)\eta^\alpha \\ &\leq c_2(k)e^{\frac{2\lambda\eta}{1-\mu}}\eta^\alpha \\ &= c_2(k+1)\eta^\alpha\end{aligned}$$

where the last inequality is because $e^x > 1 + x$ for $x > 0$. Similarly,

$$\begin{aligned}\|\varepsilon_{k+2} - \varepsilon_{k+1}\| &\leq \mu\|\varepsilon_{k+1} - \varepsilon_k\| + \eta\lambda\|\varepsilon_{k+1}\| + c_1\eta^{\alpha+1} \\ &\leq \left[\mu c_3(k) + \lambda c_2(k+1) + c_1\right]\eta^{\alpha+1} \\ &= \left[\mu e^{-\frac{2\lambda\eta}{1-\mu}} + \frac{1-\mu}{2} + \frac{1-\mu}{2}e^{-\frac{2\lambda\eta}{1-\mu}(k+1)}\right]c_3(k+1)\eta^{\alpha+1} \\ &\leq \left[\mu + \frac{1-\mu}{2} + \frac{1-\mu}{2}\right]c_3(k+1)\eta^{\alpha+1} \\ &= c_3(k+1)\eta^{\alpha+1}.\end{aligned}$$

□

A.3 $\mathcal{O}(\eta^\alpha)$-close HBF for a specific α

In this section, we derive the form of $\mathcal{O}(\alpha)$-close HBF for given a specific α. There are basically three steps to find a HBF that is $\mathcal{O}(\eta^\alpha)$-close to HB:

1. truncate γ_k to the desired order α, i.e, $\gamma_k = \sum_{\sigma=0}^{\alpha-2}\gamma_k^{(\sigma)}$;
2. from the smallest σ, find all $\chi_j^{(\sigma)}$ with $j \leq k$ by finding the corresponding $\mathcal{S}_{m,\sigma}$ with $m = \{2, \ldots, \sigma+2\}$ for each σ;
3. derive the expression of $\gamma_k^{(\sigma)}$ for all $\sigma \leq \alpha - 2$ in a recursive manner using the relation $\gamma_k^{(\sigma)} = \sum_{j=0}^{k}\mu^{k-j}\chi_j^{(\sigma)}$.

In the following, we give the cases for $\alpha = 2$ and 3 as examples. With this approach, one can in fact find HBF with arbitrary order of closeness to HB.

A.3.1 $\alpha = 2$.

According to Theorem 2.1, the series of γ_k is truncated to the first term, i.e., $\gamma_k = \eta^0\gamma_k^{(0)}$, where $\gamma_k = \sum_{j=0}^{k}\mu^{k-j}\chi_j^{(0)}$. Thus the first step is to find $\chi_j^{(0)}$, which can be given by first identifying the set $\mathcal{S}$:

$$\mathcal{S}_{m=2,\sigma=0} = \{(\sigma_1 = 0, \sigma_2 = 0)\}, \tag{58}$$

therefore there is only one term in $\chi_j^{(0)}$:

$$\chi_j^{(0)} = \frac{1}{2}\left[\mathbf{L}_\beta^{j,0}\gamma_j^{(-1)} + \mu\mathbf{L}_\beta^{j-1,0}\gamma_{j-1}^{(-1)}\right].$$

Recall that

$$\gamma_j^{(-1)} = G_j = \frac{1-\mu^{j+1}}{1-\mu}\nabla L, \tag{59}$$

which, according to our definition in Theorem 2.1, leads to

$$\mathbf{L}_\beta^{j,0} = \gamma_j^{(-1)} \cdot \nabla = G_j \cdot \nabla,$$

we obtain that

$$\begin{aligned}\chi_j^{(0)} &= \frac{1}{2}\left[G_j \cdot \nabla G_j + \mu G_{j-1} \cdot \nabla G_{j-1}\right] \\ &= \frac{1}{2(1-\mu)^2}\left[(1-\mu^{j+1})^2 + \mu(1-\mu^j)^2\right]\nabla L \cdot \nabla^2 L. \end{aligned} \tag{60}$$

Thus

$$\begin{aligned}\gamma_k^{(0)} &= \frac{1}{2}\sum_{j=0}^{k}\mu^{k-j}\left[G_j \cdot \nabla G_j + \mu G_{j-1} \cdot \nabla G_{j-1}\right] \\ &= \frac{\nabla L \cdot \nabla^2 L}{2(1-\mu)^2}\sum_{j=0}^{k}\mu^{k-j}\left[(1-\mu^{j+1})^2 + \mu(1-\mu^j)^2\right] \\ &= \frac{\nabla L \cdot \nabla^2 L}{2(1-\mu)^2}\sum_{j=0}^{k}\left[(1+\mu)\mu^{k-j} + \mu^{k+1}(\mu^j(1+\mu) - 4)\right]. \end{aligned} \tag{61}$$

When k is larege, the above expression can be simplified as

$$\gamma_k^{(0)} \approx \frac{(1+\mu)\sum_{j=0}^{k}\mu^j}{2(1-\mu)^2}\nabla L \cdot \nabla^2 L \approx \frac{1+\mu}{2(1-\mu)^3}\nabla L \cdot \nabla^2 L.$$

A.3.2 $\alpha = 3$.

Similarly, in this case we first truncate the series of γ_k to the desired order, i.e., $\gamma_k = \gamma_k^{(0)} + \eta\gamma_k^{(1)}$ where we have already obtained $\gamma_k^{(0)}$ in the last section, thus we only need to find $\gamma_k^{(1)}$ and $\chi_k^{(1)}$, which can be done by first finding the set $\mathcal{S}_{m=2,\sigma=1}$ and $\mathcal{S}_{m=3,\sigma=1}$:

$$\begin{aligned}\mathcal{S}_{2,1} &= \{(\sigma_1 = 1, \sigma_2 = 0), (\sigma_1 = 0, \sigma_2 = 1)\}, \\ \mathcal{S}_{3,1} &= \{(\sigma_1 = 0, \sigma_2 = 0, \sigma_3 = 0)\}.\end{aligned}$$

Therefore there are three terms of $\chi_j^{(1)}$:

$$\begin{aligned}\chi_j^{(1)} = &\frac{1}{2}\left[\mathbf{L}_\beta^{j,1}\gamma_j^{(-1)} + \mu\mathbf{L}_\beta^{j-1,1}\gamma_{j-1}^{(-1)}\right] + \frac{1}{2}\left[\mathbf{L}_\beta^{j,0}\gamma_j^{(0)} + \mu\mathbf{L}_\beta^{j-1,0}\gamma_{j-1}^{(0)}\right] \\ &- \frac{1}{6}\left[\mathbf{L}_\beta^{j,0}\mathbf{L}_\beta^{j,0}\gamma_j^{(-1)} - \mu\mathbf{L}_\beta^{j-1,0}\mathbf{L}_\beta^{j-1,0}\gamma_{j-1}^{(-1)}\right]. \end{aligned} \tag{62}$$

Recall that $\gamma_j^{(-1)} = G_j$, $\mathbf{L}_\beta^{j,0} = G_j \cdot \nabla$, and $\mathbf{L}_\beta^{j,1} = \gamma_j^{(0)} \cdot \nabla$, the first line of Eq. (62) is

$$\frac{1}{2}\left[\gamma_j^{(0)} \cdot \nabla G_j + \mu\gamma_{j-1}^{(0)} \cdot \nabla G_{j-1} + G_j \cdot \nabla\gamma_j^{(0)} + \mu G_{j-1} \cdot \nabla\gamma_{j-1}^{(0)}\right] \tag{63}$$

while the second line is

$$-\frac{1}{6}\left[G_j \cdot \nabla\left(G_j \cdot \nabla G_j\right) - \mu G_{j-1} \cdot \nabla\left(G_{j-1} \cdot \nabla G_{j-1}\right)\right]. \tag{64}$$

To simplify these terms, we can either replace all $\gamma_j^{(0)}$ with the expression in Eq. (61) and write G_j explicitly, or notice the recursive relation between G_j and G_{j-1} in Theorem 2.1, i.e. ,$G_j = \mu G_{j-1} + \nabla L$, then Eq. (63) becomes

$$\frac{1}{2}\left[\gamma_j^{(0)} \cdot \nabla^2 L + \nabla L \cdot \nabla\gamma_j^{(0)}\right] + \frac{\mu}{2}\left[\left(\gamma_j^{(0)} + \gamma_{j-1}^{(0)}\right) \cdot \nabla G_{j-1} + G_{j-1} \cdot \nabla\left(\gamma_j^{(0)} + \gamma_{j-1}^{(0)}\right)\right]$$

and Eq. (64) is now

$$-\frac{1}{6}\nabla L \cdot \nabla\left(G_j \cdot \nabla G_j\right) - \frac{\mu}{6}G_{j-1} \cdot \nabla\left(G_j \cdot \nabla G_j - G_{j-1} \cdot \nabla G_{j-1}\right). \tag{65}$$

Summing over these terms gives us $\chi_j^{(1)}$:

$$\chi_j^{(1)} = \Psi_j^{(1)} + \mu\Theta_j^{(1)} \tag{66}$$

where

$$\Psi_j^{(1)} = \frac{1}{2}\left(\gamma_j^{(0)} \cdot \nabla^2 L + \nabla L \cdot \nabla \gamma_j^{(0)}\right) - \frac{1}{6}\nabla L \cdot \nabla\left(G_j \cdot \nabla G_j\right)$$
$$\Theta_j^{(1)} = \frac{1}{2}\left[\left(\gamma_j^{(0)} + \gamma_{j-1}^{(0)}\right) \cdot \nabla G_{j-1} + G_{j-1} \cdot \nabla\left(\gamma_j^{(0)} + \gamma_{j-1}^{(0)}\right)\right] - \frac{1}{6} G_{j-1} \cdot \nabla\left(G_j \cdot \nabla G_j - G_{j-1} \cdot \nabla G_{j-1}\right).$$

We can now find $\gamma_k^{(1)}$ through its definition:

$$\gamma_k^{(1)} = \sum_{j=0}^{k} \mu^{k-j} \chi_j^{(1)} = \sum_{j=0}^{k} \mu^{k-j} \Psi_j^{(1)} + \mu \sum_{j=0}^{k} \mu^{k-j} \Theta_j^{(1)}. \tag{67}$$

In the following, we derive the form of $\gamma_k^{(1)}$ When k is large. According to Eq. (61), we have

$$\begin{aligned}
\mu^{k-j}\gamma_j^{(0)} &= \mu^{k-j}\frac{\nabla L \cdot \nabla^2 L}{2(1-\mu)^2}\sum_{i=0}^{j}\left[(1+\mu)\mu^{j-i} + \mu^{j+1}(\mu^i(1+\mu) - 4)\right] \\
&= \mu^{k-j}\frac{\nabla L \cdot \nabla^2 L}{2(1-\mu)^2}\left[\frac{(1+\mu)(1-\mu^{j+1})}{1-\mu} + \frac{\mu^{j+1}(1+\mu)(1-\mu^{j+1})}{1-\mu} - 4(j+1)\mu^{j+1}\right] \\
&= \frac{\nabla L \cdot \nabla^2 L}{2(1-\mu)^2}\left[\frac{(1+\mu)(\mu^{k-j} - \mu^{k+1})}{1-\mu} + \frac{\mu^{k+1}(1+\mu)(1-\mu^{j+1})}{1-\mu} - 4(j+1)\mu^{k+1}\right] \\
&= \frac{\nabla L \cdot \nabla^2 L}{2(1-\mu)^2}\left[\frac{(1+\mu)\mu^{k-j}}{1-\mu} - \frac{\mu^{k+j+1}(1+\mu)}{1-\mu} - 4(j+1)\mu^{k+1}\right] \\
&\approx \mu^{k-j}\frac{(1+\mu)}{2(1-\mu)^3}\nabla L \cdot \nabla^2 L
\end{aligned} \tag{68}$$

and, according to Eq. (59),

$$\mu^{k-j} G_j \cdot \nabla G_j = \frac{\mu^{k-j}(1 - 2\mu^{j+1} + \mu^{2(j+1)})}{(1-\mu)^2}\nabla L \cdot \nabla^2 L \approx \frac{\mu^{k-j}}{(1-\mu)^2}\nabla L \cdot \nabla^2 L. \tag{69}$$

Combining Eq. (68) and (69) gives the form of $\mu^{k-j}\Psi_j^{(1)}$ when k is large:

$$\mu^{k-j}\Psi_j^{(1)} \approx \frac{\mu^{k-j}(1+\mu)}{4(1-\mu)^3}\left[(\nabla L \cdot \nabla^2 L) \cdot \nabla^2 L + \nabla L \cdot \nabla\left(\nabla L \cdot \nabla^2 L\right)\right] - \frac{\mu^{k-j}}{6(1-\mu)^2}\nabla L \cdot \nabla\left(\nabla L \cdot \nabla^2 L\right)$$

which immediately leads to

$$\begin{aligned}
&\sum_{j=0}^{k}\mu^{k-j}\Psi_j^{(1)} \\
\approx& \frac{(1+\mu)}{4(1-\mu)^4}\left[(\nabla L \cdot \nabla^2 L) \cdot \nabla^2 L + \nabla L \cdot \nabla\left(\nabla L \cdot \nabla^2 L\right)\right] - \frac{\nabla L \cdot \nabla\left(\nabla L \cdot \nabla^2 L\right)}{6(1-\mu)^3} \\
=& \frac{1}{4(1-\mu)^4}\left[(1+\mu)(\nabla L \cdot \nabla^2 L) \cdot \nabla^2 L + \frac{(1+5\mu)}{3}\nabla L \cdot \nabla\left(\nabla L \cdot \nabla^2 L\right)\right].
\end{aligned} \tag{70}$$

The left part is now deriving the form of $\mu^{k-j}\Theta_j^{(1)}$, which can be done by first finding

$$\begin{aligned}
\mu^{k-j}\gamma_j^{(0)} \cdot \nabla G_{j-1} &\approx \mu^{k-j}\frac{(1+\mu)}{2(1-\mu)^3}(\nabla L \cdot \nabla^2 L) \cdot \nabla G_{j-1} \\
&\approx \mu^{k-j}\frac{(1+\mu)}{2(1-\mu)^4}(\nabla L \cdot \nabla^2 L) \cdot \nabla^2 L \approx \mu^{k-j}\gamma_{j-1}^{(0)} \cdot \nabla G_{j-1}
\end{aligned} \tag{71}$$

and

$$\mu^{k-j} G_{j-1} \cdot \nabla \left(G_j \cdot \nabla G_j - G_{j-1} \cdot \nabla G_{j-1} \right) \approx \frac{2\mu^{k-j}}{(1-\mu)^3} \nabla L \cdot \nabla \left(\nabla L \cdot \nabla^2 L \right), \tag{72}$$

thus

$$\sum_{j=0}^{k} \mu^{k-j} \Theta_j^{(1)} \approx \frac{(1+\mu)}{2(1-\mu)^5} \left[(\nabla L \cdot \nabla^2 L) \cdot \nabla^2 L + \nabla L \cdot \nabla \left(\nabla L \cdot \nabla^2 L \right) \right]. \tag{73}$$

Combing this equation with Eq. (70), we can now conclude the form of $\gamma_k^{(1)}$ when k is large:

$$\begin{aligned}
\gamma_k^{(1)} &= \sum_{j=0}^{k} \mu^{k-j} \left(\Psi_j^{(1)} + \mu \Theta_j^{(1)} \right) \\
&\quad \frac{1}{4(1-\mu)^4} \left[(1+\mu)(\nabla L \cdot \nabla^2 L) \cdot \nabla^2 L + \frac{(1+5\mu)}{3} \nabla L \cdot \nabla \left(\nabla L \cdot \nabla^2 L \right) \right] \\
&\qquad + \frac{\mu(1+\mu)}{2(1-\mu)^5} \left[(\nabla L \cdot \nabla^2 L) \cdot \nabla^2 L + \nabla L \cdot \nabla \left(\nabla L \cdot \nabla^2 L \right) \right] \\
&= \frac{(1+\mu)^2}{4(1-\mu)^5} \left[(\nabla L \cdot \nabla^2 L) \cdot \nabla^2 L + \frac{1 + 10\mu + \mu^2}{3(1+\mu)^2} \nabla L \cdot \nabla \left(\nabla L \cdot \nabla^2 L \right) \right]
\end{aligned} \tag{74}$$

Note that when $\mu = 0$ we recover the result of GD, i.e., $\gamma_k^{(1)} = \frac{(\nabla L \cdot \nabla^2 L) \cdot \nabla^2 L}{4} + \frac{\nabla L \cdot \nabla \left(\nabla L \cdot \nabla^2 L \right)}{12}$.

B Proofs for Section 3

Given data (x_i, y_i), the architecture of 2-layer diagonal linear network is

$$f(x_i; \mathbf{w}) = x_i^T (\mathbf{w}_+ \odot \mathbf{w}_+ - \mathbf{w}_- \odot \mathbf{w}_-) = \sum_{j=1}^{d} x_{i;j} \left(\mathbf{w}_{+;j}^2 - \mathbf{w}_{-;j}^2 \right) \tag{75}$$

and the empirical loss function is

$$L(\mathbf{w}) = \frac{1}{2n} \sum_{i=1}^{n} (f(x_i; \mathbf{w}) - y_i)^2.$$

We let $r = (r_1, \ldots, r_n)^T \in \mathbb{R}^n$ be the residual where $\forall i : r_i = f(x_i; \mathbf{w}) - y_i$. According to Theorem 2.1, the HBF learning dynamics of model parameters $\mathbf{w}_+$ and $\mathbf{w}_-$ will be

$$\dot{\mathbf{w}}_+ = -\frac{\nabla_{\mathbf{w}_+} L}{1-\mu} - \eta \gamma_k^{\mathbf{w}_+}, \quad \dot{\mathbf{w}}_- = -\frac{\nabla_{\mathbf{w}_-} L}{1-\mu} - \eta \gamma_k^{\mathbf{w}_-} \tag{76}$$

where we use $\gamma_k^{\mathbf{w}_+} \in \mathbb{R}^d$ and $\gamma_k^{\mathbf{w}_-} \in \mathbb{R}^d$ to represent the error terms for HBF of $\mathbf{w}_+$ and $\mathbf{w}_-$, respectively, and the gradients are

$$\nabla_{\mathbf{w}} L = \frac{1}{n} X^T r, \tag{77}$$

$$\nabla_{\mathbf{w}_+} L = 2\mathbf{w}_+ \odot \nabla_{\mathbf{w}} L, \quad \nabla_{\mathbf{w}_-} L = -2\mathbf{w}_- \odot \nabla_{\mathbf{w}} L. \tag{78}$$

Using the expressions above, it can be easily verified that

$$\mathbf{w}_- \odot \nabla_{w_+} L + \mathbf{w}_+ \odot \nabla_{w_-} L = 0, \tag{79}$$

and we will frequently use this relation later. Recall the definition of $\kappa_j = \mathbf{w}_{+;j} \mathbf{w}_{-;j}$, we now present useful lemmas before proving Theorem 3.1.

Lemma B.1. *Let $\kappa_j(t) = \mathbf{w}_{+;j}(t) \mathbf{w}_{-;j}(t)$, $\gamma_{k;j}^{\mathbf{w}_\pm}$ denote the j-th component of $\gamma_k^{\mathbf{w}_\pm}$, and*

$$\epsilon_j(t) = \int_0^t ds \left(\frac{\gamma_{k;j}^{\mathbf{w}_+}(s)}{\mathbf{w}_{+;j}(s)} + \frac{\gamma_{k;j}^{\mathbf{w}_-}(s)}{\mathbf{w}_{-;j}(s)} \right), \tag{80}$$

then we have

$$\kappa_j(t) = \kappa_j(0) e^{-\eta \epsilon_j(t)}. \tag{81}$$

Proof. The proof applies the dynamics of $\mathbf{w}_+$ and that of $\mathbf{w}_-$:

$$\begin{aligned}\frac{d\kappa}{dt} &= \dot{\mathbf{w}}_+ \odot \mathbf{w}_- + \dot{\mathbf{w}}_- \odot \mathbf{w}_+ \\ &= \left(-\frac{\nabla_{\mathbf{w}_+} L}{1-\mu} - \eta\gamma_k^{\mathbf{w}_+}\right) \odot \mathbf{w}_- + \mathbf{w}_+ \odot \left(-\frac{\nabla_{\mathbf{w}_-} L}{1-\mu} - \eta\gamma_k^{\mathbf{w}_-}\right) \\ &= -\eta\left(\gamma_k^{\mathbf{w}_+} \odot \mathbf{w}_- + \mathbf{w}_+ \odot \gamma_k^{\mathbf{w}_-}\right), \end{aligned} \tag{82}$$

where we use Eq. (26) in the second equality and Eq. (79) in the third equality. As a result, for the j-th component of κ, we have

$$\begin{aligned}\dot{\kappa}_j &= -\eta\kappa_j\left(\frac{\gamma_{k;j}^{\mathbf{w}_+}}{\mathbf{w}_{+;j}} + \frac{\gamma_{k;j}^{\mathbf{w}_-}}{\mathbf{w}_{-;j}}\right) \\ \implies \kappa_j(t) &= \kappa_j(0)e^{-\eta\epsilon_j(t)}. \end{aligned} \tag{83}$$

□

It is also interesting to investigate the dynamics of $\mathbf{w}$ as shown below.

Lemma B.2. *If $\mathbf{w}_\pm$ is run with HBF, then the dynamics of $\mathbf{w}$ satisfies that*

$$\dot{\mathbf{w}} = -4\mathbf{v} \odot \frac{\nabla_{\mathbf{w}} L}{1-\mu} - \eta\Gamma_k^{\mathbf{w}} \tag{84}$$

where we let

$$\mathbf{v} = \left(\mathbf{w}_+ \odot \mathbf{w}_+ + \mathbf{w}_- \odot \mathbf{w}_-\right), \quad \Gamma_k^{\mathbf{w}} = 2\left(\gamma_k^{\mathbf{w}_+} \odot \mathbf{w}_+ - \gamma_k^{\mathbf{w}_-} \odot \mathbf{w}_-\right). \tag{85}$$

Proof. Using the dynamics of $\mathbf{w}_\pm$ Eq. (26), we can show that

$$\begin{aligned}\dot{\mathbf{w}} &= 2\dot{\mathbf{w}}_+ \odot \mathbf{w}_+ - 2\dot{\mathbf{w}}_- \odot \mathbf{w}_- \\ &= 2\left(-\frac{\nabla_{\mathbf{w}_+} L}{1-\mu} - \eta\gamma_k^{\mathbf{w}_+}\right) \odot \mathbf{w}_+ - 2\left(-\frac{\nabla_{\mathbf{w}_-} L}{1-\mu} - \eta\gamma_k^{\mathbf{w}_-}\right) \odot \mathbf{w}_- \\ &= -4\left(\mathbf{w}_+ \odot \mathbf{w}_+ + \mathbf{w}_- \odot \mathbf{w}_-\right) \odot \frac{\nabla_{\mathbf{w}} L}{1-\mu} - 2\eta\left(\gamma_k^{\mathbf{w}_+} \odot \mathbf{w}_+ - \gamma_k^{\mathbf{w}_-} \odot \mathbf{w}_-\right). \end{aligned} \tag{86}$$

□

To show the implicit bias of HBF, we need to first explore the dynamics of $\mathbf{w}$, which is present in the following lemma.

Lemma B.3 (Dynamics of $\mathbf{w}$ for diagonal linear networks under HBF)**.** *Under conditions of Theorem 3.1, if the diagonal linear network $f(x;\mathbf{w})$ is trained with HBF (Theorem 2.1), let*

$$\begin{aligned}\Lambda_j^{\mathrm{GF}}(\mathbf{w};\kappa(t)) &= \frac{2\kappa_j(t)}{4}\left[\frac{\mathbf{w}_j(t)}{2\kappa_j(t)}\operatorname{arcsinh}\left(\frac{\mathbf{w}_j(t)}{2\kappa_j(t)}\right) - \sqrt{1+\frac{\mathbf{w}_j^2(t)}{4\kappa_j^2(t)}} + 1\right] \\ \varphi_j(t) &= \frac{\eta}{4}\int_0^t ds\left[\frac{\gamma_{k;j}^{\mathbf{w}_+}(s)}{\mathbf{w}_{+;j}(s)} - \frac{\gamma_{k;j}^{\mathbf{w}_-}(s)}{\mathbf{w}_{-;j}(s)}\right] \\ \Lambda_j(\mathbf{w},t;\kappa) &= \Lambda_j^{\mathrm{GF}}(\mathbf{w};\kappa(t)) + \mathbf{w}_j(t)\varphi_j(t), \end{aligned} \tag{87}$$

then the learning dynamics of the parameter $\mathbf{w}$ satisfies that

$$\frac{d}{dt}\partial_{\mathbf{w}_j}\Lambda_j + \frac{\partial_{\mathbf{w}_j} L}{1-\mu} = 0. \tag{88}$$

The proof of this lemma can be found in Appendix B.2. In the following we first focus on the proof of Theorem 3.1.

B.1 Proof of Theorem 3.1

Now we can prove Theorem 3.1 with above helper lemmas.

Proof. Recall the definition of Λ_j in Lemma B.3 and we further define

$$\Lambda(\mathbf{w}, t; \kappa) = \sum_{j=1}^{d} \Lambda_j(\mathbf{w}, t; \kappa), \tag{89}$$

then Lemma B.3 gives us

$$\begin{aligned}
\frac{d}{dt}\nabla_{\mathbf{w}}\Lambda(\mathbf{w}, t; \kappa) &= \left(\frac{d}{dt}\partial_{\mathbf{w}_1}\Lambda_1(\mathbf{w}, t; \kappa), \ldots, \frac{d}{dt}\partial_{\mathbf{w}_d}\Lambda_d(\mathbf{w}, t; \kappa)\right)^T \\
&= -\frac{X^T r}{n(1-\mu)} \\
\implies \nabla_{\mathbf{w}}\Lambda(\mathbf{w}(\infty), \infty; \kappa(\infty)) \; - \nabla_{\mathbf{w}}\Lambda(\mathbf{w}(0), 0; \kappa(0)) &= -\sum_{i=1}^{n} \frac{x_i \int_0^\infty r_i(\tau) d\tau}{n(1-\mu)} = \sum_{i=1}^{n} x_i c_i
\end{aligned} \tag{90}$$

where we let $c_i = -\frac{\int_0^\infty r_i(\tau)d\tau}{n(1-\mu)}$. Let $\nabla_{\mathbf{w}}\Lambda(\mathbf{w}(0), 0; \kappa(0)) = 0$ and recall the definition of $\Lambda(\mathbf{w}; \kappa)$ in Theorem 3.1., then Eq. (90) is equivalent to

$$\nabla_{\mathbf{w}}\Lambda(\mathbf{w}; \kappa) - \sum_{i=1}^{n} x_i c_i = 0,$$

which is exactly the KKT condition of $\arg\min_{\mathbf{w}: X\mathbf{w}=y} \Lambda(\mathbf{w}; \kappa)$ proposed in Theorem 3.1. Therefore, we finish the proof. □

B.2 Proof of Lemma B.3

In this section we present the proof of Lemma B.3.

Proof. For simplicity, in the following we write the subscripts explicitly. According to Lemma B.2, the dynamics of $\mathbf{w}_j$ can be written as

$$\dot{\mathbf{w}}_j = -\frac{4}{1-\mu}\mathbf{v}_j \partial_{\mathbf{w}_j} L - \eta \Gamma^{\mathbf{w}}_{k;j}. \tag{91}$$

Note that

$$\mathbf{v}_j^2 - \mathbf{w}_j^2 = 4\mathbf{w}_{+;j}^2 \mathbf{w}_{-;j}^2 \implies \mathbf{v}_j^2 = \sqrt{\mathbf{w}_j^2 + 4\kappa_j^2}, \tag{92}$$

then Eq. (91) can be written as

$$\frac{\dot{\mathbf{w}}_j}{4\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2}} = -\frac{\partial_{\mathbf{w}_j} L}{1-\mu} - \eta \frac{\Gamma^{\mathbf{w}}_{k;j}}{4\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2}}. \tag{93}$$

We now define a function

$$\Lambda_j(\mathbf{w}, t; \kappa) = \bar{\Lambda}_j(\mathbf{w}, t; \kappa) + \mathbf{w}_j \varphi_j(t) \tag{94}$$

for some $\bar{\Lambda}_j(\mathbf{w}, t; \kappa)$ and $\varphi_j(t)$ such that

$$\frac{d}{dt}\partial_{\mathbf{w}_j}\Lambda_j(\mathbf{w}, t; \kappa) = \frac{\dot{\mathbf{w}}_j + \eta \Gamma^{\mathbf{w}}_{k;j}}{4\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2}}, \tag{95}$$

the we can prove this lemma. Now we continue to find the $\bar{\Lambda}_j(\mathbf{w}, t; \kappa)$ and $\varphi_j(t)$. By definition,

$$\frac{d}{dt}\partial_{\mathbf{w}_j}\Lambda_j(\mathbf{w}, t; \kappa) = \partial^2_{\mathbf{w}_j}\bar{\Lambda}_j \dot{\mathbf{w}}_j + \partial_t \partial_{\mathbf{w}_j}\bar{\Lambda}_j + \dot{\varphi}_j, \tag{96}$$

which, when compared with Eq. (95), implies that

$$\partial^2_{\mathbf{w}_j}\bar{\Lambda}_j = \frac{1}{4\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2}}. \tag{97}$$

Solving this equation gives us

$$\partial_{\mathbf{w}_j}\bar{\Lambda}_j = \frac{1}{4}\int \frac{d\mathbf{w}_j}{\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2}} = \frac{\ln\left(\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2} + \mathbf{w}_j\right)}{4} + c \tag{98}$$

where c is a constant and can be determined by requiring $\partial_{\mathbf{w}_j}\bar{\Lambda}_j|_{t=0} + \varphi_j(0) = 0 \implies c = -\ln(2\kappa_j(0))/4$. Thus Eq. (98) becomes

$$\partial_{\mathbf{w}_j}\bar{\Lambda}_j = \frac{1}{4}\ln\left(\frac{\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2(t)} + \mathbf{w}_j}{2\kappa_j(t)}\right) - \frac{\eta\epsilon_j(t)}{4}$$

where we have used the definition of $\epsilon_j(t)$ in Lemma B.1. Solving the above equation will give us the form of $\bar{\Lambda}_j$

$$\begin{aligned}
\bar{\Lambda}_j &= \frac{1}{4}\int d\mathbf{w}_j \operatorname{arcsinh}\left(\frac{\mathbf{w}_j}{2\kappa_j(t)}\right) - \frac{\eta\epsilon_j(t)\mathbf{w}_j}{4} \\
&= \frac{1}{4}\left[\mathbf{w}_j \operatorname{arcsinh}\left(\frac{\mathbf{w}_j}{2\kappa_j(t)}\right) - \sqrt{\mathbf{w}_j^2 + 4\kappa_j^2(t)} + 2\kappa_j(t)\right] - \frac{\eta\epsilon_j(t)\mathbf{w}_j}{4} \\
&= \Lambda_j^{\mathrm{GF}}(\mathbf{w};\kappa(t)) - \frac{\eta\epsilon_j(t)\mathbf{w}_j}{4}
\end{aligned} \tag{99}$$

where we use the definition of Λ^{GF} in Eq. (25). Comparing the rest parts of Eq. (96) with Eq. (95) requires that

$$\begin{aligned}
&\partial_t\partial_{\mathbf{w}_j}\bar{\Lambda}_j + \dot{\varphi}_j = \eta\frac{\Gamma^{\mathbf{w}}_{k;j}}{4\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2(t)}} \\
&\implies \dot{\varphi}_j(t) = \frac{\eta\kappa_j^2(t)\dot{\epsilon}_j(t)}{\left(\mathbf{w}_j + \sqrt{\mathbf{w}_j^2 + 4\kappa_j^2(t)}\right)\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2(t)}} + \eta\frac{\Gamma^{\mathbf{w}}_{k;j}}{4\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2(t)}}.
\end{aligned} \tag{100}$$

When combined with the form of $\bar{\Lambda}_j$, we can find the form of Λ_j:

$$\begin{aligned}
\Lambda_j(\mathbf{w},t;\kappa) &= \Lambda_j^{\mathrm{GF}}(\mathbf{w};\kappa(t)) + \eta\mathbf{w}_j\int \frac{ds}{\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2(s)}}\left[\frac{\kappa_j^2(s)}{\mathbf{w}_j + \sqrt{\mathbf{w}_j^2 + 4\kappa_j^2(s)}}\dot{\epsilon}_j\right. \\
&\qquad \left. - \frac{\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2(s)}\dot{\epsilon}_j}{4} + \frac{\Gamma^{\mathbf{w}}_{k;j}}{4}\right] \\
&= \Lambda_j^{\mathrm{GF}}(\mathbf{w};\kappa(t)) + \eta\mathbf{w}_j\int \frac{ds}{\sqrt{\mathbf{w}_j^2 + 4\kappa_j^2(s)}}\left[-\frac{\mathbf{w}_j\dot{\epsilon}_j}{4} + \frac{\Gamma^{\mathbf{w}}_{k;j}}{4}\right] \\
&= \Lambda_j^{\mathrm{GF}}(\mathbf{w};\kappa(t)) + \eta\mathbf{w}_j\int ds\left[\frac{\gamma^{\mathbf{w}_+}_{k;j}}{\mathbf{w}_{+;j}} - \frac{\gamma^{\mathbf{w}_-}_{k;j}}{\mathbf{w}_{-;j}}\right]
\end{aligned} \tag{101}$$

where we use the definition of ϵ_j (Lemma B.1) and Eq. (92) in the last equality. □

B.3 Implicit Bias of HBF for Diagonal Linear Networks when $\alpha = 2$

In this case, the correction term $\gamma^{\mathbf{w}_\pm}$ will be

$$\gamma^{\mathbf{w}_\pm} = \frac{1+\mu}{2(1-\mu)^3}\nabla_{\mathbf{w}_\pm}L \cdot \nabla^2_{\mathbf{w}_\pm}L.$$

We need to first find the Hessian $\nabla^2_{\mathbf{w}_\pm} L$. Due to the element-wise product, it will be convenient to derive the Hessian by writing the subscripts explicitly. We start with $\mathbf{w}_+$.

$$\begin{aligned}
\partial_{\mathbf{w}_{+;i}} \partial_{\mathbf{w}_{+;j}} L &= \frac{2}{n} \partial_{\mathbf{w}_{+;i}} \left(\mathbf{w}_{+;j} (X^T r)_j \right) \\
&= \frac{2}{n} \left[\delta_{ij} (X^T r)_j + \sum_{c=1}^{n} \mathbf{w}_{+;j} \partial_{\mathbf{w}_{+;i}} \left(x_{c;j} (x_c^T \mathbf{w} - y_c) \right) \right] \\
&= \frac{2}{n} \left[\delta_{ij} (X^T r)_j + 2 \sum_{c=1}^{n} \mathbf{w}_{+;j} x_{c;j} x_{c;i} \mathbf{w}_{+;i} \right], \qquad (102)
\end{aligned}$$

where we use the delta symbol $\delta_{ij} = 1$ if $i = j$ otherwise $\delta_{ij} = 0$. Therefore, we can conclude that

$$\nabla^2_{\mathbf{w}_+} L = \frac{2}{n} \left[\operatorname{diag}(X^T r) + 2 \sum_{c=1}^{n} (\mathbf{w}_+ \odot x_c)(\mathbf{w}_+ \odot x_c)^T \right]. \qquad (103)$$

Following a similar approach, we obtain that for $\mathbf{w}_-$

$$\begin{aligned}
\partial_{\mathbf{w}_{-;i}} \partial_{\mathbf{w}_{-;j}} L &= -\frac{2}{n} \partial_{\mathbf{w}_{-;i}} \left(\mathbf{w}_{-;j} (X^T r)_j \right) \\
&= \frac{2}{n} \left[-\delta_{ij} (X^T r)_j + 2 \sum_{c=1}^{n} \mathbf{w}_{-;j} x_{c;j} x_{c;i} \mathbf{w}_{-;i} \right] \qquad (104) \\
\implies \nabla^2_{\mathbf{w}_-} L &= \frac{2}{n} \left[-\operatorname{diag}(X^T r) + 2 \sum_{c=1}^{n} (\mathbf{w}_- \odot x_c)(\mathbf{w}_- \odot x_c)^T \right]. \qquad (105)
\end{aligned}$$

It is now left for us to find the form of $\nabla_{\mathbf{w}_\pm} L \cdot \nabla^2_{\mathbf{w}} L$. Again, it is convenient to write the subscripts explicitly:

$$\begin{aligned}
\left(\nabla_{\mathbf{w}_+} L \cdot \nabla^2_{\mathbf{w}_+} L \right)_j &= \sum_{i=1}^{d} \partial_{\mathbf{w}_{+;i}} \partial_{\mathbf{w}_{+;j}} L \partial_{\mathbf{w}_{+;i}} L \\
&= \frac{4}{n^2} \sum_{i=1}^{d} \left[\delta_{ij} (X^T r)_j + 2 \sum_{c=1}^{n} \mathbf{w}_{+;j} x_{c;j} x_{c;i} \mathbf{w}_{+;i} \right] \mathbf{w}_{+;i} (X^T r)_i \\
&= \frac{4}{n^2} \left[\mathbf{w}_{+;j} ((X^T r)_j)^2 + 2 \sum_{c=1}^{n} \mathbf{w}_{+;j} x_{c;j} \left(x_c \odot \mathbf{w}_+ \odot \mathbf{w}_+ \right)^T X^T r \right]. \qquad (106)
\end{aligned}$$

Similarly,

$$\left(\nabla_{\mathbf{w}_-} L \cdot \nabla^2_{\mathbf{w}_-} L \right)_j = \frac{4}{n^2} \left[\mathbf{w}_{-;j} ((X^T r)_j)^2 - 2 \sum_{c=1}^{n} \mathbf{w}_{-;j} x_{c;j} \left(x_c \odot \mathbf{w}_- \odot \mathbf{w}_- \right)^T X^T r \right]. \qquad (107)$$

Using Eq. (106) and (107), we can derive that

$$\frac{\gamma_j^{\mathbf{w}_\pm}}{\mathbf{w}_{\pm;j}} = \frac{2(1+\mu)}{(1-\mu)^3 n^2} \left[((X^T r)_j)^2 \pm 2 \sum_{c=1}^{n} x_{c;j} \left(x_c \odot \mathbf{w}_\pm \odot \mathbf{w}_\pm \right)^T X^T r \right], \qquad (108)$$

which further gives us the integral ϵ_j:

$$\begin{aligned}
\dot{\epsilon}_j &= \frac{\gamma_j^{\mathbf{w}_+}}{\mathbf{w}_{+;j}} + \frac{\gamma_j^{\mathbf{w}_-}}{\mathbf{w}_{-;j}} \\
&= \frac{4(1+\mu)}{(1-\mu)^3 n^2} \left[((X^T r)_j)^2 + \sum_{c=1}^{n} \sum_{i=1}^{d} x_{c;j} x_{c;i} (X^T r)_i \left(\mathbf{w}_{+;i}^2 - \mathbf{w}_{-;i}^2 \right) \right] \\
&= \frac{4(1+\mu)}{(1-\mu)^3 n^2} \left[((X^T r)_j)^2 + \sum_{c=1}^{n} x_{c;j} x_c^T (\mathbf{w} \odot (X^T r)) \right] \\
&= \frac{4(1+\mu)}{(1-\mu)^3} \left[(\nabla_{\mathbf{w}} L)_j^2 + \frac{1}{n} \left(X^T X (\mathbf{w} \odot \nabla_{\mathbf{w}} L) \right)_j \right]. \qquad (109)
\end{aligned}$$

On the other hand, according to Lemma B.2, $\partial_{\mathbf{w}_i} L$ can be written as

$$-(1-\mu)\frac{\dot{\mathbf{w}}_i}{4\mathbf{v}_i} - \eta(1-\mu)\frac{\Gamma_i^{\mathbf{w}}}{4\mathbf{v}_i}, \tag{110}$$

which further gives us that

$$\begin{aligned}
\eta\int_0^t ds\left(X^TX(\mathbf{w}\odot\nabla_{\mathbf{w}}L)\right)_j &= -\eta(1-\mu)\sum_{c=1}^n\sum_{i=1}^d x_{c;j}x_{c;i}\int_{\mathbf{w}_i(0)}^{\mathbf{w}_i(t)} d\mathbf{w}_i\frac{\mathbf{w}_i(s)}{4\mathbf{v}_i(s)} + \mathcal{O}\left(\eta^2\right)\\
&= -\eta(1-\mu)\sum_{c=1}^n\sum_{i=1}^d x_{c;j}x_{c;i}\int_{\mathbf{w}_i(0)}^{\mathbf{w}_i(t)} d\mathbf{w}_i\frac{\mathbf{w}_i(s)}{4\sqrt{\mathbf{w}_i^2(s)+4\kappa_i^2(s)}} + \mathcal{O}\left(\eta^2\right)\\
&= -\frac{\eta(1-\mu)}{4}\sum_{c=1}^n\sum_{i=1}^d x_{c;j}x_{c;i}\left(\sqrt{\mathbf{w}_i^2(t)+4\kappa_i^2(t)}-\sqrt{\mathbf{w}_i^2(0)+4\kappa_i^2(0)}\right).
\end{aligned}$$

where we use Lemma B.2 in the first equality and Eq. (92) in the second equality. Since $\mathbf{w}(0)=0$ and Lemma B.1, we obtain

$$\begin{aligned}
\eta\int_0^t ds\left(X^TX(\mathbf{w}\odot\nabla_{\mathbf{w}}L)\right)_j &= -\frac{\eta(1-\mu)}{4}\sum_{c=1}^n\sum_{i=1}^d x_{c;j}x_{c;i}\left(\sqrt{\mathbf{w}_i^2(t)+4\kappa_i^2(0)}-2\kappa_i(0)\right)\\
&= -\frac{\eta(1-\mu)}{4}\left(X^TX\mathbf{q}(t)\right)_j
\end{aligned} \tag{111}$$

where we let $\mathbf{q}\in\mathbb{R}^d$ with

$$\mathbf{q}_i(t) = \sqrt{\mathbf{w}_i^2(t)+4\kappa_i^2(0)}-2\kappa_i(0)\geq 0.$$

Now combining Eq. (109) and Eq. (111), we can derive

$$\eta\epsilon_j(t) = \frac{4\eta(1+\mu)}{(1-\mu)^3}\int_0^t ds(\partial_{\mathbf{w}_j}L)^2 - \frac{\eta(1+\mu)}{(1-\mu)^2 n}\left(X^TX\mathbf{q}\right)_j + \mathcal{O}\left(\eta^2\right). \tag{112}$$

To obtain the full potential function, we still need to find the form of φ_j. According to the definition of $\mathbf{v}$ and ϵ_j and Eq. (108), we can derive

$$\begin{aligned}
2\gamma_{k;j}^{\mathbf{w}_+}\mathbf{w}_{+;j} - 2\gamma_{k;j}^{\mathbf{w}_-}\mathbf{w}_{-;j} - \mathbf{w}_j\dot{\epsilon}_j &= \mathbf{v}_j\left(\frac{\gamma_{k;j}^{\mathbf{w}_+}}{\mathbf{w}_+} - \frac{\gamma_{k;j}^{\mathbf{w}_-}}{\mathbf{w}_-}\right)\\
&= \frac{4(1+\mu)}{(1-\mu)^3 n}\sum_{c=1}^n\sum_{i=1}^d \mathbf{v}_j x_{c;j}x_{c;i}\mathbf{v}_i\partial_{\mathbf{w}_i}L,
\end{aligned} \tag{113}$$

which, when combined with the definition of φ_j in Lemma B.3, further gives us

$$\begin{aligned}
\dot{\varphi}_j &= \eta\frac{(1+\mu)}{(1-\mu)^3 n}\sum_{c=1}^n\sum_{i=1}^d x_{c;j}x_{c;i}\mathbf{v}_i\partial_{\mathbf{w}_i}L\\
&= -\frac{\eta(1+\mu)}{4(1-\mu)^2 n}\sum_{c=1}^n\sum_{i=1}^d x_{c;j}x_{c;i}\dot{\mathbf{w}}_i + \mathcal{O}\left(\eta^2\right)
\end{aligned} \tag{114}$$

where we use Eq. (110) in the second equality. As a result,

$$\varphi_j(\infty) = -\frac{\eta(1+\mu)}{4(1-\mu)^2 n}\sum_{c=1}^n\sum_{i=1}^d x_{c;j}x_{c;i}\mathbf{w}_i(t) = \frac{\eta(1+\mu)}{4(1-\mu)^2 n}\left(X^TX\mathbf{w}\right)_j. \tag{115}$$

One interesting thing aspect of φ_j if $\mathbf{w}$ converges to an interpolation solution where $X\mathbf{w}(\infty)=y$ is

$$\varphi_j(\infty) = \frac{\eta(1+\mu)}{4(1-\mu)^2}\partial_{\mathbf{w}_j}L(0). \tag{116}$$

In summary, the potential function $\mathcal{O}\left(\eta^2\right)$-close HBF is

$$\begin{aligned}
\kappa_j(\infty) &= \kappa_j(0)\exp\left(-\frac{4\eta(1+\mu)}{(1-\mu)^3}\int_0^\infty ds(\partial_{\mathbf{w}_j}L)^2 + \frac{\eta(1+\mu)}{(1-\mu)^2 n}\left(X^TX\mathbf{q}(\infty)\right)_j\right)\\
\Lambda_j(\mathbf{w},\infty;\kappa) &= \Lambda_j^{\mathrm{GF}}(\mathbf{w},\kappa(\infty)) + \frac{\eta(1+\mu)}{4(1-\mu)^2}\mathbf{w}_j\partial_{\mathbf{w}_j}L(0).
\end{aligned} \tag{117}$$

C Details for Numerical experiments

The experiments are conducted on a CentOS Linux 7.9.2 platform equipped with an Intel(R) Xeon CPU E5-2683 at 3.00 GHz, 256GB of RAM, and an NVIDIA Tesla A100 graphics card.

C.1 Details for Section 2.2.2

For the experiment of Figure 1, we conduct observation on the comparison of directional smoothness for HB and GD on the CIFAR-10 dataset Krizhevsky et al. (2009). A multilayer perceptron with two hidden layers (each of width 200) is trained for 2000 epochs using full-batch GD and HB ($\mu = 0.9$), and the step size is set to 0.1.

C.2 Details for Section 4

For the discrete learning dynamics of HB and GD, we set the step size as η and the momentum factor is μ. For the continuous approximations, we use $\eta_{\text{Euler}} = \eta/10$ as the Euler step sizes to approximate the dynamics. These hyper-parameters are listed in Table 2.

x, y	$1, 0.6$
Starting point	$a_1 = 2.8, a_2 = 3.5$
η	5×10^{-3}
μ	0.7
η_{Euler}	5×10^{-4}

Table 2: Hyper-parameters for 2-d model.

We let the model parameter be $\beta = (a_1, a_2)^T \in \mathbb{R}^2$. For RGF, we use the ODE

$$\dot{\beta} = -\frac{\nabla_\beta L}{1-\mu} \implies \beta_{k+1} = \beta_k - \eta_{\text{Euler}} \frac{\nabla_\beta L}{1-\mu}.$$

Formulations of HBFs with $\alpha = 2, 3$ are denoted in Table 1. We denote $\mathbf{1}_d = (1, \ldots, 1)^T \in \mathbb{R}^d$. For the dataset $\{(x_i, y_i)\}_{i=1}^d$, we set $n = 40, d = 100$. The data point follows a Gaussian distribution $\mathcal{N}(0, I_d)$. To make the ground truth solution $\mathbf{w}^*$ sparse, we let 5 components of it be nonzero. Recall that the initialization is $\kappa(0) = s^2 \mathbf{1}_d$ where s controls the initialization scale. In Fig. 6(b), we make the initialization as $\mathbf{w}_+ = \mathbf{w}_- = s\mathbf{1}_d$ with $s = 0.01$. We set the step size η for HB as 10^{-3}. For RGF and HBF, we let the Euler step size $\eta_{\text{Euler}} = 10^{-4}$ to simulate the continuous dynamics. In Fig. 6(a) and 6(b), we set $\eta = 10^{-2}$. For the initialization, to make the task slightly harder, we let $\mathbf{w}_+ = \vartheta s \mathbf{1}_d$ and $\mathbf{w}_- = s\mathbf{1}_d/\vartheta$ with $\vartheta = 0.9$ such that we still have $\kappa(0) = s^2 \mathbf{1}_d$ while the initialization symmetry is slightly broken.

Addition experiments for different η_{Euler} We additionally run the experiments in Fig. 2 for each of $\eta_{\text{Euler}} = \{\eta/10, \eta/100, \eta/1000\}$, and confirm that the observations and conclusions hold in all different η_{Euler} Fig. 3.

C.3 Additional Numerical Experiments for Non-Linear Networks

We conduct experiments of Fig. 2(b) in the MNIST dataset, where we now train a three-layer fully-connected neural networks (FCNN). The FCNN has a structure of `Linear(784×128)` $\to$ `SiLU`$\to$`Linear(128×128)`$\to$`BatchNormalization`$\to$`Linear(128×10)`. Cross-entropy is used for the loss function. The batch size is 60,000 and the momentum factor $\mu \in (0.7, 0.8)$. The learning rate is $\eta = 0.01$. The results (reported in Fig. 4) well align with the results for the toy model in Fig. 2(b): HBF with $\alpha = 3$ has lower discretization error compared to HBF with $\alpha = 2$, and both of them are better than the RGF.

C.4 Additional Numerical Experiments for Diagonal Linear Networks

We now investigate the implicit bias of HB for 2-layer diagonal linear networks

$$f(x; \mathbf{w}) = \mathbf{w}^T x = (\mathbf{w}_+ \odot \mathbf{w}_+ - \mathbf{w}_- \odot \mathbf{w}_-)^T x \tag{118}$$

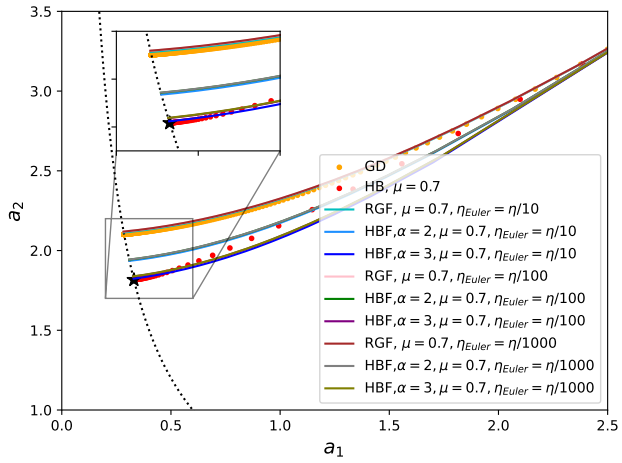


Figure 3: Trajectories of continuous approximations with different η_{Euler}.

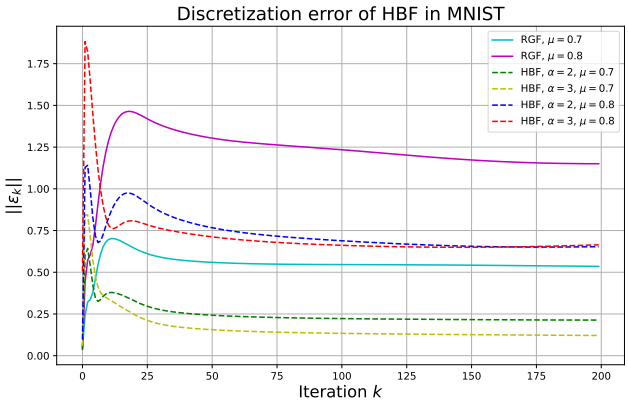


Figure 4: Discretization error of HBF in MNIST when training a three non-linear networks.

for a dataset $\{(x_i, y_i)\}_{i=1}^n$ where $x \in \mathbb{R}^d, y \in \mathbb{R}$. The empirical loss is

$$L(\mathbf{w}_+, \mathbf{w}_-) = \sum_i (f(x_i; \mathbf{w}) - y_i)^2. \tag{119}$$

We let $n < d$ and denote the ground truth solution as $\mathbf{w}^*$ such that $\mathbf{w}^{*T} x = y$. We let $\mathbf{w}^*$ be sparse. For a given scale s we let

$$\kappa(0) = \mathbf{w}_+(0) \odot \mathbf{w}_-(0) = s^2(1, \ldots, 1)^T \in \mathbb{R}^d. \tag{120}$$

C.4.1 Discretization Error for Different Approximations

Our first experiment explores the discretization error, where we let k denote the iteration count and first obtain $\mathbf{w}_k^{\text{HB}}$ by training $f(x; \mathbf{w})$ with HB. In addition, we also train $f(x; \mathbf{w})$ with RGF (Eq. (1)) and HBF (Corollary 3.2), respectively. We calculate the discretization error as

$$\|\mathbf{w}_k^{\text{HB}} - \mathbf{w}(t_k)\|_2^2 \tag{121}$$

for $\mathbf{w}(t_k)$ obtained from HBF or RGF and present the results in Fig. 5, where HBF enjoys smaller discretization error than RGF for different μ, supporting our theoretical claims.

C.4.2 Implication for difference of implicit bias between HB and GD

We first discuss the implication obtained from our HBF. Setting $\mu = 0$ in Corollary 3.2 gives us the implicit bias of $\mathcal{O}\left(\eta^2\right)$-close continuous approximation of GD, i.e., IGR (Barrett and Dherin, 2022)

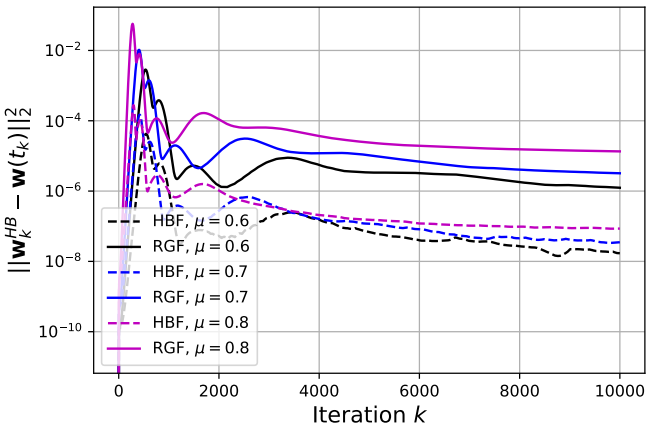


Figure 5: Discretization errors $\|\mathbf{w}_k^{\mathrm{HB}} - \mathbf{w}(t_k)\|_2^2$ for HBF (dotted lines) and RGF (solid lines), respectively, when training the 2-layer diagonal linear networks.

Flow (IGRF). Both IGRF and HBF have the initialization rescaling effect. The difference between them is closely connected with the composed parameter

$$\psi := \frac{\eta(1+\mu)}{(1-\mu)^2} \tag{122}$$

and the value of $\Phi/(1-\mu)$. And the discrepancy between the implicit bias of HB and that of GD will be more obvious for large value of μ. These observations stand in contrast to the case for $\mathcal{O}(\eta)$-close RGF, which cannot distinguish the implicit bias of HB from that of GD.

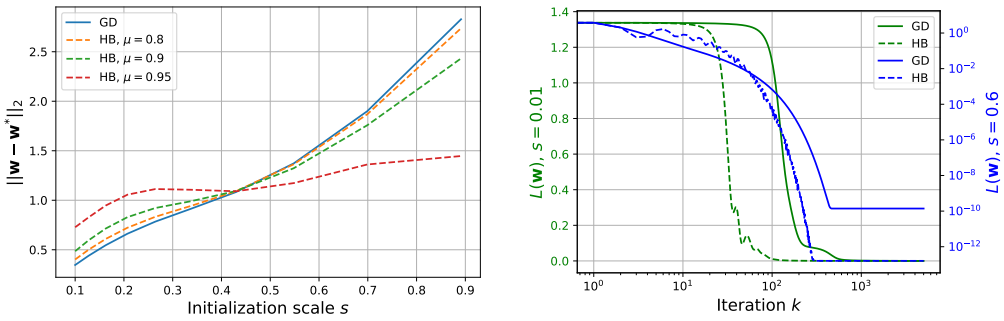
In addition, we note the following observations: (i). the value of

$$\kappa \propto \exp(-\frac{\eta 4(1+\mu)}{(1-\mu)^3}\int_0^\infty ds(\partial L)^2) \tag{123}$$

is related to the speed of convergence since it depends on the integral of gradient along the training trajectory, implying that a faster speed of convergence would possibly lead to a smaller $\int_0^\infty ds(\partial L)^2$ which then leads to a larger κ; (ii). a smaller κ implies a better sparsity and generalization performance for sparse regression, because the objective function in Corollary 3.2 will be closer to the ℓ_1-norm.

Now let κ^{HBF} and κ^{IGRF} be κ obtained from HBF and IGRF, respectively. As HB converges faster than GD in practice, then ∂L becomes neglectable very quickly for HBF if it converges too fast, and, according to our observation (i) above, we conclude that $\kappa^{HBF} > \kappa^{IGRF}$. Then according to our observation (ii) above, IGRF will generalize better than HBF. On the other hand, if the speed of convergence of HBF and that of IGRF are similar (e.g., blue line in Fig. 6(b)), then HBF and IGRF will have similar values of $\Phi = \int_0^\infty ds(\partial L)^2$ while κ^{HBF} additionally depends on a coefficient $\frac{1+\mu}{(1-\mu)^3} > 1$, thus it is possible that in this cae $\kappa^{HBF} < \kappa^{IGRF}$, which implies that HBF will generalize better in this case. In summary, there might exist a tension between the speed of convergence and the generalization for HB, i.e., if HB converges too fast, κ would be larger for HB hence solutions of GD would enjoy better generalization properties.

Below we conduct experiments for the above claims. In particular, we compare the implicit bias of HB with that of GD. Given s, we train $f(x;\mathbf{w})$ with GD and HB, respectively. We calculate the distance between the returned solution $\mathbf{w}(\infty)$ and the ground truth solution $\mathbf{w}^*$, i.e., $\|\mathbf{w}(\infty) - \mathbf{w}^*\|_2$, as a measure of generalization performance and report the results in Fig. 6(a). It can be seen that, when the initialization scale s is small, solutions of GD generalize better than those of HB. This can be explained by Corollary 3.2: compared to GD, when s is small, $L(\mathbf{w})$ decreases much faster for HB (green lines in Fig. 6(b)), which leads to a smaller $\int dsL(\mathbf{w})$ and weaker initialization mitigation effect, thus the solutions of HB generalize worse than GD solutions. Recall that in Corollary 3.2, as $\kappa_j(0) = s^2$ increases, Φ determines the generalization performances for HB and GD since it controls the extent of the initialization mitigation effect. Furthermore, $L(\mathbf{w})$ does not decrease much faster



(a) (b)

Figure 6: (a). Generalization performances $\|\mathbf{w}(\infty) - \mathbf{w}^*\|_2$ for different initialization scales s when $f(x; \mathbf{w})$ is trained by GD and HB with different values of μ. (b). $L(\mathbf{w})$ during training processes of HB ($\mu = 0.9$) and GD for different s.

for HB than for GD (blue lines in Fig. 6(b)), thus GD and HB have a similar value of Φ, which is further enhanced by a factor of $(1 + \mu)/(1 - \mu)^3$ for HB according to Corollary 3.2. As a result, HB solutions will generalize better than GD and the discrepancy between them is more significant for large μ (large $(1 + \mu)/(1 - \mu)^3$) as shown in Fig. 6(a).

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: This paper provides a new continuous time model for HB method, HBF, and investigates several properties of the discrete HB method through lens of the HBF. These contributions exactly match those in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Please see the separate paragraph "Limitation" in Section 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: Proofs for all theorems are presented in Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have presented numerical experiments in Section 2.2.2 and Section 4 with the corresponding experimental details in Appendix C.

   Guidelines:

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

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

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

Answer: [NA]

Justification: We use synthetic data and open-source data, which are discussed in Appendix C.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Please see Appendix C.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [NA]

   Justification: We do not have error bars to report.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Please see Appendix C.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conform with the NeurIPS Code of Ethis in every respect.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

    Justification: This is a theoretical paper, which dose not have direct potential societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: This is a theoretical paper, which does not provide new data or models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use the open-source data CIFAR-10, which are properly credited in this paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: This paper does not provide new assets.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: These topics are not covered by this paper.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [Yes]

    Justification: This paper does not involve any of these subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.