Gradient Flow Dynamics and Implicit Bias of Diagonal Linear Networks under Infinitesimal Initialization

Jiajie Zhao [1 2] **Jianxing Wang** [1 2] **Junjie Yang** [1 2] **Zhiwei Bai** [1 2] **Yaoyu Zhang** [1 2 3]

Abstract

We study the gradient flow dynamics of diagonal linear networks for regression tasks under infinitesimal initialization. Extending Theorem 1 from Pesme & Flammarion (2023), we generalize the analysis to both deep diagonal linear networks and a broader class of two-layer diagonal linear networks (as defined in Definition 4.1). Specifically, we demonstrate that the training trajectories of these models can be equivalently characterized by the proposed Algorithm 1. We further prove that this algorithm converges to the solution of a modified ℓ_1 norm minimization problem. As a result, we establish that the implicit bias of both network architectures corresponds to a modified ℓ_1 norm in the regime of infinitesimal initialization. Additionally, we provide insights into the underlying mechanisms governing these dynamics by identifying the Structural Invariant Manifold (SIM) (Zhao et al., 2026) as the key geometric structure that shapes the learning process.

1. Introduction

The remarkable success of deep learning relies on the ability of overparameterized neural networks to generalize well to unseen data, even when they possess enough capacity to memorize the training set with random labels (Zhang et al., 2017). Classical learning theory, which relies on uniform convergence and capacity measures like VC-dimension (Vapnik & Chervonenkis, 2015; Mohri et al., 2018), often fails to explain this phenomenon (Zhang et al., 2017; Neyshabur et al., 2017). Consequently, attention has shifted toward the concept of implicit bias (Neyshabur et al., 2017; Vardi, 2023), which posits that gradient-based optimization methods favor solutions that generalize well.

The implicit bias in linear models is well-understood—gradient descent converges to the minimum Euclidean norm solution in regression (Gunasekar et al., 2018; Zhang et al., 2017) and max-margin solution in classification (Soudry et al., 2018). As for nonlinear model, diagonal linear network, where each layer applies a diagonal linear transformation to the input, is a simple testbed for studying implicit bias. In Woodworth et al. (2020), the authors showed that for a two-layer diagonal linear network, the implicit bias of gradient descent is the ℓ_1 norm under infinitesimal initialization, and to the L_2 norm under large (infinite) initialization. They further demonstrated that similar results hold for deep diagonal linear networks under infinitesimal initialization of a specific direction.

Beyond characterizing convergence points, there is growing interest in understanding the complete trajectory of the dynamics. It is observed that under small initialization, saddle-to-saddle dynamics is present in many models, such as the diagonal linear network (Pesme & Flammarion, 2023; Jacot et al., 2021; Berthier, 2023), matrix factorization (Li et al., 2021; Bai et al., 2024), ReLU network (Boursier et al., 2022; Bantzis et al., 2025). The most relevant work is Pesme & Flammarion (2023), which, building on the mirror flow technique from Azulay et al. (2021), proves that under infinitesimal initialization, the gradient flow trajectory of a two-layer diagonal linear network successively transitions from one saddle point to another, ultimately converging to the minimum ℓ_1-norm solution. These saddle-to-saddle transitions are further characterized through a recursive algorithm.

The analysis in Pesme & Flammarion (2023) relies on the explicit form of mirror flow. In Li et al. (2022), the author establishes the existence of mirror flow in a broader setting known as commuting parametrization, which encompasses diagonal linear networks as a special case. This suggests that the results of Pesme & Flammarion (2023) could potentially be extended to more general architectures, although such a generalization remains an open question.

In this paper, we achieve this generalization via a novel technique called the Structural Invariant Manifold (Zhao

[1]School of Mathematical Sciences, Shanghai Jiao Tong University [2], Institute of Natural Sciences, MOE-LSC, Shanghai Jiao Tong University [3]School of Artificial Intelligence, Shanghai Jiao Tong University. Correspondence to: Yaoyu Zhang <zhyy.sjtu@sjtu.edu.cn>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

et al., 2026), which investigates data-independent properties of parametric models. We extend the analysis of saddle-to-saddle dynamics in Pesme & Flammarion (2023) to both deep diagonal linear networks and general two-layer diagonal linear networks (defined in Definition 4.1). As a direct corollary, we characterize the implicit bias of the two types of diagonal linear network.

1.1. Contributions

Training Trajectory: Theorem 1 in Pesme & Flammarion (2023) addresses the saddle-to-saddle dynamics of two-layer diagonal linear networks and deep networks restricted to specific initialization directions. We generalize this in Theorem 4.5 by extending the analysis to general two-layer networks (Definition 4.1) and establishing results for deep networks under generic initialization directions.

Implicit bias: Gunasekar et al. (2017); Woodworth et al. (2020) established that the implicit bias corresponds to ℓ_1-norm minimization for standard two-layer diagonal linear networks and deep diagonal linear networks under specific initialization directions. We generalize these findings by characterizing the implicit bias as a modified ℓ_1-norm minimization for general two-layer diagonal linear networks and deep diagonal linear networks under generic initialization directions (Corollary 4.6).

Mechanism: We provide a theoretical analysis of the underlying mechanism that leads to the equivalence between the dynamics and the recursive algorithm, identifying the Structural Invariant Manifold (Zhao et al., 2026) as the key for this behavior (Section 6).

2. Related Works

Diagonal Linear Network: Early work on diagonal linear networks focused on establishing the implicit bias of gradient flow (Gunasekar et al., 2017; Woodworth et al., 2020; Azulay et al., 2021) and mirror flow (Labarrière et al., 2024; Azulay et al., 2021). Subsequent research has elucidated the geometric nature of these optimization paths, specifically saddle-to-saddle dynamics (Pesme & Flammarion, 2023; Berthier, 2023; Jacot et al., 2021; Berthier, 2025). More recently, literature has addressed the algorithmic impact of discrete hyperparameters, such as step size (Even et al., 2023; Nacson et al., 2022), stochastic noise (Pesme et al., 2021), and momentum (Papazov et al., 2024).

Matrix Factorization: Since diagonal linear networks can be viewed as a special case of matrix factorization, related literature is particularly relevant. Gunasekar et al. (2017) prove that for commuting observations, gradient descent exhibits an implicit bias toward low nuclear norm solutions. Li et al. (2021) provide both theoretical and empirical evidence that gradient flow with infinitesimally small initialization behaves like a greedy low-rank algorithm. However, Pesme & Flammarion (2023) demonstrate that this greedy low-rank behavior does not extend to diagonal linear networks. Bai et al. (2024) further identify the "connectivity" of observations as a key factor influencing the dynamics. Extending the technique in this paper to the matrix factorization setting is an interesting direction for future research.

3. Preliminary

Analytic Parametric Model and Its Training: In this paper, we use $F(\boldsymbol{\theta})(\boldsymbol{x})$ to denote a parametric model. $\boldsymbol{\theta} \in \mathbb{R}^M$ is the parameter of the model, $\boldsymbol{x} \in \mathbb{R}^d$ is the input of the model. The output of the model is $F(\boldsymbol{\theta})(\boldsymbol{x}) \in \mathbb{R}$. If F is real-analytic for $(\boldsymbol{\theta}, \boldsymbol{x}) \in \mathbb{R}^M \times \mathbb{R}^d$, we say F is an analytic parametric model. To train the parametric model, we collect data $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. We define MSE loss $L(\boldsymbol{\theta}) = \sum_{i=1}^n (F(\boldsymbol{\theta})(\boldsymbol{x}_i) - y_i)^2$. We consider its gradient flow

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}), \boldsymbol{\theta}(0) = \boldsymbol{\theta}_0. \tag{1}$$

Here, $\boldsymbol{\theta}_0$ is the initialization.

Diagonal Linear Network: An L-layer diagonal linear model is $F(\boldsymbol{\theta})(\boldsymbol{x}) = \sum_{i=1}^n a_{i,1}a_{i,2}\cdots a_{i,L}x_i$, where $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, $\boldsymbol{\theta} = (a_{i,1}, a_{i,2} \ldots, a_{i,L})_{i=1}^n \in \mathbb{R}^{nL}$. The training data of diagonal linear model is denoted as $\boldsymbol{X}, \boldsymbol{y}$, where $\boldsymbol{X}$ is an $m \times n$ matrix, $\boldsymbol{y}$ is an $m \times 1$ vector. Each row of $\boldsymbol{X}$ and $\boldsymbol{y}$ is a sample. The MSE loss is $L(\boldsymbol{\theta}) = \|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}) - \boldsymbol{y}\|_2^2$. Here, $\boldsymbol{k}(\boldsymbol{\theta}) = (k_i(\boldsymbol{\theta}_i))_{i=1}^n$, where $k_i(\boldsymbol{\theta}_i) = a_{i,1} \times a_{i,2} \times \cdots \times a_{i,L}$. If $L \geq 3$, we call it the deep diagonal linear network.

Notation: We use $[n]$ to denote the set $\{1, \ldots, n\}$. We use bold letters to denote vectors, and non-bold letters to denote scalars. For a vector $\boldsymbol{v}$, we use v_i denotes the i-th coordinate of $\boldsymbol{v}$. For a matrix $\boldsymbol{X}$, we use $\boldsymbol{X}_{ij}$ to denote the element in the i-th row and j-th column, $\boldsymbol{X}_i$ to denote the i-th row of $\boldsymbol{X}$, and $\boldsymbol{X}_{:,j}$ to denote the j-th column of $\boldsymbol{X}$.

Infinitesimal initialization: Let $\boldsymbol{\theta}_0$ denote a fixed base parameter vector. We set the initialization of the model weights as: $\boldsymbol{\theta}(0) = s\boldsymbol{\theta}_0$ where $s > 0$ is a scalar multiplier. The "infinitesimal initialization" refers to the asymptotic limit as $s \to 0$.

ℓ_1 Implicit Bias: Two-layer diagonal linear model is known to exhibit implicit regularization of ℓ_1 norm under infinitesimal initialization (Gunasekar et al., 2017). Mathematically, if we denote the solution of Equation (1) as $\phi(\boldsymbol{\theta}_0, t)$, then we have

$$\lim_{s\to 0} \lim_{t\to+\infty} \phi(s\boldsymbol{\theta}_0, t) = \boldsymbol{k}^*,$$

where $\boldsymbol{k}^*$ is a solution of the following ℓ_1 minimization problem:

$$\min_{\boldsymbol{k}\in\mathbb{R}^n} \|\boldsymbol{k}\|_1 \quad s.t. \quad \boldsymbol{X}\boldsymbol{k} = \boldsymbol{y}. \tag{2}$$

4. Main Results

Definition 4.1 (General Two-Layer Diagonal Linear Network)**.** Consider the model defined by

$$F(\boldsymbol{\theta})(\boldsymbol{x}) = \sum_{i=1}^{n} k_i(\boldsymbol{\theta}_i) x_i,$$

where $\boldsymbol{\theta}_i$ is a vector for each $i \in [n]$, $\boldsymbol{\theta} = (\boldsymbol{\theta}_i)_{i=1}^n$ and $\boldsymbol{x} = (x_i)_{i=1}^n$. Define the vector $\boldsymbol{k}(\boldsymbol{\theta}) = (k_i(\boldsymbol{\theta}_i))_{i=1}^n$. Given a data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ and a target vector $\boldsymbol{y} \in \mathbb{R}^m$, we define the loss function as

$$L(\boldsymbol{\theta}) = \|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}) - \boldsymbol{y}\|_2^2.$$

The parameter $\boldsymbol{\theta}$ is optimized using gradient flow.

For each $i \in [n]$, we impose the following assumptions on the function $k_i(\boldsymbol{\theta}_i)$:

- The function $k_i(\boldsymbol{\theta}_i)$ is real analytic, satisfies $k_i(\mathbf{0}) = 0$, and has a unique critical point at $\boldsymbol{\theta}_i = \mathbf{0}$.
- Let $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_d$ denote the eigenvalues of the Hessian $\nabla^2 k_i(\mathbf{0})$. We assume that
$$\lambda_1 < \min\{0, \lambda_2\}, \quad \lambda_d > \max\{0, \lambda_{d-1}\}.$$
- The function $k_i(\boldsymbol{\theta}_i)$ is unbounded along each of its limit trajectories[1].

Summary of Main Results: Briefly speaking, in Corollary 4.6, we prove the implicit bias of deep diagonal linear network and general two-layer diagonal linear network (defined in Definition 4.1) under infinitesimal initialization. The implicit bias is a modified ℓ_1 norm:

$$R(\boldsymbol{k}) = \sum_{i=1}^{n} (t_i^+ k_i^+ + t_i^- k_i^-).$$

The proof of Corollary 4.6 consists of two steps. In Theorem 4.5, we prove that the dynamics of the two models is equivalent to Algorithm 1. In Theorem 4.2, we prove that Algorithm 1 will converge to the solution of

$$\min_{\boldsymbol{k}} R(\boldsymbol{k}) \quad s.t. \quad \boldsymbol{X}\boldsymbol{k} = \boldsymbol{y}.$$

Then the implicit bias is a direct corollary.

4.1. The Algorithm

Theorem 4.2 (Convergence and Well-Posedness of Algorithm 1)**.** *Assume the following conditions hold for Algorithm 1:*

- *The linear system $\boldsymbol{X}\boldsymbol{k} = \boldsymbol{y}$ admits at least one solution.*
- *At each iteration, the index $j = \arg\min_{i\in I_2} \delta_i$ is uniquely defined.*
- *At each iteration p, if there exists an index $i \in [n]$ such that: (i) $k_i^{(p)} = 0$, and (ii) $s_i^{(p)} = t_i^+$ or $s_i^{(p)} = -t_i^-$, then u_i is assumed to be nonzero.*

Under these assumptions, Algorithm 1 terminates in a finite number of iterations. Furthermore, the output $\boldsymbol{k}$ of the algorithm is a solution to the following optimization problem:

$$\min \sum_{i=1}^{n} \left(t_i^+ k_i^+ + t_i^- k_i^-\right) \quad \textit{subject to} \quad \boldsymbol{X}\boldsymbol{k} = \boldsymbol{y}, \tag{3}$$

where k_i^+ and k_i^- denote the positive and negative parts of k_i, respectively.

Proof. The proof is provided in Appendix A.1. □

Algorithm 1 Algorithm($\boldsymbol{X}, \boldsymbol{y}, \{t_i^+\}_{i=1}^n, \{t_i^-\}_{i=1}^n$)

Require: Data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$.
Require: Positive values $\{t_i^+\}_{i=1}^n$ and $\{t_i^-\}_{i=1}^n$.
Initialization: $p \leftarrow 0$, $\boldsymbol{k}^{(0)} \leftarrow \mathbf{0}$, $\boldsymbol{s}^{(0)} \leftarrow \mathbf{0}$.
while $\boldsymbol{X}\boldsymbol{k}^{(p)} \neq \boldsymbol{y}$ **do**
  $\boldsymbol{u} \leftarrow \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}^{(p)})$
  $I_1 \leftarrow \{i \in \{1,\ldots,n\} \mid k_i^{(p)} > 0\}$
  $I_2 \leftarrow \{i \in \{1,\ldots,n\} \mid k_i^{(p)} = 0\}$
  $I_3 \leftarrow \{i \in \{1,\ldots,n\} \mid k_i^{(p)} < 0\}$
  for $i \in I_2$ **do**
    if $u_i > 0$, **then** $\delta_i \leftarrow (t_i^+ - s_i^{(p)})/u_i$
    else if $u_i < 0$, **then** $\delta_i \leftarrow (t_i^- + s_i^{(p)})/|u_i|$
    else $\delta_i \leftarrow +\infty$
    end if
  end for
  $j \leftarrow \operatorname{argmin}_{i\in I_2} \delta_i$
  $\boldsymbol{s}^{(p+1)} \leftarrow \boldsymbol{s}^{(p)} + \delta_j \boldsymbol{u}$
  Find $\boldsymbol{k}^{(p+1)}$ as the solution to:

$$\begin{aligned} \min_{\boldsymbol{k}} \quad & \|\boldsymbol{X}\boldsymbol{k} - \boldsymbol{y}\|_2^2 \\ \text{s.t.} \quad & k_i \ge 0, \quad \forall i \in I_1 \\ & k_i \le 0, \quad \forall i \in I_3 \\ & k_i = 0, \quad \forall i \in I_2 \setminus \{j\} \end{aligned}$$

  $p \leftarrow p + 1$
end while
return $\boldsymbol{k}^{(p)}$

[1] See Lemma A.2 and Definition A.3 for detailed explanations. This assumption primarily fails if $k_i(\boldsymbol{\theta}_i)$ is globally bounded from above (i.e., $k_i(\boldsymbol{\theta}_i) < M$ for all $\boldsymbol{\theta}_i$) or below.

Remark 4.3. Algorithm 1 and Theorem 4.2 are straightforward generalizations of the corresponding result in Pesme & Flammarion (2023), where the authors assume symmetric growth times, i.e., $t_i^- = t_i^+ = 1$ for all $i \in [n]$.

4.2. Dynamics of diagonal linear network

Assumption 4.4 (Assumptions and Notation for Theorem 4.5)**. Shared Notation:** We define the loss function as

$$L(\boldsymbol{\theta}) = \|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}) - \boldsymbol{y}\|_2^2.$$

Let $\boldsymbol{\theta}(t)$ evolve according to the gradient flow dynamics:

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\nabla L(\boldsymbol{\theta}), \quad \boldsymbol{\theta}(0) = \boldsymbol{\theta}_0,$$

and denote the solution by $\phi(\boldsymbol{\theta}_0, t)$.

We assume throughout that the conditions of Theorem 4.2 are satisfied. In particular, by Theorem 4.2, Algorithm 1, when applied to input $(\boldsymbol{X}, \boldsymbol{y}, \{t_i^+\}_{i=1}^n, \{t_i^-\}_{i=1}^n)$, terminates after a finite number of iterations. Let $\{\boldsymbol{k}^{(p)}\}_{p=0}^{p_{\max}}$ denote the sequence of vectors generated at each iteration.

Assumptions and Notation for Deep Diagonal Linear Network: Consider an L-layer diagonal linear network with data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ and target vector $\boldsymbol{y} \in \mathbb{R}^m$. Assume $L \geq 3$. Fix the initialization direction

$$\boldsymbol{\theta}^* = (a_{i,1}^*, \ldots, a_{i,L}^*)_{i=1}^n \in \mathbb{R}^{Ln}.$$

For each $i \in [n]$, assume that the minimizer

$$\mathrm{argmin}_{k\in[L]}(a_{i,k}^*)^2$$

is unique. Without loss of generality, we assume that this minimum is attained at $k = L$.

Define the following quantities:

- For $i \in [n]$, $j \in [L-1]$, define
$$\mu_{i,j}^2 = (a_{i,j}^*)^2 - (a_{i,L}^*)^2.$$
- For each $i \in [n]$, define the function
$$F_i(z) = \int_{a_{i,L}^*}^z \frac{1}{\prod_{j=1}^{L-1}\sqrt{x^2+\mu_{i,j}^2}}\,\mathrm{d}x.$$
- Define
$$t_i^+ = F_i(+\infty), \quad t_i^- = -F_i(-\infty).$$

Assumptions and Notation for the General Two-Layer Diagonal Linear Network: For each $i \in [n]$, let λ_i^+ and λ_i^- denote the largest and smallest eigenvalues, respectively, of the Hessian $\nabla^2 k_i(\boldsymbol{0})$. Define

$$t_i^+ = \frac{1}{\lambda_i^+}, \quad t_i^- = -\frac{1}{\lambda_i^-}.$$

Fix an initialization direction $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_i^*)_{i=1}^n$. Assume that for each $i \in [n]$, the vector $\boldsymbol{\theta}_i^*$ is not orthogonal to the eigenvectors corresponding to λ_i^+ and λ_i^-.

Theorem 4.5 (Dynamics of Deep Diagonal Linear Networks)**.** *Consider either a deep diagonal linear network or a general two-layer diagonal linear network. Suppose Assumption 4.4 holds, and let Γ^s denote the trajectory of gradient flow initialized at $s\boldsymbol{\theta}^*$. Then the following statements hold:*

- *For each $p = 0, 1, \ldots, p_{\max}$, there exists a point $\boldsymbol{\theta}^s \in \Gamma^s$ such that*
$$\lim_{s\to 0} \boldsymbol{k}(\boldsymbol{\theta}^s) = \boldsymbol{k}^{(p)}.$$
- *The iterated limit*
$$\boldsymbol{\theta}' = \lim_{s\to 0}\lim_{t\to+\infty} \phi(s\boldsymbol{\theta}^*, t)$$
exists, and satisfies
$$\boldsymbol{k}(\boldsymbol{\theta}') = \boldsymbol{k}^{(p_{\max})}.$$

Proof. The proof is provided in Appendix A.2. □

Corollary 4.6 (Implicit Bias of Diagonal Linear Networks)**.** *Under Assumption 4.4, the implicit bias of a deep diagonal linear network or a general two-layer diagonal linear network with infinitesimal initialization is given by*

$$R(\boldsymbol{k}) = \sum_{i=1}^n \left(t_i^+ k_i^+ + t_i^- k_i^-\right).$$

Proof. This is a direct corollary of Theorem 4.2 and Theorem 4.5. □

Remark 4.7. We provide a detailed comparison of Theorem 4.5 and Corollary 4.6 with the existing literature in Section 1.1.

5. Intuition of the Dynamics

In this section, we use a three-layer diagonal linear network as an illustrative example. The proof of Theorem 4.5 serves as a rigorous formalization of the intuition developed in this section.

Specifically, we analyze the gradient flow dynamics of the model $F(\boldsymbol{\theta})(\boldsymbol{x}) = \sum_{i=1}^n a_i b_i c_i x_i$ under a squared error

loss $L(\boldsymbol{\theta})$. For a single neuron (dropping the index i), the dynamics for its parameters (a, b, c) are given by:

$$\frac{\mathrm{d}a}{\mathrm{d}t} = bc \cdot v, \qquad \frac{\mathrm{d}b}{\mathrm{d}t} = ac \cdot v, \qquad \frac{\mathrm{d}c}{\mathrm{d}t} = ab \cdot v,$$

where $v = \sum_{k=1}^{m} \boldsymbol{X}_k(y_k - F(\boldsymbol{\theta})(\boldsymbol{X}_k))$ is a term related to the correlation between the neuron's feature and the prediction error.

5.1. The Feature Selection Phase and Distance-Speed Argument

In the early phase of training, with infinitesimally small initial parameters $\boldsymbol{\theta}(0)$, the model output $F(\boldsymbol{\theta})(\boldsymbol{x})$ is negligible. Consequently, $v(t)$ remains approximately constant, $v(t) \approx v_0 = \sum_{k=1}^{m} \boldsymbol{X}_k y_k$.

This constant, v_0, can be interpreted as the **speed** of the neuron's evolution. By rescaling time as $\tau = v_0 t$, we can isolate the geometry of the parameter trajectory:

$$\frac{\mathrm{d}a}{\mathrm{d}\tau} = bc, \qquad \frac{\mathrm{d}b}{\mathrm{d}\tau} = ac, \qquad \frac{\mathrm{d}c}{\mathrm{d}\tau} = ab.$$

These dynamics possess two conserved quantities:

$$a^2(\tau) - c^2(\tau) = a^2(0) - c^2(0) := \mu_1^2$$
$$b^2(\tau) - c^2(\tau) = b^2(0) - c^2(0) := \mu_2^2$$

Without loss of generality, let us assume $a(0)^2 > b(0)^2 > c(0)^2 > 0$ and $a(0)b(0) > 0$. Substituting the conserved quantities into the dynamic for c yields:

$$\frac{\mathrm{d}c}{\mathrm{d}\tau} = \sqrt{c^2 + \mu_1^2}\sqrt{c^2 + \mu_2^2}.$$

Now, consider an initialization of scale $s \ll 1$: $a(0) = a_0 s, b(0) = b_0 s, c(0) = c_0 s$. The conserved quantities become $\mu_1^2 = (a_0^2 - c_0^2)s^2 := \tilde{\mu}_1^2 s^2$ and $\mu_2^2 = (b_0^2 - c_0^2)s^2 := \tilde{\mu}_2^2 s^2$. By defining a scaled parameter $\tilde{c}(\tau) = c(\tau)/s$, its dynamic is:

$$\frac{\mathrm{d}\tilde{c}}{\mathrm{d}\tau} = s\sqrt{\tilde{c}^2 + \tilde{\mu}_1^2}\sqrt{\tilde{c}^2 + \tilde{\mu}_2^2}, \quad \text{with} \quad \tilde{c}(0) = c_0.$$

A neuron becomes "significant" when its parameters grow from $\mathcal{O}(s)$ to $\mathcal{O}(1)$. In the scaled frame, this corresponds to $\tilde{c}$ growing from c_0 to $\pm\infty$. The time required for this growth (the blow-up time) can be found by separating variables. For instance, the time to grow to positive significance is:

$$\tau^+ = \int_{c_0}^{\infty} \frac{\mathrm{d}\tilde{c}}{s\sqrt{\tilde{c}^2 + \tilde{\mu}_1^2}\sqrt{\tilde{c}^2 + \tilde{\mu}_2^2}} = \frac{1}{s}\int_{c_0}^{\infty} \frac{\mathrm{d}x}{\sqrt{x^2 + \tilde{\mu}_1^2}\sqrt{x^2 + \tilde{\mu}_2^2}}.$$

This structure motivates the core intuition. We define the **distance** a neuron must travel as the integral, which depends only on the initial *direction* (a_0, b_0, c_0):

$$D^+ = \int_{c_0}^{\infty} \frac{\mathrm{d}x}{\sqrt{x^2 + \tilde{\mu}_1^2}\sqrt{x^2 + \tilde{\mu}_2^2}}.$$

The **time** to grow to significance in the original time t is obtained by un-scaling τ^+:

$$t^+ = \frac{\tau^+}{|v_0|} = \frac{1}{|v_0|s} D^+. \tag{4}$$

Distance-Speed Argument: Equation (4) reveals that the time for a neuron to become significant is proportional to a "distance" D^+ (determined by its initialization) and inversely proportional to its "speed" $|v_0|$ (determined by its feature's correlation with the target). The neurons with **shortest time** will be $\mathcal{O}(1)$ first, and therefore it will be chosen as the grown feature.

5.2. The Learning Phase and Sign-Locking of Features

Learning Phase: Once a feature, say $k_j = a_j b_j c_j$, has grown to a significant magnitude at time $t_{\min}$, the dynamics of the system change. The approximation that the learned vector $\boldsymbol{k}$ is near zero no longer holds. The system now enters a "learning phase" where the gradient descent dynamics actively adjust the significant features to minimize the loss function $L(\boldsymbol{\theta})$.

Mismatch of Time Scales of Two Phases: There exists a mismatch in the time scales of the feature selection and learning phases. Specifically, the feature selection phase persists for a duration on the order of $\mathcal{O}\left(\frac{1}{s}\right)$, whereas the duration of the learning phase is independent of s. Consequently, when s is sufficiently small, the time spent in the learning phase becomes negligible in comparison to that of the feature selection phase. This implies that, during training, no new features are developed away from zero.

Sign-Locking: The key insight is that once $k_j(t)$ has become significantly non-zero, its sign is effectively locked for a very long time. Consider the case where $k_j(t)$ has grown to be positive, meaning $a_j(t), b_j(t), c_j(t)$ are all large. Since we have conserved quantities

$$a_j^2(\tau) - c_j^2(\tau) = a_j^2(0) - c_j^2(0)$$
$$b_j^2(\tau) - c_j^2(\tau) = b_j^2(0) - c_j^2(0),$$

therefore, the parameter $\boldsymbol{\theta}_i(\tau) := (a_j(\tau), b_j(\tau), c_j(\tau))$ is **restricted to a curve** during training. As a consequence, for $k_j(t)$ to flip its sign and become negative, $a_j(t), b_j(t), c_j(t)$ would first have to decrease back to its initialization, which is $\mathcal{O}(s)$ scale. So the time for $k_j(t)$ to flip its sign and become negative is of order $\mathcal{O}(\frac{1}{s})$. Since we assume s is a sufficiently small number representing the scale of initialization, this required time is enormous.

This reveals a mismatch in time scales during the learning dynamics. Consider the second learning phase. Let k_j and k_l denote the feature chosen at the first and second feature selection phase, respectively. We compare the following two time scales:

- The time required for k_j to flip its sign is of order $\frac{1}{s}$, which diverges to infinity as $s \to 0$.
- The time required for k_j and k_l to reach a minimum of the loss function $L(\boldsymbol{\theta})$ without changing their signs is approximately independent of s, since both coordinates have already moved away from zero.

Consequently, when s is sufficiently small, k_j and k_l reach the minimum long before k_j is able to flip its sign. This results in a learning phase that can be effectively modeled as a sign-constrained optimization problem. For instance, if k_j becomes positive during the first learning phase, then the second learning phase can be described by the following constrained optimization:

$$\min_{k_j, k_l} \|\boldsymbol{X}_{:,j}k_j + \boldsymbol{X}_{:,l}k_l - \boldsymbol{y}\|_2^2 \quad \text{subject to} \quad k_j \geq 0.$$

5.3. Comprehensive Dynamics

Inter-Phase Continuity of Feature Selection: Although certain neurons are not selected during a given feature selection phase, they still undergo non-trivial movement and settle at specific positions within the interval $(-t^-, t^+)$, as characterized by Equation (4). Given that the time scale of the subsequent feature learning phase is negligible, it does not significantly affect the final positions of these neurons. As a result, in the next feature selection phase, each non-activated neuron resumes its dynamics from the position reached at the end of the previous selection phase.

Feature Selection Drives Subsequent Learning: The training process alternates between feature selection and feature learning phases, each exerting influence on the other. During a feature selection phase, a new feature is typically activated. Consequently, the subsequent learning phase jointly optimizes this newly selected feature along with those previously selected.

Learning Shapes Future Selection Dynamics: Conversely, each learning phase modifies the correlation vector $\boldsymbol{v} = \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}^{(p)})$, which governs the "speed" of neurons in the ensuing feature selection phase. As a result, the selection behavior in each feature selection phase is inherently influenced by the convergence point of the preceding learning phase.

Comprehensive Dynamics: The training process proceeds iteratively through alternating phases. At each iteration p, the following steps are performed:

1. **Feature Selection Phase:** A new neuron j_p is selected from the inactive set based on the minimal activation time, computed as the ratio of "distance" to "speed". The distance corresponds to the remaining gap inherited from the end of the previous selection phase, while the speed is determined by the correlation vector $\boldsymbol{v} = \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}^{(p)})$.
2. **Learning Phase:** The current active set of neurons $\mathcal{A}_p = \{j_1, \ldots, j_p\}$ jointly minimizes the loss function, subject to individual sign constraints on their corresponding coefficients.
3. **Termination:** The procedure repeats until convergence, i.e., the gradient flow converges to its global minimum.

This framework of sequential feature activation and joint optimization defines the overall training trajectory and is exactly Algorithm 1.

6. Mechanism Behind the Dynamics

In Section 5, we used a three-layer diagonal linear network as an illustrative example to explain the validity of Theorem 4.5. In this section, we aim to uncover a more general underlying mechanism that helps us understand the dynamics of diagonal linear networks.

6.1. Structural Invariant Manifold (SIM)

Definition 6.1 (Structural Invariant Manifold (SIM) (Zhao et al., 2026))**.** Let $F(\boldsymbol{\theta})(\boldsymbol{x}), \boldsymbol{\theta} \in \mathbb{R}^M, \boldsymbol{x} \in \mathbb{R}^d$ be an analytic parametric model. For an immersed submanifold $\mathcal{M} \subset \mathbb{R}^M$, we say $\mathcal{M}$ is a **structural invariant manifold** if it is invariant under $-\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ in Equation (1) for any real analytic loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and dataset S[2].

Definition 6.2 (orbit, page 33 of Jurdjevic (1997))**.** Let $\mathcal{F}$ be a family of analytic vector fields on an analytic manifold $\mathcal{M}$. Let $G = G(\mathcal{F})$ be the group (pseudogroup) of diffeomorphisms (local diffeomorphisms) generated by $\{e^{tX} \mid t \in \mathbb{R}, X \in \mathcal{F}\}$ under composition. For any $\boldsymbol{\theta} \in \mathcal{M}$, we define the **orbit** of $\mathcal{F}$ through $\boldsymbol{\theta}$ as $\{g(\boldsymbol{\theta}) \mid g \in G\}$, which we denote by $O_{\mathcal{F}}(\boldsymbol{\theta})$. [3]

Theorem 6.3 (SIMs of F are orbit unions of $\mathcal{F}$ (Zhao et al., 2026))**.** *Let $F(\boldsymbol{\theta})(\boldsymbol{x}), \boldsymbol{\theta} \in \mathbb{R}^M, \boldsymbol{x} \in \mathbb{R}^d$ be an analytic parametric model. Let $\mathcal{F} = \left\{\nabla_{\boldsymbol{\theta}} F(\cdot)(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathbb{R}^d\right\}$. Then*

- *Each SIM of F is union of orbits of $\mathcal{F}$.*
- *Each orbit of $\mathcal{F}$ is an SIM.*

[2] See Definition B.3 for the definition of invariant manifold.

[3] There is a detailed explanation of orbit and its properties in chapter 2 of Jurdjevic (1997).

Remark 6.4. Theorem 6.3 implies that orbit is the "smallest unit" of SIM.

Theorem 6.5 (SIM of diagonal linear network)**.** *Consider the model $F(\boldsymbol{\theta})(\boldsymbol{x}) = \sum_{i=1}^{n} k_i(\boldsymbol{\theta}_i)x_i, \boldsymbol{\theta} = (\boldsymbol{\theta}_i)_{i=1}^{n}, \boldsymbol{x} = (x_i)_{i=1}^{n}$. Assume that for all $i \in [n]$, the function $k_i(\boldsymbol{\theta}_i)$ is real analytic, and has a unique critical point at $\mathbf{0}$. Define*

- $\mathcal{F} = \left\{\nabla_{\boldsymbol{\theta}} F(\cdot)(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathbb{R}^d\right\}$
- $\mathcal{F}_i = \{\nabla k_i(\cdot)\}, i \in [n]$.

Then for each $\boldsymbol{\theta} = (\boldsymbol{\theta}_i)_{i=1}^{n}$, we have

$$O_{\mathcal{F}}(\boldsymbol{\theta}) = \prod_{i=1}^{n} O_{\mathcal{F}_i}(\boldsymbol{\theta}_i).$$

Besides, for each $i \in [n]$, the following holds:

- *If $\boldsymbol{\theta}_i = \mathbf{0}$, then $O_{\mathcal{F}_i}(\boldsymbol{\theta}_i) = \{\mathbf{0}\}$.*
- *If $\boldsymbol{\theta}_i = \mathbf{0}$, then $O_{\mathcal{F}_i}(\boldsymbol{\theta}_i)$ is a simple curve*[4]*.*

Proof. By calculation, we have

$$\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})(\boldsymbol{x}) = (\nabla k_i(\boldsymbol{\theta}_i)x_i)_{i=1}^{n}.$$

So we have

$$\mathcal{F} = \prod_{i=1}^{n} \mathcal{F}_i.$$

Therefore, it holds that

$$O_{\mathcal{F}}(\boldsymbol{\theta}) = \prod_{i=1}^{n} O_{\mathcal{F}_i}(\boldsymbol{\theta}_i).$$

If $\boldsymbol{\theta}_i = 0$, then $\nabla k_i(\boldsymbol{\theta}_i) = \mathbf{0}$. So $O_{\mathcal{F}_i}(\boldsymbol{\theta}_i) = \{\mathbf{0}\}$. If $\boldsymbol{\theta}_i \neq 0$, by assumption, $\nabla k_i(\boldsymbol{\theta}_i) \neq 0$. By Hermann–Nagano Theorem (Theorem 6 in Section 2 of Jurdjevic (1997)), $O_{\mathcal{F}_i}(\boldsymbol{\theta}_i)$ is a real analytic submanifold, and $\dim(O_{\mathcal{F}_i}(\boldsymbol{\theta}_i)) = 1$.

It is readily to verify that, if $\boldsymbol{\theta}_i \neq \mathbf{0}$, on $O_{\mathcal{F}_i}(\boldsymbol{\theta}_i)$, the value of $k_i(\cdot)$ is strictly increasing. So $O_{\mathcal{F}_i}(\boldsymbol{\theta}_i)$ is non-self intersecting.

□

6.2. Implications of SIM in Dynamics

SIM Induces Two-Phase Dynamics: In Theorem 6.5, we show that for each $i \in [n]$, if the initial parameter satisfies $\boldsymbol{\theta}_i = \mathbf{0}$, then $O_{\mathcal{F}_i}(\boldsymbol{\theta}_i) = \{\mathbf{0}\}$. This invariance implies that parameters initialized exactly at zero remain stationary under gradient flow indefinitely. In our setting, however, parameters are initialized infinitesimally close to zero rather than exactly at zero. Then such parameters require an infinite amount of time to move significantly away from the origin. Consequently, the learning dynamics exhibit a natural separation into two distinct phases: an initial feature selection phase, during which parameters remain near zero for an extended period, followed by a learning phase that unfolds over a finite time scale.

SIM Induces Sign Constraint During the Learning Phase: By Theorem 6.5, under gradient flow dynamics, if the initialization $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_i^*)_{i=1}^{n}$ satisfies $\boldsymbol{\theta}_i^* \neq \mathbf{0}$, then for each $i \in [n]$, the trajectory $\boldsymbol{\theta}_i(t)$ evolves along a simple curve for all $t \in \mathbb{R}$. Suppose, for contradiction, that $\boldsymbol{\theta}_i(t)$ undergoes a sign change during training; that is, there exist times $t_1 < t_2$ such that $\boldsymbol{\theta}_i(t_1) > \delta > 0$ and $\boldsymbol{\theta}_i(t_2) < \delta' < 0$. Since $\boldsymbol{\theta}_i(t)$ evolves along a continuous simple curve, the intermediate value theorem implies the existence of some $\hat{t} \in (t_1, t_2)$ such that $\boldsymbol{\theta}_i(\hat{t}) = \boldsymbol{\theta}_i^*$.

Now, if the initialization $\boldsymbol{\theta}_i^*$ is chosen to be infinitesimally small, then reaching this value along the trajectory requires an infinite amount of time. However, the learning phase proceeds over a finite time horizon. As a result, such a sign change cannot occur within the learning phase. This implies that, due to the structure imposed by SIM, the sign of each $\boldsymbol{\theta}_i(t)$ is effectively preserved throughout the learning phase.

6.3. Mechanism of Incremental Learning

In this context, *incremental learning* refers to the phenomenon whereby *only one neuron* is selected during the feature selection phase.

Consider model of the form $F(\boldsymbol{\theta})(\boldsymbol{x}) = \sum_{i=1}^{n} k_i(\boldsymbol{\theta}_i)x_i$. For each $i \in [n]$, the gradient flow dynamics of the parameter $\boldsymbol{\theta}_i$ are given by:

$$\frac{\mathrm{d}\boldsymbol{\theta}_i}{\mathrm{d}t} = \nabla k_i(\boldsymbol{\theta}_i)\, u_i(\boldsymbol{\theta}), \tag{5}$$

where $\boldsymbol{u}(\boldsymbol{\theta}) = \boldsymbol{X}^{\intercal}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}))$. During the feature selection phase, the change in $\boldsymbol{\theta}$ is small due to the scale of initialization. Consequently, $u_i(\boldsymbol{\theta})$ remains approximately constant and can be approximated by a fixed value u_i^*. Therefore, the dynamics simplify to:

$$\frac{\mathrm{d}\boldsymbol{\theta}_i}{\mathrm{d}t} \approx \nabla k_i(\boldsymbol{\theta}_i)\, u_i^*. \tag{6}$$

Under this approximation, u_i^* can be interpreted as the effective "speed" at which $\boldsymbol{\theta}_i$ evolves along the curve $O_{\mathcal{F}_i}(\boldsymbol{\theta}_i)$.

As established in Section 5.1 and Lemma A.9, for deep diagonal linear networks and general two-layer diagonal linear networks with an initialization scale s, there exists a scaling function $h(s)$ such that $\lim_{s\to 0} h(s) = +\infty$. This

[4] Here, a simple curve refers to an analytic curve that does not intersect itself.

function governs the temporal dynamics required for a neuron to escape the initialization regime. Specifically, the time required for a neuron to attain an $O(1)$ magnitude in the positive or negative direction is approximately $t_i^+ h(s)$ and $t_i^- h(s)$, respectively. For instance:

- $h(s) = \frac{1}{s}$ for three-layer diagonal linear networks,
- $h(s) = -\log s$ for general diagonal linear networks.

Define the quantity

$$\delta_i := \begin{cases} \frac{t_i^+}{u_i^*}, & \text{if } u_i^* > 0, \\ \frac{t_i^-}{|u_i^*|}, & \text{if } u_i^* < 0. \end{cases}$$

δ_i represents the effective time required for neuron i to $O(1)$, taking into account both its growth direction and speed. The neuron with the smallest δ_i will be the one that activates first and is thus selected during the feature selection phase. Under generic conditions on the data $\boldsymbol{X}$ and $\boldsymbol{y}$, the minimum of δ_i is attained uniquely, implying that only one neuron is selected in this phase.

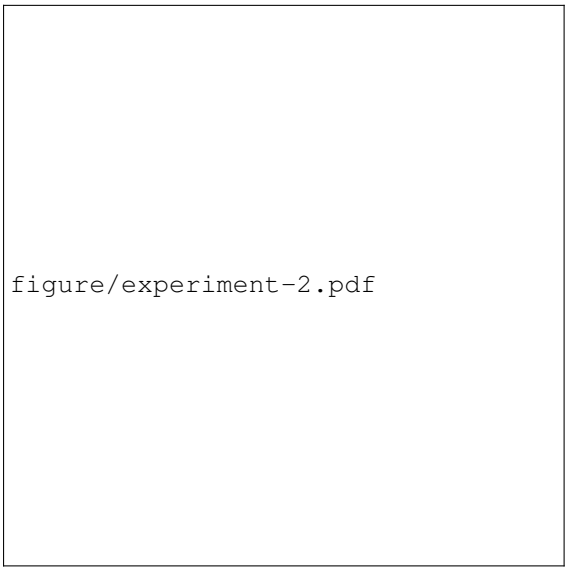


Figure 1. The model is trained via gradient descent. The initialization scale and learning rate are set to 10^{-60} and 0.1, respectively. We utilize the *mpmath* library to support high-precision computations. In the second figure, the boxes indicate the values of k_i at the positions of the vertical dashed lines.

7. Experiments

We consider the model $F(\boldsymbol{\theta})(\boldsymbol{x}) = \sum_{i=1}^4 k_i(\boldsymbol{\theta}_i) x_i$, where

$$k_i(\boldsymbol{\theta}_i) = 2a_i b_i + (a_i^2 + b_i^2 + c_i^2) c_i \,, i = 1, 2;$$
$$k_i(\boldsymbol{\theta}_i) = a_i \tanh(a_i) - (e^{b_i} - 1)^2 \,, i = 3, 4.$$

One can readily verify that this model satisfies Definition 4.1. Ideally, to verify Theorem 4.5, we would require infinitesimal initialization and gradient flow, which are impractical to implement in experimental settings. As a practical alternative, we initialize the model with a sufficiently small scale of 10^{-60}, and employ a relatively large learning rate of 0.1 to accelerate the training of gradient descent.

The data matrix and label vector are given by:

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0.5 & 0.7 & 0 \\ 0.5 & 1 & 0.1 & 0.7 \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{7}$$

As illustrated in Figure 1, the training dynamics demonstrate transitions between successive saddle points. The three dashed lines in the second panel indicate the values of $k_i(\cdot)$ following each learning phase. In Appendix C.1, we compute the values of $\boldsymbol{k}^{(p)}$ in Algorithm 1 for $p = 1, 2, 3$, and confirm that these computed values closely correspond to those represented by the dashed lines in the figure.

8. Limitations and Discussion

8.1. Limitations

Infinitesimal Initialization: Our analysis is confined to the regime of infinitesimal initialization. While this setting facilitates theoretical tractability, it presents two notable drawbacks: (i) Infinitesimal initialization is not practical in real-world applications, where parameters are typically initialized with finite variance; (ii) Prior works on implicit bias, such as Woodworth et al. (2020), is able to characterize the implicit bias under all initialization scales.

Gradient Flow Assumption: The theoretical analysis in this paper is conducted under the assumption of continuous-time gradient flow dynamics. However, in practice, optimization is performed using discrete-time algorithms such as gradient descent or adaptive methods like Adam. The extent to which our results extend to these more realistic optimization settings remains an open question.

8.2. Discussion

Technique: We highlight that the technique employed in this work relies on the concept of the Structural Invariant Manifold, which generally arises in nonlinear models such as matrix sensing and neural networks (Zhao et al., 2026; Simsek et al., 2021; Liu, 2024; Marcotte et al., 2023). The potential outcomes of this study may be extendable to more complex models such as matrix sensing. Investigating this generalization is an interesting direction for future research.

Similarity and difference between ℓ_1 and modified ℓ_1 minimization: Both the standard and modified ℓ_1 norms fundamentally promote sparsity. Their difference is that the modified ℓ_1 norm can alter the recovered support (i.e., which specific features the model chooses to select). To illustrate

this, consider the data matrix presented in Equation (7).

- Standard ℓ_1 minimization yields the solution: $k_3 = 1.43, k_4 = -0.20$, with $k_1 = k_2 = 0$ (feature support: $\{3, 4\}$).
- Modified ℓ_1 minimization (applying a 0.5 penalty weight to $|k_1|$) shifts the optimal solution to: $k_1 = 1.0, k_4 = -0.71$, with $k_2 = k_3 = 0$ (feature support: $\{1, 4\}$).

This demonstrates that though the solution remains sparse, the actual features learned by the network may differ.

Gap between gradient flow (GF) and gradient descent (GD): Whether the SIM remains exactly invariant under GD or SGD depends on its geometric curvature:

- Affine (Flat) SIMs: If a SIM is an affine subspace, it remains strictly invariant under GD/SGD. Because the tangent space is globally constant, discrete steps do not cause the trajectory to leave the manifold. For instance, in a standard diagonal linear network parameterized by $k_i(\boldsymbol{\theta}_i) = a_i b_i$, the affine SIM defined by $\{(\boldsymbol{\theta}_i)_{i=1}^n \mid a_1 = b_1 = 0\}$ is invariant under GD and SGD.
- Curved SIMs: If a SIM is curved, it generally loses its exact invariance under GD/SGD. Taking a finite discrete step along the tangent space of a curved manifold naturally causes the parameter to diverge slightly from the exact manifold. For example, the curved SIM defined by $\{(\boldsymbol{\theta}_i)_{i=1}^n \mid a_1^2 - b_1^2 = 1\}$ is not exactly invariant under GD.

This theoretical gap explains a specific phenomenon observed in the training dynamics of Figure 1 (optimized via GD, learning rate $\eta = 0.1$). Between epochs 3000 and 4000, k_1 changes sign but fails to return to its exact initial scale (10^{-120}). This behavior is directly driven by the loss of exact invariance on the curved SIM caused by the discretization.

9. Conclusion

In this paper, we investigated the training dynamics of deep diagonal linear networks and general two-layer diagonal linear networks. We showed that their training dynamics can be equivalently described by Algorithm 1. Furthermore, we proved that Algorithm 1 converges to the solution of a modified ℓ_1 norm minimization problem. As a result, we established that the implicit bias of both types of diagonal linear networks under infinitesimal initialization corresponds to a modified ℓ_1 norm.

In addition, we analyzed the underlying mechanisms driving these dynamics and identified the Structural Invariant Manifold (SIM) as the key geometric structure governing the learning process.

Acknowledgements

This work was supported by the National Key R&D Program of China (Grant No. 2022YFA1008200), the National Natural Science Foundation of China (Grant No.12571567), the Natural Science Foundation of Shanghai (Grant No. 25ZR1402280).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Azulay, S., Moroshko, E., Nacson, M. S., Woodworth, B. E., Srebro, N., Globerson, A., and Soudry, D. On the implicit bias of initialization shape: Beyond infinitesimal mirror descent. In *International Conference on Machine Learning*, pp. 468–477. PMLR, 2021.

Bai, Z., Zhao, J., and Zhang, Y. Connectivity shapes implicit regularization in matrix factorization models for matrix completion. *Advances in Neural Information Processing Systems*, 37:45914–45955, 2024.

Bantzis, I., Simon, J. B., and Jacot, A. Saddle-to-saddle dynamics in deep relu networks: Low-rank bias in the first saddle escape. *arXiv preprint arXiv:2505.21722*, 2025.

Berthier, R. Incremental learning in diagonal linear networks. *Journal of Machine Learning Research*, 24(171): 1–26, 2023.

Berthier, R. Diagonal linear networks and the lasso regularization path. *arXiv preprint arXiv:2509.18766*, 2025.

Boursier, E., Pillaud-Vivien, L., and Flammarion, N. Gradient flow dynamics of shallow relu networks for square loss and orthogonal inputs. *Advances in Neural Information Processing Systems*, 35:20105–20118, 2022.

Even, M., Pesme, S., Gunasekar, S., and Flammarion, N. (s) gd over diagonal linear networks: Implicit bias, large stepsizes and edge of stability. *Advances in Neural Information Processing Systems*, 36:29406–29448, 2023.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018.

Jacot, A., Ged, F., Şimşek, B., Hongler, C., and Gabriel, F. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. *arXiv preprint arXiv:2106.15933*, 2021.

Jurdjevic, V. *Geometric control theory*. Cambridge university press, 1997.

Labarrière, H., Molinari, C., Rosasco, L., Villa, S., and Vega, C. Optimization insights into deep diagonal linear networks. *arXiv preprint arXiv:2412.16765*, 2024.

Li, Z., Luo, Y., and Lyu, K. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. In *International Conference on Learning Representations*, 2021.

Li, Z., Wang, T., Lee, J. D., and Arora, S. Implicit bias of gradient descent on reparametrized models: On equivalence to mirror descent. *Advances in Neural Information Processing Systems*, 35:34626–34640, 2022.

Liu, Z. Symmetry induces structure and constraint of learning. In *International Conference on Machine Learning*, 2024.

Lojasiewicz, S. Ensembles semi-analytiques. *Lectures Notes IHES (Bures-sur-Yvette)*, 1965.

Marcotte, S., Gribonval, R., and Peyré, G. Abide by the law and follow the flow: Conservation laws for gradient flows. *Advances in Neural Information Processing Systems*, 36: 63210–63221, 2023.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

Nacson, M. S., Ravichandran, K., Srebro, N., and Soudry, D. Implicit bias of the step size in linear diagonal neural networks. In *International Conference on Machine Learning*, pp. 16270–16295. PMLR, 2022.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Papazov, H., Pesme, S., and Flammarion, N. Leveraging continuous time to understand momentum when training diagonal linear networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 3556–3564. PMLR, 2024.

Pesme, S. and Flammarion, N. Saddle-to-saddle dynamics in diagonal linear networks. *Advances in Neural Information Processing Systems*, 36:7475–7505, 2023.

Pesme, S., Pillaud-Vivien, L., and Flammarion, N. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. *Advances in Neural Information Processing Systems*, 34:29218–29230, 2021.

Simsek, B., Ged, F., Jacot, A., Spadaro, F., Hongler, C., Gerstner, W., and Brea, J. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. *International Conference on Machine Learning*, 2021.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19 (70):1–57, 2018.

Vapnik, V. N. and Chervonenkis, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity: festschrift for alexey chervonenkis*, pp. 11–30. Springer, 2015.

Vardi, G. On the implicit bias in deep-learning algorithms. *Communications of the ACM*, 66(6):86–93, 2023.

Woodworth, B., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673. PMLR, 2020.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Zhao, J., Luo, T., and Zhang, Y. Architecture induces structural invariant manifolds of neural network training dynamics. *Mathematical Models and Methods in Applied Sciences*, pp. 1–45, 2026.

Ziyin, L., Wang, M., Li, H., and Wu, L. Parameter symmetry and noise equilibrium of stochastic gradient descent. *Advances in Neural Information Processing Systems*, 37: 93874–93906, 2024.

A. Proof of Theorems

A.1. Proof of Theorem 4.2

We first prove a lemma.

Lemma A.1. *Consider Algorithm 1 under assumptions of Theorem 4.2. Let $J^{(p)}$ be the union of index set $I_1 \cup I_3 \cup \{j\}$ at step p. Then $\{\boldsymbol{X}_{:,j}\}_{j\in J^{(p)}}$ is linearly independent for each p.*

Proof. We prove by induction. For $p = 0$, $J^{(p)} = \{j\}$, where $j = \operatorname{argmin}_{i\in[n]}\delta_i$. Then we have $u_j \neq 0$. Since

$$u_j = (\boldsymbol{X}^\intercal \boldsymbol{y})_j = \langle \boldsymbol{X}_{:,j}, \boldsymbol{y}\rangle,$$

we have $\boldsymbol{X}_{:,j} \neq \boldsymbol{0}$. Therefore the statement holds for $p = 0$. Assume the statement holds for some natural number p. We want to prove it for $p + 1$.

Let j be the neuron chosen at step p. Let $I_1^{(p)} = \{i \in [n] \mid k_i^{(p)} > 0\}$, $I_2^{(p)} = \{i \in [n] \mid k_i^{(p)} = 0\}$, $I_3^{(p)} = \{i \in [n] \mid k_i^{(p)} < 0\}$. By the definition of Algorithm 1, $\boldsymbol{k}^{(p+1)}$ is the solution of the optimization problem

$$\min_{\boldsymbol{k}} \quad \|\boldsymbol{X}\boldsymbol{k} - \boldsymbol{y}\|_2^2 \quad \text{s.t.} \quad k_i \geq 0, \forall i \in I_1^{(p)}, \quad k_i \leq 0, \forall i \in I_3^{(p)}, \quad k_i = 0, \forall i \in I_2^{(p)} \setminus \{j\} \tag{8}$$

Define $I_1^{(p+1)} = \{i \in [n] \mid k_i^{(p+1)} > 0\}$, $I_2^{(p+1)} = \{i \in [n] \mid k_i^{(p+1)} = 0\}$, $I_3^{(p+1)} = \{i \in [n] \mid k_i^{(p+1)} < 0\}$. Define $\boldsymbol{u} = \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}^{(p+1)})$.

Since $\boldsymbol{k}^{(p+1)}$ solves optimization problem 8, therefore for each $i \in I_1^{(p+1)} \cup I_3^{(p+1)}$, we have $u_i = 0$. Let j' be the neuron chosen at step $p + 1$.

If $\boldsymbol{X}_{:,j'}$ is in linear span of $\{\boldsymbol{X}_{:,i}\}_{i\in I_1^{(p+1)}\cup I_3^{(p+1)}}$, then it is readily verifiable that $u_{j'} = 0$. So $\delta_{j'} = +\infty$. This contradicts the fact that j'-th neuron is chosen at step $p + 1$. So $\boldsymbol{X}_{:,j'}$ is not in linear span of $\{\boldsymbol{X}_{:,i}\}_{i\in I_1^{(p+1)}\cup I_3^{(p+1)}}$.

By our assumption, $\{\boldsymbol{X}_{:,j}\}_{j\in J^{(p)}}$ is linearly independent. Since we have $I_1^{(p+1)} \cup I_3^{(p+1)} \subset J^{(p)}$, $\{\boldsymbol{X}_{:,i}\}_{i\in I_1^{(p+1)}\cup I_3^{(p+1)}}$ is linearly independent.

Since $\boldsymbol{X}_{:,j'}$ is not in the linear span of $\{\boldsymbol{X}_{:,i}\}_{i\in I_1^{(p+1)}\cup I_3^{(p+1)}}$, $\{\boldsymbol{X}_{:,j}\}_{j\in J^{(p+1)}}$ is linearly independent. By mathematical induction, the lemma is proved. □

In the following we prove Theorem 4.2.

Proof. **Well-Posedness:** At each step p, by assumption, the index $j = \arg\min_{i\in I_2} \delta_i$ is uniquely defined. So j is well-posed.

Let $J = I_1^{(p)} \cup I_3^{(p)}$. By Lemma A.1, $\{X_{:,j}\}_{j\in J}$ is linearly independent. Recall that $\boldsymbol{k}^{(p)}$ is the solution of

$$\min_{\boldsymbol{k}} \quad \|\boldsymbol{X}\boldsymbol{k} - \boldsymbol{y}\|_2^2 \quad \text{s.t.} \quad k_i \geq 0, \forall i \in I_1^{(p)}, \quad k_i \leq 0, \forall i \in I_3^{(p)}, \quad k_i = 0, \forall i \in I_2^{(p)} \setminus \{j\}. \tag{9}$$

Then $\boldsymbol{k}_J^{(p)}$ is the solution of

$$\min_{\boldsymbol{k}_J} \quad \|\boldsymbol{X}_{:,J}\boldsymbol{k}_J - \boldsymbol{y}\|_2^2 \quad \text{s.t.} \quad k_i \geq 0, \forall i \in I_1^{(p)}, \quad k_i \leq 0, \forall i \in I_3^{(p)}. \tag{10}$$

Since $\{X_{:,j}\}_{j\in J}$ is linearly independent, optimization problem (10) is strictly convex. So the solution of (10) is unique. Therefore, the solution of (9) is unique. So $\boldsymbol{k}^{(p+1)}$ is well defined.

Terminates in finite iterations:

Note that the error $\epsilon^{(p)} = \|\boldsymbol{X}\boldsymbol{k}^{(p)} - \boldsymbol{y}\|_2^2$ strictly decreases over p. Besides, there only exists finite possible ϵ of the minimization problem

$$\min \|\boldsymbol{X}\boldsymbol{k} - \boldsymbol{y}\|_2^2, \quad s.t. \quad \boldsymbol{k}_i \geq 0, i \in I_1; \boldsymbol{k}_i \leq 0, i \in I_3; \boldsymbol{k}_i = 0, i \in I_2 \setminus j$$

for different I_1, I_3, I_2, j. So $\epsilon^{(p)}$ must reach its minimum after finite iterations. Since $\boldsymbol{X}\boldsymbol{k} = \boldsymbol{y}$ has solution, so $\epsilon^{(p)}$ reach 0 after finite iterations.

Convergence to Optimization Problem (11)**:** Consider the optimization probelm

$$\min \sum_{i=1}^{n} \left(t_i^+ k_i^+ + t_i^- k_i^-\right) \quad \text{subject to} \quad \boldsymbol{X}\boldsymbol{k} = \boldsymbol{y}. \tag{11}$$

For $\boldsymbol{k} = (k_1, \ldots, k_n) \in \mathbb{R}^n$, define $\partial(\boldsymbol{k}) = (\partial_i(k_i))_{i=1}^n$, where

$$\partial_i(k_i) = \begin{cases} t_i^+, & \text{if } k_i > 0 \\ -t_i^-, & \text{if } k_i < 0 \\ [-t_i^-,\ t_i^+], & \text{if } k_i = 0 \end{cases}$$

By calculation, the KKT condition of (11) is

$$\mathbf{0} \in \partial(\boldsymbol{k}) + \boldsymbol{X}^\intercal \boldsymbol{\lambda}, \quad \boldsymbol{X}\boldsymbol{k} = \boldsymbol{y}.$$

Since the optimization problem described in Equation (11) is convex, and it has only equal constraint, then the solution of KKT condition is equivalent to solution of original problem.

For each p, let $\boldsymbol{k}^{(p)}$ and $\boldsymbol{s}^{(p)}$ be the value of $\boldsymbol{k}$ and $\boldsymbol{s}$ at step p. Let $\boldsymbol{k}^*$ be the output of the algorithm. We now verify that $\boldsymbol{k}^*$ satisfies the KKT condition.

From the algorithm, one sees that for each p, we have the following:

- If $s_i^{(p)} = t_i^+$, then $k_i^{(p)} \geq 0$.
- If $s_i^{(p)} = -t_i^-$, then $k_i^{(p)} \leq 0$.
- If $s_i^{(p)} \in (-t_i^-, t_i^+)$, then $k_i^{(p)} = 0$.

So $\boldsymbol{s}^{(p)} \in \partial(\boldsymbol{k}^{(p)})$. Besides, by the update of $\boldsymbol{s}^{(p)}$, one sees that $\boldsymbol{s}^{(p)} \in \mathrm{Img}(\boldsymbol{X}^\intercal)$. So there exists $\boldsymbol{\lambda}^{(p)} \in \mathbb{R}^m$ such that $\boldsymbol{s}^{(p)} = \boldsymbol{X}^\intercal(-\boldsymbol{\lambda}^{(p)})$. So

$$\boldsymbol{s}^{(p)} + \boldsymbol{X}^\intercal \boldsymbol{\lambda}^{(p)} = \mathbf{0}, \quad \boldsymbol{s}^{(p)} \in \partial(\boldsymbol{k}^{(p)}).$$

So the condition

$$\mathbf{0} \in \partial(\boldsymbol{k}) + \boldsymbol{X}^\intercal \boldsymbol{\lambda}$$

holds in each step. Moreover, we have $\boldsymbol{X}\boldsymbol{k}^* = \boldsymbol{y}$. So $\boldsymbol{k}^*$ satisfies the KKT condition. So $\boldsymbol{k}^*$ is a solution of (11). □

A.2. Proof of Theorem 4.5

Lemma A.2 (Li et al. (2021))**.** *Assume $g(\boldsymbol{\theta}) : \mathbb{R}^M \to \mathbb{R}$ is real analytic, and $g(\mathbf{0}) = 0, \nabla g(\mathbf{0}) = \mathbf{0}$. Let $\boldsymbol{H} = \nabla^2 g(\mathbf{0})$. Assume $\boldsymbol{H}$ has a unique largest eigenvalue $\lambda_1 > 0$ with corresponding eigenvectors $\boldsymbol{v}_1$. Denote the solution of*

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \nabla g(\boldsymbol{\theta}), \boldsymbol{\theta}(0) = \boldsymbol{\theta}_0$$

as $\phi(\boldsymbol{\theta}_0, t)$. Then for $\boldsymbol{\theta}_1 \in \mathbb{R}^d$ such that $\langle \boldsymbol{\theta}_1, \boldsymbol{v}_1 \rangle > 0$, the limit $h(\boldsymbol{\theta}_1, t) := \lim_{s \to 0} \phi(s\boldsymbol{\theta}_1, t + \frac{1}{\lambda_1} \log \frac{1}{s})$ exists, and $h(\boldsymbol{\theta}_1, t) \neq \mathbf{0}$. Moreover, the trajectory $\Gamma(\boldsymbol{\theta}_1) := \{h(\boldsymbol{\theta}_1, t) \mid t \in \mathbb{R}\}$ is independent of $\boldsymbol{\theta}_1$ as long as $\langle \boldsymbol{\theta}_1, \boldsymbol{v}_1 \rangle > 0$. Besides, the same statements also hold for $\langle \boldsymbol{\theta}_1, \boldsymbol{v}_1 \rangle < 0$.

Proof. The lemma is proved in Li et al. (2021) in their Theorem 5.3. □

Definition A.3 (limit trajectory)**.** In general two-layer diagonal linear network, we apply Lemma A.2 by setting $g(\boldsymbol{\theta}) = k_i(\boldsymbol{\theta}_i)$. We use $\Gamma_i^{++}, \Gamma_i^{+-}$ to denote the $\Gamma(\boldsymbol{v}_1)$ and $\Gamma(-\boldsymbol{v}_1)$ in Lemma A.2, respectively. One can also consider the limit trajectory of

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\nabla k_i(\boldsymbol{\theta}_i), \boldsymbol{\theta}(0) = \boldsymbol{\theta}_0$$

when we assume $\boldsymbol{H}$ has a unique smallest eigenvalue $\lambda_2 < 0$ and corresponding eigenvector $\boldsymbol{v}_2$. We use $\Gamma_i^{-+}, \Gamma_i^{--}$ to denote the $\Gamma(\boldsymbol{v}_2)$ and $\Gamma(-\boldsymbol{v}_2)$. The four trajectories, $\Gamma_i^{++}, \Gamma_i^{+-}, \Gamma_i^{-+}, \Gamma_i^{--}$ are called limit trajectories of neuron i.

Example of limit trajectory: Consider $k_i(\boldsymbol{\theta}) = a^2 - b^2, \boldsymbol{\theta} = (a, b) \in \mathbb{R}^2$. Then $\Gamma_i^{++} = \{(a, b) \mid a > 0, b = 0\}$, $\Gamma_i^{+-} = \{(a, b) \mid a < 0, b = 0\}$, $\Gamma_i^{-+} = \{(a, b) \mid a = 0, b > 0\}$, $\Gamma_i^{--} = \{(a, b) \mid a = 0, b < 0\}$. In this case of $k_i(\boldsymbol{\theta}) = a^2 - b^2$, the limit trajectory can also be derived from the conserved quantity $a(t)b(t) = a(0)b(0)$. Under infinitesimal initialization, we have $a(t)b(t) = 0$. This results in the four limit trajectories.

Lemma A.4 (convergence to limit trajectory under infinitesimal initialization)**.** *Consider the general two-layer diagonal linear network in Definition 4.1. For each $i \in [n]$, let $\boldsymbol{H}_i = \nabla^2 k_i(\boldsymbol{\theta}_i)$. Let $\lambda_i^+, \boldsymbol{v}_i^+$ be the largest eigenvalue and corresponding eigenvector. Let $\lambda_i^-, \boldsymbol{v}_i^-$ be the smallest eigenvalue and corresponding eigenvector. Fix $\boldsymbol{\theta} = (\boldsymbol{\theta}_i)_{i=1}^n$. Assume that $\langle \boldsymbol{\theta}_i, \boldsymbol{v}_i^+ \rangle \neq 0, \langle \boldsymbol{\theta}_i, \boldsymbol{v}_i^- \rangle \neq 0$. Let Γ^s be the trajectory of gradient flow under initialization $s\boldsymbol{\theta}$. If there exists $l \in [n]$, $\mu \neq 0$, $\boldsymbol{\theta}^s = (\boldsymbol{\theta}_i^s)_{i=1}^n \in \Gamma^s$ such that $\lim_{s\to 0} k_l(\boldsymbol{\theta}_l^s) = \mu$, then $\boldsymbol{\theta}_l^s$ converges to a point in $\Gamma^{\mathrm{sign}(\mu), \mathrm{sign}(\langle \boldsymbol{\theta}_i, \boldsymbol{v}_i^{\mathrm{sign}(\mu)} \rangle)}$.*

Proof. For simplicity, we assume $\mu > 0$, $\langle \boldsymbol{\theta}_i, \boldsymbol{v}_i^+ \rangle > 0$. The proofs of other cases are similar. Under this assumption, $\Gamma^{\mathrm{sign}(\mu), \mathrm{sign}(\langle \boldsymbol{\theta}_i, \boldsymbol{v}_i^{\mathrm{sign}(\mu)} \rangle)} = \Gamma^{++}$. By calculation, the gradient flow of general two-layer diagonal linear network is

$$\frac{\mathrm{d}\boldsymbol{\theta}_l^s}{\mathrm{d}t} = \nabla k_l(\boldsymbol{\theta}_l^s) \cdot u(\boldsymbol{\theta}^s), \boldsymbol{\theta}_l^s(0) = s\boldsymbol{\theta}_l.$$

Here, $u(\boldsymbol{\theta}^s)$ is a scalar. Let $\phi(\boldsymbol{\theta}', t)$ be the solution of

$$\frac{\mathrm{d}\boldsymbol{\theta}_l}{\mathrm{d}t} = \nabla k_l(\boldsymbol{\theta}_l), \boldsymbol{\theta}(0) = \boldsymbol{\theta}'. \tag{12}$$

Let $\boldsymbol{\theta}^s = (\boldsymbol{\theta}_i^s)_{i=1}^n \in \Gamma^s$, and assume $k_l(\boldsymbol{\theta}_l^s) = \delta$. Then $\boldsymbol{\theta}_l^s$ is on the trajectory of Equation (12). Define $\boldsymbol{\theta}_{\mathrm{ref}}^s = \phi(s\boldsymbol{\theta}_l, \frac{1}{\lambda_1} \log \frac{1}{s})$. Since $\boldsymbol{\theta}_l^s$ is on the trajectory of Equation (12), there exists $t(s)$ such that $\boldsymbol{\theta}_l^s = \phi(\boldsymbol{\theta}_{\mathrm{ref}}^s, t(s))$. By Lemma A.2, the limit $\boldsymbol{\theta}_{\mathrm{ref}} := \lim_{s\to 0} \boldsymbol{\theta}_{\mathrm{ref}}^s$ exists, and $\boldsymbol{\theta}_{\mathrm{ref}} \neq \mathbf{0}$.

By assumption of general diagonal linear network, on Γ^{++}, the value of $k_l(\cdot)$ is unbounded. It is readily verifiable that on Γ^{++}, $k_l(\cdot)$ is positive and strictly increasing. Moreover, we have $\mu > 0$. Therefore, there exists $\boldsymbol{\theta}_l^* \in \Gamma^{++}$ such that $k_l(\boldsymbol{\theta}_l^*) = \mu$. Since $\boldsymbol{\theta}_l^* \in \Gamma^{++}$, there exists $T \in \mathbb{R}$ such that $\boldsymbol{\theta}_l^* = \phi(\boldsymbol{\theta}_{\mathrm{ref}}, T)$. Since $\boldsymbol{\theta}_{\mathrm{ref}} = \lim_{s\to 0} \boldsymbol{\theta}_{\mathrm{ref}}^s$, we have $\lim_{s\to 0} \phi(\boldsymbol{\theta}_{\mathrm{ref}}^s, T) = \boldsymbol{\theta}_l^*$. Therefore, for any $\epsilon > 0$, there exists $\delta > 0$ such that for all $0 < s < \delta$, we have $\|\phi(\boldsymbol{\theta}_{\mathrm{ref}}^s, T) - \boldsymbol{\theta}_l^*\|_2 < \epsilon$, and $|k_l(\phi(\boldsymbol{\theta}_{\mathrm{ref}}^s, T)) - k_l(\boldsymbol{\theta}_l^*)| < \epsilon$. Denote $\boldsymbol{h}^s := \phi(\boldsymbol{\theta}_{\mathrm{ref}}^s, T)$. Since we have $\lim_{s\to 0} k_l(\boldsymbol{\theta}_l^s) = k_l(\boldsymbol{\theta}_l^*) = \mu$, we may assume that $|k_l(\boldsymbol{\theta}_l^s) - \mu| < \epsilon$. Then we have

$$|k_l(\boldsymbol{\theta}_l^s) - k_l(\boldsymbol{h}^s)| \leq |k_l(\boldsymbol{\theta}_l^s) - \mu| + |k_l(\boldsymbol{h}^s) - \mu| < 2\epsilon.$$

Since

$$\boldsymbol{\theta}_l^s = \phi(\boldsymbol{\theta}_{\mathrm{ref}}^s, t(s)),$$

we have

$$\boldsymbol{\theta}_l^s = \phi(\boldsymbol{h}^s, t(s) - T).$$

By calculation,

$$\left.\frac{\mathrm{d}k_l(\phi(\boldsymbol{\theta}_{\mathrm{ref}}^s, t))}{\mathrm{d}t}\right|_{t=T} = \|\nabla k_l(\boldsymbol{h}^s)\|_2^2 > 0.$$

Since $\|\boldsymbol{h}^s - \boldsymbol{\theta}_l^*\|_2 < \epsilon$, then we may assume

$$\frac{\mathrm{d}k_l(\phi(\boldsymbol{\theta}_{\mathrm{ref}}^s, t))}{\mathrm{d}t} > \frac{1}{2}\|\nabla k_l(\boldsymbol{\theta}_l^*)\|_2^2$$

for t near T.

Since

$$|k_l(\boldsymbol{\theta}^s) - k_l(\boldsymbol{h}^s)| < 2\epsilon,$$

the following holds:

$$|t(s) - T| < \frac{4\epsilon}{\|\nabla k_l(\boldsymbol{\theta}_l^*)\|_2^2}$$

when ϵ is sufficiently small. Therefore

$$\boldsymbol{\theta}_l^s = \phi(\boldsymbol{h}^s, T), t(s) - T) = \phi(\boldsymbol{\theta}_l^* + \mathcal{O}(\epsilon), \mathcal{O}(\epsilon)).$$

Therefore we have

$$\|\boldsymbol{\theta}_l^s - \boldsymbol{\theta}_l^*\|_2 = \mathcal{O}(\epsilon).$$

So $\lim_{s\to 0} \boldsymbol{\theta}_l^s = \boldsymbol{\theta}^*$. The lemma is proved.

□

Lemma A.5 (convergence guaranty if initialized at the limit trajectory)**.** *Consider the general two-layer diagonal linear network in Definition 4.1 with data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{y} \in \mathbb{R}^m$. For each $i \in [n]$, let $\Gamma_i^{++}, \Gamma_i^{+-}, \Gamma_i^{-+}, \Gamma_i^{--}$ be the limit trajectories defined in Definition A.3. Let $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_i^*)_{i=1}^n$ be the initialization. Consider the gradient flow of general two-layer diagonal linear network:*

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\nabla L(\boldsymbol{\theta}), \boldsymbol{\theta}(0) = \boldsymbol{\theta}^*, \tag{13}$$

where $L(\boldsymbol{\theta}) = \|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}) - \boldsymbol{y}\|_2^2$. Let $\boldsymbol{s} = (s_1, \ldots, s_n)$ be a signed vector, i.e. $\boldsymbol{s}_i \in \{+, -\}$ for all $i \in [n]$. We define $\Gamma_i^+ := \Gamma_i^{++} \cup \Gamma_i^{+-}$, and $\Gamma_i^- := \Gamma_i^{-+} \cup \Gamma_i^{--}$. Assume that for each $i \in [n]$, we have $\boldsymbol{\theta}_i^* \in \Gamma^{s_i}$. Assume that $\boldsymbol{X}$ has full column rank. Denote the solution of Equation (13) by $\boldsymbol{\theta}(t)$. Then $\boldsymbol{k}(\boldsymbol{\theta}(t))$ converges to the solution of the following optimization problem:

$$\min_{\boldsymbol{k}\in\mathbb{R}^n} \|\boldsymbol{X}\boldsymbol{k} - \boldsymbol{y}\|_2^2 \quad s.t. \quad k_i \geq 0 \text{ if } s_i = +, \quad k_i \leq 0 \text{ if } s_i = -. \tag{14}$$

Proof. Define $I = \{i \in [n] \mid s_i = +\}$, and $J = [n] \setminus I$. Without loss of generality, we assume $\boldsymbol{\theta}_i^* \in \Gamma^{++}$ for all $i \in I$, and $\boldsymbol{\theta}_i^* \in \Gamma^{-+}$ for all $i \in J$. By Theorem 6.5, $\boldsymbol{\theta}_i(t) \in \Gamma^{++}$ for all $i \in I$, and $\boldsymbol{\theta}_j(t) \in \Gamma^{-+}$ for all $j \in J$. It is readily verifiable that on Γ_i^{++}, the value of k_i is positive. On Γ_i^{-+}, the value of k_i is negative. So $k_i(\boldsymbol{\theta}_i(t)) > 0, \forall i \in I, t \in \mathbb{R}$, and $k_i(\boldsymbol{\theta}_j(t)) < 0, \forall j \in J, t \in \mathbb{R}$.

By assumption, $\boldsymbol{X}$ has full column rank. Therefore, for any $\delta > 0$, the set $\{\boldsymbol{k} \in \mathbb{R}^n \mid \|\boldsymbol{X}\boldsymbol{k} - \boldsymbol{y}\|_2 < \delta\}$ is bounded. Let $\boldsymbol{\theta}(t)$ be the solution of Equation (13). Since the loss of gradient flow is decreasing, we have $\|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}(t)) - \boldsymbol{y}\|_2 \leq \|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}(0)) - \boldsymbol{y}\|_2$. So $\boldsymbol{k}(\boldsymbol{\theta}(t))$ is bounded. So $k_i(\boldsymbol{\theta}_i(t))$ is bounded for each $i \in [n]$. It is readily verifiable that on Γ_i^{++}, $\boldsymbol{k}_i$ is positive and monotonically increasing. Besides, by assumption, on Γ_i^{++}, $k_i(\cdot)$ is unbounded. For all $i \in I$, we have $\boldsymbol{\theta}_i(t) \in \Gamma^{++}$. Therefore, since $k_i(\boldsymbol{\theta}_i(t))$ is bounded, $\boldsymbol{\theta}_i(t)$ is also bounded. Similarly, one may prove that for $j \in J$, $\boldsymbol{\theta}_j(t)$ is bounded. So $\boldsymbol{\theta}(t)$ is bounded.

By assumption, for each $i \in [n]$, $k_i(\boldsymbol{\theta}_i)$ is a real analytic function. So $L(\boldsymbol{\theta})$ is a real analytic function. By standard Łojasiewicz's inequality (Lojasiewicz, 1965) argument, $\boldsymbol{\theta}(t)$ converges to a critical point of $L(\boldsymbol{\theta})$. Let $\boldsymbol{\theta}' = (\boldsymbol{\theta}_i')_{i=1}^n$ be the critical point that $\boldsymbol{\theta}(t)$ converges to. Since $\boldsymbol{\theta}' = \lim_{t\to+\infty} \boldsymbol{\theta}(t)$, we have $k_i(\boldsymbol{\theta}_i'(t)) \geq 0, \forall i \in I$, and $k_i(\boldsymbol{\theta}_j'(t)) \leq 0, \forall j \in J$.

Define $\boldsymbol{u}(\boldsymbol{\theta}) = \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}))$, and $\boldsymbol{u} = \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}'))$. Since $\boldsymbol{\theta}'$ is a critical point of $L(\boldsymbol{\theta})$, so we have $\nabla L(\boldsymbol{\theta}') = 0$. This is equivalent to

$$\nabla k_i(\boldsymbol{\theta}')u_i = \boldsymbol{0}, \forall i \in [n].$$

Since for each $i \in [n]$, $\nabla k_i(\cdot)$ has unique critical point at $\boldsymbol{0}$, so $\nabla L(\boldsymbol{\theta}') = 0$ is equivalent to

$$\boldsymbol{\theta}' = \boldsymbol{0} \text{ or } u_i = 0, \forall i \in [n].$$

Let $K = \{i \in [n] \mid \boldsymbol{\theta}_i' = \boldsymbol{0}\}$. For any $i \in I \cup K$, it is readily to verify that $u_i \leq 0$. Otherwise assume $u_i > 0$. By calculation, $\boldsymbol{\theta}_i$ follows the dynamics of

$$\frac{\mathrm{d}\boldsymbol{\theta}_i}{\mathrm{d}t} = \nabla k_i(\boldsymbol{\theta}_i) \cdot u_i(\boldsymbol{\theta}).$$

So we have

$$\frac{\mathrm{d}k_i(\boldsymbol{\theta}_i)}{\mathrm{d}t} = \|\nabla k_i(\boldsymbol{\theta}_i)\|_2^2 \cdot u_i(\boldsymbol{\theta}).$$

Therefore, for sufficiently large t, the value of $k_i(\boldsymbol{\theta}_i)$ increases monotonically. This contradicts the fact that $k_i(\boldsymbol{\theta}_i(t)) > 0$ and $k_i(\boldsymbol{\theta}_i(t)) \to 0$. So $u_i \leq 0$ for all $i \in I \cup K$. Similarly, we have $u_i \geq 0$ for all $i \in J \cup K$. Together with the condition that $\boldsymbol{\theta}' = \mathbf{0}$ or $u_i = 0, \forall i \in [n]$, it is readily verifiable that $\boldsymbol{k}(\boldsymbol{\theta}')$ satisfies the KKT conditions of optimization problem (14). Moreover, since optimization problem (14) is convex and satisfies Slater's condition, $\boldsymbol{k}(\boldsymbol{\theta}')$ is its solution. □

Lemma A.6. *Let $\boldsymbol{y} \in \mathbb{R}^m$. Let U be a subspace of $\mathbb{R}^m$. Let $\boldsymbol{x}^*$ be the solution of the optimization problem:*

$$\min_{\boldsymbol{x} \in U} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$

Fix $L > 0$. Then for any $\epsilon > 0$, there exists $\delta > 0$, such that if $\boldsymbol{x} \in U$ satisfies $\|\boldsymbol{x} - \boldsymbol{y}'\|_2 < L\delta + \|\boldsymbol{x}^* - \boldsymbol{y}\|_2$ and $\|\boldsymbol{y}' - \boldsymbol{y}\|_2 < \delta$, then $\|\boldsymbol{x} - \boldsymbol{x}^*\|_2 < \epsilon$.

Proof. Let $d = \|\boldsymbol{x}^* - \boldsymbol{y}\|_2$. If $d = 0$, the statement is trivial. So we assume $d > 0$. As $\boldsymbol{x}^*$ is the orthogonal projection of $\boldsymbol{y}$ onto U, the vector $\boldsymbol{x}^* - \boldsymbol{y}$ is orthogonal to the subspace U. Since $\boldsymbol{x} - \boldsymbol{x}^* \in U$, it follows that $\langle \boldsymbol{x} - \boldsymbol{x}^*, \boldsymbol{x}^* - \boldsymbol{y} \rangle = 0$.

By the Pythagorean theorem:

$$\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 = \|(\boldsymbol{x} - \boldsymbol{x}^*) + (\boldsymbol{x}^* - \boldsymbol{y})\|_2^2 = \|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 + \|\boldsymbol{x}^* - \boldsymbol{y}\|_2^2.$$

Rearranging this identity, we get an expression for the term we wish to bound:

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 - \|\boldsymbol{x}^* - \boldsymbol{y}\|_2^2 = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 - d^2.$$

We introduce $\boldsymbol{y}'$ by writing $\boldsymbol{x} - \boldsymbol{y} = (\boldsymbol{x} - \boldsymbol{y}') + (\boldsymbol{y}' - \boldsymbol{y})$. Substituting this gives:

$$\begin{aligned}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 &= \|(\boldsymbol{x} - \boldsymbol{y}') + (\boldsymbol{y}' - \boldsymbol{y})\|_2^2 - d^2 \\ &= \|\boldsymbol{x} - \boldsymbol{y}'\|_2^2 + 2\langle \boldsymbol{x} - \boldsymbol{y}', \boldsymbol{y}' - \boldsymbol{y} \rangle + \|\boldsymbol{y}' - \boldsymbol{y}\|_2^2 - d^2.\end{aligned}$$

Using the given conditions $\|\boldsymbol{x} - \boldsymbol{y}'\|_2 < d$ and $\|\boldsymbol{y}' - \boldsymbol{y}\|_2 < \delta$:

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 < (L\delta + d)^2 + 2\langle \boldsymbol{x} - \boldsymbol{y}', \boldsymbol{y}' - \boldsymbol{y} \rangle + \delta^2 - d^2 = 2\langle \boldsymbol{x} - \boldsymbol{y}', \boldsymbol{y}' - \boldsymbol{y} \rangle + (L^2 + 1)\delta^2 + 2dL\delta.$$

By the Cauchy-Schwarz inequality and the given conditions again:

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 < 2\|\boldsymbol{x} - \boldsymbol{y}'\|_2\|\boldsymbol{y}' - \boldsymbol{y}\|_2 + (L^2 + 1)\delta^2 + 2dL\delta < 2d(L + 1)\delta + (L + 1)^2\delta^2$$

For a given $\epsilon > 0$, we could pick $\delta > 0$ sufficiently small such that $d(L + 1)\delta + (L + 1)^2\delta^2 < \epsilon^2$. For this choice of δ:

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 < \epsilon^2.$$

Taking the square root yields $\|\boldsymbol{x} - \boldsymbol{x}^*\|_2 < \epsilon$. □

Lemma A.7. *Let $\boldsymbol{k}(\boldsymbol{\theta}) = (k_i(\boldsymbol{\theta}_i))_{i=1}^n \in \mathbb{R}^n$, where $\boldsymbol{\theta} = (\boldsymbol{\theta}_i)_{i=1}^n$. Assume that for each $i \in [n]$, $\boldsymbol{k}_i$ is a real analytic activation, and $k_i(\mathbf{0}) = \nabla k_i(\mathbf{0}) = 0$. Let $\boldsymbol{X} \in \mathbb{R}^{m \times n}, \boldsymbol{y} \in \mathbb{R}^m$. Define $L(\boldsymbol{\theta}) = \|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}) - \boldsymbol{y}\|_2^2$. Fix $\boldsymbol{\theta}^*$, and define $\boldsymbol{k}^* = \boldsymbol{k}(\boldsymbol{\theta}^*)$. Assume that $\boldsymbol{k}^*$ solves the following optimization problem for some $I_1, I_2, I_3, \{j\} \subset [n]$,:*

$$\min \|\boldsymbol{X}\boldsymbol{k} - \boldsymbol{y}\|_2^2, \quad s.t. \quad \boldsymbol{k}_i \geq 0, i \in I_1; \boldsymbol{k}_i \leq 0, i \in I_3; \boldsymbol{k}_i = 0, i \in I_2 \setminus \{j\}.$$

Define $I = \{i \in [n] \mid \boldsymbol{k}_i^* = 0\}$. Then for any $\epsilon > 0$, there exists $\delta > 0$ such that the following holds: "For all initialization $\boldsymbol{\theta}'$ such that $\|\boldsymbol{k}(\boldsymbol{\theta}') - \boldsymbol{k}^*\|_2 < \delta$, let $\boldsymbol{\theta}(t)$ be the solution of $\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\nabla L(\boldsymbol{\theta}), \boldsymbol{\theta}(0) = \boldsymbol{\theta}'$. Then $\|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}(t)) - \boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}^*)\|_\infty < \epsilon$ for all $0 \leq t < T$, where $T = \inf_{t \geq 0}\{t \mid \|\boldsymbol{k}_I(\boldsymbol{\theta}(t))\|_\infty > \delta\}$. "

Proof. Define $J = [n] \setminus I$. Define $U = \mathrm{span}(\{\boldsymbol{X}_{:,i} \mid i \in J\})$. Since $\boldsymbol{k}^*$ solves the optimization problem

$$\min \|\boldsymbol{X}\boldsymbol{k} - \boldsymbol{y}\|_2^2, \quad s.t. \quad \boldsymbol{k}_i \geq 0, i \in I_1; \boldsymbol{k}_i \leq 0, i \in I_3; \boldsymbol{k}_i = 0, i \in I_2 \setminus j,$$

So $\boldsymbol{k}_J^*$ must solve the following problem:

$$\min \|\boldsymbol{X}_{:,J}\boldsymbol{k}_J - \boldsymbol{y}\|_2^2.$$

Let $\boldsymbol{z}^* = \boldsymbol{X}_{:,J}\boldsymbol{k}_J \in U$. Then $\boldsymbol{z}$ is the solution of the optimization problem

$$\min_{\boldsymbol{z} \in U} \|\boldsymbol{z} - \boldsymbol{y}\|_2^2.$$

Since the loss of gradient flow is decreasing, so

$$\|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}(t)) - \boldsymbol{y}\|_2 < \|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}') - \boldsymbol{y}\|_2 \leq \|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}') - \boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}^*)\|_2 + \|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}^*) - \boldsymbol{y}\|_2 \leq L\|\boldsymbol{k}(\boldsymbol{\theta}') - \boldsymbol{k}^*\|_2 + \|\boldsymbol{z}^* - \boldsymbol{y}\|_2.$$

Here, L is $\|\boldsymbol{X}\|_2$. By Lemma A.6, for any $\epsilon > 0$, there exists $\delta > 0$ such that if $\boldsymbol{z} \in U$ satisfies $\|\boldsymbol{z} - \boldsymbol{y}'\|_2 < L\delta + \|\boldsymbol{z}^* - \boldsymbol{y}\|_2$ and $\|\boldsymbol{y}' - \boldsymbol{y}\| < \delta$, then $\|\boldsymbol{z} - \boldsymbol{z}^*\| < \epsilon$. Assume $\|\boldsymbol{k}(\boldsymbol{\theta}') - \boldsymbol{k}^*\| < \delta'$ for some $\delta' > 0$ to be determined. Then we have

$$\|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}(t)) - \boldsymbol{y}\|_2 < L\delta' + \|\boldsymbol{z}^* - \boldsymbol{y}\|_2.$$

We may pick $\delta' < \delta$, so that

$$\|\boldsymbol{O}(\boldsymbol{\theta}(t)) - \boldsymbol{y}\|_2 < L\delta + \|\boldsymbol{z}^* - \boldsymbol{y}\|_2.$$

Besides, we have

$$\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}(t)) = \sum_{i \in I} k_i(\boldsymbol{\theta}_i(t))\boldsymbol{X}_{:,i} + \sum_{i \in J} k_i(\boldsymbol{\theta}_i(t))\boldsymbol{X}_{:,i}.$$

We may pick δ' sufficiently small such that

$$\|\boldsymbol{k}_I(\boldsymbol{\theta}(t))\|_2 < \delta' \Longrightarrow \|\sum_{i \in I} k_i(\boldsymbol{\theta}_i(t))\boldsymbol{X}_{:,i}\|_2 < \delta.$$

So as long as $\|\boldsymbol{k}_I(\boldsymbol{\theta}(t))\|_2 < \delta'$, we have

$$\|\sum_{i \in J} k_i(\boldsymbol{\theta}_i(t))\boldsymbol{X}_{:,i} - (\boldsymbol{y} - \sum_{i \in I} k_i(\boldsymbol{\theta}_i(t))\boldsymbol{X}_{:,i})\|_2 < L\delta + \|\boldsymbol{z}^* - \boldsymbol{y}\|_2.$$

Besides, we have $\|\sum_{i \in I} k_i(\boldsymbol{\theta}_i(t))\boldsymbol{X}_{:,i})\|_2 < \delta$, and $\sum_{i \in J} k_i(\boldsymbol{\theta}_i(t))\boldsymbol{X}_{:,i} \in U$. By Lemma A.6, we have

$$\|\sum_{i \in J} k_i(\boldsymbol{\theta}_i(t))\boldsymbol{X}_{:,i} - \boldsymbol{z}^*\|_2 < \epsilon.$$

So we have

$$\|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}(t)) - \boldsymbol{X}\boldsymbol{k}^*\|_2 \leq \|\sum_{i \in J} k_i(\boldsymbol{\theta}_i(t))\boldsymbol{X}_{:,i} - \boldsymbol{z}^*\|_2 + \|\sum_{i \in I} k_i(\boldsymbol{\theta}_i(t))\boldsymbol{X}_{:,i}\|_2 < \epsilon + \delta'.$$

We may pick $\delta' < \epsilon$, then we have

$$\|\boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}(t)) - \boldsymbol{X}\boldsymbol{k}^*\|_2 < 2\epsilon.$$

Since in L_2 norm is equivalent to L_∞ norm, the lemma holds.

□

Definition A.8 (time mapping)**.** Consider the general two-layer diagonal linear network, and let Assumption 4.4 holds. For each $i \in [n]$, let $\boldsymbol{\theta}_i^s(t)$ be the solution of

$$\frac{\mathrm{d}\boldsymbol{\theta}_i}{\mathrm{d}t} = \nabla k_i(\boldsymbol{\theta}_i), \boldsymbol{\theta}_i(0) = s\boldsymbol{\theta}_i^*.$$

Define $\Gamma_i^s = \{\boldsymbol{\theta}_i^s(t) \mid t \in \mathbb{R}\}$. It is readily verifiable that for any $\boldsymbol{\theta}_i \in \Gamma_i(\boldsymbol{\theta}_i^*)$, there exists unique $t \in \mathbb{R}$ such that $\boldsymbol{\theta}_i = \boldsymbol{\theta}_i^s(t)$. The map from $\boldsymbol{\theta}_i$ to $\frac{t}{-\log s}$ is denoted by $\frac{t}{-\log s} = \tau_i^s(\boldsymbol{\theta}_i)$. We call $\tau_i^s(\boldsymbol{\theta}_i)$ as the time mapping of neuron i. The time mapping $\tau_i^s(\boldsymbol{\theta}_i)$ makes sense only if $\boldsymbol{\theta}_i \in \Gamma_i^s$. Besides, we define $\boldsymbol{\tau}^s(\boldsymbol{\theta}) = (\tau_i^s(\boldsymbol{\theta}_i))_{i=1}^n$. We call $\boldsymbol{\tau}^s$ as the time mapping.

Lemma A.9. *Consider the notation and assumptions in Definition A.8. Fix $i \in [n]$. Let $\boldsymbol{H} = \nabla^2 k_i(\mathbf{0})$. Assume $\boldsymbol{H}$ has a unique largest eigenvalue $\lambda_1 > 0$ with corresponding eigenvectors $\boldsymbol{v}_1$. Assume $\boldsymbol{H}$ has a unique smallest eigenvalue $\lambda_2 < 0$ with the corresponding eigenvectors $\boldsymbol{v}_2$. Assume $\langle \boldsymbol{\theta}_i^*, \boldsymbol{v}_1 \rangle, \langle \boldsymbol{\theta}_i^*, \boldsymbol{v}_2 \rangle \neq 0$. Then the following holds:*

- *Fix $0 < \delta < 1$. Pick $\boldsymbol{\theta}_i^s \in \Gamma_i^s(\boldsymbol{\theta}_i^*)$ for each $0 < s < \delta$. Assume there exists $\epsilon > 0$ such that $\frac{1}{\lambda_2} + \epsilon < \tau_i^s(\boldsymbol{\theta}_i^s) < \frac{1}{\lambda_1} - \epsilon$ for all $0 < s < \delta$. Then $\lim_{s\to 0} \boldsymbol{\theta}_i^s = \mathbf{0}$.*
- *Fix $0 < \delta < 1$. Pick $\boldsymbol{\theta}_i^s \in \Gamma_i^s(\boldsymbol{\theta}_i^*)$ for each $0 < s < \delta$. Assume there exists $\epsilon > 0$ such that $\tau_i^s(\boldsymbol{\theta}_i^s) > \frac{1}{\lambda_1} + \epsilon$ for all $0 < s < \delta$. Then $\lim_{s\to 0} k_i(\boldsymbol{\theta}_i^s) = +\infty$.*
- *Fix $0 < \delta < 1$. Pick $\boldsymbol{\theta}_i^s \in \Gamma_i^s(\boldsymbol{\theta}_i^*)$ for each $0 < s < \delta$. Assume there exists $\epsilon > 0$ such that $\tau_i^s(\boldsymbol{\theta}_i^s) < \frac{1}{\lambda_2} - \epsilon$ for all $0 < s < \delta$. Then $\lim_{s\to 0} k_i(\boldsymbol{\theta}_i^s) = -\infty$.*

Proof. This lemma is a direct corollary of Lemma A.2. We leave out the proof. □

Theorem A.10 (Dynamics of General Two-layer Diagonal Linear Network)**.** *Consider the general two-layer diagonal linear network, and let Assumption 4.4 holds. Let Γ^s to be the trajectory of GF initialized at $s\boldsymbol{\theta}^*$. Let $\boldsymbol{\tau}^s$ be the time mapping defined in Definition A.8. Then the following holds:*

- *For each $p = 0, 1, \ldots, p_{\max}$, there exists $\boldsymbol{\theta}^s \in \Gamma^s$ such that $\lim_{s\to 0} \boldsymbol{k}(\boldsymbol{\theta}^s) = \boldsymbol{k}^{(p)}$ and $\lim_{s\to 0} \boldsymbol{\tau}^s(\boldsymbol{\theta}^s) = \boldsymbol{s}^{(p)}$.*
- *The limit $\boldsymbol{\theta}' = \lim_{s\to+\infty} \lim_{t\to+\infty} \phi(s\boldsymbol{\theta}^*, t)$ exists, and $\boldsymbol{k}(\boldsymbol{\theta}') = \boldsymbol{k}^{(p_{\max})}$.*

Proof. For $p = 0$, the statements obviously hold. Assume the statements hold for some natural number p. We want to prove that the statements hold for $p + 1$.

Define $I = \{i \in [n] \mid k_i^{(p)} = 0\}$, $J_1 = \{i \in [n] \mid k_i^{(p)} > 0\}$, , $J_2 = \{i \in [n] \mid k_i^{(p)} < 0\}$. Define $\boldsymbol{u}(\boldsymbol{\theta}) = \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}))$. Let $\boldsymbol{\theta}^s(t)$ be the solution of

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\nabla L(\boldsymbol{\theta}), \boldsymbol{\theta}(0) = \boldsymbol{\theta}^s.$$

By calculation, we have

$$\frac{\mathrm{d}\boldsymbol{\theta}_i^s(t)}{\mathrm{d}t} = \nabla k_i(\boldsymbol{\theta}_i^s(t)) u_i(\boldsymbol{\theta}^s(t)).$$

For each $i \in [n]$, let $T_i^s(t)$ to be the solution of

$$\frac{\mathrm{d}T_i^s(t)}{\mathrm{d}t} = u_i(\boldsymbol{\theta}^s(t)), T_i^s(0) = 0.$$

Then we have

$$\frac{\mathrm{d}\boldsymbol{\theta}_i^s(T_i^s)}{\mathrm{d}T_i^s} = \nabla k_i(\boldsymbol{\theta}_i^s(T_i^s)), \boldsymbol{\theta}_i^s(0) = \boldsymbol{\theta}_i^s.$$

Let $\boldsymbol{\tau}^s$ be the time mapping defined in Definition A.8. By Theorem 6.5, $\tau_i^s(\boldsymbol{\theta}_i^s(t))$ is well defined for each $i \in [n]$. By definition of time mapping, for all $i \in [n]$ and $s > 0$, we have

$$\tau_i^s(\boldsymbol{\theta}_i^s(t)) = \frac{T_i^s(t)}{-\log s} + \tau_i^s(\boldsymbol{\theta}_i^s). \tag{15}$$

Let $\boldsymbol{u}^* := \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}^{(p)})$. By Lemma A.7, for any $\epsilon > 0$, there exists $\delta > 0$ such that for sufficiently large s, we have $\|\boldsymbol{u}(\boldsymbol{\theta}^s(t)) - \boldsymbol{u}^*\|_\infty < \epsilon$, where $T^s = \inf_{t\geq 0}\{t \mid \|\boldsymbol{k}_I(\boldsymbol{\theta}(t))\|_\infty > \delta\}$.

For $0 \leq t < T^s$, we have

$$u_i^* - \epsilon < \frac{\mathrm{d}T_i^s(t)}{\mathrm{d}t} < u_i^* + \epsilon.$$

So for $0 \leq t < T^s$, it holds that

$$(u_i^* - \epsilon)t < T_i^s(t) < (u_i^* + \epsilon)t.$$

By Equation (15) and the assumption that $\lim_{s\to 0}\boldsymbol{\tau}^s(\boldsymbol{\theta}^s)=\boldsymbol{s}^{(p)}$, for sufficiently small s, we have

$$(u_i^*-\epsilon)\frac{t}{-\log s}+(s_i^{(p)}-\epsilon)<\tau_i^s(\boldsymbol{\theta}_i^s(t))<(u_i^*+\epsilon)\frac{t}{-\log s}+(s_i^{(p)}+\epsilon),\quad \forall 0<t<T^s.$$

For simplicity of notation, we assume $u_i^*>0$ for each $i\in I$. The case of $u_i^*<0$ for some i is similar. For each $i\in I$, define $\mu_i(\epsilon)$ to be the interval $[\frac{t_i^+-(s_i^{(p)}+\epsilon)}{u_i^*+\epsilon},\frac{t_i^+-(s_i^{(p)}-\epsilon)}{u_i^*-\epsilon}]$. We use $\mu_i^-(\epsilon)$ and $\mu_i^+(\epsilon)$ to denote the left endpoint and right endpoint of $\mu(\epsilon)$, respectively.

Fix $i\in I, 0<a<\mu_i^-(\epsilon)$. By Lemma A.9, if $0<t^s<\min\{a\log\frac{1}{s},T^s\}$ for each $s>0$, then $\lim_{s\to 0}\boldsymbol{\theta}_i^s(t)=\mathbf{0}$. Besides, if $a>\mu_i^+(\epsilon)$ and $0<t^s<\min\{a\log\frac{1}{s},T^s\}$ for each $s>0$, then $\lim_{s\to 0}k_i(\boldsymbol{\theta}_i^s(t))=+\infty$.

Since $T^s=\inf_{t\geq 0}\{t\mid\|\boldsymbol{k}_I(\boldsymbol{\theta}(t))\|_\infty>\delta\}$, we have $\|\boldsymbol{k}_I(\boldsymbol{\theta}(T^s))\|_\infty=\delta$. Therefore, there exists $l\in I$, such that

$$k_l(\boldsymbol{\theta}_l(T^s))=\pm\delta.$$

Since $u_i^*>0$ for each i, it holds that

$$k_l(\boldsymbol{\theta}_l(T^s))=\delta.$$

For each $i\in I$, we define $\mu_i(0)=(t_i^+-s_i^{(p)})/u_i^*$. Let $j\in\operatorname{argmin}_{i\in I_2}\mu_i(0)$. By assumption, j is unique. Therefore, for sufficiently small ϵ, the intervals $\mu_i(\epsilon), i\in[n]$ has no intersection.

It is readily verifiable that for sufficiently small ϵ, we have $l=j$. Otherwise let $l\neq j$. By Lemma A.9, for any $\hat{\epsilon}>0$, if $T^s/\log\frac{1}{s}<\mu_l^-(\epsilon)-\hat{\epsilon}$ holds for sequence $s_n\to 0$, then $k_l(\boldsymbol{\theta}_l(T^s))\to 0$, which contradicts $k_l(\boldsymbol{\theta}_l(T^s))=\delta>0$. So for sufficiently small s, we have

$$T^s/\log\frac{1}{s}>\mu_l^-(\epsilon)-\hat{\epsilon}.$$

We may pick $\hat{\epsilon}$ sufficiently small such that

$$\mu_l^-(\epsilon)-\hat{\epsilon}>\mu_j^+(\epsilon).$$

Therefore, we have

$$T^s/\log\frac{1}{s}>\mu_l^-(\epsilon)-\hat{\epsilon}>\mu_j^+(\epsilon).$$

By Lemma A.9, we have $k_j(\boldsymbol{\theta}_j^s(T^s))\to+\infty$, which contradicts that $|k_j(\boldsymbol{\theta}_j^s(T^s))|\leq\delta$.

Moreover, we have $T^s/\log\frac{1}{s}\in\mu_j(\epsilon)$. Therefore, for all $i\in I$, it holds that

$$\tau_i^s(\boldsymbol{\theta}_i^s(T^s))=u_i^*\mu_j(0)+s_i^{(p)}+\mathcal{O}(\epsilon).$$

As a consequence, for sufficiently small ϵ, for each $i\in I\setminus\{j\}$, we have

$$\lim_{s\to 0}\boldsymbol{\theta}_i^s(T^s)=\mathbf{0}.$$

By Lemma A.4, $\boldsymbol{\theta}_j^s(T^s)$ converges to one point in Γ^{++} or Γ^{+-}. Define $J=J_1\cup J_2$. For $i\in J$, we already have the control $\|\boldsymbol{X}_{:,J}\boldsymbol{k}_J(\boldsymbol{\theta}_J^s(T^s))-\boldsymbol{X}_{:,J}\boldsymbol{k}_J^{(p)}\|_\infty<\epsilon$. By Lemma A.1, $\boldsymbol{X}_{:,J}$ has full column rank. So we have

$$\|\boldsymbol{k}_J(\boldsymbol{\theta}_J^s(T^s))-\boldsymbol{k}_J^{(p)}\|_\infty<C\epsilon,$$

where C depends only on $\boldsymbol{X}_{:,J}$. By Lemma A.4, for each $i\in J$, the limit $\lim_{s\to 0}\boldsymbol{\theta}_i^s(T^s)$ exists, and $\lim_{s\to 0}\boldsymbol{\theta}_i^s(T^s)$ is on the limit trajectories in Definition A.3.

Therefore, for all $i\in[n]$, the limit $\boldsymbol{\theta}_i^s(T^s)$ exists. So

$$\hat{\boldsymbol{\theta}}_0:=\lim_{s\to 0}\boldsymbol{\theta}^s(T^s)$$

exists. Let $\hat{\boldsymbol{\theta}}(t) = \phi(\hat{\boldsymbol{\theta}}_0, t)$. By Theorem 6.5, for each $i \in I \setminus \{j\}$, for all $t > 0$, we have $\hat{\boldsymbol{\theta}}_i(t) = \mathbf{0}$. Then apply Lemma A.5, the limit $\boldsymbol{k}^* = \lim_{t\to\infty} \boldsymbol{k}(\hat{\boldsymbol{\theta}}(t))$ exists, and solves the optimization problem:

$$\min_{\boldsymbol{k}\in\mathbb{R}^n} \|\boldsymbol{X}\boldsymbol{k} - \boldsymbol{y}\|_2^2 \quad s.t. \quad k_i \geq 0 \text{ if } s_i = +, \quad k_i \leq 0 \text{ if } s_i = -. \tag{16}$$

By definition, $\boldsymbol{k}^* = \boldsymbol{k}^{(p+1)}$. So for any $\epsilon' > 0$, there exists $T > 0$ such that

$$\|\boldsymbol{k}^* - \boldsymbol{k}(\hat{\boldsymbol{\theta}}(T))\|_2 < \epsilon'.$$

Since $\hat{\boldsymbol{\theta}}_0 := \lim_{s\to 0} \boldsymbol{\theta}^s(T^s)$, for sufficiently small s, we have

$$\|\phi(\boldsymbol{\theta}^s(T^s), T) - \hat{\boldsymbol{\theta}}(T)\|_2 < \epsilon',$$

and

$$\|\boldsymbol{k}(\phi(\boldsymbol{\theta}^s(T^s), T)) - \boldsymbol{k}(\hat{\boldsymbol{\theta}}(T))\|_2 < \epsilon'.$$

Let $\boldsymbol{\theta}_s := \phi(\boldsymbol{\theta}^s(T^s), T)$. Then we have

$$\|\boldsymbol{k}(\boldsymbol{\theta}_s) - \boldsymbol{k}^{(p+1)}\|_2 < 2\epsilon'.$$

Therefore, we have

$$\lim_{s\to 0} \boldsymbol{k}(\boldsymbol{\theta}_s) = \boldsymbol{k}^{(p+1)}.$$

By definition,

$$\boldsymbol{\tau}^s(\boldsymbol{\theta}_s) = \boldsymbol{\tau}^s(\boldsymbol{\theta}^s(T^s)) + \frac{T}{-\log s}.$$

Let $s \to 0$, one gets

$$\lim_{s\to 0} \boldsymbol{\tau}^s(\boldsymbol{\theta}_s) = \lim_{s\to 0} \boldsymbol{\tau}^s(\boldsymbol{\theta}^s(T^s)) = \boldsymbol{u}^* \mu_j(0) + \boldsymbol{s}^{(p)}.$$

By definition, $\boldsymbol{u}^* \mu_j(0) + \boldsymbol{s}^{(p)} = \boldsymbol{s}^{(p+1)}$. So

$$\lim_{s\to 0} \boldsymbol{\tau}^s(\boldsymbol{\theta}_s) = \boldsymbol{s}^{(p+1)}.$$

So the statements of induction hold for $p+1$. By mathematical induction, the statements hold for any natural number p.

In the following we prove that the limit $\boldsymbol{\theta}' = \lim_{s\to+\infty} \lim_{t\to+\infty} \phi(s\boldsymbol{\theta}^*, t)$ exists, and $\boldsymbol{k}(\boldsymbol{\theta}') = \boldsymbol{k}^{(p_{\max})}$. We have already proved that there exists $\boldsymbol{\theta}_s$ such that $\boldsymbol{k}(\boldsymbol{\theta}_s) \to \boldsymbol{k}^{p_{\max}}$. By Lemma A.4, the limit $\tilde{\boldsymbol{\theta}} := \lim_{s\to 0} \boldsymbol{\theta}_s$ also exists.

By definition of $\boldsymbol{k}^{p_{\max}}$, $L(\tilde{\boldsymbol{\theta}}) = 0$. So $\tilde{\boldsymbol{\theta}}$ is a global minimizer of $L(\boldsymbol{\theta})$. Since $L(\boldsymbol{\theta})$ is real analytic, Łojasiewicz inequality Lojasiewicz (1965) holds for $L(\boldsymbol{\theta})$.

By standard Łojasiewicz inequality argument (for example, see Lemma G.1 in Li et al. (2021)) , there exists $C, \alpha, \delta > 0$, such that for any $\boldsymbol{\theta}_0$ in $B_\delta(\tilde{\boldsymbol{\theta}})$, the limit $\lim_{t\to\infty} \phi(\boldsymbol{\theta}_0, t)$ exists, and

$$\|\lim_{t\to\infty} \phi(\boldsymbol{\theta}_0, t) - \tilde{\boldsymbol{\theta}}\|_2 < C|\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}|_2^\alpha.$$

Since $\lim_{s\to+\infty} \boldsymbol{\theta}_s = \tilde{\boldsymbol{\theta}}$, for sufficiently small s, we have

$$\|\lim_{t\to\infty} \phi(\boldsymbol{\theta}_s, t) - \hat{\boldsymbol{\theta}}\|_2 < C\|\boldsymbol{\theta}_s - \hat{\boldsymbol{\theta}}\|_2^\alpha.$$

Let $s \to 0$, we get

$$\lim_{s\to 0} \lim_{t\to+\infty} \phi(s\boldsymbol{\theta}^*, t) = \tilde{\boldsymbol{\theta}}.$$

□

Theorem A.11. *Consider the deep diagonal linear network, and let Assumption 4.4 holds. Let Γ_s to be the trajectory of GF initialized at $s\boldsymbol{\theta}^*$. We use $a_{i,j}(\boldsymbol{\theta})$ to denote the value of $a_{i,j}$ at $\boldsymbol{\theta}$. Then the following holds:*

- *For each $p = 0, 1, \ldots, p_{\max}$, there exists $\boldsymbol{\theta}^s \in \Gamma^s$ such that*

  - $\lim_{s\to 0} \boldsymbol{k}(\boldsymbol{\theta}^s) = \boldsymbol{k}^{(p)}$, *and*
  - $\lim_{s\to+\infty} \frac{a_{i,L}(\boldsymbol{\theta}_s)}{s^{L-2}} = F_k^{-1}(s_k^{(p)})$.
- *The limit* $\boldsymbol{\theta}' = \lim_{s\to+\infty}\lim_{t\to+\infty}\phi(s\boldsymbol{\theta}^*, t)$ *exists, and* $\boldsymbol{k}(\boldsymbol{\theta}') = \boldsymbol{k}^{(p_{\max})}$.

Proof. The main idea of proof is the same with Theorem A.10. By calculation, we have

$$\frac{\mathrm{d}a_{i,\ell}}{\mathrm{d}t} = \Big(\prod_{r\neq\ell} a_{i,r}\Big)\, u_i(\boldsymbol{\theta}), \qquad \boldsymbol{u}(\boldsymbol{\theta}) = \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta})).$$

For any $i = 1, \dots, n$, $j = 1, \dots, L-1$, we have the preserved quantity

$$a_{i,j}^2(t) - a_{i,L}^2(t) = a_{i,j}^2(0) - a_{i,L}^2(0).$$

By assumption, $a_{i,j}^2(0) - a_{i,L}^2(0) = s^2\mu_{i,j}^2$. So we have

$$a_{i,j}^2(t) - a_{i,L}^2(t) = s^2\mu_{ij}^2 > 0.$$

Therefore, for any $t \geq 0$, the value of $a_{i,j}^2(t)$ can not be zero. So for $i = 1, \dots, n$, $j = 1, \dots, L-1$, the sign of $a_{i,j}(t)$ does not change during training. As a consequence, we have

$$\frac{\mathrm{d}a_{i,L}}{\mathrm{d}t} = \pm\Big(\prod_{r\neq L}\sqrt{a_{i,L}^2 + s^2\mu_{i,r}^2}\Big)\, u_i(\boldsymbol{\theta}),$$

The sign of the $\pm$ is same with the sign of $\prod_{r\neq L} a_{i,r}(0)$. Without loss of generality we assume that $\prod_{r\neq L} a_{i,r}^* > 0$ for all i. Otherwise one can replace $a_{i,L}(t)$ with $-a_{i,L}(t)$. Under the assumption, we have

$$\frac{\mathrm{d}a_{i,L}}{\mathrm{d}t} = \Big(\prod_{r\neq L}\sqrt{a_{i,L}^2 + s^2\mu_{i,r}^2}\Big)\, u_i(\boldsymbol{\theta}). \tag{17}$$

For $p = 0$, it is readily verifiable that the statements of the theorem hold. Now assume the statements of the theorem hold for natural number p. We want to prove the statements for $p+1$.

Define $I_1 = \{i \in [n] \mid k_i^{(p)} > 0\}$, $I_2 = \{i \in [n] \mid k_i^{(p)} = 0\}$, $I_3 = \{i \in [n] \mid k_i^{(p)} < 0\}$. Denote $\boldsymbol{\theta}^s(t) = \phi(\boldsymbol{\theta}^s, t)$, and denote $a_{ij}^s(t)$ to be the value of a_{ij} of $\boldsymbol{\theta}^s(t)$.

Define $\boldsymbol{u}(\boldsymbol{\theta}) = \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}(\boldsymbol{\theta}))$, and define $\boldsymbol{u}^* = \boldsymbol{X}^\intercal(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{k}^{(p)})$. By Lemma A.7, for any $\epsilon > 0$, there exists $\delta > 0$ such that for sufficiently small s, we have

$$\|\boldsymbol{u}(\boldsymbol{\theta}^s(t)) - \boldsymbol{u}^*\|_\infty < \epsilon$$

for all $0 \leq t \leq T^s$, where $T^s = \inf_{t\geq 0}\{t \mid \|\boldsymbol{k}_{I_2}(\boldsymbol{\theta}^s(t))\|_\infty > \delta\}$.

By Equation (17), we have

$$\frac{\mathrm{d}a_{i,L}^s}{\prod_{r\neq L}\sqrt{(a_{i,L}^s)^2 + s^2\mu_{i,r}^2}} = u_i(\boldsymbol{\theta}^s(t))\mathrm{d}t.$$

Take the integration, and we get

$$\int_{a_{i,L}^s(0)}^{a_{i,L}^s(t)} \frac{\mathrm{d}x}{\prod_{r\neq L}\sqrt{x^2 + s^2\mu_{i,r}^2}} = \int_0^t u_i(\boldsymbol{\theta}^s(r))\mathrm{d}r.$$

Change variables by defining $x = sx'$. Then

$$\int_{a_{i,L}(0)}^{a_{i,L}^s(t)} \frac{\mathrm{d}x}{\prod_{r\neq L}\sqrt{x^2 + s^2\mu_{i,r}^2}} = \int_{a_{i,L}(0)/s}^{a_{i,L}^s(t)/s} s^{2-L}\frac{\mathrm{d}x'}{\prod_{r\neq L}\sqrt{(x')^2 + \mu_{i,r}^2}}.$$

Therefore, we have

$$\int_{a_{i,L}(0)/s}^{a_{i,L}^s(t)/s} s^{2-L} \frac{\mathrm{d}x}{\prod_{r\neq L}\sqrt{x^2+\mu_{i,r}^2}} = \int_0^t u_i(\boldsymbol{\theta}^s(r))\mathrm{d}r.$$

By the definition of $F_i(z)$, we have

$$F_i(\frac{a_{i,L}^s(t)}{s^{L-2}}) - F_i(\frac{a_{i,L}^s(0)}{s^{L-2}}) = \int_0^t u_i(\boldsymbol{\theta}^s(r))\mathrm{d}r.$$

Since $\|\boldsymbol{u}(\boldsymbol{\theta}^s(t)) - \boldsymbol{u}^*\|_\infty < \epsilon$ for all $0 \le t \le T^s$, so we have

$$(u_i^* - \epsilon)t < F_i(\frac{a_{i,L}^s(t)}{s}) - F_i(\frac{a_{i,L}^s(0)}{s}) < (u_i^* + \epsilon)t, \forall t \in [0, T^s]. \tag{18}$$

Since $\|\boldsymbol{k}_{I_2}(\boldsymbol{\theta}^s(T^s))\|_\infty = \delta$, then there exists $j \in I_2$ such that

$$k_j(\boldsymbol{\theta}^s(T^s)) = \pm\delta.$$

Without loss of generality, let us assume that $u_j^* > 0$. Then $k_j(\boldsymbol{\theta}^s(T^s)) > 0$. So we have

$$k_j(\boldsymbol{\theta}^s(T^s)) = \delta.$$

Since we have the conserved quantities

$$(a_{j,l}^s)^2(t) - (a_{j,L}^s)^2(t) = s^2\mu_{jl}^2 \quad , \quad \forall l \in [L]$$

So we have

$$\lim_{s\to 0} a_{j,L}^s = \delta^{1/L}.$$

Apply Equation (18) to neuron j and $t = T^s$, we get

$$T^s = \frac{t_j^+ - F_j^{-1}(\boldsymbol{s}_j^{(p)})}{u_j^*} + h(\epsilon, s)$$

Here, $h(\epsilon, s) \to 0$ as $(\epsilon, s) \to \mathbf{0}$. For each $i \in I_2$, define

$$T_i = \begin{cases} \frac{t_i^+ - F_i^{-1}(s_i^{(p)})}{|u_i|}, & \text{if } u_i > 0 \\ \frac{t_i^- + F_i^{-1}(s_i^{(p)})}{|u_i|}, & \text{if } u_i < 0 \\ +\infty, & \text{if } u_i = 0 \end{cases}$$

Let $l \in \operatorname{argmin}_{k\in I_2} T_k$. By assumption, l is unique. Therefore T_l is strictly smaller than T_k if $k \neq l$.

It is readily verifiable that $j = l$ for sufficiently small ϵ and s, otherwise $a_{j,L}(t)$ will first reach δ.

For $i \in I_2 \setminus j$, by Equation (18), we have

$$(u_i - \epsilon)T^s < F_i(\frac{a_{i,L}^s(T^s)}{s}) - F_i(\frac{a_{i,L}^s(0)}{s}) < (u_i + \epsilon)T^s.$$

So we have

$$F_i(\frac{a_{i,L}^s(T^s)}{s}) = F_i(s_i^{(p)}) + u_i T_j + o(1).$$

Here, $o(1) \to 0$ as $\epsilon, s \to 0$. Since $F_i(s_i^{(p)}) + u_i T_j \in (-t_i^-, t_i^+)$ for all $i \in I_2 \setminus \{j\}$, then $a_{i,L}^s(T^s) = O(s)$. So we have

$$\lim_{s\to 0} a_{i,L}^s(T^s) = 0, \forall i \in I_2 \setminus \{j\}.$$

Similar to the proof of Theorem A.10, the limit $\boldsymbol{\theta}' = \lim_{s\to 0} \boldsymbol{\theta}^s(T^s)$ exists. Besides, $a_{i,l}(\boldsymbol{\theta}^s) = 0, \forall i \in I_2 \setminus \{j\}, l \in [L]$. The following part of proof is the same with Theorem A.10, we leave out the details. □

B. Concepts in Differential Geometry

In the appendix, we present several definitions and concepts in differential geometry that are pertinent to the content of this paper.

Definition B.1 (analytic manifold, page 3 and 4 of Jurdjevic (1997))**.** $\mathcal{M}$ is called an n dimensional analytic manifold if $\mathcal{M}$ is a topology space such that at each point $p \in \mathcal{M}$ there exists a neighbourhood U of p and a homeomorphism ϕ from U onto an open subset of $\mathbb{R}^n$. It is assumed that n does not vary with the choice of a point p on $\mathcal{M}$. The pair (ϕ, U) is called a chart at p. Moreover:

1. There exists a countable collection of charts $\{(\phi_i, U_i)\}_{i=1}^{\infty}$ such that $\mathcal{M} = \bigcup_{i=1}^{\infty} U_i$.
2. For each pair of points p_1 and p_2, there exist charts (ϕ_1, U_1) and (ϕ_2, U_2) such that $p_1 \in U_1$, $p_2 \in U_2$, and $U_1 \cap U_2 = \emptyset$. That is, points of $\mathcal{M}$ are separated by coordinate neighborhoods (i.e., $\mathcal{M}$ is Hausdorff).
3. For any charts (ϕ_1, U_1) and (ϕ_2, U_2) such that $U_1 \cap U_2 \neq \emptyset$, the mapping $\phi_1 \circ \phi_2^{-1}$ is analytic as a mapping from an open set in $\mathbb{R}^n$ into $\mathbb{R}^n$.

Definition B.2 (analytic vector fields, Definition 1 in Chapter 1 of Jurdjevic (1997))**.** Let $\mathcal{M}$ be an analytic manifold. The totality of $(p, v), p \in \mathcal{M}, v \in T_p\mathcal{M}$, is called the tangent bundle of $\mathcal{M}$ and is denoted by $T\mathcal{M}$. A vector field is a mapping $X : \mathcal{M} \to T\mathcal{M}$ such that for each $p \in \mathcal{M}$, if $\pi : T\mathcal{M} \to \mathcal{M}$ denotes the natural projection, then $\pi\big(X(p)\big) = p$. We say that X is an analytic vector field if X is an analytic map from $\mathcal{M}$ (as an analytic manifold) into $T\mathcal{M}$ (another analytic manifold).

Definition B.3 (Invariant manifold)**.** Let $\mathcal{M}$ be an immersed submanifold of $\mathbb{R}^M$. Let X be an analytic vector field on $\mathbb{R}^M$, and let $\boldsymbol{\theta}(t)$ denote the solution to the Cauchy problem $\dot{\boldsymbol{\theta}} = X(\boldsymbol{\theta}), \boldsymbol{\theta}(0) = \boldsymbol{\theta}_0$. We say that $\mathcal{M}$ is an **invariant manifold** of X if for every $\boldsymbol{\theta}_0 \in \mathcal{M}$, the solution $\boldsymbol{\theta}(t)$ remains in $\mathcal{M}$ for all t in its maximal interval of existence. We also say $\mathcal{M}$ **is invariant under** X.

C. Experiments

C.1. Details of Figure 1

We consider the model $F(\boldsymbol{\theta})(\boldsymbol{x}) = \sum_{i=1}^{4} k_i(\boldsymbol{\theta}_i)$ such that:

$$k_i(\boldsymbol{\theta}_i) = 2a_ib_i + (a_i^2 + b_i^2 + c_i^2)c_i \,, i = 1, 2$$

$$k_i(\boldsymbol{\theta}) = a_i \tanh(a_i) - (e^{b_i} - 1)^2 \,, i = 3, 4.$$

One can readily verify that this model satisfies Definition 4.1, and $t_i^+ = t_i^- = 1$ for all $i \in [n]$. We have the data matrix

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0.5 & 0.7 & 0 \\ 0.5 & 1 & 0.1 & 0.7 \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Experiment details: The parameters are initialized uniformly in the interval $[-10^{-60}, 10^{-60}]$. The learning rate is 0.1. We use *mpmath* library to support calculation of high-precision. In the code, we set mp.mp.dps = 200, which means that we use 200 decimal places of precision for calculations.

Output of Algorithm 1: In the following we calculate the output of Algorithm 1 under the input $\boldsymbol{X}, \boldsymbol{y}, t_i^+, t_i^-$.

- $p = 0$; The initialization is $\boldsymbol{k}^{(0)} = \boldsymbol{s}^{(0)} = \boldsymbol{0}$.
- $p = 1$; We have $I_2 = \{1, 2, 3, 4\}$. By calculation, $\boldsymbol{u} = \boldsymbol{X}^\intercal \boldsymbol{y} = (1, 0.5, 0.7, 0)^\intercal$. So the time vector $\boldsymbol{\delta} = (1, 2, \frac{10}{7}, +\infty)$. Therefore, the first neuron will be chosen as the feature. So $\boldsymbol{s}^{(1)} = (1, 0.5, 0.7, 0)^\intercal$. $k_1^{(1)}$ is the solution of
$$\min_{k_1 \in \mathbb{R}} \|\boldsymbol{X}_{:,1} k_1 - \boldsymbol{y}\|_2^2.$$
This leads to $k_1^{(1)} = 0.8$. So $\boldsymbol{k}^{(1)} = (0.8, 0, 0, 0)^\intercal$. To summarize, we have
$$\boldsymbol{k}^{(1)} = (0.8, 0, 0, 0)^\intercal \,, \; \boldsymbol{s}^{(1)} = (1, 0.5, 0.7, 0)^\intercal.$$

- $p = 2$; We have $I_2 = \{2, 3, 4\}$. By calculation, we have

$$u_2, u_3, u_4 = -0.3, 0.1, -0.28.$$

So we have

$$\delta_2, \delta_3, \delta_4 = 5, 3, \frac{25}{7}.$$

Therefore, δ_3 is the smallest. So the third neuron is chosen. Then $\boldsymbol{s}^{(2)} = (1, -0.4, 1, -0.84)^\intercal$.

Besides, $k_1^{(2)}$ and $k_3^{(2)}$ are the solution of

$$\min_{(k_1,k_3)\in\mathbb{R}^2} \|\boldsymbol{X}_{:,1}k_1 + \boldsymbol{X}_{:,3}k_3 - \boldsymbol{y}\|_2^2 \quad s.t. \quad k_1 \geq 0.$$

The solution is $k_1 = 0, k_3 = 1.4$. So we have $\boldsymbol{k}^{(2)} = (0, 0, 1.4, 0)^\intercal$. To summarize, we have

$$\boldsymbol{k}^{(2)} = (0, 0, 1.4, 0)^\intercal\ ,\ \boldsymbol{s}^{(2)} = (1, -0.4, 1, -0.84)^\intercal.$$

- $p = 3$; We have $I_2 = \{1, 2, 4\}$. By calculation, we have

$$u_1, u_2, u_4 = -0.05, -0.13, -0.098.$$

So we have

$$\delta_1, \delta_2, \delta_4 = 20, \frac{60}{13}, \frac{80}{49}.$$

Therefore, δ_4 is the smallest. So the fourth neuron is chosen. $k_3^{(3)}$ and $k_4^{(3)}$ are the solution of

$$\min_{(k_3,k_4)\in\mathbb{R}^2} \|\boldsymbol{X}_{:,3}k_3 + \boldsymbol{X}_{:,4}k_4 - \boldsymbol{y}\|_2^2 \quad s.t. \quad k_3 \geq 0.$$

The solution is $k_3 = \frac{10}{7}, k_4 = -\frac{10}{49}$. So we have $\boldsymbol{k}^{(3)} = (0, 0, \frac{10}{7}, -\frac{10}{49})^\intercal$. Besides, it is readily to verify that $\boldsymbol{X}\boldsymbol{k}^{(3)} = \boldsymbol{y}$. So the iteration stops.

To summarize, Algorithm 1 ends at $p = 3$, and we have

$$\boldsymbol{k}^{(3)} = (0, 0, \frac{10}{7}, -\frac{10}{49})^\intercal.$$

Comparison of $\boldsymbol{k}^{(p)}$ to the dashed line in Figure 1:

- In the first dashed line, we have $(k_1, k_2, k_3, k_4)^\intercal = (0.800, 8.34e-70, 1.51e-28, -1.35e-100)^\intercal$. This value is close to $\boldsymbol{k}^{(1)} = (0.8, 0, 0, 0)^\intercal$.
- In the second dashed line, we have $(k_1, k_2, k_3, k_4)^\intercal = (-1.01e-6, -6.23e-78, 1.400, -1.34e-17)^\intercal$. This value is close to $\boldsymbol{k}^{(2)} = (0, 0, 1.4, 0)^\intercal$.
- In the second dashed line, we have $(k_1, k_2, k_3, k_4)^\intercal = (-1.01e-6, -6.23e-78, 1.400, -1.34e-17)^\intercal$. This value is close to $\boldsymbol{k}^{(2)} = (0, 0, 1.4, 0)^\intercal$.
- In the third dashed line, we have $(k_1, k_2, k_3, k_4)^\intercal = (-1.71e-6, -2.48e-57, 1.429, -0.204)^\intercal$. This value is close to $\boldsymbol{k}^{(3)} = (0, 0, \frac{10}{7}, -\frac{10}{49})^\intercal$.

Difference between gradient flow and gradient descent: In gradient flow, as shown in Theorem 6.5, the parameter $\boldsymbol{\theta}_i$ is strictly restricted to a simple curve. As a consequence, if $k_i(\boldsymbol{\theta}_i)$ wants to change its sign, then $\boldsymbol{\theta}_i$ must must pass its initialization. Therefore, if $k_i(\boldsymbol{\theta}_i)$ wants to change its sign, $|k_i|$ must decrease to its initialization scale.

However, in Figure 1, we see that the value of k_1 is 0.800 at the first dashed line, and $-1.01e-6$ at the second dashed line. Between the two dashed lines, the value of $|k_1|$ does not decrease to its initialization scale $1e-60$. This is different from the dynamics of gradient flow. We attribute this phenomenon to the fact that Structural Invariant Manifold in Theorem 6.5 is not invariant under gradient descent (since $O_{\mathcal{F}_i}(\boldsymbol{\theta}_i)$ is not a straight line). The failure of Theorem 6.5 in gradient descent is also evidenced by Ziyin et al. (2024).

C.2. Deep Diagonal Linear Network

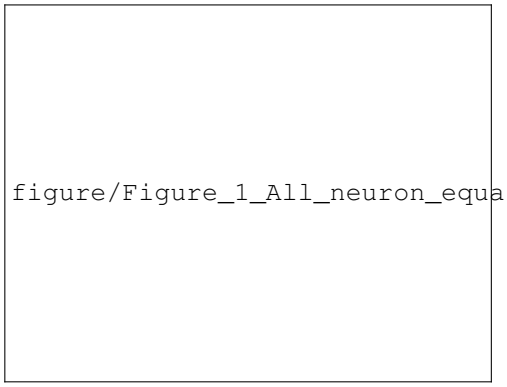
Figure 2 illustrates the training dynamics of a Deep Diagonal Linear Network ($w = a \odot b \odot c$) optimized via gradient descent. The experiment utilizes a synthetic dataset with $M = 3$ samples and $N = 8$ features. We employ a small initialization scale ($\approx 10^{-4}$) while maintaining a strict layer-wise hierarchy ($a > b > c$) to facilitate theoretical analysis.

Figures 2a and 2b demonstrate a clear synchronization between the descent of the loss curve and the sequential growth of neural parameters. The learning process exhibits distinct stage-wise behavior, where the system transitions between saddle points. We define the end of each stage as the point where the active parameters stabilize and the gradient of the residual becomes negligible.

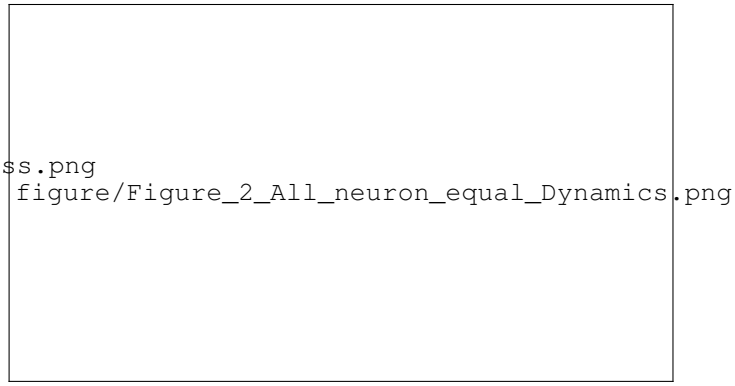
Figure 2c presents the theoretical "growth time" for each neuron at the onset of every stage, calculated by our algorithm as the integrated virtual distance required to escape the saddle point. Notably, these theoretical predictions align perfectly with the empirical emergence order and relative timing observed in Figure 1.

Figure 3 demonstrates the controllability of feature selection through initialization. While the network naturally prioritizes features based on data correlation, we show that this order can be manipulated. By selectively amplifying the initialization magnitude of the 5th neuron by a factor of 5 (while keeping others at the original scale), we successfully induce this neuron to emerge first, overriding the natural data-driven order. Crucially, our theoretical framework accurately captures this perturbation, correctly predicting the altered emergence sequence in Figure 3.

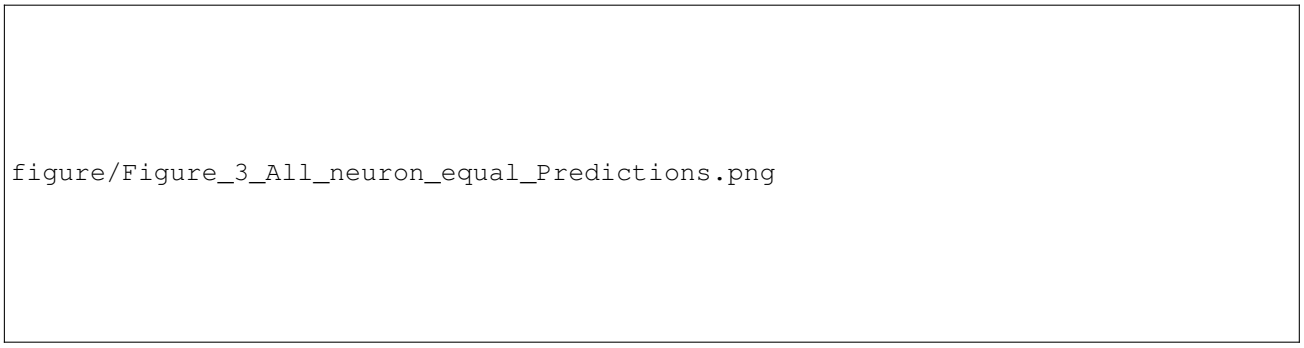
Figure 4 investigates the implicit bias under a specific "Zero-Balanced" initialization scheme ($a = b, c = 0$, with scale $\approx 10^{-4}$). This setting creates a strict saddle point at initialization. To verify the regularization properties, we computed the mathematical minimum ℓ_1-norm solution for the generated dataset. As shown in Figure 3, as the loss approaches zero, the parameters converge precisely to the ℓ_1 solution rather than the sparse ground truth (ℓ_0 solution). This confirms that the $c = 0$ initialization induces a strong inductive bias towards ℓ_1 regularization, effectively driving the network to select the solution with the minimum norm among infinite possibilities.



(a) Training loss of the Deep Diagonal Linear Network

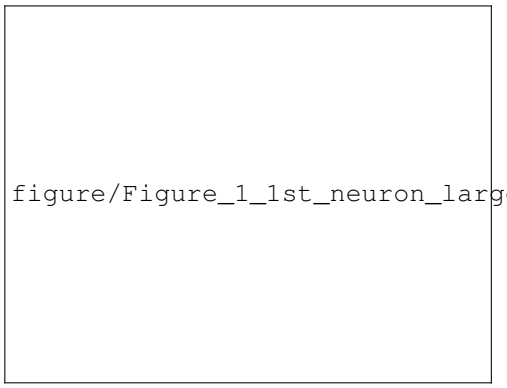


(b) Parameter trajectories of individual neurons across training epochs

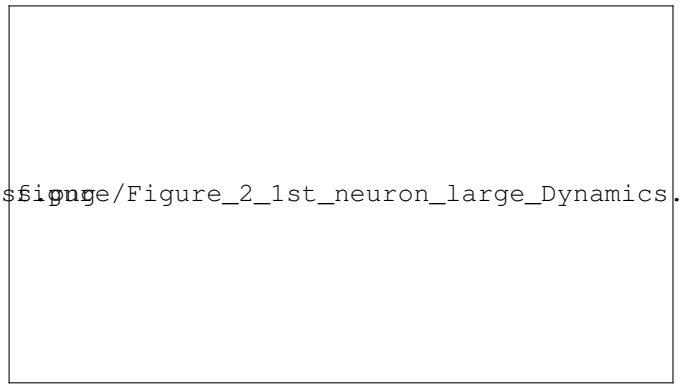


(c) Theoretical prediction of neuron growth times at each stage

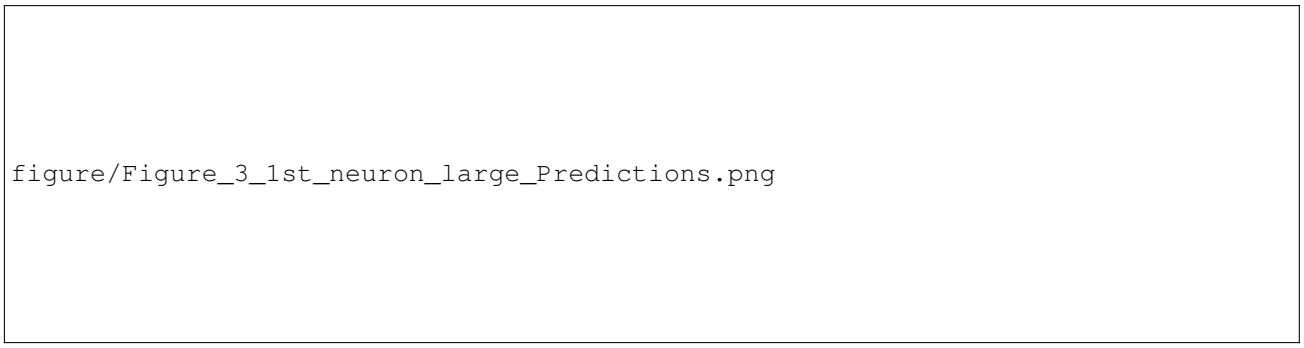
Figure 2. Training dynamics of the Deep Diagonal Linear Network with uniform initialization scale. (a) The training loss curve. The vertical red dashed lines correspond to the same epochs marked in (b), indicating stage transitions. (b) Evolution of the absolute parameter magnitudes $|w_i| = |a_i b_i c_i|$ for the 8 neurons throughout training. (c) Theoretical predictions of the growth time calculated at the beginning of each stage. The algorithm identifies the neuron with the shortest predicted growth time as the "winner" for the subsequent stage.



(a) Training loss curve of the Deep Diagonal network



(b) Parameter trajectories of individual neurons across training epochs



(c) Theoretical prediction of neuron growth times at each stage

Figure 3. Training dynamics of the Deep Diagonal Linear Network with biased initialization. The initialization scale of the first neuron is amplified by a factor of 5, while others remain unchanged. (a) The training loss curve. The vertical red dashed lines correspond to the same epochs marked in (b). (b) Evolution of the absolute parameter magnitudes $|w_i|$ throughout training. Note that the first neuron emerges earliest due to its larger initialization scale. (c) Theoretical predictions of the growth time. The algorithm correctly identifies the first neuron as the "winner" of the initial stage, capturing the effect of the initialization bias.

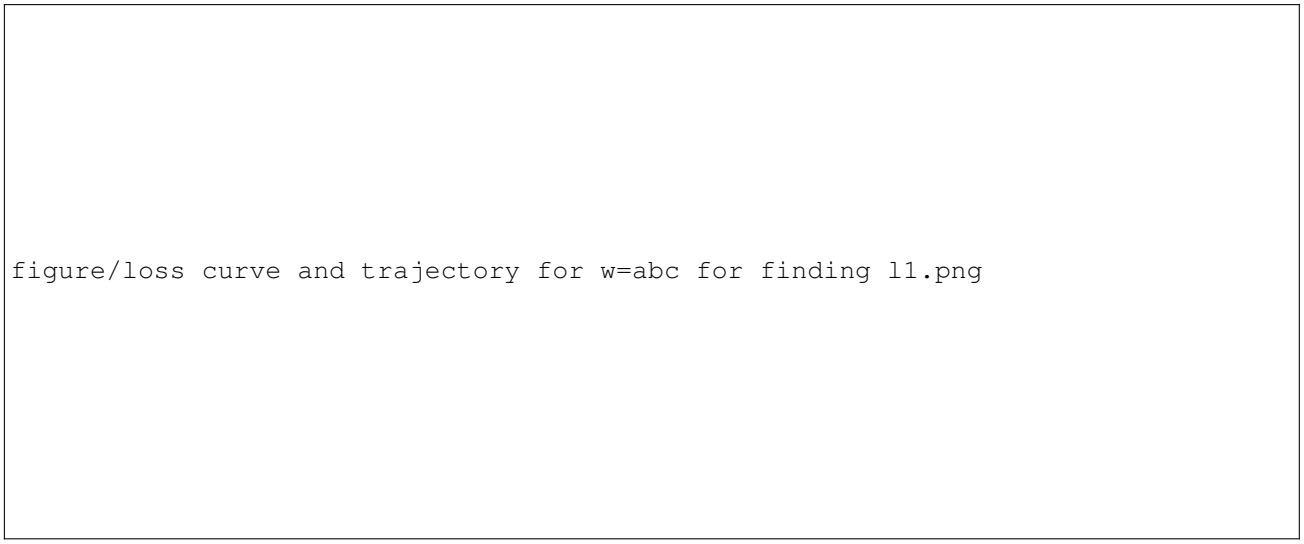


Figure 4. Verification of implicit ℓ_1 regularization under "Zero-Balanced" initialization ($a = b, c = 0$). The left figure is Training loss curve. The right figure depicts parameter evolution trajectories compared against the theoretical baseline. The red dashed lines represent the mathematical minimum ℓ_1-norm solution for the dataset.