Optimization of Graph Neural Networks:
Implicit Acceleration by Skip Connections and More Depth

Keyulu Xu 1 Mozhi Zhang 2 Stefanie Jegelka 1 Kenji Kawaguchi 3

Abstract

Graph Neural Networks (GNNs) have been studied through the lens of expressive power and generalization. However, their optimization properties are less well understood. We take the first step towards analyzing GNN training by studying the gradient dynamics of GNNs. First, we analyze linearized GNNs and prove that despite the non-convexity of training, convergence to a global minimum at a linear rate is guaranteed under mild assumptions that we validate on real-world graphs. Second, we study what may affect the GNNs’ training speed. Our results show that the training of GNNs is implicitly accelerated by skip connections, more depth, and/or a good label distribution. Empirical results confirm that our theoretical results for linearized GNNs align with the training behavior of nonlinear GNNs. Our results provide the first theoretical support for the success of GNNs with skip connections in terms of optimization, and suggest that deep GNNs with skip connections would be promising in practice.

1. Introduction

Graph Neural Networks (GNNs) (Gori et al., 2005; Scarselli et al., 2009) are an effective framework for learning with graphs. GNNs learn node representations on a graph by extracting high-level features not only from a node itself but also from a node’s surrounding subgraph. Specifically, the node representations are recursively aggregated and updated using neighbor representations (Merkwirth & Lengauer, 2005; Duvenaud et al., 2015; Defferrard et al., 2016; Kearnes et al., 2016; Gilmer et al., 2017; Hamilton et al., 2017; Velickovic et al., 2018; Liao et al., 2020).

Recently, there has been a surge of interest in studying the theoretical aspects of GNNs to understand their success and limitations. Existing works have studied GNNs’ expressive power (Keriven & Peyré, 2019; Maron et al., 2019; Chen et al., 2019; Xu et al., 2019; Sato et al., 2019; Loukas, 2020), generalization capability (Scarselli et al., 2018; Du et al., 2019b; Xu et al., 2020; Garg et al., 2020), and extrapolation properties (Xu et al., 2021). However, the understanding of the optimization properties of GNNs has remained limited. For example, researchers working on the fundamental problem of designing more expressive GNNs hope and often empirically observe that more powerful GNNs better fit the training set (Xu et al., 2019; Sato et al., 2020; Vignac et al., 2020). Theoretically, given the non-convexity of GNN training, it is still an open question whether better representational power always translates into smaller training loss. This motivates the more general questions:

* Can gradient descent find a global minimum for GNNs?
* What affects the speed of convergence in training?

In this work, we take an initial step towards answering the questions above by analyzing the trajectory of gradient descent, i.e., gradient dynamics or optimization dynamics. A complete understanding of the dynamics of GNNs, and deep learning in general, is challenging. Following prior works on gradient dynamics (Saxe et al., 2014; Arora et al., 2019a; Bartlett et al., 2019), we consider the linearized regime, i.e., GNNs with linear activation. Despite the linearity, key properties of nonlinear GNNs are present: The objective function is non-convex and the dynamics are nonlinear (Saxe et al., 2014; Kawaguchi, 2016). Moreover, we observe the learning curves of linear GNNs and ReLU GNNs are surprisingly similar, both converging to nearly zero training loss at the same linear rate (Figure 1). Similarly, prior works report comparable performance in node classification benchmarks even if we remove the non-linearities (Thekumparampil et al., 2018; Wu et al., 2019). Hence, understanding the dynamics of linearized GNNs is a valuable step towards understanding the general GNNs.

Our analysis leads to an affirmative answer to the first question. We establish that gradient descent training of a linearized GNN with squared loss converges to a global minimum at a linear rate. Experiments confirm that the assump-

---

1Equal contribution 1Massachusetts Institute of Technology 2The University of Maryland 3Harvard University. Correspondence to: Keyulu Xu <keyulu@mit.edu>, Kenji Kawaguchi <kkawaguchi@fas.harvard.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).
We give an explanation from the lens of optimization. The training losses of deeper GNNs may be worse due to over-smoothing, and such architectures are put of intermediate GNN layers, and thus the resulting dynamics are more intricate. To our knowledge, our results are the first convergence results for GNNs with more than one hidden layer, with or without skip connections.

We then study what may affect the training speed of GNNs. First, for any fixed depth, GNNs with skip connections train faster. Second, increasing the depth further accelerates the training of GNNs. Third, faster training is obtained when the labels are more correlated with the graph features, i.e., labels contain “signal” instead of “noise”. Overall, experiments for nonlinear GNNs agree with the prediction of our theory for linearized GNNs.

We begin by introducing our notation. Let $G = (V,E)$ be a graph with $n$ vertices $V = \{v_1,v_2,\cdots,v_n\}$. Its adjacency matrix $A \in \mathbb{R}^{n \times n}$ has entries $A_{ij} = 1$ if $(v_i,v_j) \in E$ and 0 otherwise. The degree matrix associated with $A$ is $D = \text{diag}(d_1, d_2, \cdots, d_n)$ with $d_i = \sum_{j=1}^{n} A_{ij}$. For any matrix $M \in \mathbb{R}^{m \times m'}$, we denote its $j$-th column vector by $M_{sj} \in \mathbb{R}^{m'}$, its $i$-th row vector by $M_{i*} \in \mathbb{R}^{m'}$, and its largest and smallest (i.e., $\min(m,m')$-th largest) singular values by $\sigma_{\max}(M)$ and $\sigma_{\min}(M)$, respectively. The data matrix $X \in \mathbb{R}^{s \times n}$ has columns $X_{j*}$ corresponding to the feature vector of node $v_j$, with input dimension $m_x$.

The task of interest is node classification or regression. Each node $v_i \in V$ has an associated label $y_{i} \in \mathbb{R}^{m_y}$. In the transductive (semi-supervised) setting, we have access to training labels for only a subset $I \subseteq [n]$ of nodes on $G$, and the goal is to predict the labels for the other nodes in $[n] \\setminus I$. Our problem formulation easily extends to the inductive setting by letting $I = [n]$, and we can use the trained model for prediction on unseen graphs. Hence, we have access to $\bar{n} = |I| \leq n$ training labels $Y = [y_{i}]_{i \in I} \in \mathbb{R}^{m_y \times \bar{n}}$ and we train the GNN using $X, Y, G$. Additionally, for any $M \in \mathbb{R}^{m \times m'}, \ I$ may index sub-matrices $M_{s*} = [M_{s*}]_{i \in I} \in \mathbb{R}^{m' \times \bar{n}}$ (when $m' \geq n$) and $M_{*s} = [M_{*s}]_{i \in I} \in \mathbb{R}^{s \times m}$ (when $m \geq n$).

Graph Neural Networks (GNNs) use the graph structure and node features to learn representations of nodes (Scarselli et al., 2009). GNNs maintain hidden representations $h^i_{(l)} \in \mathbb{R}^{m_l}$ for each node $v$, where $m_l$ is the hidden dimension on the $l$-th layer. We let $X_{(0)} = [h_{(0)}^1, h_{(0)}^2, \cdots, h_{(0)}^{\bar{n}}] \in \mathbb{R}^{m_0 \times \bar{n}}$, and set $X_{(0)}$ as the input feature $X$. The node hidden representations $X_{(l)}$ are updated by aggregating and transforming the neighbor representations:

$$X_{(l)} = \sigma(B_{(l)} X_{(l-1)} S) \in \mathbb{R}^{m_l \times n}, \quad (1)$$

where $\sigma$ is a nonlinearity such as ReLU, $B_{(l)} \in \mathbb{R}^{m_l \times m_{l-1}}$ is the weight matrix, and $S \in \mathbb{R}^{n \times n}$ is the GNN aggregation matrix, whose formula depends on the exact variant of GNN. In Graph Isomorphism Networks (GIN) (Xu et al., 2018),...
Throughout this paper, we refer multiscale GNNs to the commonly used Jumping Knowledge Network (JK-Net) (Xu et al., 2018), which connects the output of all intermediate GNN layers to the final layer with skip connections:

**Definition 2.** (Multiscale linear GNN). Given data matrix \(X \in \mathbb{R}^{m_x \times n}\), aggregation matrix \(S \in \mathbb{R}^{n \times n}\), weight matrices \(W_l \in \mathbb{R}^{m_x \times m_u}\), \(B_l \in \mathbb{R}^{m_l \times m_{l-1}}\) with \(W = (W_0, W_1, \ldots, W_H)\), a multiscale linear GNN with \(H\) layers \(f(X, W, B) \in \mathbb{R}^{m_x \times n}\) is defined as

\[
f(X, W, B) = \sum_{l=0}^{H} W_l X_l, \quad X_l = B_l X_{l-1} S. \tag{3}
\]

Given a GNN \(f(\cdot)\) and a loss function \(\ell(\cdot, Y)\), we can train the GNN by minimizing the training loss \(L(W, B)\):

\[
L(W, B) = \ell(f(X, W, B)_{\mathcal{I}}, Y), \tag{4}
\]

where \(f(X, W, B)_{\mathcal{I}}\) corresponds to the GNN’s predictions on nodes that have training labels and thus incur training losses. The pair \((W, B)\) represents the trainable weights:

\[
L(W, B) = L(W_1, \ldots, W_H, B_1, \ldots, B_H) .
\]

For completeness, we define the global minimum of GNNs.

**Definition 3.** (Global minimum). For any \(H \in \mathbb{N}_0\), \(L^*_H\) is the global minimum value of the \(H\)-layer linear GNN \(f\):

\[
L^*_H = \inf_{W, B} \ell(f(X, W, B)_{\mathcal{I}}, Y). \tag{6}
\]

Similarly, we define \(L^*_1\) as the global minimum value of the multiscale linear GNN \(f\) with \(H\) layers.

We are ready to present our main results on global convergence for linear GNNs and multiscale linear GNNs.

### 3. Convergence Analysis

In this section, we show that gradient descent training a linear GNN with squared loss, with or without skip connections, converges linearly to a global minimum. Our conditions for global convergence hold on real-world datasets and provably hold under assumptions, e.g., initialization.

In linearized GNNs, the loss \(L(W, B)\) is non-convex (and non-invex) despite the linearity. The graph aggregation \(S\) creates interaction among the data and poses additional challenges in the analysis. We show a fine-grained analysis of the GNN’s gradient dynamics can overcome these challenges. Following previous works on gradient dynamics (Saxe et al., 2014; Huang & Yau, 2020; Ji & Telgarsky, 2020; Kawaguchi, 2021), we analyze the GNN learning process via the gradient flow, i.e., gradient descent with infinitesimal steps: \(\forall t \geq 0\), the network weights evolve as

\[
\frac{d}{dt}W_t = -\frac{\partial L}{\partial W}(W_t, B_t), \quad \frac{d}{dt}B_t = -\frac{\partial L}{\partial B}(W_t, B_t), \tag{7}
\]

where \((W_t, B_t)\) represents the trainable parameters at time \(t\) with initialization \((W_0, B_0)\).
3.1. Linearized GNNs

Theorem 1 states our result on global convergence for linearized GNNs without skip connections.

**Theorem 1.** Let \( f \) be an \( H \)-layer linear GNN and \( \ell(q,Y) = \|q - Y\|_2^2 \) where \( q,Y \in \mathbb{R}^{m_x \times n} \). Then, for any \( T > 0 \),

\[
L(W_T, B_T) - L^*_H \leq (L(W_0, B_0) - L^*_H) e^{-4 \lambda_T^{(H)} \sigma_{\min}^2(X(S^H)_{+}, z) T},
\]

where \( \lambda_T^{(H)} \) is the smallest eigenvalue \( \lambda^{(H)} := \inf_{t \in [0,T]} \lambda_{\min}(B_{(t)}^{(1:H)}) \) and \( B_{(t)}^{(1:H)} := B_{(1)}^{(1:H)} \cdots B_{(1)}^{(1:H)} \) for any \( l \in \{0, \ldots, H\} \) with \( B_{(1)}^{(1:H)} := I \).

**Proof.** (Sketch) We decompose the gradient dynamics into three components: the graph interaction, non-convex factors, and convex factors. We then bound the effects of the graph interaction and non-convex factors through \( \sigma_{\min}^2(X(S^H)_{+}, z) \) and \( \lambda_{\min}(B_{(t)}^{(1:H)}) \) respectively. The complete proof is in Appendix A.1.

Theorem 1 implies that convergence to a global minimum at a linear rate is guaranteed if \( \sigma_{\min}^2(X(S^H)_{+}, z) > 0 \) and \( \lambda_T > 0 \). The first condition on the product of \( X \) and \( S^H \) indexed by \( z \) only depends on the node features \( X \) and the GNN aggregation matrix \( S \). It is satisfied if \( \text{rank}(X(S^H)_{+}, z) = \min(m_x, \bar{n}) \), because \( \sigma_{\min}(X(S^H)_{+}, z) \) is the \( \min(m_x, \bar{n}) \)-th largest singular value of \( X(S^H)_{+}, z \in \mathbb{R}^{m_x \times \bar{n}} \). The second condition \( \lambda_T^{(H)} > 0 \) is time-dependent and requires a more careful treatment. Linear convergence is implied as long as \( \lambda_{\min}(B_{(t)}^{(1:H)}) \) and \( \lambda_{\min}(B_{(t)}^{(1:H)}) \) respectively.

**Empirical validation of conditions.** We verify both the graph condition \( \sigma_{\min}^2(X(S^H)_{+}, z) > 0 \) and the time-dependent condition \( \lambda_T^{(H)} > 0 \) for (discretized) \( T > 0 \). First, on the popular graph datasets, Cora and Citeseer (Sen et al., 2008), and the GNN models, GCN (Kipf & Welling, 2017) and GIN (Xu et al., 2019), we have \( \sigma_{\min}^2(X(S^H)_{+}, z) > 0 \) (Figure 2a). Second, we train linear GCN and GIN on Cora and Citeseer to plot an example of how the \( \lambda_T^{(H)} = \inf_{t \in [0,T]} \lambda_{\min}((B_{(t)}^{(1:H)})^{\top} B_{(t)}^{(1:H)}) \) changes with respect to time \( T \) (Figure 2b). We further confirm that \( \lambda_T^{(H)} > 0 \) until convergence, \( \lim_{T \to \infty} \lambda_T^{(H)} > 0 \) across different settings, e.g., datasets, depths, models (Figure 2c). Our experiments use the squared loss, random initialization, learning rate 1e-4, and set the hidden dimension to the input dimension (note that Theorem 1 assumes the hidden dimension is at least the input dimension). Further experimental details are in Appendix C. Along with Theorem 1, we conclude that linear GNNs converge linearly to a global minimum. Empirically, we indeed see both linear and ReLU GNNs converging at the same linear rate to nearly zero training loss in node classification tasks (Figure 1).

**Guarantee via initialization.** Besides the empirical verification, we theoretically show that a good initialization guarantees the time-dependent condition \( \lambda_T > 0 \) for any \( T > 0 \). Indeed, like other neural networks, GNNs do not converge to a global optimum with certain initializations: e.g., initializing all weights to zero leads to zero gradients and \( \lambda_T^{(H)} = 0 \) for all \( T \), and hence no learning. We introduce a notion of singular margin and say an initialization is good if it has a positive singular margin. Intuitively, a good initialization starts with an already small loss.

**Definition 4.** (Singular margin) The initialization \( (W_0, B_0) \) is said to have singular margin \( \gamma > 0 \) with respect to a layer \( l \in \{1, \ldots, H\} \) if \( \sigma_{\min}(B_{(l)}^{(1:H)}) \geq \gamma \) for all \( (W, B) \) such that \( L(W, B) \leq L(W_0, B_0) \).

Proposition 1 then states that an initialization with positive singular margin \( \gamma \) guarantees \( \lambda_T^{(H)} \geq \gamma^2 > 0 \) for all \( T \):

**Proposition 1.** Let \( f \) be a linear GNN with \( H \) layers and \( \ell(q,Y) = \|q - Y\|_2^2 \). If the initialization \( (W_0, B_0) \) has singular margin \( \gamma > 0 \) with respect to the layer \( H \) and \( m_H \geq m_x \), then \( \lambda_T^{(H)} \geq \gamma^2 \) for all \( T \in [0, \infty) \).

Proposition 1 follows since \( L(W_t, B_t) \) is non-increasing with respect to time \( t \) (proof in Appendix A.2).

Relating to previous works, our singular margin is a generalized variant of the deficiency margin of linear feedforward networks (Arora et al., 2019a, Definition 2 and Theorem 1):

**Proposition 2.** (Informal) If initialization \( (W_0, B_0) \) has deficiency margin \( c > 0 \), then it has singular margin \( \gamma > 0 \).

The formal version of Proposition 2 is in Appendix A.3.

To summarize, Theorem 1 along with Proposition 1 implies that we have a prior guarantee of linear convergence to a global minimum for any graph with \( \text{rank}(X(S^H)_{+}, z) = \min(m_x, \bar{n}) \) and initialization \( (W_0, B_0) \) with singular margin \( \gamma > 0 \); i.e., for any desired \( \epsilon > 0 \), we have that

\[
T \geq \frac{1}{4 \gamma^2 \sigma_{\min}^2(X(S^H)_{+}, z)} \log \frac{L(A_0, B_0) - L^*_H}{\epsilon}, \tag{9}
\]

While the margin condition theoretically guarantees linear convergence, empirically, we have already seen that the convergence conditions of across different training settings for widely used random initialization.

Theorem 1 suggests that the convergence rate depends on a combination of data features \( X \), the GNN architecture and graph structure via \( S \) and \( H \), the label distribution and initialization via \( \lambda_T \). For example, GIN has better such constants than GCN on the Cora dataset with everything else
Without skip connections, the GNNs under linearization still behave like linear feedforward networks with augmented graph features. With skip connections, the dynamics and analysis become much more intricate. The expressive power of multiscale linear GNNs changes significantly as depth increases. Moreover, the skip connections create complex interactions among different layers and graph structures of various scales in the optimization dynamics. Theorem 2 states our convergence results for multiscale linear GNNs in three cases: (i) a general form; (ii) a weaker condition for boundary cases that uses \( \lambda_{TH}^f \) instead of \( \lambda_{TH}^{(l)} \); (iii) a faster rate if we have monotonic expressive power as depth increases.

**Theorem 2.** Let \( f \) be a multiscale linear GNN with \( H \) layers and \( \ell(q, Y) = \|q - Y\|_F^2 \), where \( q, Y \in \mathbb{R}^{m_0 \times n} \). Let \( \lambda_{TH}^{(1:H)} := \min_{0 \leq l \leq H} \lambda_{TH}^{(l)} \). For any \( T > 0 \), the following hold:

1. *(General).* Let \( G_{TH} := [X^T, (XS)^T, \ldots, (XS^H)^T]^T \in \mathbb{R}^{(H+1)m_0 \times n} \). Then
   \[
   L(W_T, B_T) - L_{TH}^* \leq (L(W_0, B_0) - L_{TH}^*) e^{-4\lambda_{TH}^{(1:H)} \sigma_{TH}^2 ((G_{TH})_{xz}) T}. 
   \]

2. *(Boundary cases).* For any \( H' \in \{0, 1, \ldots, H\} \),
   \[
   L(W_T, B_T) - L_{H'}^* \leq (L(W_0, B_0) - L_{H'}^*) e^{-4\lambda_{TH'}^{(H')} \sigma_{TH'}^2 ((X(S^H'))_{xz}) T}. 
   \]

3. *(Monotonic expressive power).* If there exist \( l, l' \in \{0, \ldots, H\} \) with \( l < l' \) such that \( L_l^* \geq L_{l+1}^* \geq \cdots \geq L_{l'}^* \), or \( L_l^* \leq L_{l+1}^* \leq \cdots \leq L_{l'}^* \), then
   \[
   L(W_T, B_T) - L_{l'}^* \leq (L(W_0, B_0) - L_{l'}^*) e^{-4\sum_{k=0}^{l-1} \lambda_{TH}^{(k)} \sigma_{TH}^2 ((X(S^k))_{xz}) T},
   \]
   where \( l'' = l \) if \( L_l^* = L_{l+1}^* = \cdots = L_{l'}^* \), and \( l'' = l' \) if \( L_l^* \leq L_{l+1}^* \leq \cdots \leq L_{l'}^* \).

**Proof.** (Sketch) A key observation in our proof is that the interactions of different scales cancel out to point towards a specific direction in the gradient dynamics induced in a space of the loss value. The complete proof is in Appendix A.4.

Similar to Theorem 1 for linear GNNs, the most general form (i) of Theorem 2 implies that convergence to the global minimum value of the entire multiscale linear GNN \( L_{1:H}^* \) at linear rate is guaranteed when \( \sigma_{TH}^2 ((G_{TH})_{xz}) > 0 \) and \( \lambda_{TH}^{(1:H)} > 0 \). The graph condition \( \sigma_{TH}^2 ((G_{TH})_{xz}) > 0 \) is satisfied if \( \text{rank}((G_{TH})_{xz}) = \min(m_x(H+1), n) \). The time-dependent condition \( \lambda_{TH}^{(1:H)} > 0 \) is guaranteed if the initialization \( (W_0, B_0) \) has singular margin \( \gamma > 0 \) with respect to every layer (Proposition 3 is proved in Appendix A.5):

**Proposition 3.** Let \( f \) be a multiscale linear GNN and \( \ell(q, Y) = \|q - Y\|_F^2 \). If the initialization \( (W_0, B_0) \) has singular margin \( \gamma > 0 \) with respect to every layer \( T \) (Figure 3b), and we confirm that at convergence, \( \lambda_{TH}^{(1:H)} > 0 \) across different settings (Figure 3c). Experimental details are in Appendix C.

**Boundary cases.** Because the global minimum value of multiscale linear GNNs \( L_{1:H}^* \) can be smaller than that of linear GNNs \( L_{TH}^* \), the conditions in Theorem 2(i) may sometimes be stricter than those of Theorem 1. For example, in Theorem 2(i), we require \( \lambda_{TH}^{(1:H)} := \min_{0 \leq l \leq H} \lambda_{TH}^{(l)} \) rather than \( \lambda_{TH}^{(H)} \) to be positive. If \( \lambda_{TH}^{(l)} = 0 \) for some \( l \), then Theorem 2(i) will not guarantee convergence to \( L_{1:H}^* \).

Although the boundary cases above did not occur on the tested real-world graphs (Figure 3), for theoretical interest, Theorem 2(ii) guarantees that in such cases, multiscale linear GNNs still converge to a value no worse than the global minimum value of non-multiscale linear GNNs. For any intermediate layer \( H' \), assuming \( \sigma_{TH'}^2 ((X(S^H'))_{xz}) > 0 \) and \( \lambda_{TH'}^{(H')} > 0 \), Theorem 2(ii) bounds the loss of the multiscale linear GNN \( L(W_{TH}, B_{TH}) \) at convergence by the global minimum value \( L_{H'}^* \) of the corresponding linear GNN with \( H' \) layers.

**Faster rate under monotonic expressive power.** Theorem 2(iii) considers a special case that is likely in real graphs: the global minimum value of the non-multiscale linear GNN \( L_{H'}^* \) is monotonic as \( H' \) increases. Then (iii) gives a faster rate than (ii) and linear GNNs. For example, if the globally optimal value decreases as linear GNNs get deeper. i.e., \( L_0^* \geq L_1^* \geq \cdots \geq L_H^* \), or vice versa, \( L_0^* \leq L_1^* \leq \cdots \leq L_H^* \).
Optimization of Graph Neural Networks: Implicit Acceleration by Skip Connections and More Depth

Figure 3. Empirical validation of assumptions for global convergence of multiscale linear GNNs. Left panel confirms the graph condition \( \sigma_{\min}^2((G_H)_x) > 0 \) for Cora and Citeseer, and for GCN and GIN. Middle panel shows the time-dependent \( \lambda_T^{(1:H)} \) for one training setting (multiscale linear GCN on Cora). Each point in right panel is \( \lambda_T^{(1:H)} \) at the last iteration for different training settings.

\[ \int_{t=0}^{\infty} \lambda_T = \int_{t=0}^{\infty} \frac{dt}{L(W_t, B_t) - L^*_H} = \int_{t=0}^{\infty} \frac{dt}{\sigma_{\min}^2((X(S^h)_{x})_T)} \]

where \( l = 0 \) if \( L_0^* \geq L_1^* \geq \cdots \geq L_H^* \) and \( l = H \) if \( L_0^* \leq L_1^* \leq \cdots \leq L_H^* \). Moreover, if the globally optimal value does not change with respect to the depth as \( L_1^* = L_2^* = \cdots = L_H^* \), then we have

\[ L(W_t, B_T) - L^*_H \leq \frac{e^{-4\sum_{k=0}^{H} \sigma_{\min}^2((X(S^k)_{x})_T)}}{2H} \]

We obtain a faster rate for multiscale linear GNNs than for linear GNNs, as \( e^{-4\sum_{k=0}^{H} \sigma_{\min}^2((X(S^k)_{x})_T)} \leq e^{-4\lambda_{T}^{(1:H)} \sigma_{\min}^2((X(S^h)_{x})_T)} \). Interestingly, unlike linear GNNs, multiscale linear GNNs in this case do not require any condition on initialization to obtain a priori guarantee on global convergence since \( e^{-4\lambda_{T}^{(1:H)} \sigma_{\min}^2((X(S^h)_{x})_T)} \leq e^{-4\lambda_{T}^{(1:0)} \sigma_{\min}^2((X(S^h)_{x})_T)} \) with \( \lambda_{T}^{(0)} = 1 \) and \( (X(S^0))_x = X_x \).

To summarize, we prove global convergence rates for multiscale linear GNNs (Thm. 2.2(i)) and experimentally validate the conditions. Part (ii) addresses boundary cases where the conditions of Part (i) do not hold. Part (iii) gives faster rates assuming monotonic expressive power with respect to depth. So far, we have shown multiscale linear GNNs converge faster than linear GNNs in the case of (iii). Next, we compare the training speed for more general cases.

4. Implicit Acceleration

In this section, we study how the skip connections, depth of GNN, and label distribution may affect the speed of training for GNNs. Similar to previous works (Arora et al., 2018), we compare the training speed by comparing the per step loss reduction \( \frac{d}{dt} L(W_t, B_t) \) for arbitrary differentiable loss functions \( \ell(Y) : \mathbb{R}^m \rightarrow \mathbb{R} \). Smaller \( \frac{d}{dt} L(W_t, B_t) \) implies faster training. Loss reduction offers a complementary view to the convergence rates in Section 3, since it is instantaneous and not an upper bound.

We present an analytical form of the loss reduction \( \frac{d}{dt} L(W_t, B_t) \) for linear GNNs and multiscale linear GNNs. The comparison of training speed then follows from our formula for \( \frac{d}{dt} L(W_t, B_t) \). For better exposition, we first introduce several notations. We let \( \bar{B}^{(1:l)} = B^{(1)}B^{(l-1)} \cdots B^{(l)} \) for all \( l' \) and \( l \) where \( \bar{B}^{(l')} = I \) if \( l' > l \). We also define

\[ J_{(i,l),t} := \left[ \bar{B}^{(1:l-1)} \right] (W_{(l,t)} B^{(1:l+1)}_{(l)})_T, \]

\[ F_{(i,t)} := \left[ \bar{B}^{(1:l)} \right] T \bar{B}^{(1:l)}_t (I_m) \geq 0, \]

\[ V_t := \frac{\partial L(W_t, B_t)}{\partial Y_t}, \]

where \( \bar{Y}_t := f(X, W_t, B_t) \). For any vector \( v \in \mathbb{R}^m \) and positive semidefinite matrix \( M \in \mathbb{R}^{m \times m} \), we use \( \|v\|_M^2 := v^T M v \). Intuitively, \( V_t \) represents the derivative of the loss \( L(W_t, B_t) \) with respect to the model output \( \bar{Y} = f(X, W_t, B_t) \). \( J_{(i,l),t} \) and \( F_{(i,t)} \) represent matrices that describe how the errors are propagated through the weights of the networks.

Theorem 3, proved in Appendix A.6, gives an analytical formula of loss reduction for linear GNNs and multiscale linear GNNs.

**Theorem 3.** For any differentiable loss function \( q \rightarrow \ell(q, Y) \), the following hold for any \( H \geq 0 \) and \( t \geq 0 \):

(i) (Non-multiscale) For \( f \) as in Definition 1:

\[
\frac{d}{dt} L(W_t, B_t) = -\left\| \text{vec} \left[ V_t (X (S^H)_x) \right] \right\|_{F^H, t}^2 - \sum_{i=1}^H \left\| J_{(i,H),t} \text{vec} \left[ V_t (X (S^H)_x) \right] \right\|_2^2.
\]

We use this Mahalanobis norm notation for conciseness without assuming it to be a norm, since \( M \) may be low rank.
Figure 4. **Comparison of the training speed of GNNs.** Left: Multiscale GNNs train faster than non-multiscale GNNs. Middle: Deeper GNNs train faster. Right: GNNs train faster when the labels have signals instead of random noise. The patterns above hold for both ReLU and linear GNNs. Additional results are in Appendix B.

(ii) (Multiscale) For \( f \) as in Definition 2:

\[
\frac{d}{dt} L_2(W_t, B_t) = - \sum_{i=0}^{H} \left\| \text{vec} \left[ V_i(X(S^i)_{+})^T \right] \right\|_2^2, \tag{16}
\]

\[
- \sum_{i=1}^{H} \sum_{l=1}^{H} J_{i,l,t} \text{vec} \left[ V_i(X(S^i)_{+}Z)^T \right] \right\|_2^2.
\]

In what follows, we apply Theorem 3 to predict how different factors affect the training speed of GNNs.

### 4.1. Acceleration with Skip Connections

We first show that multiscale linear GNNs tend to achieve faster loss reduction \( \frac{d}{dt} L_2(W_t, B_t) \) compared to the corresponding linear GNN without skip connections, \( \frac{d}{dt} L_1(W_t, B_t) \). It follows from Theorem 3 that

\[
\frac{d}{dt} L_2(W_t, B_t) - \frac{d}{dt} L_1(W_t, B_t) \leq - \sum_{l=0}^{H-1} \left\| \text{vec} \left[ V_l(X(S^l)_{+}Z)^T \right] \right\|_2^2,
\]

where \( a_i = \sum_{l=1}^{H-1} \left\| J_{i,l,t} \text{vec} [V_l(X(S^l)_{+}Z)] \right\|_2^2 + 2b_i^T a_i \geq 0 \), and \( b_i = J_{i,l,t} \text{vec} [V_l(X(S^l)_{+}Z)] \).

The assumption of \( \sum_{l=1}^{H-1} \left\| J_{i,l,t} \text{vec} [V_l(X(S^l)_{+}Z)] \right\|_2^2 \geq 0 \) is satisfied in various ways: for example, it is satisfied if the last layer's term \( b_i \) and the other layers' terms \( a_i \) are aligned as \( b_i^T a_i \geq 0 \), or if the last layer's term \( b_i \) is dominated by the other layers' terms \( a_i \) as \( 2\|b_i\|_2 \leq \|a_i\|_2 \).

Then equation (17) shows that the multiscale linear GNN decreases the loss value with strictly many more negative terms, suggesting faster training.

Empirically, we indeed observe that multiscale GNNs train faster (Figure 4a), both for (nonlinear) ReLU and linear GNNs. We verify this by training multiscale and non-multiscale, ReLU and linear GCNs on the Cora and Citeseer datasets with cross-entropy loss, learning rate 5e-5, and hidden dimension 32. Results are in Appendix B.

### 4.2. Acceleration with More Depth

Our second finding is that deeper GNNs, with or without skip connections, train faster. For any differentiable loss function \( q \mapsto \ell(q, Y) \), Theorem 3 states that the loss of the multiscale linear GNN decreases as

\[
\frac{d}{dt} L(W_t, B_t) = - \sum_{i=0}^{H} \left\| \text{vec} \left[ V_i(X(S^i)_{+}Z)^T \right] \right\|_2^2, \tag{18}
\]

\[
- \sum_{i=1}^{H} \sum_{l=1}^{H} J_{i,l,t} \text{vec} \left[ V_i(X(S^i)_{+}Z)^T \right] \right\|_2^2.
\]

In equation (18), we can see that the multiscale linear GNN achieves faster loss reduction as depth \( H \) increases. A similar argument applies to non-multiscale linear GNNs.

Empirically too, deeper GNNs train faster (Figure 4b). Again, the acceleration applies to both (nonlinear) ReLU GNNs and linear GNNs. We verify this by training multiscale and non-multiscale, ReLU and linear GCNs with 2, 4, and 6 layers on the Cora and Citeseer datasets with learning rate 5e-5, hidden dimension 32, and cross-entropy loss. Results are in Appendix B.

### 4.3. Label Distribution: Signal vs. Noise

Finally, we studied how the labels affect the training speed. For the loss reduction (15) and (16), we argue that the term of \( V_i(X(S^i)_{+}Z)^T \) tends to be larger for labels \( Y \) that are more correlated with the graph features \( X(S^i)_{+}Z \), e.g., labels are signals instead of “noise”.

Without loss of generality, we assume \( Y \) is normalized, e.g., one-hot labels. Here, \( \frac{\partial L(A_t, B_t)}{\partial Y_t} \) is the derivative of the loss with respect to the model output, e.g., \( V_t = 2(\hat{Y}_t - Y) \) for squared loss. If the rows of \( Y \) are random noise vectors,
We train GNNs with the original labels of the dataset and work is inspired by these works but differs in that our analysis applies to all learnable weights and does not require these specific assumptions, and we study the more complex GNN architecture with skip connections. GNNs consider the interaction of graph structures via the recursive message passing, but such structured, locally varying interaction is not present in feedforward networks. Furthermore, linear feedforward networks, even with skip connections, have the same expressive power as shallow linear models, a crucial condition in previous proofs (Bartlett et al., 2019; Du & Hu, 2019; Arora et al., 2019a; Zou et al., 2020). In contrast, the expressive power of multiscale linear GNNs can change significantly as depth increases. Accordingly, our proofs significantly differ from previous studies.

Another line of works studies the gradient dynamics of neural networks in the neural tangent kernel (NTK) regime (Jacot et al., 2018; Li & Liang, 2018; Allen-Zhu et al., 2019; Arora et al., 2019b; Chizat et al., 2019; Du et al., 2019a; Kawaguchi & Huang, 2019; Nitanda & Suzuki, 2021). With over-parameterization, the NTK remains almost constant during training. Hence, the corresponding neural network is implicitly linearized with respect to random features of the NTK at initialization (Lee et al., 2019; Yehudai & Shamir, 2019; Liu et al., 2020). On the other hand, our work needs to address nonlinear dynamics and changing expressive power.

Learning dynamics and optimization of GNNs. Closely related to our work, Du et al. (2019b); Xu et al. (2021) study the gradient dynamics of GNNs via the Graph NTK but focus on GNNs’ generalization and extrapolation properties. We instead analyze optimization. Only Zhang et al. (2020) also prove global convergence for GNNs, but for the one-hidden-layer case, and they assume a specialized tensor initialization and training algorithms. In contrast, our results work for any finite depth with no assumptions on specialized training. Other works aim to accelerate and stabilize the training of GNNs through normalization techniques (Cai et al., 2020) and importance sampling (Chen et al., 2018a,b; Huang et al., 2018; Chiang et al., 2019; Zou et al., 2019). Our work complements these practical works with a better theoretical understanding of GNN training.

5. Related Work

Theoretical analysis of linearized networks. The theoretical study of neural networks with some linearized components has recently drawn much attention. Tremendous efforts have been made to understand linear feedforward networks, in terms of their loss landscape (Kawaguchi, 2016; Hardt & Ma, 2017; Laurent & Brecht, 2018) and optimization dynamics (Saxe et al., 2014; Arora et al., 2019a; Bartlett et al., 2019; Du & Hu, 2019; Zou et al., 2020). Recent works prove global convergence rates for deep linear networks under certain conditions (Bartlett et al., 2019; Du & Hu, 2019; Arora et al., 2019a; Zou et al., 2020). For example, Arora et al. (2019a) assume the data to be whitened. Zou et al. (2020) fix the weights of certain layers during training. Our work is inspired by these works but differs in that our analysis applies to all learnable weights and does not require...
Acknowledgements

KX and SJ were supported by NSF CAREER award 1553284 and NSF III 1900933. MZ was supported by ODNI, IARPA, via the BETTER Program contract 2019-1905160005. The research of KK was partially supported by the Center of Mathematical Sciences and Applications at Harvard University. The views, opinions, and/or findings contained in this article are those of the author and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the Department of Defense, ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References


Optimization of Graph Neural Networks: Implicit Acceleration by Skip Connections and More Depth


A. Proofs

In this section, we complete the proofs of our theoretical results. We show the proofs of Theorem 1 in Appendix A.1, Proposition 1 in Appendix A.2, Proposition 2 in Appendix A.3, Theorem 2 in Appendix A.4, Proposition 3 in Appendix A.5, and Theorem 3 in Appendix A.6.

Before starting our proofs, we first introduce additional notation used in the proofs. We define the corner cases on the products of $B$ as:

$$
B_{(H)} B_{(H-1)} \cdots B_{(l+1)} := I_{m_l} \quad \text{if } H = l \quad (19)
$$
$$
B_{(H)} B_{(H-1)} \cdots B_{(1)} := I_{m_e} \quad \text{if } H = 0 \quad (20)
$$
$$
B_{(l-1)} B_{(l-2)} \cdots B_{(1)} := I_{m_e} \quad \text{if } l = 1 \quad (21)
$$

Similarly, for any matrices $M_{(l)}$, we define $M_{(l)} M_{(l-1)} \cdots M_{(k)} := I_{m}$ if $l < k$, and $M_{(l)} M_{(l-1)} \cdots M_{(k)} := M_{(k)} = M_{(l)}$ if $l = k$. Given a scalar-valued variable $a \in \mathbb{R}$ and a matrix $M \in \mathbb{R}^{d \times d'}$, we define

$$
\frac{\partial a}{\partial M} = \begin{bmatrix} \frac{\partial a}{\partial M_{11}} & \cdots & \frac{\partial a}{\partial M_{1d'}} \\ \vdots & \ddots & \vdots \\ \frac{\partial a}{\partial M_{d1}} & \cdots & \frac{\partial a}{\partial M_{dd'}} \end{bmatrix} \in \mathbb{R}^{d \times d'},
$$

(22)

where $M_{ij}$ represents the $(i, j)$-th entry of the matrix $M$. Given a vector-valued variable $a \in \mathbb{R}^d$ and a column vector $b \in \mathbb{R}^{d'}$, we let

$$
\frac{\partial a}{\partial b} = \begin{bmatrix} \frac{\partial a}{\partial b_1} & \cdots & \frac{\partial a}{\partial b_{d'}} \end{bmatrix} \in \mathbb{R}^{d \times d'},
$$

(23)

where $b_i$ represents the $i$-th entry of the column vector $b$. Similarly, given a vector-valued variable $a \in \mathbb{R}^d$ and a row vector $b \in \mathbb{R}^{1 \times d'}$, we write

$$
\frac{\partial a}{\partial b} = \begin{bmatrix} \frac{\partial a}{\partial b_{11}} & \cdots & \frac{\partial a}{\partial b_{1d'}} \\ \vdots & \ddots & \vdots \\ \frac{\partial a}{\partial b_{d1}} & \cdots & \frac{\partial a}{\partial b_{dd'}} \end{bmatrix} \in \mathbb{R}^{d \times d'},
$$

(24)

where $b_{1i}$ represents the $i$-th entry of the row vector $b$. Finally, we recall the definition of the Kronecker product product of two matrices: for matrices $M \in \mathbb{R}^{d_M \times d_M'}$ and $\tilde{M} \in \mathbb{R}^{d_M' \times d_M}$,

$$
M \otimes \tilde{M} = \begin{bmatrix} M_{11} \tilde{M} & \cdots & M_{1d_M'} \tilde{M} \\ \vdots & \ddots & \vdots \\ M_{d_M1} \tilde{M} & \cdots & M_{d_Md_M'} \tilde{M} \end{bmatrix} \in \mathbb{R}^{d_M \times d_M' \times d_M \times d_M'}.
$$

(25)

A.1. Proof of Theorem 1

We begin with a proof overview of Theorem 1. We first relate the gradients $\nabla W(H) L$ and $\nabla B_{(l)} L$ to the gradient $\nabla (H) L$, which is defined by

$$
\nabla (H) L(W, B) := \frac{\partial L(W, B)}{\partial Y} (X(S^H)_{\cdot L})^T \in \mathbb{R}^{m_y \times m_e}.
$$

Using the proven relation of $\nabla W(H) L$, $\nabla B_{(l)} L$ and $\nabla (H) L$, we first analyze the dynamics induced in the space of $W_{(l)} B_{(l)} B_{(l-1)} \cdots B_{(1)}$ in Appendix A.1.1, and then the dynamics induced in the space of loss value $L(W, B)$ in Appendix A.1.2. Finally, we complete the proof by using the assumption of employing the square loss in Appendix A.1.3.

Let $W_{(H)} = W$ (during the proof of Theorem 1). We first prove the relationship of the gradients $\nabla W(H) L$, $\nabla B_{(l)} L$ and $\nabla (H) L$ in the following lemma:
Lemma 1. Let $f$ be an $H$-layer linear GNN and $\ell(q, Y) = \|q - Y\|_F^2$ where $q, Y \in \mathbb{R}^{m_y \times n}$. Then, for any $(W, B)$,

$$\nabla_{W(H)} L(W, B) = \nabla_{(H)} L(W, B)(B(H)B(H-1) \cdots B(1))^\top \in \mathbb{R}^{m_y \times m_l},$$

(26)

and

$$\nabla_{B(i)} L(W, B) = (W(H)B(H)B(H-1) \cdots B(i+1))^\top \nabla_{(H)} L(W, B)(B(i-1)B(i-2) \cdots B(1))^\top \in \mathbb{R}^{m_l \times m_{m-1}},$$

(27)

Proof of Lemma 1. From Definition 1, we have $\hat{Y} = f(X, W, B) = W(H)(X(H))_{s} Z$ where $X(l) = B(l)X(l-1)S$. Using this definition, we can derive the formula of $\frac{\partial \text{vec}[^{\hat{Y}}]}{\partial \text{vec}[W(H)]} \in \mathbb{R}^{m_y \times m_y \times m}$ as:

$$\frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[W(H)]} = \frac{\partial}{\partial \text{vec}[W(H)]} \text{vec}[W(H)(X(H))_{s} Z]$$

$$= \frac{\partial}{\partial \text{vec}[W(H)]} [(X(H))_{s} Z]^\top \otimes I_{m_y} \text{vec}[W(H)] = [(X(H))_{s} Z]^\top \otimes I_{m_y} \in \mathbb{R}^{m_y \times m_y \times m}$$

(28)

We will now derive the formula of $\frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[B(i)]} \in \mathbb{R}^{m_y \times m_l \times m_{m-1}}$:

$$\frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[B(i)]} = \frac{\partial}{\partial \text{vec}[B(i)]} \text{vec}[W(H)(X(H))_{s} Z]$$

$$= \frac{\partial}{\partial \text{vec}[B(i)]} [I_n \otimes W(H)] \text{vec}[(X(H))_{s} Z]$$

$$= [I_n \otimes W(H)] \frac{\partial \text{vec}[(X(H))_{s} Z]}{\partial \text{vec}[X(l)]}$$

$$= [I_n \otimes W(H)] \frac{\partial \text{vec}[(X(H))_{s} Z]}{\partial \text{vec}[X(l)]} \frac{\partial \text{vec}[X(l)]}{\partial \text{vec}[B(i)]}$$

$$= [I_n \otimes W(H)] \frac{\partial \text{vec}[(X(H))_{s} Z]}{\partial \text{vec}[X(l)]} \frac{\partial \text{vec}[B(l)X(l-1)S]}{\partial \text{vec}[B(l)]}$$

$$= [I_n \otimes W(H)] \frac{\partial \text{vec}[(X(H))_{s} Z]}{\partial \text{vec}[X(l)]} \frac{\partial ((X(l-1)S)^\top \otimes I_{m_l}) \text{vec}[B(l)]}{\partial \text{vec}[B(l)]}$$

$$= [I_n \otimes W(H)] \frac{\partial \text{vec}[(X(H))_{s} Z]}{\partial \text{vec}[X(l)]} [(X(l-1)S)^\top \otimes I_{m_l}]$$

(29)

Here, we have that

$$\text{vec}[(X(H))_{s} Z] = \text{vec}[B(H)X(H-1)S_{s} Z] = \text{vec}[(S^\top)_{s} \otimes B(H)] \text{vec}[X(H-1)].$$

(30)

and

$$\text{vec}[X(H)] = \text{vec}[B(H)X(H-1)S_{s} Z] = \text{vec}[S \otimes B(H)] \text{vec}[X(H-1)].$$

(31)

By recursively applying (31), we have that

$$\text{vec}[(X(H))_{s} Z] = \text{vec}[(S^\top)_{s} \otimes B(H)] \text{vec}[S^\top \otimes B(H-1)] \cdots \text{vec}[S^\top \otimes B(l+1)] \text{vec}[X(l)]$$

$$= \text{vec}[(S^{H-1})^\top_{s} \otimes B(H)B(H-1) \cdots B(l+1)] \text{vec}[X(l)],$$

where

$$B(H)B(H-1) \cdots B(l+1) := I_{m_l} \text{ if } H = l.$$

Therefore,
Thus, with \( \frac{\partial \text{vec}[(X_{(H)})_s^T]}{\partial \text{vec}[X_{(l)}]} = \text{vec}[(S^{H-l})^T]_{s^T} \otimes B_{(H)}B_{(H-1)} \cdots B_{(l+1)}] \). (32)

Combining (29) and (32) yields
\[
\frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[B_{(l)}]} = [I_n \otimes W_{(H)}]_{s^T} \frac{\partial \text{vec}[(X_{(H)})_s^T]}{\partial \text{vec}[X_{(l)}]} [(X_{(l-1)} S)^T_{s^T} \otimes I_{m_l}]
\]
\[
= [I_n \otimes W_{(H)}]_{s^T} \text{vec}[(S^{H-l})^T_{s^T} \otimes B_{(H)}B_{(H-1)} \cdots B_{(l+1)}][(X_{(l-1)} S)^T_{s^T} \otimes I_{m_l}]
\]
\[
= [(X_{(l-1)} S^{H-l+1})^T_{s^T} \otimes W_{(H)}B_{(H)}B_{(H-1)} \cdots B_{(l+1)}] \in \mathbb{R}^{m_n \times m_1 m_{l-1}}.
\] (33)

Using (28), we will now derive the formula of \( \nabla_{W_{(H)}} L(W, B) \in \mathbb{R}^{m_n \times m_H} \):
\[
\frac{\partial L(W, B)}{\partial \text{vec}[W_{(H)}]} = \frac{\partial L(W, B)}{\partial \text{vec}[\hat{Y}]} \frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[W_{(H)}]} = \frac{\partial L(W, B)}{\partial \text{vec}[\hat{Y}]} [(X_{(H)})_{s_{s^T}} \otimes I_{m_n}]
\]

Thus, with \( \frac{\partial L(W, B)}{\partial \hat{Y}} \in \mathbb{R}^{m_n \times n} \),
\[
\nabla_{\text{vec}[W_{(H)}]} L(W, B) = \frac{\partial L(W, B)}{\partial \hat{Y}} (X_{(H)})_{s_{s^T}} \in \mathbb{R}^{m_n \times m_H}.
\] (34)

Using (33), we will now derive the formula of \( \nabla_{B_{(l)}} L(W, B) \in \mathbb{R}^{m_l \times m_{l-1}} \):
\[
\frac{\partial L(W, B)}{\partial \text{vec}[B_{(l)}]} = \frac{\partial L(W, B)}{\partial \text{vec}[\hat{Y}]} \frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[B_{(l)}]} = \frac{\partial L(W, B)}{\partial \text{vec}[\hat{Y}]} [(X_{(l-1)} (S^{H-l+1})_{s^T} \otimes W_{(H)}B_{(H)}B_{(H-1)} \cdots B_{(l+1)}]_{s^T}]
\]

Thus, with \( \frac{\partial L(W, B)}{\partial \hat{Y}} \in \mathbb{R}^{m_n \times n} \),
\[
\nabla_{\text{vec}[B_{(l)}]} L(W, B) = \frac{\partial L(W, B)}{\partial \hat{Y}} (X_{(l-1)} (S^{H-l+1})_{s^T})_{s^T} \in \mathbb{R}^{m_l \times m_{l-1}}.
\]
Therefore,
\[ \nabla B_{(l)} L(W, B) = (W(H)B(H)B(H-1) \cdots B(l+1))^\top \frac{\partial L(W, B)}{\partial Y} \left( X_{(l-1)}(S^{l-1}S^{l+1})_x \right)^\top \in \mathbb{R}^{m_1 \times m_1}. \tag{35} \]

With (34) and (35), we are now ready to prove the statement of this lemma by introducing the following notation:
\[ \nabla (l) L(W, B) := \frac{\partial L(W, B)}{\partial Y} \left( X(S^l)_x \right)^\top \in \mathbb{R}^{m_2 \times m_2}. \]

Using this notation along with (34)
\[ \nabla W(\mu) L(W, B) = \frac{\partial L(W, B)}{\partial Y} \left( X(H)_x \right)^\top = \frac{\partial L(W, B)}{\partial Y} \left( B(1)X(H) - B(H)B(H-1) \cdots B(1)X(S^H)_x \right)^\top \]
\[ = \nabla (H) L(W, B)(B(H)B(H-1) \cdots B(1))^\top, \]

Similarly, using (35),
\[ \nabla B_{(l)} L(W, B) = (W(H)B(H)B(H-1) \cdots B(l+1))^\top \frac{\partial L(W, B)}{\partial Y} \left( X_{(l-1)}(S^{l-1}S^{l+1})_x \right)^\top = \left( W(H)B(H)B(H-1) \cdots B(l+1)^\top \right) \frac{\partial L(W, B)}{\partial Y} \left( B(l+1)B(l) \cdots B(1)X(S^H)_x \right)^\top \]
\[ = (W(H)B(H)B(H-1) \cdots B(l+1)) ^\top \nabla (H) L(W, B)(B(l-1)B(l-2) \cdots B(1))^\top \]
where \( B(l-1)B(l-2) \cdots B(1) := I_{m_2} \) if \( l = 1 \).

By using Lemma 1, we complete the proof of Theorem 1 in the following.

A.1.1. Dynamics Induced in the Space of \( W(l)B(l)B(l-1) \cdots B(1) \)

We now consider the dynamics induced in the space of \( W(l)B(l)B(l-1) \cdots B(1) \). We first consider the following discrete version of the dynamics:
\[ W'(H) = W(H) - \alpha \nabla W(H)L(W, B) \]
\[ B'(l) = B(l) - \alpha \nabla B(l) L(W, B). \]

This dynamics induces the following dynamics:
\[ W'(H)B'(H)B'(H-1) \cdots B'(1) = (W(H) - \alpha \nabla W(H)L(W, B))(B(H) - \alpha \nabla B(H)L(W, B)) \cdots (B(1) - \alpha \nabla B(1)L(W, B)). \]

Define
\[ Z(H) := W(H)B(H)B(H-1) \cdots B(1), \]
and
\[ Z'(H) := W'(H)B'(H)B'(H-1) \cdots B'(1). \]
Then, we can rewrite
\[ Z'(H) = (W(H) - \alpha \nabla W(H)L(W, B))(B(H) - \alpha \nabla B(H)L(W, B)) \cdots (B(1) - \alpha \nabla B(1)L(W, B)). \]
By expanding the multiplications, this can be written as:

\[ Z'_H = Z_H - \alpha \nabla_{W(H)} L(W, B) B_H \cdots B_{(1)} - \alpha \sum_{i=1}^{H} W_{(H)} B_{(H)} \cdots B_{(i+1)} \nabla_{B_{(i)}} L(W, B) B_{(i-1)} \cdots B_{(1)} + O(\alpha^2). \]

By vectorizing both sides,

\[ \text{vec}[Z'_H] - \text{vec}[Z_H] = -\alpha \text{vec}[\nabla_{W(H)} L(W, B) B_H \cdots B_{(1)}] - \alpha \sum_{i=1}^{H} \text{vec}[W_{(H)} B_{(H)} \cdots B_{(i+1)} \nabla_{B_{(i)}} L(W, B) B_{(i-1)} \cdots B_{(1)}] + O(\alpha^2). \]

Here, using the formula of \( \nabla_{W(H)} L(W, B) \) and \( \nabla_{B(H)} L(W, B) \), we have that

\[ \text{vec}[\nabla_{W(H)} L(W, B) B_H \cdots B_{(1)}] = \text{vec}[\nabla (H) L(W, B)(B_H \cdots B_{(1)})^\top B_H \cdots B_{(1)}] = [(B_H \cdots B_{(1)})^\top B_H \cdots B_{(1)} \otimes I_{m_y}] \text{vec}[\nabla (H) L(W, B)], \]

and

\[ \sum_{i=1}^{H} \text{vec}[W_{(H)} B_{(H)} \cdots B_{(i+1)} \nabla_{B_{(i)}} L(W, B) B_{(i-1)} \cdots B_{(1)}] = \sum_{i=1}^{H} [(B_{(i-1)} \cdots B_{(1)})^\top B_{(i-1)} \cdots B_{(1)} \otimes W_{(H)} B_{(H)} \cdots B_{(i+1)} (W_{(H)} B_{(H)} \cdots B_{(i+1)})^\top] \text{vec}[\nabla (H) L(W, B)]. \]

Summarizing above,

\[ \text{vec}[Z'_H] - \text{vec}[Z_H] = -\alpha [(B_H \cdots B_{(1)})^\top B_H \cdots B_{(1)} \otimes I_{m_y}] \text{vec}[\nabla (H) L(W, B)] - \alpha \sum_{i=1}^{H} [(B_{(i-1)} \cdots B_{(1)})^\top B_{(i-1)} \cdots B_{(1)} \otimes W_{(H)} B_{(H)} \cdots B_{(i+1)} (W_{(H)} B_{(H)} \cdots B_{(i+1)})^\top] \text{vec}[\nabla (H) L(W, B)] + O(\alpha^2). \]

Therefore, the induced continuous dynamics of \( Z_H = W_{(H)} B_{(H)} B_{(H-1)} \cdots B_{(1)} \) is

\[ \frac{d}{dt} \text{vec}[Z_H] = -F_H \text{vec}[\nabla (H) L(W, B)] - \left( \sum_{i=1}^{H} J_{(i,H)}^\top J_{(i,H)} \right) \text{vec}[\nabla (H) L(W, B)], \]

where

\[ F_H = [(B_H \cdots B_{(1)})^\top B_H \cdots B_{(1)} \otimes I_{m_y}], \]

and

\[ J_{(i,H)} = [B_{(i-1)} \cdots B_{(1)} \otimes (W_{(H)} B_{(H)} \cdots B_{(i+1)})^\top]. \]

This is because

\[ J_{(i,H)}^\top J_{(i,H)} = [(B_{(i-1)} \cdots B_{(1)})^\top \otimes W_{(H)} B_{(H)} \cdots B_{(i+1)}][B_{(i-1)} \cdots B_{(1)} \otimes (W_{(H)} B_{(H)} \cdots B_{(i+1)})^\top] = [(B_{(i-1)} \cdots B_{(1)})^\top B_{(i-1)} \cdots B_{(1)} \otimes W_{(H)} B_{(H)} \cdots B_{(i+1)} (W_{(H)} B_{(H)} \cdots B_{(i+1)})^\top]. \]
A.1.2. Dynamics induced in the space of loss value $L(W, B)$

We now analyze the dynamics induced in the space of loss value $L(W, B)$. Using chain rule,

$$\frac{d}{dt} L(W, B) = \frac{d}{dt} L_0(Z(H)) = \frac{\partial L_0(Z(H))}{\partial \text{vec}[Z(H)]} \frac{d \text{vec}[Z(H)]}{dt},$$

where

$$L_0(Z(H)) = \ell(f_0(X, Z(H)) \circ Y), \quad f_0(X, Z(H)) = Z(H)X^{-1}Y, \quad \text{and} \quad Z(H) = W(1)^{H}B(1)^{H}B(H-1) \cdots B(1).$$

Since $f_0(X, Z(H)) = f(X, W, B) = \hat{Y}$ and $L_0(Z(H)) = L(W, B)$, we have that

$$\left(\frac{\partial L_0(Z(H))}{\partial \text{vec}[Z(H)]}\right)^T = \left(\frac{\partial L(W, B)}{\partial \text{vec}[\hat{Y}]} \frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[Z(H)]}\right)^T = \left(\frac{\partial L(W, B)}{\partial \text{vec}[\hat{Y}]} \left(\frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[Z(H)]}[(X(S^H)_{+1})^\top \otimes I_{m_y}] \text{vec}[Z(H)]\right)\right)^T = \text{vec} \left[\frac{\partial L(W, B)}{\partial \text{vec}[Z(H)]}(X(S^H)_{+1})^\top\right]$$

Combining these,

$$\frac{d}{dt} L(W, B) = \text{vec}[\nabla(H)L(W, B)]^T \frac{d \text{vec}[Z(H)]}{dt}$$

$$= - \text{vec}[\nabla(H)L(W, B)]^T F(H) \text{vec}[\nabla(H)L(W, B)] - \sum_{i=1}^{H} \text{vec}[\nabla(H)L(W, B)]^T J_{(i, H)}^\top J_{(i, H)} \text{vec}[\nabla(H)L(W, B)]$$

$$= - \text{vec}[\nabla(H)L(W, B)]^T F(H) \text{vec}[\nabla(H)L(W, B)] - \sum_{i=1}^{H} \|J_{(i, H)} \text{vec}[\nabla(H)L(W, B)]\|_2^2.$$}

Therefore,

$$\frac{d}{dt} L(W, B) = - \text{vec}[\nabla(H)L(W, B)]^T F(H) \text{vec}[\nabla(H)L(W, B)] - \sum_{i=1}^{H} \|J_{(i, H)} \text{vec}[\nabla(H)L(W, B)]\|_2^2 \tag{36}$$

Since $F(H)$ is real symmetric and positive semidefinite,

$$\frac{d}{dt} L(W, B) \leq -\lambda_{\min}(F(H)) \|\text{vec}[\nabla(H)L(W, B)]\|_2^2 - \sum_{i=1}^{H} \|J_{(i, H)} \text{vec}[\nabla(H)L(W, B)]\|_2^2,$$

With $\lambda_{W,B} = \lambda_{\min}(F(H))$,

$$\frac{d}{dt} L(W, B) \leq -\lambda_{W,B} \|\text{vec}[\nabla(H)L(W, B)]\|_2^2 - \sum_{i=1}^{H} \|J_{(i, H)} \text{vec}[\nabla(H)L(W, B)]\|_2^2 \tag{37}$$
A.1.3. COMPLETING THE PROOF BY USING THE ASSUMPTION OF THE SQUARE LOSS

Using the assumption that \( L(W, B) = \ell(f(X, W, B)_{\mathcal{I}}, Y) = \|f(X, W, B)_{\mathcal{I}} - Y\|_F^2 \) with \( \bar{Y} = f(X, W, B)_{\mathcal{I}} \), we have

\[
\frac{\partial L(W, B)}{\partial \bar{Y}} = \frac{\partial}{\partial \bar{Y}} \|\bar{Y} - Y\|_F^2 = 2(\bar{Y} - Y) \in \mathbb{R}^{m_y \times n},
\]

and

\[
\text{vec}[\nabla_H (L(W, B))] = \text{vec} \left[ \frac{\partial L(W, B)}{\partial \bar{Y}} (X(S^H)_{\mathcal{I}})^\top \right] = 2 \text{vec} \left[ (\bar{Y} - Y)(X(S^H)_{\mathcal{I}})^\top \right] = 2[X(S^H)_{\mathcal{I}} \odot I_{m_y}] \text{vec}[\bar{Y} - Y].
\]

Therefore,

\[
\|\text{vec}[\nabla_H (L(W, B))]\|_2^2 = 4 \text{vec}[\bar{Y} - Y]^\top [X(S^H)_{\mathcal{I}}]^\top X(S^H)_{\mathcal{I}} \odot I_{m_y}] \text{vec}[\bar{Y} - Y] \quad (38)
\]

Using (37) and (38),

\[
\frac{d}{dt} L(W, B) \leq -\lambda_W \|\text{vec}[\nabla_H (L(W, B))]\|_2^2 - \sum_{i=1}^H \|J(i, H) \text{vec}[\nabla_H (L(W, B))]\|_2^2
\]

\[
\leq -4\lambda_W \text{vec}[\bar{Y} - Y]^\top [X(S^H)_{\mathcal{I}}]^\top X(S^H)_{\mathcal{I}} \odot I_{m_y}] \text{vec}[\bar{Y} - Y] - \sum_{i=1}^H \|J(i, H) \text{vec}[\nabla_H (L(W, B))]\|_2^2
\]

\[
= -4\lambda_W \text{vec}[\bar{Y} - Y]^\top [\tilde{G}_H^\top \tilde{G}_H \odot I_{m_y}] \text{vec}[\bar{Y} - Y] - \sum_{i=1}^H \|J(i, H) \text{vec}[\nabla_H (L(W, B))]\|_2^2
\]

where the last line follows from the following definition:

\[
\tilde{G}_H := X(S^H)_{\mathcal{I}}.
\]

Decompose \( \text{vec}[\bar{Y} - Y] \) as \( \text{vec}[\bar{Y} - Y] = v + v^\perp \), where \( v = P_{\tilde{G}_H \odot I_{m_y}} \text{vec}[\bar{Y} - Y], v^\perp = (I_{m_y} - P_{\tilde{G}_H \odot I_{m_y}}) \text{vec}[\bar{Y} - Y], \) and \( P_{\tilde{G}_H \odot I_{m_y}} \in \mathbb{R}^{m_y \times m_y} \) represents the orthogonal projection onto the column space of \( \tilde{G}_H \odot I_{m_y} \in \mathbb{R}^{m_y \times m_y}. \)

Then,

\[
\text{vec}[\bar{Y} - Y]^\top [\tilde{G}_H^\top \tilde{G}_H \odot I_{m_y}] \text{vec}[\bar{Y} - Y] = (v + v^\perp)^\top \left[ \tilde{G}_H^\top \odot I_{m_y} \right] \left[ \tilde{G}_H \odot I_{m_y} \right] (v + v^\perp)
\]

\[
= v^\top \left[ \tilde{G}_H^\top \odot I_{m_y} \right] \left[ \tilde{G}_H \odot I_{m_y} \right] v
\]

\[
\geq \sigma_{\text{min}}^2(\tilde{G}_H)(\|P_{\tilde{G}_H \odot I_{m_y}} \text{vec}[\bar{Y} - Y]\|_2^2
\]

\[
= \sigma_{\text{min}}^2(\tilde{G}_H)(\|P_{\tilde{G}_H \odot I_{m_y}} \text{vec}[\bar{Y}] - P_{\tilde{G}_H \odot I_{m_y}} \text{vec}[Y]\|_2^2
\]

\[
= \sigma_{\text{min}}^2(\tilde{G}_H)(\|\text{vec}[\bar{Y}] \|_2^2 - \|\text{vec}[Y]\|_2^2)
\]

\[
= \sigma_{\text{min}}^2(\tilde{G}_H)((\|\text{vec}[\bar{Y} - Y]\|_2^2 - \|\text{vec}[Y]\|_2^2)
\]

\[
\geq \sigma_{\text{min}}^2(\tilde{G}_H)(\|\text{vec}[\bar{Y} - Y]\|_2^2 - \|\text{vec}[Y]\|_2^2)
\]

where we used the fact that the singular values of \( \tilde{G}_H \odot I_{m_y} \) are products of singular values of \( \tilde{G}_H \) and \( I_{m_y} \).

By noticing that \( L(W, B) = \|\text{vec}[\bar{Y} - Y]\|_2^2 \) and \( L_H^* = \|I_{m_y} - P_{\tilde{G}_H \odot I_{m_y}} \text{vec}[Y]\|_2^2 \),

\[
\text{vec}[\bar{Y} - Y]^\top [\tilde{G}_H \tilde{G}_H \odot I_{m_y}] \text{vec}[\bar{Y} - Y] \geq \sigma_{\text{min}}^2(\tilde{G}_H)(L(W, B) - L_H^*).
\]
Therefore,
\[
\frac{d}{dt} L(W, B) \leq -4\lambda_{W,B} \vec{Y}^T \left[ \tilde{G}_H^T \tilde{G}_H \otimes I_{m_y} \right] \vec{Y} - \sum_{i=1}^{H} \left\| J_{(i,H)} \vec{\nabla}_{(H)} L(W, B) \right\|_2^2 \\
\leq -4\lambda_{W,B} \sigma_{\text{min}}^2 (\tilde{G}_H) (L(W, B) - L^*_H) - \sum_{i=1}^{H} \left\| J_{(i,H)} \vec{\nabla}_{(H)} L(W, B) \right\|_2^2
\]

Since \( \frac{d}{dt} L^*_H = 0 \),

\[
\frac{d}{dt} (L(W, B) - L^*_H) \leq -4\lambda_{W,B} \sigma_{\text{min}}^2 (\tilde{G}_H) (L(W, B) - L^*_H) - \sum_{i=1}^{H} \left\| J_{(i,H)} \vec{\nabla}_{(H)} L(W, B) \right\|_2^2
\]

By defining \( \mathbf{L} = L(W, B) - L^*_H \),

\[
\frac{d\mathbf{L}}{dt} \leq -4\lambda_{W,B} \sigma_{\text{min}}^2 (\tilde{G}_H) \mathbf{L} - \sum_{i=1}^{H} \left\| J_{(i,H)} \vec{\nabla}_{(H)} L(W, B) \right\|_2^2
\]

(39)

Since \( \frac{d}{dt} \mathbf{L} \leq 0 \) and \( \mathbf{L} \geq 0 \), if \( \mathbf{L} = 0 \) at some time \( \bar{t} \), then \( \mathbf{L} = 0 \) for any time \( t \geq \bar{t} \). Therefore, if \( \mathbf{L} = 0 \) at some time \( \bar{t} \), then we have the desired statement of this theorem for any time \( t \geq \bar{t} \). Thus, we can focus on the time interval \([0, \bar{t}]\) such that \( \mathbf{L} > 0 \) for any time \( t \in [0, \bar{t}] \) (here, it is allowed to have \( \bar{t} = \infty \)). Thus, focusing on the time interval with \( \mathbf{L} > 0 \), equation (39) implies that

\[
\frac{1}{\mathbf{L}} \frac{d\mathbf{L}}{dt} \leq -4\lambda_{W,B} \sigma_{\text{min}}^2 (\tilde{G}_H) - \frac{1}{\mathbf{L}} \sum_{i=1}^{H} \left\| J_{(i,H)} \vec{\nabla}_{(H)} L(W, B) \right\|_2^2
\]

By taking integral over time

\[
\int_0^T \frac{1}{\mathbf{L}} \frac{d\mathbf{L}}{dt} dt \leq - \int_0^T 4\lambda_{W,B} \sigma_{\text{min}}^2 (\tilde{G}_H) dt - \int_0^T \frac{1}{\mathbf{L}} \sum_{i=1}^{H} \left\| J_{(i,H)} \vec{\nabla}_{(H)} L(W, B) \right\|_2^2 dt
\]

By using the substitution rule for integrals, \( \int_0^T \frac{1}{\mathbf{L}} \frac{d\mathbf{L}}{dt} dt = \int_{\mathbf{L}_0}^{\mathbf{L}_T} \frac{1}{\mathbf{L}} d\mathbf{L} = \log(\mathbf{L}_T) - \log(\mathbf{L}_0) \), where \( \mathbf{L}_0 = L(W_0, B_0) - L^* \) and \( \mathbf{L}_T = L(W_T, B_T) - L^*_H \). Thus,

\[
\log(\mathbf{L}_T) - \log(\mathbf{L}_0) \leq -4\sigma_{\text{min}}^2 (\tilde{G}_H) \int_0^T \lambda_{W,B} dt - \int_0^T \frac{1}{\mathbf{L}} \sum_{i=1}^{H} \left\| J_{(i,H)} \vec{\nabla}_{(H)} L(W, B) \right\|_2^2 dt
\]

which implies that

\[
\mathbf{L}_T \leq e^{\log(\mathbf{L}_0) - 4\sigma_{\text{min}}^2 (\tilde{G}_H) \int_0^T \lambda_{W,B} dt - \int_0^T \frac{1}{\mathbf{L}} \sum_{i=1}^{H} \left\| J_{(i,H)} \vec{\nabla}_{(H)} L(W, B) \right\|_2^2 dt}
\]

\[
= \mathbf{L}_0 e^{-4\sigma_{\text{min}}^2 (\tilde{G}_H) \int_0^T \lambda_{W,B} dt - \int_0^T \frac{1}{\mathbf{L}} \sum_{i=1}^{H} \left\| J_{(i,H)} \vec{\nabla}_{(H)} L(W, B) \right\|_2^2 dt}
\]

By recalling the definition of \( \mathbf{L} = L(W, B) - L^*_H \) and that \( \frac{d}{dt} \mathbf{L} \leq 0 \), we have that if \( L(W_T, B_T) - L^*_H > 0 \), then \( L(W_t, B_t) - L^*_H > 0 \) for all \( t \in [0, T] \), and

\[
L(W_T, B_T) - L^*_H \leq (L(W_0, B_0) - L^*_H) e^{-4\sigma_{\text{min}}^2 (\tilde{G}_H) \int_0^T \lambda_{W,B} dt - \int_0^T \frac{1}{\mathbf{L}} \sum_{i=1}^{H} \left\| J_{(i,H)} \vec{\nabla}_{(H)} L(W, B) \right\|_2^2 dt}.
\]

(40)

Using the property of Kronecker product,

\[
\lambda_{\text{min}}((B_{(H),t} \cdots B_{(1),t})^T B_{(H),t} \cdots B_{(1),t} \otimes I_{m_y})) = \lambda_{\text{min}}((B_{(H),t} \cdots B_{(1),t})^T B_{(H),t} \cdots B_{(1),t}),
\]
which implies that $$\lambda_T^{(H)} = \inf_{t \in [0, T]} \lambda_{W_t, B_t}$$. Thus, by noticing that

$$\int_0^T \frac{1}{L(W_t, B_t) - L_H} \sum_{i=1}^H \|J(i, H) \text{vec}[\nabla_H L(W_t, B_t)]\|_2^2 \, dt \geq 0$$, equation (40) implies that

$$L(W_T, B_T) - L_H^* \leq (L(W_0, B_0) - L_H^*) e^{-4\lambda_Y^{(H)} \sigma_{\min}(G_H)^T T - \int_0^T \frac{1}{L(W_t, B_t) - L_H} \sum_{i=1}^H \|J(i, H) \text{vec}[\nabla_H L(W_t, B_t)]\|_2^2 \, dt}

\leq (L(W_0, B_0) - L_H^*) e^{-4\lambda_Y^{(H)} \sigma_{\min}(G_H)^T T}

= (L(W_0, B_0) - L_H^*) e^{-4\lambda_Y^{(H)} \sigma_{\min}(X(S^H)_\# 2)^T T}

\square

A.2. Proof of Proposition 1

From Definition 4, we have that $$\sigma_{\min}(\tilde{B}^{(1:H)}) = \sigma_{\min}(B(H) B(H_{-1}) \cdots B(1)) \geq \gamma$$ for all $$(W, B)$$ such that $$L(W, B) \leq L(W_0, B_0)$$. From equation (37) in the proof of Theorem 1, it holds that $$\frac{d}{dt} L(W_t, B_t) \leq 0$$ for all $$t$$. Thus, we have that $$L(W_t, B_t) \leq L(W_0, B_0)$$ and hence $$\sigma_{\min}(\tilde{B}^{(1:H)}) \geq \gamma$$ for all $$t$$. Under this problem setting ($$m_H \geq m_x$$), this implies that $$\lambda_{\min}(\tilde{B}_t^{(1:H)}) \geq \gamma^2$$ for all $$t$$ and thus $$\lambda_T^{(H)} \geq \gamma^2$$.

A.3. Proof of Proposition 2

We first give the complete version of Proposition 2. Proposition 4 is the formal version of Proposition 2 and shows that our singular margin generalizes deficiency margin proposed in Arora et al. (2019a). Using the deficiency margin assumption, Arora et al. (2019a) analyzed the following optimization problem:

$$\min_{\tilde{W}} L(W_1, \ldots, \tilde{W}_{H+1}) := \frac{1}{2} \|\tilde{W}_{(H+1)} \tilde{W}_{(H)} \cdots \tilde{W}_{(1)} - \tilde{\Phi}\|_F^2$$

where $$\tilde{\Phi} \in \mathbb{R}^{\tilde{m}_y \times \tilde{m}_x}$$ is a target matrix and the last equality follows from $$\|M\|_F = \|M^T\|_F$$ for any matrix $$M$$ by the definition of the Frobenius norm. Therefore, this optimization problem (41) from the previous work is equivalent to the following optimization problem in our notation:

$$\min_{W, B} L(W, B) := \frac{1}{2} \|WB(H) B(H_{-1}) \cdots B_{(1)} - \Phi\|_F^2$$

where $$WB(H) B(H_{-1}) \cdots B_{(1)}$$ is the end-to-end matrix (i.e., $$W = \tilde{W}_{(H+1)}$$ with $$B_{(1)} = \tilde{W}_{(1)}$$ and $$\Phi = \tilde{\Phi}$$ if $$\tilde{m}_y \geq \tilde{m}_x$$, and $$WB(H) B(H_{-1}) \cdots B_{(1)}$$ is the transpose of the above matrix with $$\tilde{m}_y \leq \tilde{m}_x$$).

To solve this optimization problem, Arora et al. (2019a) showed that the initialization $$(0, 0)$$ has deficiency margin c > 0 if the end-to-end matrix $$W_0 B_0$$ has deficiency margin c > 0 with respect to the target $$\Phi$$ (Arora et al., 2019a, Definition 2): i.e., Arora et al. (2019a) assumed that the initialization $$(0, 0)$$ has deficiency margin c > 0 (as it is also invariant to the transpose of $$\tilde{W}_{(H+1)} \tilde{W}_{(H)} \cdots \tilde{W}_{(1)} - \tilde{\Phi}$$).

**Proposition 4.** Consider the optimization problem in (Arora et al., 2019a) by setting $$\frac{1}{2} X^T I = S = I, I = [n], f(X, W, B) = WB(H) B(H_{-1}) \cdots B_{(1)}$$, and $$\ell(q, \Phi) = \frac{1}{2} \|q - \Phi\|_F^2$$, where $$\Phi \in \mathbb{R}^{m_y \times m_x}$$ is a target matrix with $$m_y \geq m_x$$ without loss of generality (since the transpose of the two dimensions leads to the equivalent optimization problem under this setting: see above). Then, if an initialization $$(W_0, B_0)$$ has deficiency margin c > 0, it has singular margin $$\gamma > 0$$.

**Proof of Proposition 4.** By the definition of the deficiency margin (Arora et al., 2019a, Definition 2) and its consequence (Arora et al., 2019a, Claim 1), if an initialization $$(W_0, B_0)$$ has deficiency margin c > 0, then any pair $$(W, B)$$ for which $$L(W, B) \leq L(W_0, B_0)$$ satisfies $$\sigma_{\min}(WB(H) B(H_{-1}) \cdots B_{(1)}) \geq c > 0$$. Since the number of nonzero singular values is
equal to the matrix rank, this implies that \( \text{rank}(WB(H)B(H-1) \cdots B(1)) \geq \min(m_y, m_x) \) for any pair \((W, B)\) for which \( L(W, B) \leq L(W_0, B_0) \). Since \( \text{rank}(MM^\top) \leq \min(\text{rank}(M), \text{rank}(M^\top)) \), this implies that

\[
m_H \geq \min(m_y, m_x), \quad (44)
\]

(as well as \( m_l \geq \min(m_y, m_x) \) for all \( l \)), and that for any pair \((W, B)\) for which \( L(W, B) \leq L(W_0, B_0) \),

\[
m_x = \min(m_y, m_x) \leq \text{rank}(WB(H)B(H-1) \cdots B(1)) \leq \text{min}(\text{rank}(W), \text{rank}(B(H)B(H-1) \cdots B(1))) \leq \text{rank}(B(H)B(H-1) \cdots B(1)) \leq m_x.
\]

This shows that \( \text{rank}(B(H)B(H-1) \cdots B(1)) = m_x \) for any pair \((W, B)\) for which \( L(W, B) \leq L(W_0, B_0) \). Since \( m_H \geq m_x \) from (44) and the number of nonzero singular values is equal to the matrix rank, this implies that \( \sigma_{\min}(B(H)B(H-1) \cdots B(1)) \geq \gamma \) for some \( \gamma > 0 \) for any pair \((W, B)\) for which \( L(W, B) \leq L(W_0, B_0) \). Thus, if an initialization \((W_0, B_0)\) has deficiency margin \( c > 0 \), then it has singular margin \( \gamma > 0 \).

\[
\square
\]

**A.4. Proof of Theorem 2**

This section completes the proof of Theorem 2. We compute the derivatives of the output of multiscale linear GNN with respect to the parameters \( W(i) \) and \( B(i) \) in Appendix A.4.1. Then using these derivatives, we compute the gradient of the loss with respect to \( W(i) \) in Appendix A.4.2 and \( B(i) \) in Appendix A.4.3. We then rearrange the formula of the gradients such that they are related to the formula of \( \nabla_l \) in Appendices A.4.4. Using the proven relation, we first analyze the dynamics induced in the space of \( W(i)B(i)B(l-1) \cdots B(1) \) in Appendix A.4.5, and then the dynamics induced into the space of loss value \( L(W, B) \) in Appendix A.4.6. Finally, we complete the proof by using the assumption of using the square loss in Appendices A.4.7–A.4.10. In the following, we first prove the statement for the case of \( I = [n] \) for the simplicity of notation and then prove the statement for the general case afterwards.

**A.4.1. Derivation of formula for** \( \frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[W(i)]} \in \mathbb{R}^{m_y \times m_y \times m_l} \) \( \text{AND} \ \frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[B(i)]} \in \mathbb{R}^{m_y \times m_y \times m_{l-1}} \)

We can easily compute \( \frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[W(i)]} \) by using the property of the Kronecker product as follows:

\[
\frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[W(i)]} = \frac{\partial}{\partial \text{vec}[W(i)]} \sum_{k=0}^{H} \text{vec}[W(k)X(k)] = \frac{\partial}{\partial \text{vec}[W(i)]} \sum_{k=0}^{H} [X(k) \otimes I_{m_y}] \text{vec}[W(k)]
\]

\[
= [X(i) \otimes I_{m_y}] \in \mathbb{R}^{m_y \times m_y \times m_l}
\]

We now compute \( \frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[B(i)]} \) by using the chain rule and the property of the Kronecker product as follows:

\[
\frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[B(i)]} = \frac{\partial}{\partial \text{vec}[B(i)]} \sum_{k=0}^{H} \text{vec}[W(k)X(k)]
\]

\[
= \frac{\partial}{\partial \text{vec}[B(i)]} \sum_{k=0}^{H} [I_{m_y} \otimes W(k)] \text{vec}[X(k)]
\]

\[
= \sum_{k=0}^{H} [I_{m_y} \otimes W(k)] \frac{\partial \text{vec}[X(k)]}{\partial \text{vec}[B(i)]}
\]

\[
= \sum_{k=0}^{H} [I_{m_y} \otimes W(k)] \frac{\partial \text{vec}[X(k)]}{\partial \text{vec}[X(i)]} \frac{\partial \text{vec}[X(i)]}{\partial \text{vec}[B(i)]}
\]

\[
= \sum_{k=0}^{H} [I_{m_y} \otimes W(k)] \frac{\partial \text{vec}[X(k)]}{\partial \text{vec}[X(i)]} \frac{\partial \text{vec}[B(i)X(l-1)S]}{\partial \text{vec}[B(i)]}
\]
\[
\frac{\partial \text{vec}[X_{(k)}]}{\partial \text{vec}[X_{(l)}]} = \text{vec}[(S^{k-l})^\top \otimes B_{(k)}B_{(k-1)} \cdots B_{(l+1)}]
\]

Combining the above equations yields
\[
\frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[B_{(l)}]} = \sum_{k=l}^H [I_n \otimes W_{(k)}] \frac{\partial \text{vec}[X_{(k)}]}{\partial \text{vec}[X_{(l)}]}[(X_{(l-1)}S)^\top \otimes I_{m_l}]
\]
\[
= \sum_{k=l}^H [I_n \otimes W_{(k)}] \text{vec}[(S^{k-l})^\top \otimes B_{(k)}B_{(k-1)} \cdots B_{(l+1)}][X_{(l-1)}] \otimes I_{m_l}]
\]
\[
= \sum_{k=l}^H [(X_{(l-1)}S^{k-l+1})^\top \otimes W_{(k)}B_{(k-1)}B_{(k-1)} \cdots B_{(l+1)}] \in \mathbb{R}^{m_n \times m_l m_{l-1}}.
\]

A.4.2. DERIVATION OF A FORMULA OF \( \nabla_{W(l)} L(W, B) \in \mathbb{R}^{m_n \times m_l} \)

Using the chain rule and (47), we have that
\[
\frac{\partial L(W, B)}{\partial \text{vec}[W_{(l)}]} = \frac{\partial L(W, B)}{\partial \text{vec}[^{\top}Y]} \frac{\partial \text{vec}[^{\top}Y]}{\partial \text{vec}[W_{(l)}]} = \frac{\partial L(W, B)}{\partial \text{vec}[^{\top}Y]} [X_{(l)}^\top \otimes I_{m_n}].
\]

Thus, with \( \frac{\partial L(W, B)}{\partial Y} \in \mathbb{R}^{m_n \times n} \), by using
\[
\nabla_{\text{vec}[W_{(l)}]} L(W, B) = \left( \frac{\partial L(W, B)}{\partial \text{vec}[W_{(l)}]} \right)^\top
\]
\[
= [X_{(l)} \otimes I_{m_n}] \left( \frac{\partial L(W, B)}{\partial \text{vec}[^{\top}Y]} \right)^\top
\]
\[
= [X_{(l)} \otimes I_{m_n}] \text{vec} \left[ \frac{\partial L(W, B)}{\partial Y} \right]
\]
\[
= \text{vec} \left[ \frac{\partial L(W, B)}{\partial Y} X_{(l)}^\top \right] \in \mathbb{R}^{m_n \times m_l}.
\]

Therefore,
\[
\nabla_{W_{(l)}} L(W, B) = \frac{\partial L(W, B)}{\partial Y} X_{(l)}^\top \in \mathbb{R}^{m_n \times m_l}.
\]
A.4.3. DERIVATION OF A FORMULA OF $\nabla_{B(l)} L(W, B) \in \mathbb{R}^{m_l \times m_{l-1}}$

Using the chain rule and (48), we have that

$$\frac{\partial L(W, B)}{\partial \text{vec}[B(l)]} = \frac{\partial L(W, B)}{\partial \text{vec}[\hat{Y}]} \frac{\partial \text{vec}[\hat{Y}]}{\partial \text{vec}[B(l)]} = \frac{\partial L(W, B)}{\partial \text{vec}[\hat{Y}]} \sum_{k=l}^{H} [(X_{(l-1)} S_{k-l+1}^{T} \otimes W(k) B(k-1) \cdots B(l+1)].$$

Thus, with $\frac{\partial L(W, B)}{\partial Y} \in \mathbb{R}^{m_y \times n}$,

$$\nabla_{\text{vec}[B(l)]} L(W, B) = \left( \frac{\partial L(W, B)}{\partial \text{vec}[B(l)]} \right)^T$$

$$= \sum_{k=l}^{H} [X_{(l-1)} S_{k-l+1}^{T} \otimes (W(k) B(k-1) \cdots B(l+1))^T] \left( \frac{\partial L(W, B)}{\partial \text{vec}[\hat{Y}]} \right)^T$$

$$= \sum_{k=l}^{H} [X_{(l-1)} S_{k-l+1}^{T} \otimes (W(k) B(k-1) \cdots B(l+1))^T] \text{vec} \left[ \frac{\partial L(W, B)}{\partial Y} \right]$$

$$= \sum_{k=l}^{H} \text{vec} \left[ (W(k) B(k-1) \cdots B(l+1))^T \frac{\partial L(W, B)}{\partial Y} (X_{(l-1)} S_{k-l+1}^{T})^T \right] \in \mathbb{R}^{m_l m_{l-1}}.$$

Therefore,

$$\nabla_{B(l)} L(W, B) = \sum_{k=l}^{H} (W(k) B(k-1) \cdots B(l+1))^T \frac{\partial L(W, B)}{\partial Y} (X_{(l-1)} S_{k-l+1}^{T})^T \in \mathbb{R}^{m_l \times m_{l-1}}. \quad (50)$$

A.4.4. RELATING GRADIENTS TO $\nabla_{(l)} L$

We now relate the gradients of the loss to $\nabla_{(l)} L$, which is defined by

$$\nabla_{(l)} L(W, B) := \frac{\partial L(W, B)}{\partial Y} (X S^l)^T \in \mathbb{R}^{m_y \times m_x}.$$

By using this definition and (49), we have that

$$\nabla_{W(l)} L(W, B) = \frac{\partial L(W, B)}{\partial Y} X_{(l)}^T$$

$$= \frac{\partial L(W, B)}{\partial Y} (B(l) X_{(l-1)} S)^T$$

$$= \frac{\partial L(W, B)}{\partial Y} (B(l) B_{(l-1)} \cdots B_{(1)} X S^l)^T$$

$$= \nabla_{(l)} L(W, B) (B(l) B_{(l-1)} \cdots B_{(1)})^T,$$

where $B(l) B_{(l-1)} \cdots B_{(1)} := I_{m_x}$ if $l = 0$. Similarly, by using the definition and (50),

$$\nabla_{B(l)} L(W, B) = \sum_{k=l}^{H} (W(k) B(k-1) \cdots B(l+1))^T \frac{\partial L(W, B)}{\partial Y} (X_{(l-1)} S_{k-l+1}^{T})^T$$

$$= \sum_{k=l}^{H} (W(k) B(k-1) \cdots B(l+1))^T \frac{\partial L(W, B)}{\partial Y} (B(l-1) B_{(l-2)} \cdots B_{(1)} X S_{l-1} S_{k-l+1}^{T})^T$$

$$= \sum_{k=l}^{H} (W(k) B(k-1) \cdots B(l+1))^T \frac{\partial L(W, B)}{\partial Y} (B(l-1) B_{(l-2)} \cdots B_{(1)} X S^k)^T.$$
\[
\n\begin{align*}
\sum_{k=1}^{H} (W(k)B(k)B(k-1) \cdots B(l+1))^T \nabla_{(k)} L(W, B)(B(l-1)B(l-2) \cdots B(1))^T
\end{align*}
\]

where \(B(l-1)B(l-2) \cdots B(1)\) := \(I_{m_z}\) if \(l = 1\). In summary thus far, we have that

\[
\nabla_{W(l)} L(W, B) = \nabla_{(l)} L(W, B)(B(l-1) \cdots B(1))^T \in \mathbb{R}^{m_y \times m_l},
\]

and

\[
\nabla_{B(l)} L(W, B) = \sum_{k=1}^{H} (W(k)B(k)B(k-1) \cdots B(l+1))^T \nabla_{(k)} L(W, B)(B(l-2) \cdots B(1))^T \in \mathbb{R}^{m_l \times m_{l-1}},
\]

where \(\nabla_{(l)} L(W, B) := \frac{\partial L(W, B)}{\partial X^l} \in \mathbb{R}^{m_y \times m_z}\), \(B(k)B(k-1) \cdots B(l) := I_{m_l}\) if \(k = l\), \(B(l)B(l-1) \cdots B(1) := I_{m_z}\) if \(l = 0\), and \(B(l-1)B(l-2) \cdots B(1) := I_{m_z}\) if \(l = 1\).

### A.4.5. Dynamics induced in the space of \(W(l)B(l)B(l-1) \cdots B(1)\)

We now consider the Dynamics induced in the space of \(W(l)B(l)B(l-1) \cdots B(1)\). We first consider the following discrete version of the dynamics:

\[
W'(l) = W(l) - \alpha \nabla_{W(l)} L(W, B)
\]

\[
B'(l) = B(l) - \alpha \nabla_{B(l)} L(W, B).
\]

This dynamics induces the following dynamics:

\[
W'(l)B'(l)B'(l-1) \cdots B'(1) = (W(l) - \alpha \nabla_{W(l)} L(W, B))(B(l) - \alpha \nabla_{B(l)} L(W, B)) \cdots (B(1) - \alpha \nabla_{B(1)} L(W, B)).
\]

Define

\[
Z(l) := W(l)B(l)B(l-1) \cdots B(1)
\]

and

\[
Z'(l) := W'(l)B'(l)B'(l-1) \cdots B'(1).
\]

Then, we can rewrite

\[
Z'(l) = (W(l) - \alpha \nabla_{W(l)} L(W, B))B(l) \cdots (B(1) - \alpha \nabla_{B(1)} L(W, B)).
\]

By expanding the multiplications, this can be written as:

\[
Z'(l) = Z(l) - \alpha \nabla_{W(l)} L(W, B)B(l) \cdots B(1) - \alpha \sum_{i=1}^{l} W(l)B(l) \cdots B(i+1) \nabla_{B(l)} L(W, B)B(l-1) \cdots B(1) + O(\alpha^2)
\]

By vectorizing both sides,

\[
\text{vec}[Z'(l)] - \text{vec}[Z(l)] = -\alpha \text{vec}[\nabla_{W(l)} L(W, B)B(l) \cdots B(1)] - \alpha \sum_{i=1}^{l} \text{vec}[W(l)B(l) \cdots B(i+1) \nabla_{B(l)} L(W, B)B(i-1) \cdots B(1)] + O(\alpha^2)
\]

Here, using the formula of \(\nabla_{W(l)} L(W, B)\) and \(\nabla_{B(l)} L(W, B)\), we have that

\[
\text{vec}[\nabla_{W(l)} L(W, B)B(l) \cdots B(1)] = \text{vec}[\nabla_{(l)} L(W, B)(B(l) \cdots B(1))^T B(l) \cdots B(1)]
\]

\[
= [(B(l) \cdots B(1))^T B(l) \cdots B(1) \otimes I_{m_y}] \text{vec}[\nabla_{(l)} L(W, B)],
\]

and

\[
\sum_{i=1}^{l} \text{vec}[W(l)B(l) \cdots B(i+1) \nabla_{B(l)} L(W, B)B(i-1) \cdots B(1)]
\]
We now analyze the dynamics induced in the space of loss value where

\[ Z \]

This is because

\[ \mathbf{vec} \left[ \sum_{i=1}^{H} \left( (W_l B_l \cdots B_{i+1})^T \nabla_{(k)} L(W, B) (B_{i-1} \cdots B_{(1)})^T B_{(i-1)} \cdots B_{(1)} \right) \right] \]

\[ = \sum_{i=1}^{H} \sum_{k=i}^{H} \mathbf{vec} \left[ (W_l B_l \cdots B_{i+1})^T \nabla_{(k)} L(W, B) (B_{i-1} \cdots B_{(1)})^T B_{(i-1)} \cdots B_{(1)} \right] \]

Summarizing above,

\[ \mathbf{vec} [Z_{(i)}'] - \mathbf{vec} [Z_{(i)}] = -\alpha [\mathbf{vec} [\nabla (B_l \cdots B_{(1)})^T (B_{(i-1)} \cdots B_{(1)} \otimes I_{m_y}) \mathbf{vec} [\nabla_{(i)} L(W, B)]]] \]

\[ - \alpha \sum_{i=1}^{H} \sum_{k=i}^{H} (B_{i-1} \cdots B_{(1)})^T B_{(i-1)} \cdots B_{(1)} \otimes W_l B_l \cdots B_{(i+1)} (W_k B_k \cdots B_{(i+1)})^T \mathbf{vec} [\nabla_{(k)} L(W, B)] \]

\[ + O(\alpha^2) \]

Therefore, the induced continuous dynamics of \( Z_{(i)} = W_l B_l B_{(i-1)} \cdots B_{(1)} \) is

\[ \frac{d}{dt} \mathbf{vec} [Z_{(i)}] = -F_{(i)} \mathbf{vec} [\nabla_{(i)} L(W, B)] - \sum_{i=1}^{H} \sum_{k=i}^{H} J_{(i,k)}^T J_{(i,k)} \mathbf{vec} [\nabla_{(k)} L(W, B)] \]

where

\[ F_{(i)} = [(B_l \cdots B_{(1)})^T B_{(i-1)} \cdots B_{(1)} \otimes I_{m_y}]] \]

and

\[ J_{(i,k)} = [B_{(i-1)} \cdots B_{(1)} \otimes (W_l B_l \cdots B_{(i+1)})^T]. \]

This is because

\[ J_{(i,k)}^T J_{(i,k)} = [(B_{(i-1)} \cdots B_{(1)})^T \otimes W_l B_l \cdots B_{(i+1)}][B_{(i-1)} \cdots B_{(1)} \otimes (W_k B_k \cdots B_{(i+1)})^T] \]

\[ = [(B_{(i-1)} \cdots B_{(1)})^T B_{(i-1)} \cdots B_{(1)} \otimes W_l B_l \cdots B_{(i+1)} (W_k B_k \cdots B_{(i+1)})^T]. \]

A.4.6. DYNAMICS INDUCED IN THE SPACE OF LOSS VALUE \( L(W, B) \)

We now analyze the dynamics induced in the space of loss value \( L(W, B) \). Define

\[ L(W, B) := \ell (f(X, W, B), Y), \]

where \( \ell \) is chosen later. Using chain rule,

\[ \frac{d}{dt} L(W, B) = \frac{d}{dt} L_0(Z_{(H)}, \ldots, Z_{(0)}) \]

\[ = \sum_{i=0}^{H} \frac{\partial L_0(Z_{(i)}, \ldots, Z_{(0)})}{\partial \mathbf{vec} [Z_{(i)}]} \frac{d \mathbf{vec} [Z_{(i)}]}{dt}, \]

where

\[ L_0(Z_{(H)}, \ldots, Z_{(0)}) = \ell (f_0(X, Z), Y), \quad f_0(X, Z) = \sum_{i=0}^{H} Z_{(i)} X S^l, \quad \text{and} \quad Z_{(i)} = W_l B_l B_{(i-1)} \cdots B_{(1)} \]

Since \( f_0(X, Z) = f(X, W, B) = \hat{Y} \) and \( L_0(Z_{(H)}, \ldots, Z_{(0)}) = L(W, B) \),

\[ \left( \frac{\partial L_0(Z_{(i)}, \ldots, Z_{(0)})}{\partial \mathbf{vec} [Z_{(i)}]} \right)^T = \left( \frac{\partial L(W, B)}{\partial \mathbf{vec} [\hat{Y}]} \frac{\partial \mathbf{vec} [\hat{Y}]}{\partial \mathbf{vec} [Z_{(i)}]} \right)^T \]
\[
\frac{d}{dt} L(W, B) = - \sum_{l=0}^{H} \text{vec} [\nabla_{(l)} L(W, B)]^\top F_{(l)} \text{vec} [\nabla_{(l)} L(W, B)] - \sum_{l=1}^{H} \sum_{i=1}^{l} \sum_{k=i}^{H} \text{vec} [\nabla_{(l)} L(W, B)]^\top J_{(i,l)} J_{(i,k)} \text{vec} [\nabla_{(k)} L(W, B)]
\]

To simplify the second term, define \( M_{(i,i)} = \sum_{k=i}^{H} \text{vec} [\nabla_{(l)} L(W, B)]^\top J_{(i,l)} J_{(i,k)} \text{vec} [\nabla_{(k)} L(W, B)] \) and note that we can expand the double sums and regroup terms as follows:

\[
\sum_{l=1}^{H} \sum_{i=1}^{l} M_{(i,i)} = \sum_{l=1}^{H} M_{(l,1)} + \sum_{l=2}^{H} M_{(l,2)} + \cdots + \sum_{l=H}^{H} M_{(l,H)} = \sum_{i=1}^{H} \sum_{l=i}^{H} M_{(i,i)}.
\]

Moreover, for each \( i \in \{1, \ldots, H\} \),

\[
\sum_{l=i}^{H} M_{(i,i)} = \sum_{l=i}^{H} \sum_{k=i}^{H} \text{vec} [\nabla_{(l)} L(W, B)]^\top J_{(i,l)} J_{(i,k)} \text{vec} [\nabla_{(k)} L(W, B)]
\]

\[
= \left( \sum_{l=i}^{H} J_{(i,l)} \text{vec} [\nabla_{(l)} L(W, B)] \right)^\top \left( \sum_{k=i}^{H} J_{(i,k)} \text{vec} [\nabla_{(k)} L(W, B)] \right)
\]

\[
= \left\| \sum_{l=i}^{H} J_{(i,l)} \text{vec} [\nabla_{(l)} L(W, B)] \right\|_2^2
\]

Using these facts, the second term can be simplified as

\[
\sum_{l=1}^{H} \sum_{i=1}^{l} \sum_{k=i}^{H} \text{vec} [\nabla_{(l)} L(W, B)]^\top J_{(i,l)} J_{(i,k)} \text{vec} [\nabla_{(k)} L(W, B)]
\]

\[
= \sum_{l=1}^{H} \sum_{i=1}^{l} M_{(l,i)}
\]

\[
= \sum_{i=1}^{H} \sum_{l=i}^{H} M_{(i,i)}
\]

\[
= \sum_{i=1}^{H} \left\| \sum_{l=i}^{H} J_{(i,l)} \text{vec} [\nabla_{(l)} L(W, B)] \right\|_2^2
\]

Combining these,

\[
\frac{d}{dt} L(W, B) = - \sum_{l=0}^{H} \text{vec} [\nabla_{(l)} L(W, B)]^\top F_{(l)} \text{vec} [\nabla_{(l)} L(W, B)] - \sum_{i=1}^{H} \left\| \sum_{l=i}^{H} J_{(i,l)} \text{vec} [\nabla_{(l)} L(W, B)] \right\|_2^2
\]

(53)
Since $F(t)$ is real symmetric and positive semidefinite,
\[
\begin{align*}
\frac{d}{dt} L(W, B) & \leq -\sum_{l=0}^{H} \lambda_{\min}(F(t)) \| \text{vec}[\nabla_{(l)} L(W, B)] \|^2 - \sum_{l=1}^{H} \sum_{i=1}^{H} J_{(i,l)} \text{vec}[\nabla_{(l)} L(W, B)] \|
\end{align*}
\]  
(54)

A.4.7. COMPLETING THE PROOF BY USING THE ASSUMPTION OF THE SQUARE LOSS

Using the assumption that $L(W, B) = \ell(f(X, W, B), Y) = \|f(X, W, B) - Y\|^2_{F}$ with $\hat{Y} = f(X, W, B)$, we have
\[
\frac{\partial L(W, B)}{\partial Y} = \frac{\partial}{\partial Y} \|\hat{Y} - Y\|^2_{F} = 2(\hat{Y} - Y) \in \mathbb{R}^{m_y \times n},
\]
and hence
\[
\text{vec}[\nabla_{(l)} L(W, B)] = \text{vec} \left[ \frac{\partial L(W, B)}{\partial \hat{Y}} (X S^l)^\top \right] = 2 \text{vec} \left[ (\hat{Y} - Y)(X S^l)^\top \right] = 2[XS^l \otimes I_{m_y}] \text{vec}[\hat{Y} - Y].
\]

Therefore,
\[
\| \text{vec}[\nabla_{(l)} L(W, B)] \|^2_{2} = 4 \| \text{vec}[\hat{Y} - Y]^\top (X S^l)^\top X S^l \otimes I_{m_y} \| \text{vec}[\hat{Y} - Y].
\]  
(55)

We are now ready to complete the proof of Theorem 2 for each cases (i), (ii) and (iii).

A.4.8. CASE (I): COMPLETING THE PROOF OF THEOREM 2 (I)

Using equation (54) and (55) with $\lambda_{W,B} = \min_{0 \leq t \leq H} \lambda_{\min}(F(t))$, we have that
\[
\frac{d}{dt} L(W, B) \leq -\lambda_{W,B} \sum_{l=0}^{H} \| \text{vec}[\nabla_{(l)} L(W, B)] \|^2_{2} - \sum_{l=1}^{H} \sum_{i=1}^{H} J_{(i,l)} \text{vec}[\nabla_{(l)} L(W, B)] \|
\]

\[
\leq -4\lambda_{W,B} \sum_{l=0}^{H} \text{vec}[\hat{Y} - Y]^\top [(X S^l)^\top X S^l \otimes I_{m_y}] \text{vec}[\hat{Y} - Y] - \sum_{l=1}^{H} \sum_{i=1}^{H} J_{(i,l)} \text{vec}[\nabla_{(l)} L(W, B)] \|
\]

\[
\leq -4\lambda_{W,B} \text{vec}[\hat{Y} - Y]^\top \left( \sum_{l=0}^{H} (X S^l)^\top X S^l \right) \otimes I_{m_y} \text{vec}[\hat{Y} - Y] - \sum_{l=1}^{H} \sum_{i=1}^{H} J_{(i,l)} \text{vec}[\nabla_{(l)} L(W, B)] \|
\]

\[
= -4\lambda_{W,B} \text{vec}[\hat{Y} - Y]^\top [G_H G_H \otimes I_{m_y}] \text{vec}[\hat{Y} - Y] - \sum_{l=1}^{H} \sum_{i=1}^{H} J_{(i,l)} \text{vec}[\nabla_{(l)} L(W, B)] \|
\]

where the last line follows from the following fact:
\[
G_H^\top G_H = \left[ \begin{array}{ccc} X & X & \vdots \\ XS & XS & \vdots \\ \vdots & \vdots & \ddots \\ XS^H & XS^H & \cdots \end{array} \right] ^\top \left[ \begin{array}{ccc} X & X & \vdots \\ XS & XS & \vdots \\ \vdots & \vdots & \ddots \\ XS^H & XS^H & \cdots \end{array} \right] = \sum_{l=0}^{H} (X S^l)^\top X S^l.
\]

Decompose $\text{vec}[\hat{Y} - Y]$ as $\text{vec}[\hat{Y} - Y] = v + v^\perp$, where $v = P_{G_H \otimes I_{m_y}} \text{vec}[\hat{Y} - Y]$, $v^\perp = (I_{m_y} - P_{G_H \otimes I_{m_y}}) \text{vec}[\hat{Y} - Y]$, and $P_{G_H \otimes I_{m_y}} \in \mathbb{R}^{m_y \times m_y}$ represents the orthogonal projection onto the column space of $G_H \otimes I_{m_y} \in \mathbb{R}^{m_y \times (H+1)m_y}$. Then,
\[
\text{vec}[\hat{Y} - Y]^\top [G_H^\top G_H \otimes I_{m_y}] \text{vec}[\hat{Y} - Y] = (v + v^\perp)^\top [G_H^\top \otimes I_{m_y}] [G_H \otimes I_{m_y}] (v + v^\perp)
\]

\[
\geq \sigma_{\min}(G_H) \| P_{G_H \otimes I_{m_y}} \text{vec}[\hat{Y} - Y] \|^2_2
\]
where we used the fact that the singular values of \([G_H^T \otimes I_{m_y}]\) are products of singular values of \(G_H\) and \(I_{m_y}\).

By noticing that \(L(W, B) = \|\text{vec}[\hat{Y} - Y]\|_2^2\) and \(L_{1:H}^* = \|(I_{m_y} - P_{G_H^T \otimes I_{m_y}}) \text{vec}[Y]\|_2^2\),

\[
\text{vec}[\hat{Y} - Y]^T [G_H^T G_H \otimes I_{m_y}] \text{vec}[\hat{Y} - Y] \geq \sigma_{\min}^2(G_H)(L(W, B) - L_{1:H}^*).
\]

Therefore,

\[
\frac{d}{dt} L(W, B) \leq -4\lambda_{W,B} \|\text{vec}[\hat{Y} - Y]\|_2^2 \|G_H^T G_H \otimes I_{m_y}\| \|\text{vec}[\hat{Y} - Y] - \frac{1}{L} \sum_{i=1}^H \sum_{l=1}^H J_{i,l} \text{vec}[\nabla_{(l)}L(W, B)]\|_2^2
\]

\[
\leq -4\lambda_{W,B} \sigma_{\min}^2(G_H)(L(W, B) - L_{1:H}^*) - \sum_{i=1}^H \sum_{l=1}^H J_{i,l} \text{vec}[\nabla_{(l)}L(W, B)]\|_2^2
\]

Since \(\frac{d}{dt} L_{1:H}^* = 0\),

\[
\frac{d}{dt} (L(W, B) - L_{1:H}^*) \leq -4\lambda_{W,B} \sigma_{\min}^2(G_H)(L(W, B) - L_{1:H}^*) - \sum_{i=1}^H \sum_{l=1}^H J_{i,l} \text{vec}[\nabla_{(l)}L(W, B)]\|_2^2
\]

By defining \(\mathbf{L} = L(W, B) - L_{1:H}^*\),

\[
\frac{d\mathbf{L}}{dt} \leq -4\lambda_{W,B} \sigma_{\min}^2(G_H)\mathbf{L} - \sum_{i=1}^H \sum_{l=1}^H J_{i,l} \text{vec}[\nabla_{(l)}L(W, B)]\|_2^2 \tag{56}
\]

Since \(\frac{d}{dt} \mathbf{L} \leq 0\) and \(\mathbf{L} \geq 0\), if \(\mathbf{L} = 0\) at some time \(\bar{t}\), then \(\mathbf{L} = 0\) for any time \(t \geq \bar{t}\). Therefore, if \(\mathbf{L} = 0\) at some time \(\bar{t}\), then we have the desired statement of this theorem for any time \(t \geq \bar{t}\). Thus, we can focus on the time interval \([0, \bar{t}]\) such that \(\mathbf{L} > 0\) for any time \(t \in [0, \bar{t}]\) (here, it is allowed to have \(\bar{t} = \infty\)). Thus, focusing on the time interval with \(\mathbf{L} > 0\), equation (56) implies that

\[
\frac{1}{\mathbf{L}} \frac{d\mathbf{L}}{dt} \leq -4\lambda_{W,B} \sigma_{\min}^2(G_H) - \frac{1}{\mathbf{L}} \sum_{i=1}^H \sum_{l=1}^H J_{i,l} \text{vec}[\nabla_{(l)}L(W, B)]\|_2^2
\]

By taking integral over time

\[
\int_0^T \frac{1}{\mathbf{L}} \frac{d\mathbf{L}}{dt} dt \leq -\int_0^T 4\lambda_{W,B} \sigma_{\min}^2(G_H) dt - \int_0^T \frac{1}{\mathbf{L}} \sum_{i=1}^H \sum_{l=1}^H J_{i,l} \text{vec}[\nabla_{(l)}L(W, B)]\|_2^2 dt
\]

By using the substitution rule for integrals, \(\int_0^T \frac{1}{\mathbf{L}} \frac{d\mathbf{L}}{dt} dt = \int_{\mathbf{L}_0}^{\mathbf{L}_T} \frac{1}{\mathbf{L}} d\mathbf{L} = \log(\mathbf{L}_T) - \log(\mathbf{L}_0)\), where \(\mathbf{L}_0 = L(W_0, B_0) - L_{1:H}^*\) and \(\mathbf{L}_T = L(W_T, B_T) - L_{1:H}^*\). Thus,

\[
\log(\mathbf{L}_T) - \log(\mathbf{L}_0) \leq -4\sigma_{\min}^2(G_H) \int_0^T \lambda_{W,B} dt - \int_0^T \frac{1}{\mathbf{L}} \sum_{i=1}^H \sum_{l=1}^H J_{i,l} \text{vec}[\nabla_{(l)}L(W, B)]\|_2^2 dt
\]
which implies that

\[
L_T \leq e^{\log(L_0) - 4\sigma_{\min}^2(G_H) f_0^T \sum_{t=1}^{T} \sum_{i=1}^{H} J_{i(t)} \mathbf{vec}[\nabla_i L(W_t, B_t)]]^2 dt}
\]

where

\[
L_0 \leq e^{-4\sigma_{\min}^2(G_H) f_0^T \sum_{t=1}^{T} \sum_{i=1}^{H} J_{i(t)} \mathbf{vec}[\nabla_i L(W_t, B_t)]]^2 dt}
\]

By recalling the definition of \( L = L(W, B) - L_{1:H}^* \) and that \( \frac{d}{dt} L \leq 0 \), we have that if \( L(W_T, B_T) - L_{1:H}^* > 0 \), then \( L(W_t, B_t) - L_{1:H}^* > 0 \) for all \( t \in [0, T] \), and

\[
L(W_T, B_T) - L_{1:H}^* \leq (L(W_0, B_0) - L_{1:H}^*) e^{-4\lambda_T^* \sigma_{\min}^2(G_H) T - f_0^T \sum_{t=1}^{T} \sum_{i=1}^{H} J_{i(t)} \mathbf{vec}[\nabla_i L(W_t, B_t)]]^2 dt}
\]

By noticing that \( \lambda_T^{(1:H)} = \inf_{t \in [0, T]} \lambda_{W_t, B_t} \) and that \( f_0^T \sum_{t=1}^{T} \sum_{i=1}^{H} J_{i(t)} \mathbf{vec}[\nabla_i L(W_t, B_t)]]^2 dt \geq 0 \), this implies that

\[
L(W_T, B_T) - L_{1:H}^* \leq (L(W_0, B_0) - L_{1:H}^*) e^{-4\lambda_T^{(1:H)} \sigma_{\min}^2(G_H) T}
\]

This completes the proof of Theorem 2 (i) for the case of \( I = [n] \). Since every step in this proof is valid when we replace \( f(X, W, B) \) by \( f(X, W, B)_{S'} \) and \( XS' \) by \( X(S')_{S'} \) without using any assumption on \( S \) or the relation between \( S' \) and \( S \), our proof also yields for the general case of \( I \) that

\[
L(W_T, B_T) - L_{1:H}^* \leq (L(W_0, B_0) - L_{1:H}^*) e^{-4\lambda_T^{(1:H)} \sigma_{\min}^2((G_H)_{S'}) T}
\]
where we used the fact that the singular values of \( \tilde{G}_{H'} \otimes I_{m_y} \) are products of singular values of \( \tilde{G}_{H'} \) and \( I_{m_y} \).

By noticing that \( L(W, B) = \| \text{vec}[\hat{Y} - Y]\|_2^2 \) and \( L_{H'}^* = \| (I_{m_y} - P_{\tilde{G}_{H'} \otimes I_{m_y}}) \text{vec}[Y]\|_2^2 \), we have that for any \( H' \in \{0, 1, \ldots, H\} \),

\[
\text{vec}[\hat{Y} - Y]^T \left[ \tilde{G}_{H'}^T \tilde{G}_{H'} \otimes I_{m_y} \right] \text{vec}[\hat{Y} - Y] \geq \sigma_{\text{min}}^2(\tilde{G}_{H'})(L(W, B) - L_{H'}^*).
\]  

Therefore,

\[
\frac{d}{dt} L(W, B) \leq -4\lambda_{W, B} \text{vec}[\hat{Y} - Y]^T \left[ \tilde{G}_{H'}^T \tilde{G}_{H'} \otimes I_{m_y} \right] \text{vec}[\hat{Y} - Y]
\]

\[
\leq -4\lambda_{W, B} \sigma_{\text{min}}^2(\tilde{G}_{H'})(L(W, B) - L_{H'}^*)
\]

Since \( \frac{d}{dt} L_{H'}^* = 0 \),

\[
\frac{d}{dt} (L(W, B) - L_{H'}^*) \leq -4\lambda_{W, B} \sigma_{\text{min}}^2(\tilde{G}_{H'})(L(W, B) - L_{H'}^*)
\]

By defining \( L = L(W, B) - L_{H'}^* \),

\[
\frac{dL}{dt} \leq -4\lambda_{W, B} \sigma_{\text{min}}^2(\tilde{G}_{H'})
\]  

(58)

Since \( \frac{d}{dt} L \leq 0 \) and \( L \geq 0 \), if \( L = 0 \) at some time \( \bar{t} \), then \( L = 0 \) for any time \( t \geq \bar{t} \). Therefore, if \( L = 0 \) at some time \( \bar{t} \), then we have the desired statement of this theorem for any time \( t \geq \bar{t} \). Thus, we can focus on the time interval \([0, \bar{t}]\) such that \( L > 0 \) for any time \( t \in [0, \bar{t}] \) (here, it is allowed to have \( \bar{t} = \infty \)). Thus, focusing on the time interval with \( L > 0 \), equation (58) implies that

\[
\frac{1}{L} \frac{dL}{dt} \leq -4\lambda_{W, B} \sigma_{\text{min}}^2(\tilde{G}_{H'})
\]

By taking integral over time

\[
\int_0^T \frac{1}{L} \frac{dL}{dt} dt \leq - \int_0^T 4\lambda_{W, B} \sigma_{\text{min}}^2(\tilde{G}_{H'}) dt
\]

By using the substitution rule for integrals, \( \int_0^T \frac{1}{L} \frac{dL}{dt} dt = \int_{L_0}^{L_T} \frac{1}{L} dL = \log(L_T) - \log(L_0) \), where \( L_0 = L(W_0, B_0) - L^* \) and \( L_T = L(W_T, B_T) - L_{H'}^* \). Thus,

\[
\log(L_T) - \log(L_0) \leq -4\sigma_{\text{min}}^2(\tilde{G}_{H'}) \int_0^T \lambda_{W, B} dt
\]

which implies that

\[
L_T \leq e^{\log(L_0) - 4\sigma_{\text{min}}^2(\tilde{G}_{H'}) \int_0^T \lambda_{W, B} dt} = L_0 e^{-4\sigma_{\text{min}}^2(\tilde{G}_{H'}) \int_0^T \lambda_{W, B} dt}
\]

By recalling the definition of \( L = L(W, B) - L_{H'}^* \) and that \( \frac{d}{dt} L \leq 0 \), we have that if \( L(W_T, B_T) - L_{H'}^* > 0 \), then \( L(W_t, B_t) - L_{H'}^* > 0 \) for all \( t \in [0, T] \), and

\[
L(W_T, B_T) - L_{H'}^* \leq (L(W_0, B_0) - L_{H'}^*) e^{-4\sigma_{\text{min}}^2(\tilde{G}_{H'}) \int_0^T \lambda_{W, B} dt}.
\]

By noticing that \( \lambda_{\text{min}}^{(H')} = \inf_{t \in [0, T]} \lambda_{W_t, B_t} \), this implies that for any \( H' \in \{0, 1, \ldots, H\} \),

\[
L(W_T, B_T) - L_{H'}^* \leq (L(W_0, B_0) - L_{H'}^*) e^{-4\lambda_{\text{min}}^{(H')} \sigma_{\text{min}}^2(\tilde{G}_{H'}) T}
\]
This completes the proof of Theorem 2 (ii) for the case of $\mathcal{I} = [n]$. Since every step in this proof is valid when we replace $f(X, W, B)$ by $f(X, W, B)_{x_2}$ and $XS^l$ by $X(S^l)_{x_2}$ without using any assumption on $S$ or the relation between $S^{l-1}$ and $S$, our proof also yields for the general case of $\mathcal{I}$ that

$$L(W_T, B_T) - L_{H'}^* \leq (L(W_0, B_0) - L_{H'}^*) e^{-4 \lambda_{H'}^2 \sigma_{\min}^2 (X S^T) T}$$

\[\square\]

### A.4.10. Case (iii): Completing the Proof of Theorem 2 (iii)

In this case, we have the following assumption: there exist $l, l' \in \{0, \ldots, H\}$ with $l < l'$ such that $L_l^* \geq L_{l+1}^* \geq \cdots \geq L_{l'}^*$ or $L_l^* \leq L_{l+1}^* \leq \cdots \leq L_{l'}^*$. Using equation (54) and (55) with $\tilde{G}_l = X S^l$, we have that

\[
\frac{d}{dt} L(W, B) \leq -\sum_{l=0}^{H} \lambda_{\min}(F_l) \| \nabla_{(l)} L(W, B) \|_2^2 \\
\leq -4 \sum_{l=0}^{H} \lambda_{\min}(F_l) \text{vec} [\tilde{Y} - Y] (\text{vec} [\tilde{Y} - Y])^T [(X S^l)^T X S^l \otimes I_{m_y}] \text{vec} [\tilde{Y} - Y] \\
= -4 \sum_{l=0}^{H} \lambda_{\min}(F_l) \text{vec} [\tilde{Y} - Y] (\text{vec} [\tilde{G}_l \otimes I_{m_y}]) (\text{vec} [\tilde{Y} - Y])
\]

Using (57), since $\text{vec} [\tilde{Y} - Y] (\text{vec} [\tilde{G}_l \otimes I_{m_y}]) \geq \sigma_{\min}^2 (\tilde{G}_l) (L(W, B) - L_l^*)$ for any $l \in \{0, 1, \ldots, H\}$,

\[
\frac{d}{dt} L(W, B) \leq -4 \sum_{l=0}^{H} \lambda_{\min}(F_l) \sigma_{\min}^2 (\tilde{G}_l) (L(W, B) - L_l^*) \tag{59}
\]

Let $l'' = l$ if $L_l^* \geq L_{l+1}^* \geq \cdots \geq L_{l''}$, and $l'' = l'$ if $L_l^* \leq L_{l+1}^* \leq \cdots \leq L_{l''}$. Then, using (59) and the assumption of $L_l^* \geq L_{l+1}^* \geq \cdots \geq L_{l''}$ or $L_l^* \leq L_{l+1}^* \leq \cdots \leq L_{l''}$ for some $l, l' \in \{0, \ldots, H\}$, we have that

\[
\frac{d}{dt} L(W, B) \leq -4(L(W, B) - L_{l''}^*) \sum_{k=l}^{l'} \lambda_{\min}(F(k)) \sigma_{\min}^2 (\tilde{G}_k) \tag{60}
\]

Since $\frac{d}{dt} L_{l''}^* = 0$,

\[
\frac{d}{dt} (L(W, B) - L_{l''}^*) \leq -4(L(W, B) - L_{l''}^*) \sum_{k=l}^{l'} \lambda_{\min}(F(k)) \sigma_{\min}^2 (\tilde{G}_k).
\]

By taking integral over time in the same way as that in the proof for the case of (i) and (ii), we have that

\[
L(W_T, B_T) - L_{l''}^* \leq (L(W_0, B_0) - L_{l''}^*) e^{-4 \sum_{k=l}^{l'} \sigma_{\min}^2 (\tilde{G}_k) \int_0^T \lambda_{\min}(F(k), t) dt} \tag{61}
\]

Using the property of Kronecker product,

\[
\lambda_{\min}(F_{(l), t}) = \lambda_{\min}([(B_{(l), t} \cdot \cdots B_{(1), t})^T B_{(l), t} \cdots B_{(1), t}) \otimes I_{m_y}]) = \lambda_{\min}((B_{(l), t} \cdot \cdots B_{(1), t})^T B_{(l), t} \cdots B_{(1), t}),
\]

which implies that $\lambda_{T}^{(k)} = \inf_{t \in [0, T]} \lambda_{\min}(F(k), t)$. Therefore, equation (61) with $\lambda_{T}^{(k)} = \inf_{t \in [0, T]} \lambda_{\min}(F(k), t)$ yields that

\[
L(W_T, B_T) - L_{l''}^* \leq (L(W_0, B_0) - L_{l''}^*) e^{-4 \sum_{k=l}^{l'} \lambda_{T}^{(k)} \sigma_{\min}^2 (\tilde{G}_k) T} \\
= (L(W_0, B_0) - L_{l''}^*) e^{-4 \sum_{k=l}^{l'} \lambda_{T}^{(k)} \sigma_{\min}^2 (X S^k) T} \tag{62}
\]
This completes the proof of Theorem 2 (iii) for the case of $\mathcal{I} = [n]$. Since every step in this proof is valid when we replace $f(X, W, B)$ by $f(X, W, B)_{xz}$ and $XS^l$ by $X(S^l)_{xz}$ without using any assumption on $S$ or the relation between $S^{l-1}$ and $S$, our proof also yields for the general case of $\mathcal{I}$ that

$$L(W_T, B_T) - L^*_T \leq (L(W_0, B_0) - L^*_0)e^{-4\sum_{k=1}^{\ell} \lambda_k^2 \min(X(s^k)_z)T}.$$ 

\[\square\]

A.5. Proof of Proposition 3

From Definition 4, for any $l \in \{1, 2, \ldots, H\}$, we have that $\sigma_{\min}(\bar{B}^{(1,l)}) = \sigma_{\min}(B_t(i)B_{(i-1)} \cdots B_{(1)}) \geq \gamma$ for all $(W, B)$ such that $L(W, B) \leq L(W_0, B_0)$. From equation (54) in the proof of Theorem 2, it holds that $\frac{d}{dt} L(W_t, B_t) \leq 0$ for all $t$. Thus, we have that $L(W_t, B_t) \leq L(W_0, B_0)$ and hence $\sigma_{\min}(B_t^{(1,l)}) \geq \gamma$ for all $t$. Under this problem setting ($m_t \geq m_x$), this implies that $\lambda_{\min}(B_t^{(1,l)}) \geq \gamma^2$ for all $t$ and thus $\lambda^{(1:H)}_{\min} \geq \gamma^2$.

A.6. Proof of Theorem 3

The proof of Theorem 3 follows from the intermediate results of the proofs of Theorems 1 and 2 as we show in the following. For the non-multiscale case, from equation (36) in the proof of Theorem 1, we have that

$$\frac{d}{dt} L_1(W, B) = -\| \text{vec}[\nabla_{(H)} L(W, B)] \|^2_{F(H)} - \sum_{i=1}^{H} \| J_{(i,H)} \text{vec}[\nabla_{(H)} L(W, B)] \|^2_2$$

where

$$\| \text{vec}[\nabla_{(H)} L(W, B)] \|^2_{F(H)} := \text{vec}[\nabla_{(H)} L(W, B)]^\top F(H) \text{vec}[\nabla_{(H)} L(W, B)].$$

Since equation (36) in the proof of Theorem 2 is derived without the assumption on the square loss, this holds for any differentiable loss $\ell$. By noticing that $\nabla_{(H)} L(W, B) = V(X(S^H)_{xz})^\top$, we have that

$$\frac{d}{dt} L_1(W, B) = -\| \text{vec}[V(X(S^H)_{xz})^\top] \|^2_{F(H)} - \sum_{i=1}^{H} \| J_{(i,H)} \text{vec}[V(X(S^H)_{xz})^\top] \|^2_2.$$ 

This proves the statement of Theorem 3 (i).

For the multiscale case, from equation (53) in the proof of Theorem 2, we have that

$$\frac{d}{dt} L_2(W, B) = -\sum_{i=0}^{H} \| \text{vec}[\nabla_{(l)} L(W, B)] \|^2_{F(l)} - \sum_{i=1}^{H} \sum_{l=1}^{H} J_{(i,l)} \text{vec}[\nabla_{(l)} L(W, B)] \|^2_2$$

where

$$\| \text{vec}[\nabla_{(l)} L(W, B)] \|^2_{F(l)} := \text{vec}[\nabla_{(l)} L(W, B)]^\top F(l) \text{vec}[\nabla_{(l)} L(W, B)].$$

Since equation (53) in the proof of Theorem 2 is derived without the assumption on the square loss, this holds for any differentiable loss $\ell$. Since every step to derive equation (53) is valid when we replace $f(X, W, B)$ by $f(X, W, B)_{xz}$ and $XS^l$ by $X(S^l)_{xz}$ without using any assumption on $S$ or the relation between $S^{l-1}$ and $S$, the steps to derive equation (53) also yields this for the general case of $\mathcal{I}$: i.e., $\nabla_{(l)} L(W, B) = V(X(S^l)_{xz})^\top$. Thus, we have that

$$\frac{d}{dt} L_1(W, B) = -\sum_{i=0}^{H} \| \text{vec}[V(X(S^l)_{xz})^\top] \|^2_{F(l)} - \sum_{i=1}^{H} \sum_{l=1}^{H} J_{(i,l)} \text{vec}[V(X(S^l)_{xz})^\top] \|^2_2.$$ 

This completes the proof of Theorem 3 (ii).

B. Additional Experimental Results

In this section, we present additional experimental results.
Figure 6. Multiscale skip connection accelerates GNN training. We plot the training curves of GNNs with ReLU and linear activation on the Cora and Citeseer dataset. We use the GCN model with learning rate $5e^{-5}$, six layers, and hidden dimension 32.
Figure 7. **Depth accelerates GNN training.** We plot the training curves of GNNs with ReLU and linear activation, multiscale and non-multiscale on the Cora dataset. We use the GCN model with learning rate $5e^{-5}$ and hidden dimension 32.
Figure 8. Depth accelerates GNN training. We plot the training curves of GNNs with ReLU and linear activation, multiscale and non-multiscale on the Citeseer dataset. We use the GCN model with learning rate $5e^{-5}$ and hidden dimension 32.
Figure 9. GNNs train faster when the labels have signal instead of random noise. We plot the training curves of multiscale and non-multiscale GNNs with ReLU and linear activation, on the Cora dataset. We use the two-layer GCN model with learning rate $1e^{-4}$ and hidden dimension 32.
Figure 10. GNNs train faster when the labels have signal instead of random noise. We plot the training curves of multiscale and non-multiscale GNNs with ReLU and linear activation, on the Citeseer dataset. We use the two-layer GCN model with learning rate $1e^{-4}$ and hidden dimension 32.
Figure 11. Linear GNNs vs. ReLU GNNs. We plot the training curves of GCN and GIN with ReLU and linear activation on the Cora dataset. The training curves of linear GNNs and ReLU GNNs are similar, both converging to nearly zero training loss with the same linear rate. Moreover, GIN trains faster than GCN, which agrees with our bound in Theorem 1. We use the learning rate $1e^{-4}$, two layers, and hidden dimension 32.
C. Experimental Setup

In this section, we describe the experimental setup for reproducing our experiments.

Dataset. We perform all experiments on the Cora and Citeseer datasets (Sen et al., 2008). Cora and Citee are citation networks and the goal is to classify academic documents into different subjects. The dataset contains bag-of-words features for each document (node) and citation links (edges) between documents. The tasks are semi-supervised node classification. Only a subset of nodes have training labels. In our experiments, we use the default dataset split, i.e., which nodes have training labels, and minimize the training loss accordingly. Table 1 shows an overview of the dataset statistics.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Nodes</th>
<th>Edges</th>
<th>Classes</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citeseer</td>
<td>3,327</td>
<td>4,732</td>
<td>6</td>
<td>3,703</td>
</tr>
<tr>
<td>Cora</td>
<td>2,708</td>
<td>5,429</td>
<td>7</td>
<td>1,433</td>
</tr>
</tbody>
</table>

Table 1. Dataset statistics

Training details. We describe the training settings for our experiments. Let us first describe some common hyperparameters and settings, and then for each experiment or figure we describe the other hyperparameters. For our experiments, to more closely align with the common practice in GNN training, we use the Adam optimizer and keep optimizer-specific hyperparameters except initial learning rate default. We set weight decay to zero. Next, we describe the settings for each experiment respectively.

For the experiment in Figure 1, i.e., the training curves of linear vs. ReLU GNNs, we train the GCN and GIN with two layers on Cora with cross-entropy loss and learning rate 1e-4. We set the hidden dimension to 32.

For the experiment in Figure 2a, i.e., computing the graph condition for linear GNNs, we use the linear GCN and GIN model with three layers on Cora and Citeseer. For linear GIN, we set $\epsilon$ to zero and MLP layer to one.

For the experiment in Figure 2b, i.e., computing and plotting the time-dependent condition for linear GNNs, we train a linear GCN with two layers on Cora with squared loss and learning rate 1e-4. We set the hidden dimension the input dimension for both Cora and for CiteSeer, because the global convergence theorem requires the hidden dimension to be at least the same as input dimension. Note that this requirement is standard in previous works as well, such as Arora et al. (2019a). We use the default random initialization of PyTorch. The formula for computing the time-dependent $\lambda_T$ is given in the main paper.

For the experiment in Figure 2c, i.e., computing and plotting the time-dependent condition for linear GNNs across multiple training settings, we consider the following settings:

1. Dataset: Cora and Citeseer.
2. Model: GCN and GIN.
3. Depth: Two and four layers.
4. Activation: Linear and ReLU.

We train the GNN with the settings above with squared loss and learning rate 1e-4. We set the hidden dimension to input dimension for Cora and CiteSeer. We use the default random initialization of PyTorch. The formula for computing the time-dependent $\lambda_T$ is given in the main paper. For each point, we report the $\lambda_T$ at last epoch.

For the experiment in Figure 3a, i.e., computing the graph condition for multiscale linear GNNs, we use the linear GCN and GIN model with three layers on Cora and Citeseer. For linear GIN, we set $\epsilon$ to zero and MLP layer to one.

For the experiment in Figure 3b, i.e., computing and plotting the time-dependent condition for multiscale linear GNNs, we train a linear GCN with two layers on Cora with squared loss and learning rate 1e-4. We set the hidden dimension to 2000 for Cora and 4000 for CiteSeer. We use the default random initialization of PyTorch. The formula for computing the time-dependent $\lambda_T$ is given in the main paper.

For the experiment in Figure 3c, i.e., computing and plotting the time-dependent condition for multiscale linear GNNs across multiple training settings, we consider the following settings:
1. Dataset: Cora and Citeseer.
2. Model: Multiscale GCN and GIN.
3. Depth: Two and four layers.
4. Activation: Linear and ReLU.

We train the multiscale GNN with the settings above with squared loss and learning rate 1e-4. We set the hidden dimension to 2000 for Cora and 4000 for Citeseer. We use the default random initialization of PyTorch. The formula for computing the time-dependent $\lambda_T$ is given in the main paper. For each point, we report the $\lambda_T$ at last epoch.

For the experiment in Figure 4a, i.e., multiscale vs. non-multiscale, we train the GCN with six layers and ReLU activation on Cora with cross-entropy loss and learning rate 5e-5. We set the hidden dimension to 32.

We perform more extensive experiments to verify the conclusion for multiscale vs. non-multiscale in Figure 11. There, we train the GCN with six layers with both ReLU and linear activation on both Cora and Citeseer with cross-entropy loss and learning rate 5e-5. We set the hidden dimension to 32.

For the experiment in Figure 4b, i.e., acceleration with depth, we train the non-multiscale GCN with two, four, six layers and ReLU activation on Cora with cross-entropy loss and learning rate 5e-5. We set the hidden dimension to 32.

We perform more extensive experiments to verify the conclusion for acceleration with depth in Figure 7 and Figure 8. There, we train both multiscale and non-multiscale GCN with 2, 4, 6 layers with both ReLU and linear activation on both Cora and Citeseer with cross-entropy loss and learning rate 5e-5. We set the hidden dimension to 32.

For the experiment in Figure 4c, i.e., signal vs. noise, we train the non-multiscale GCN with two layers and ReLU activation on Cora with cross-entropy loss and learning rate 1e-4. We set the hidden dimension to 32. For signal, we use the default labels of Cora. For noise, we randomly choose a class as the label.

We perform more extensive experiments to verify the conclusion for signal vs. noise in Figure 9 and Figure 10. There, we train both multiscale and non-multiscale GCN with two layers with both ReLU and linear activation on both Cora and Citeseer with cross-entropy loss and learning rate 1e-4. We set the hidden dimension to 32.

For the experiment in Figure 5, i.e., first term vs. second term, we use the same setting as in Figure 4c. We use the formula of our Theorem in the main paper.

**Computing resources.** The computing hardware is based on the CPU and the NVIDIA GeForce RTX 1080 Ti GPU. The software implementation is based on PyTorch and PyTorch Geometric (Fey & Lenssen, 2019). For all experiments, we train the GNNs with CPU and compute the eigenvalues with GPU.