Linear Loss Classification: Efficient Training Through Neural Collapse

Wonyeong Song LCRE47@KAIST.AC.KR
Korea Advanced Institute of Science & Technology (KAIST), Korea

Donghwan Kim DONGHWANKIM@KAIST.AC.KR
Korea Advanced Institute of Science & Technology (KAIST), Korea

Abstract

Logistic loss is the de facto standard for classification. This success is commonly attributed to inductive biases, such as the directional convergence of the last-layer classifier toward the max-margin solution for a given feature representation, and neural collapse (NC), a terminal-phase phenomenon characterized by structural simplification of last-layer features and classifiers. However, both emerge slowly because the gradients of logistic loss decay exponentially. In this work, we introduce a *linear loss* $\ell(u) = -u$, which avoids exponentially decaying gradients and leads to faster training dynamics. Under this loss, the last-layer classifier trained by GD no longer directionally converges to the max-margin solution, but instead aligns with the difference between class means. Although this direction differs from the max-margin solution and may therefore appear suboptimal, once NC occurs, the two directions become closely aligned, reducing the discrepancy. Empirically, we demonstrate that GD with linear loss, combined with weight normalization, exhibits NC, and that both NC and generalization occur faster than with logistic loss. On the theoretical side, we prove that the unconstrained feature model (UFM) exhibits NC, while shallow ReLU networks exhibit a weaker directional collapse of features and classifiers. Together, these results demonstrate that linear loss, despite being unbounded below, provides a simple yet effective alternative to logistic loss by accelerating both neural collapse and generalization.

1. Introduction

Gradient-based methods are used extensively to solve machine learning problems, including classification tasks. Among various classification losses, the logistic loss (cross-entropy loss) is the most widely used, as neural networks trained by gradient descent on this loss exhibit remarkably strong generalization performance. This success is often attributed to inductive biases induced by such training dynamics. For instance, Soudry et al. [14], Ji and Telgarsky [6] and Chizat and Bach [3] showed that the last-layer classifier converges to a particular solution that generalizes well, namely, the max-margin direction for separable datasets. Moreover, Papyan et al. [12] identified an empirical phenomenon termed *neural collapse*, characterized by the structural simplification of last-layer features and classifiers during the terminal phase of training. This simplified structure has been associated with strong generalization performance and improved adversarial robustness.

Despite these advantages, one limitation of the logistic loss is that the emergence of inductive biases is slowed by the exponentially decaying gradients. In particular, Soudry et al. [14] showed that the loss decreases at a rate $\mathcal{O}(1/t)$, but the classifier converges toward the max-margin direction only logarithmically, at a rate $\mathcal{O}(1/\log t)$. To accelerate this convergence, Ji and Telgarsky [8] and Zhang et al. [15] proposed using an adaptive/aggressive step size inversely proportional to the

© W. Song & D. Kim.

current loss value. This approach yields an $\mathcal{O}(1/t)$ convergence rate for the classifier direction, but those works require estimation of current loss value, or careful stepsize scheduling. Moreover, these works focus exclusively on the max-margin bias of single-layer classifiers, without addressing more general representation dynamics such as neural collapse.

In this paper, we introduce the linear loss $\ell(u) = -u$, which avoids exponentially decaying gradients. At first glance, this loss may appear unsuitable for classification, as it is unbounded below and its minimization leads to a divergent trajectory of neural weights. However, this alone does not preclude its use for classification; minimizing the logistic loss also results in a divergent yet directionally meaningful trajectory. Another apparent drawback is that the resulting classifier does not exhibit an implicit bias toward the max-margin direction. Nevertheless, we show that linear loss is well suited for classification and can lead to faster training than logistic loss.

In particular, we demonstrate that under the unconstrained feature model (UFM) [11], an abstraction of sufficiently overparameterized neural networks, linear loss induces NC, including the collapse of the within-class variance of last-layer features. As within-class feature representations collapse toward a single point, the role of the implicit bias of the last-layer classifier becomes less important. We also show that, for two-layer ReLU networks, feature mappings collapse into a one-dimensional subspace. Moreover, our empirical results show that ReLU networks, trained with linear loss, combined with appropriate normalization techniques, exhibit NC more rapidly than those trained with logistic loss, allowing them to achieve the same test accuracy much faster.

1.1. Our Contribution

- In Section 3, we introduce the linear loss and analyze its properties and advantages as a classification loss.
- In Section 4, we provide empirical evidence that neural collapse (NC) emerges under linear loss when combined with appropriate normalization techniques. These results further indicate that linear loss accelerates the emergence of NC, thereby allowing neural networks to achieve the same test accuracy substantially faster.
- In Section 5, we theoretically show that linear loss induces NC under the unconstrained feature model (UFM), where the last-layer features are treated as free variables. Going beyond this restrictive assumption, we prove that feature mappings of two-layer ReLU networks collapse into a one-dimensional subspace without relying on the UFM.

2. Background of Neural Collapse

Recent theoretical and empirical studies have shown that continuing the training beyond zero training error can further improve generalization performance. A notable observation in this regime is *neural collapse* (NC) [12], which refers to the collapse of within-class variability of last-layer features, along with a structured alignment of features and classifiers. This structural simplification has been associated with improved generalization performance and adversarial robustness.

To formally define NC, we consider training a deep neural network on a dataset containing C classes with N examples per class. Let $\boldsymbol{x}_{i,c}$ denote the ith example of class c, and let $\boldsymbol{h}_{i,c}$ be its last-layer feature obtained by passing $\boldsymbol{x}_{i,c}$ through all but the last layer. The last-layer classifier predicts labels according to the decision rule $\arg\max_{c'} \langle \boldsymbol{v}_{c'}, \boldsymbol{h}_{i,c} \rangle + \boldsymbol{b}_{c'}$. We define the class means

and the global mean of the features as $\boldsymbol{\mu}_c := \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{h}_{i,c}$ and $\boldsymbol{\mu}_G := \frac{1}{C}\sum_{c=1}^{C} \boldsymbol{\mu}_c$, respectively. We further define the within-class covariance and between-class covariance as:

$$\boldsymbol{\Sigma}_W := \frac{1}{CN}\sum_{c=1}^{C}\sum_{i=1}^{N}(\boldsymbol{h}_{i,c} - \boldsymbol{\mu}_c)(\boldsymbol{h}_{i,c} - \boldsymbol{\mu}_c)^\top, \quad \boldsymbol{\Sigma}_B := \frac{1}{C}\sum_{c=1}^{C}(\boldsymbol{\mu}_c - \boldsymbol{\mu}_G)(\boldsymbol{\mu}_c - \boldsymbol{\mu}_G)^\top$$

During the terminal phase of training, which begins once the training accuracy reaches 100%, neural collapse is said to occur when the last-layer features and classifiers exhibit the following properties:

- **(NC1) Within-class variability collapse: $\boldsymbol{\Sigma}_W \to \mathbf{0}$**
- **(NC2) Convergence to a simplex ETF[1]:** For any pair of distinct classes $1 \le c_1 \ne c_2 \le C$, $\|\boldsymbol{\mu}_{c_1} - \boldsymbol{\mu}_G\| - \|\boldsymbol{\mu}_{c_2} - \boldsymbol{\mu}_G\| \to 0$ and $\cos(\boldsymbol{\mu}_{c_1} - \boldsymbol{\mu}_G, \boldsymbol{\mu}_{c_2} - \boldsymbol{\mu}_G) \to -\frac{1}{C-1}$
- **(NC3) Convergence to self-duality:** For any class $c = 1, 2, \ldots, C$, $\frac{\boldsymbol{v}_c}{\|\boldsymbol{v}_c\|} - \frac{\boldsymbol{\mu}_c - \boldsymbol{\mu}_G}{\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|} \to \mathbf{0}$.

In the remainder of this paper, we focus on binary classification ($C = 2$).

3. The Linear Loss

We introduce the linear loss $\ell(u) = -u$, which is Bayes consistent (see Appendix A); its minimizer induces the Bayes' optimal decision rule. Although this loss function is unbounded below and thus its minimization leads to a divergent trajectory, this does not pose a fundamental issue. Indeed, minimizing the logistic loss also yields a divergent yet directionally meaningful trajectory [14].

For a neural network $f(\cdot; \boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$, and a dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, the empirical risk under linear loss is

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ell(y_i f(\boldsymbol{x}_i, \boldsymbol{\theta})) = \sum_{i=1}^{n} -y_i f(\boldsymbol{x}_i, \boldsymbol{\theta}). \tag{1}$$

Analyzing this objective provides several insights into the behavior of linear loss.

Implicit bias toward mean-difference direction Let $\boldsymbol{v}$ denote the parameter of the last-layer classifier. If we denote the mean vectors of positive/negative class by $\boldsymbol{\mu}_\pm$, the empirical risk minimization with respect to $\boldsymbol{v}$ yields $\arg\max_{\boldsymbol{v}} \boldsymbol{v}^\top(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)$. This implies that the last-layer classifier converges in direction to the mean-difference direction $\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-$. Although this differs from the max-margin direction induced by logistic loss, we argue that this does not pose a problem since two directions coincide once NC occurs. Moreover, since NC3 reduces to $\boldsymbol{v}/\|\boldsymbol{v}\| \to (\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)/\|\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-\|$ for binary classification, the implicit bias induced by linear loss already points the classifier toward its eventual NC3 solution, naturally accelerating the emergence of NC3.

Connection to Fisher's LDA Finding $\arg\max_{\boldsymbol{v}}\ \boldsymbol{v}^\top(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)$ can also be viewed as maximizing between-class variance. Although this is not exactly the same as Fisher's linear discriminant [5], which simultaneously maximizes between-class variance and minimizes within-class variance, our linear loss can be viewed as playing a similar role once within-class variance vanishes by NC1.

1. A simplex equiangular tight frame (ETF) is the collection of vectors that have same length and maximally pairwise-distanced.

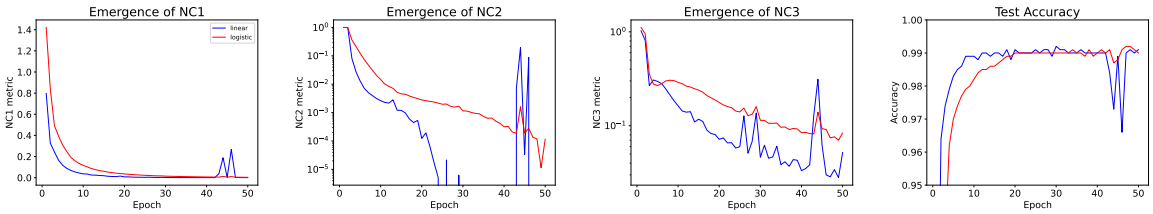

Figure 1: NC metrics (See Appendix B for a formal definition) and test accuracy of neural networks trained by gradient descent with logistic and linear losses

4. Numerical Experiments: NC under Linear Loss

We conducted numerical experiments to evaluate the training efficiency of linear loss. We trained a three-layer neural network to classify digits 0 and 8 from the MNIST dataset. Figure 1 compares the evolution of NC metrics and test accuracy for networks trained under logistic and linear losses. Despite occasional numerical instabilities, neural networks trained with linear loss exhibit NC much faster than those trained with logistic loss. Consequently, networks trained with linear loss reach the same test accuracy (≈ 0.99) much faster than those trained with logistic loss.

For both losses, batch normalization was used to stabilize feature representations. In addition, we found weight normalization particularly useful for linear loss. Since the linear loss does not saturate, gradient norms grow much faster than under logistic loss, leading to numerical instability. Moreover, since the loss is homogeneous with respect to network weights, a gradient step on a normalized network can be interpreted as a small-step update on the unnormalized network. This suggests that weight normalization preserves the learning direction. Detailed experimental setups and definitions of NC metrics are provided in Appendix B.

5. Theoretical Analysis of NC under Linear Loss

In this section, we provide theoretical guarantees for the emergence of NC under linear loss. We also explain empirical benefits of batch and weight normalization under the linear loss from a theoretical perspective.

5.1. NC on Unconstrained Feature Model

The unconstrained feature model (UFM) is an abstraction of deep neural networks that treats the last-layer feature as a free variable [11]. This assumption is well justified in overparameterized regimes, where models can interpolate arbitrary training data. Under the UFM, we show that the last-layer classifier $\boldsymbol{v}$ and features $\boldsymbol{h}$ converge in direction under gradient flow

$$\frac{d\boldsymbol{v}}{dt} = -\nabla_{\boldsymbol{v}}\mathcal{L}, \quad \frac{d\boldsymbol{h}}{dt} = -\nabla_{\boldsymbol{h}}\mathcal{L}.$$

Furthermore, they satisfy NC2, NC3 and a weaker form of NC1: $\boldsymbol{\Sigma}_W\boldsymbol{\Sigma}_B^{\dagger} \to \mathbf{0}$ (see Appendix C.1). We further show that weight normalization facilitates the complete collapse of within-class variance, thereby inducing full NC (see Appendix C.2). These results suggest that NC can arise under the linear loss, when the model is sufficiently overparameterized.

5.2. NC on Two-Layer ReLU Network

Going beyond the restrictive UFM assumption, we consider a two-layer ReLU network parametrized by $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{v})$:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{v}^T \sigma(\boldsymbol{W}^\top \boldsymbol{x}) = \sum_{j=1}^{h} v_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle), \tag{2}$$

where $\boldsymbol{v} = [v_1, \cdots, v_h]^\top$ and $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_h]^\top$, trained with linear loss. We denote the feature mapping as $\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}) := [\sigma(\langle \boldsymbol{w}_1, \boldsymbol{x} \rangle), \cdots, \sigma(\langle \boldsymbol{w}_h, \boldsymbol{x} \rangle)]^\top$, and the trajectory of weights under gradient flow as $\boldsymbol{v}(t), \boldsymbol{W}(t)$. We impose the following mild assumptions.

Assumption 1 (Orthogonal Separability) *There exist positive number $0 < \mu \leq 1$ such that for any $1 \leq i, j \leq n$, $\cos(y_i \boldsymbol{x}_i, y_j \boldsymbol{x}_j) \geq \mu$.*

Assumption 2 (Balanced Initialization) $\|\boldsymbol{w}_j(0)\| = |v_j(0)|, \ \forall 1 \leq j \leq h$.

Assumption 3 (Non-Extreme Initialization) *If $v_j(0)$ is positive, $\cos(\boldsymbol{w}_j(0), \boldsymbol{x}_-) < 1$ where $\boldsymbol{x}_- = \sum_{y_i=-1} \boldsymbol{x}_i$. Conversely, if $v_j(0)$ is negative, $\cos(\boldsymbol{w}_j(0), \boldsymbol{x}_+) < 1$ where $\boldsymbol{x}_+ = \sum_{y_i=+1} \boldsymbol{x}_i$.*

Now we formally state the neural collapse of a two-later ReLU network trained by gradient flow (GF) with linear loss.

Theorem 4 (Neural Collapse of GF on a Two-Layer ReLU Network Trained with Linear Loss) *Under Assumptions 1, 2 and 3, the limits $\bar{\boldsymbol{v}} = \lim_{t\to\infty} \frac{\boldsymbol{v}(t)}{\|\boldsymbol{v}(t)\|}$, $\bar{\boldsymbol{W}} = \lim_{t\to\infty} \frac{\boldsymbol{W}(t)}{\|\boldsymbol{W}(t)\|}$ exist. Moreover, there exist nonnegative constants $s_+(t)$, $s_-(t)$ and unit vectors $\boldsymbol{u}_+(t), \boldsymbol{u}_-(t)$ such that*

- *(Directional collapse of features) The feature mappings of each class satisfy*

$$\boldsymbol{h}_{\boldsymbol{\theta}(t)}(\boldsymbol{x}_i) - \langle s_+(t)\boldsymbol{x}_+, \boldsymbol{x}_i \rangle \boldsymbol{u}_+(t) \underset{t\to\infty}{\to} \boldsymbol{0}, \quad \forall \boldsymbol{x}_i \text{ s.t. } y_i = +1$$

$$\boldsymbol{h}_{\boldsymbol{\theta}(t)}(\boldsymbol{x}_i) - \langle s_-(t)\boldsymbol{x}_-, \boldsymbol{x}_i \rangle \boldsymbol{u}_-(t) \underset{t\to\infty}{\to} \boldsymbol{0}, \quad \forall \boldsymbol{x}_i \text{ s.t. } y_i = -1.$$

  Moreover, the limits $\bar{\boldsymbol{u}}_+ = \lim_{t\to\infty} \boldsymbol{u}_+(t)$, $\bar{\boldsymbol{u}}_- = \lim_{t\to\infty} \boldsymbol{u}_-(t)$ exist.

- *(Orthogonal class means) $\langle \bar{\boldsymbol{u}}_+, \bar{\boldsymbol{u}}_- \rangle = 0$*

- *(Alignment of classifier and features) $\boldsymbol{v}(t) = s_+(t)\boldsymbol{u}_+(t) - s_-(t)\boldsymbol{u}_-(t)$*

There are several caveats about this theorem. First, two-layer ReLU networks do not exhibit 'complete' neural collapse, as its features lie in a one-dimensional subspace, instead of shrinking to a single point. This is not a flaw of linear loss: it comes from the limited expressiveness of the shallow structure. Two-layer ReLU networks trained by logistic loss also experience such 'directional' collapse [10]. Second, unlike logistic loss where $\lim_{t\to\infty} s_+(t)/s_-(t)$ exists, the linear loss would not allow this limit. More concretely, for linear loss, $s_+(t) = \Theta(e^{\|\boldsymbol{x}_+\|t}), \quad s_-(t) = \Theta(e^{\|\boldsymbol{x}_-\|t})$. This may cause severe asymmetry between two classes when $\|\boldsymbol{x}_+\| \neq \|\boldsymbol{x}_-\|$, although $\bar{\boldsymbol{v}}$, $\bar{\boldsymbol{W}}$ still exist in that case. Nevertheless, we claim that such an asymmetry can be prevented in a deep network equipped with batch normalization, since batch normalization balances two class means of next layer's input. We also argue that the directional collapse can also appear under weight normalization, and in that case the restriction in magnitudes prevents the severe asymmetry $\max\{s_+(t)/s_-(t), s_-(t)/s_+(t)\} \underset{t\to\infty}{\to} \infty$ (see Appendix D.4).

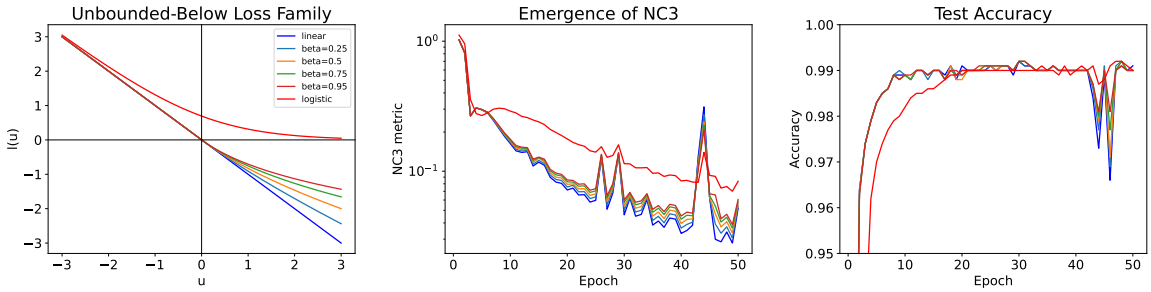

Figure 2: (Left) Visualization of linear loss, logistic loss, and unbounded-below loss family defined in (3). (Middle and Right) Emergence of NC3 and test accuracy under different losses. NC1 and NC2 exhibit similar behavior (see Appendix E).

6. Discussion: Towards Numerically Stable Unbounded-Below Losses

Figure 1 demonstrates that linear loss is more efficient than logistic loss, but also shows that training with linear loss is numerically less stable than training with logistic loss, even with weight normalization. We conjecture that this reflects a tradeoff between convergence speed and numerical stability arising from larger gradient norms. To investigate this conjecture, we further consider a loss family whose gradients decay more slowly than those of logistic loss, but more rapidly than those of linear loss. This family is defined by

$$\ell_\beta(u) = \begin{cases} -u, & u \le 0, \\ -\frac{(1+u)^{1-\beta}-1}{1-\beta}, & u \ge 0, \end{cases} \quad \text{where } 0 < \beta < 1. \tag{3}$$

Figure 2 compares this loss family with the linear and logistic losses. This family achieves nearly the same performance as linear loss while being considerably more numerically stable, even without weight normalization. These results suggest that numerically stable unbounded-below losses can retain the fast training dynamics of linear loss while substantially improving numerical stability, making them a promising direction for future research.

7. Conclusion

In this paper, we introduced the linear loss $\ell(u) = -u$ and studied its role in binary classification. On the theoretical side, we showed that the unconstrained feature model (UFM) exhibits neural collapse (NC), while shallow ReLU networks exhibit a weaker directional collapse of features and classifiers. Empirically, we demonstrated that linear loss, when combined with weight normalization, accelerates the emergence of NC and enables faster convergence to the same generalization performance. Taken together, these results establish linear loss as a simple yet effective alternative to standard classification losses. Furthermore, our study suggests that numerically stable unbounded-below losses constitute a promising direction for developing efficient and numerically stable classification losses.

Acknowledgement

This work was supported in part by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (Nos. RS-2019-NR040050 and RS-2025-00515159), and by the KAIST Grand Challenge 30 Project (KC30) funded by MSIT and KAIST (No. N11260046).

References

[1] Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17:1205–1223, 2007.

[2] Etienne Boursier, Loucas Pillaud-Vivien, and Nicolas Flammarion. Gradient flow dynamics of shallow ReLU networks for square loss and orthogonal inputs. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[3] Lénaïc Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125, pages 1305–1338, 2020.

[4] Simon S. Du, Wei Hu, and Jason D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[5] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[6] Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99, pages 1772–1798, 2019.

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

[8] Ziwei Ji and Matus Telgarsky. Characterizing the implicit bias via a primal-dual analysis. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132, pages 772–804, 2021.

[9] Hancheng Min, Enrique Mallada, and René Vidal. Early neuron alignment in two-layer ReLU networks with small initialization. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[10] Hancheng Min, Zhihui Zhu, and René Vidal. Neural collapse under gradient flow on shallow ReLU networks for orthogonally separable data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[11] Dustin G. Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *Sampling Theory, Signal Processing, and Data Analysis*, 20(2):11, 2022.

[12] Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[13] Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[14] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018.

[15] Ruiqi Zhang, Jingfeng Wu, and Peter Bartlett. Gradient descent converges arbitrarily fast for logistic regression via large and adaptive stepsizes. In *Forty-second International Conference on Machine Learning*, 2025.

Appendix A. Proof of Bayes Consistency of Linear Loss

Let $\eta(\boldsymbol{x}) = P(Y = 1 \mid X = \boldsymbol{x})$. The expected risk under the linear loss is

$$\mathcal{R}(f) = \mathbb{E}_{X,Y}[-Yf(X)].$$

Conditioning on $X = \boldsymbol{x}$, we have

$$\begin{aligned}\mathcal{R}(f \mid \boldsymbol{x}) &= \mathbb{E}[-Yf(\boldsymbol{x}) \mid X = \boldsymbol{x}] \\ &= -\eta(\boldsymbol{x})f(\boldsymbol{x}) + (1 - \eta(\boldsymbol{x}))f(\boldsymbol{x}) \\ &= (1 - 2\eta(\boldsymbol{x}))f(\boldsymbol{x}).\end{aligned}$$

Since the linear loss is unbounded below, the conditional risk does not admit a finite minimizer. Instead, its infimum is approached by

$$f(\boldsymbol{x}) \to \begin{cases} +\infty, & \eta(\boldsymbol{x}) > \frac{1}{2}, \\ -\infty, & \eta(\boldsymbol{x}) < \frac{1}{2}. \end{cases}$$

Therefore,

$$\operatorname{sgn}(f(\boldsymbol{x})) = \begin{cases} +1, & P(Y = 1 \mid X = \boldsymbol{x}) > P(Y = -1 \mid X = \boldsymbol{x}), \\ -1, & P(Y = 1 \mid X = \boldsymbol{x}) < P(Y = -1 \mid X = \boldsymbol{x}), \end{cases}$$

which coincides with the Bayes optimal decision rule. Hence, the linear loss is Bayes consistent.

Appendix B. Detailed Setup for Numerical Experiment

B.1. Problem Setup

We consider a binary classification of digit 0 and digit 8 of MNIST dataset. We use only 2000 train examples and 400 test examples, to focus on the terminal phase of training.

B.2. Model and Architecture

We train a fully connected three-layer (two hidden layers) ReLU network with width 32. Batch normalization is applied before every ReLU activation. We use near-zero initialization and train both networks by using GD with stepsize 0.0001. For linear loss, we normalize the norm of each neuron to be a constant (output layer : 0.2, other layers : 0.5).

B.3. NC Metrics

We define several metrics to measure the emergence of NC1∼NC3, based on their definition. We calculate $\operatorname{tr}(\boldsymbol{\Sigma}_W \boldsymbol{\Sigma}_B^{\dagger})$ to measure NC1, and calculate $\| \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} - \frac{\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-}{\|\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-\|} \|$ to measure NC3. In binary classification, NC2 is always true, since in that case $\boldsymbol{\mu}_G = \frac{1}{2}(\boldsymbol{\mu}_+ + \boldsymbol{\mu}_-)$ and hence automatically $\boldsymbol{\mu}_+ - \boldsymbol{\mu}_G = -(\boldsymbol{\mu}_- - \boldsymbol{\mu}_G)$. However, we use $\langle \frac{\boldsymbol{\mu}_+}{\|\boldsymbol{\mu}_+\|}, \frac{\boldsymbol{\mu}_-}{\|\boldsymbol{\mu}_-\|} \rangle$ to measure NC2 as [10] does, since it can measure the progress of neural collapse (second part of Theorem 4), and the orthogonality of class means implies the formulation of simplex ETF consists of centered class means, even for the multi-class problem.

Appendix C. Proof of Results in Section 5.1

C.1. Emergence of Weaker Form of NC under UFM

Let $\boldsymbol{H}$ be a collection of $\boldsymbol{h}_i$, which is the feature mapping of $\boldsymbol{x}_i$. Then our problem becomes

$$\min_{\boldsymbol{H},\boldsymbol{v},b} \mathcal{L}(\boldsymbol{H},\boldsymbol{v},b) = -\sum_{i=1}^{n} y_i(\boldsymbol{v}^\top \boldsymbol{h}_i + b) \tag{4}$$

Let $\mathcal{I}_\pm$ be index sets of positive/negative labels respectively. Then the gradient flow yields

$$\dot{\boldsymbol{v}} = \sum_{i=1}^{n} y_i \boldsymbol{h}_i = \sum_{i\in\mathcal{I}^+} \boldsymbol{h}_i - \sum_{i\in\mathcal{I}^-} \boldsymbol{h}_i, \tag{5}$$

$$\dot{\boldsymbol{h}_i} = y_i \boldsymbol{v} = \begin{cases} \boldsymbol{v} & (i \in \mathcal{I}_+) \\ -\boldsymbol{v} & (i \in \mathcal{I}_-) \end{cases} \tag{6}$$

Combining Equation (5) and Equation (6) yields

$$\ddot{\boldsymbol{v}} = n\boldsymbol{v}, \boldsymbol{v}(0) = \boldsymbol{v}_0, \dot{\boldsymbol{v}}(0) = \boldsymbol{h}_0 \tag{7}$$

where $\boldsymbol{h}_0 = [\sum_{i\in\mathcal{I}^+} \boldsymbol{h}_i - \sum_{i\in\mathcal{I}^-} \boldsymbol{h}_i]_{t=0}$. The solution is

$$\boldsymbol{v}(t) = \frac{1}{2}\left(\boldsymbol{v}_0 + \frac{\boldsymbol{h}_0}{\sqrt{n}}\right) e^{\sqrt{n}t} + \frac{1}{2}\left(\boldsymbol{v}_0 - \frac{\boldsymbol{h}_0}{\sqrt{n}}\right) e^{-\sqrt{n}t} \tag{8}$$

and

$$\begin{aligned} \boldsymbol{h}_i(t) &= \boldsymbol{h}_i(0) + y_i \int_0^t \boldsymbol{v}(\tau) d\tau \\ &= \boldsymbol{h}_i(0) + y_i \left[\frac{1}{2\sqrt{n}}\left(\boldsymbol{v}_0 + \frac{\boldsymbol{h}_0}{\sqrt{n}}\right) e^{\sqrt{n}t} - \frac{1}{2\sqrt{n}}\left(\boldsymbol{v}_0 - \frac{\boldsymbol{h}_0}{\sqrt{n}}\right) e^{-\sqrt{n}t}\right] \end{aligned}$$

Note that this is not a 'complete NC', because features of each class does not shrink into a single point. However, since each class is pushed indefinitely toward the direction $\pm\left(\boldsymbol{v}_0 + \frac{\boldsymbol{h}_0}{\sqrt{n}}\right)$, the within-class variability becomes relatively negligible, compared to the between-class varaibility. This yields similar phenomena to NC.

- Since all points in the same class move identically, within-class variability remains constant. Nevertheless, the between-class variability grows indefinitely, so $\mathrm{tr}(\boldsymbol{\Sigma}_W \boldsymbol{\Sigma}_B^\dagger) \to 0$, which is a weaker form of NC1.
- In binary classification, NC2 is automatically satisfied, as described in Appendix B.
- Since $\boldsymbol{v}, \boldsymbol{h}_+$ converges in direction to $\boldsymbol{v}_0 + \frac{\boldsymbol{h}_0}{\sqrt{n}}$ and $\boldsymbol{h}_-$ converges in direction to $-\left(\boldsymbol{v}_0 + \frac{\boldsymbol{h}_0}{\sqrt{n}}\right)$, we can deduce that

$$\|\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} - \frac{\boldsymbol{h}_+ - \boldsymbol{h}_-}{\|\boldsymbol{h}_+ - \boldsymbol{h}_-\|}\| \to 0, \tag{9}$$

  which is exactly NC3.

C.2. Complete NC under UFM with Weight Normalization

We already saw that within-class variance does not vanish, despite the UFM assumption. This is because, when $\boldsymbol{h}_i, \boldsymbol{h}_j$ are in same class, $\boldsymbol{h}_i - \boldsymbol{h}_j$ always remains constant, while $\frac{\|\boldsymbol{h}_i - \boldsymbol{h}_j\|}{\|\boldsymbol{h}_i\|}$ converges to 0. However, weight normalization suppresses the norms of the norms of $\boldsymbol{h}_i$ $\boldsymbol{h}_j$ while preserving their directions, causing $\boldsymbol{h}_i - \boldsymbol{h}_j$ to eventually converge to $\mathbf{0}$.

To formalize, suppose that the model is sufficiently overparameterized, but the norm of features is limited due to the weight normalization. Then these hidden layers can interpolate any feature $\boldsymbol{h}$ with norm restriction $\|\boldsymbol{h}\| \leq C$. In short, we consider the following minimization problem :

$$\min_{\boldsymbol{H},\boldsymbol{v}} -\sum_{i=1}^{n} y_i \boldsymbol{v}^\top \boldsymbol{h}_i, \quad \text{s.t. } \|\boldsymbol{v}\|, \|\boldsymbol{h}_i\| \leq 1. \tag{10}$$

In this case we have the following differential inclusion :

$$\frac{d\boldsymbol{v}}{dt} \in \sum_{i=1}^{n} y_i \boldsymbol{h}_i - \mathcal{N}_{B_{\boldsymbol{v}}}(\boldsymbol{v}), \quad \frac{d\boldsymbol{h}_i}{dt} \in y_i \boldsymbol{v} - \mathcal{N}_{B_{\boldsymbol{h}}}(\boldsymbol{v}). \tag{11}$$

Here, $B_{\boldsymbol{x}} = \{\boldsymbol{x} \mid \|\boldsymbol{x}\| \leq 1\}$. Since $\mathcal{N}_{B_{\boldsymbol{x}}} = \begin{cases} 0 & \|\boldsymbol{x}\| < 1 \\ \{\lambda \boldsymbol{x} : \lambda \geq 0\} & \|\boldsymbol{x}\| = 1 \end{cases}$, the global minima should satisfy

$$\exists \lambda \geq 0 \text{ s.t. } \lambda \boldsymbol{v} = \sum_{i=1}^{n} y_i \boldsymbol{h}_i, \quad \forall i,\ \exists \mu_i \geq 0 \text{ s.t. } \mu_i \boldsymbol{h}_i = y_i \boldsymbol{v} \tag{12}$$

This implies $\boldsymbol{h}_i = y_i \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}$ and $\boldsymbol{v} = \sum_{i=1}^{n} y_i \boldsymbol{h}_i = \boldsymbol{h}_+ - \boldsymbol{h}_-$, which is the complete NC.

Appendix D. Proof of Results in Section 5.2

In this section, we prove Theorem 4. The proofs, especially those in Appendix D.2, are based on [9], [10], whose idea is roughly as follows.

- **(Gradient flow analysis)** By calculating $\frac{d}{dt}\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$, one can prove the following: if the initial sign of v_j is positive/negative, then $\boldsymbol{w}_j$ will be attracted by positively/negatively labeled data.
- **(Early neural alignment)** As a result, if the initial sign of v_j is positive/negative, then $\boldsymbol{w}_j$ will eventually activate only positively/negatively labeled data, respectively.
- **(Asymptotic analysis)** Due to neural alignment, we can decompose the entire network into two linear networks, each exclusively activated by positively or negatively labeled data. The existing asymptotic convergence results [7] for linear networks imply the directional collapse of features and subsequent NC phenomena.

There are two major differences between the analysis of [10] and ours. First, surprisingly, our linear loss removes the error term when analyzing $\frac{d}{dt}\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$. This not only simplifies the proof of neural alignment, but also reduces the iteration complexity required to achieve it. Second, since we consider a different loss function, we cannot directly apply the result of [7]. In Appendix D.3 we proved that the linear loss also induces the directional collapse asymptotically.

D.1. Gradient Flow Analysis

Recall that

$$\mathcal{L}(\boldsymbol{v}, \boldsymbol{W}) = \sum_{i=1}^{n} -y_i \boldsymbol{v}^\top \sigma(\boldsymbol{W}^\top \boldsymbol{x}_i) = \sum_{i=1}^{n} \sum_{j=1}^{h} -y_i v_j \sigma(\boldsymbol{w}_j^\top \boldsymbol{x}_i). \tag{13}$$

Hence,

$$\frac{d}{dt}\boldsymbol{w}_j = -\partial_{\boldsymbol{w}_j} \mathcal{L}(\boldsymbol{v}, \boldsymbol{W}) = v_j \sum_{\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle > 0} y_i \boldsymbol{x}_i \tag{14}$$

and

$$\begin{aligned} \frac{d}{dt}\frac{\boldsymbol{w}_j}{\|\boldsymbol{w}_j\|} &= \frac{1}{\|\boldsymbol{w}_j\|}\left(\boldsymbol{I} - \frac{\boldsymbol{w}_j \boldsymbol{w}_j^\top}{\|\boldsymbol{w}_j\|^2}\right)\left(v_j \sum_{\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle > 0} y_i \boldsymbol{x}_i\right) \\ &= \operatorname{sgn}(v_j)\left(\boldsymbol{I} - \frac{\boldsymbol{w}_j \boldsymbol{w}_j^\top}{\|\boldsymbol{w}_j\|^2}\right) \sum_{\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle > 0} y_i \boldsymbol{x}_i \end{aligned} \tag{15}$$

Where the last equality is due to balanced initialization (Assumption 2) : under balanced initialization, the balance is maintained throughout the GF training [4]. We recall that the sign of v_j does not change throughout the training [2].

This formula gives us intuition about the change of the input neuron $\boldsymbol{w}_j$. If $v_j(t) > 0$, then $\boldsymbol{w}_j$ moves toward $\sum_{\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle > 0} y_i \boldsymbol{x}_i$. As a result, $\boldsymbol{w}_j$ is attracted by positively labeled data and repelled by negatively labeled data. In contrast, if $v_j(t) < 0$, $\boldsymbol{w}_j$ is repelled by positively labeled data and attracted by negatively labeled data. Since $\operatorname{sgn}(v_j)$ does not change over time, one can expect that the input neurons are separated into two groups, according to the sign of the corresponding output neuron v_j. This observation is a key idea in proving neural alignment. Since the quantity $\sum_{\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle > 0} y_i \boldsymbol{x}_i$ will appear repeatedly in the proof, so we denote it by $\boldsymbol{x}_a(\boldsymbol{w}_j)$ for notational convenience.

D.2. Early Neuron Alignment

Throughout this section, we need several notations for simplification. We first separate datapoints and output neurons according to their labels and initial signs, respectively.

$$\mathcal{I}_+ = \{i \in [n] \mid y_i = +1\}, \quad \mathcal{I}_- = \{i \in [n] \mid y_i = -1\} \tag{16}$$

$$\mathcal{N}_+(t) = \{j \in [h] \mid v_j(t) > 0\}, \quad \mathcal{N}_-(t) = \{j \in [h] \mid v_j(t) < 0\} \tag{17}$$

Recall that for balanced initialization, the sign of $v_j(t)$ does not change over time, so we simply denote $\mathcal{N}_+ = \mathcal{N}_+(0)$, $\mathcal{N}_- = \mathcal{N}_-(0)$ and consider only those sets. We additionally use the following notation:

$$x_{\min} := \min_{1 \le i \le n} \|\boldsymbol{x}_i\|, \quad \boldsymbol{x}_+ := \sum_{i \in \mathcal{I}_+} \boldsymbol{x}_i, \quad \boldsymbol{x}_- := \sum_{i \in \mathcal{I}_-} \boldsymbol{x}_i \tag{18}$$

We also define

$$\mathcal{S}_+ := \{\boldsymbol{w} \mid \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle > 0 \Leftrightarrow y_i = +1,\ \forall i\}, \quad \mathcal{S}_- := \{\boldsymbol{w} \mid \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle > 0 \Leftrightarrow y_i = -1,\ \forall i\} \tag{19}$$

and

$$\mathcal{S}_{dead} := \{\boldsymbol{w} \mid \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle < 0,\ \forall i\} \tag{20}$$

The following observations will be used throughout the proof.

- $\mathcal{S}_+, \mathcal{S}_-, \mathcal{S}_{dead}$ are cones except for the absence of the origin. Moreover, $\operatorname{Int}(\mathcal{S}_+) = -\operatorname{Int}(\mathcal{S}_-)$.
- If $y_i = +1$, then $\boldsymbol{x}_i \in \operatorname{Int} \mathcal{S}_+$ and $-\boldsymbol{x}_i \in \operatorname{Int} \mathcal{S}_-$ due to orthogonal separability. Similarly, if $y_i = -1$, $\boldsymbol{x}_i \in \operatorname{Int} \mathcal{S}_-$ and $-\boldsymbol{x}_i \in \operatorname{Int} \mathcal{S}_+$. As a result, $\boldsymbol{x}_a(\boldsymbol{w}) \in \operatorname{Int} \mathcal{S}_+$, $-\boldsymbol{x}_a(\boldsymbol{w}) \in \operatorname{Int} \mathcal{S}_-$ for any $\boldsymbol{w}$.

Now we present our first important observation.

Claim 1. $\boldsymbol{x}_a(\boldsymbol{w}_j)$ will change its value only finitely many times.

To prove this claim, we introduce several lemmas. The following lemma quantifies the alignment of the input neuron weights $\{\boldsymbol{w}_j\}$ and data $\{\{\boldsymbol{x}_i\}$, described in Section D.1.

Lemma 5 *Let $\boldsymbol{x}_r \in \mathbb{S}^{D-1}$ be a fixed reference vector. Then*

$$\frac{d}{dt} \cos(\boldsymbol{x}_r, \boldsymbol{w}_j) = \operatorname{sgn}(v_j)(\cos(\boldsymbol{x}_r, \boldsymbol{x}_a(\boldsymbol{w}_j)) - \cos(\boldsymbol{x}_r, \boldsymbol{w}_j) \cos(\boldsymbol{w}_j, \boldsymbol{x}_a(\boldsymbol{w}_j)))\|\boldsymbol{x}_a(\boldsymbol{w}_j)\| \tag{21}$$

Note that this lemma is a direct consequence of Equation (15). The following lemma establishes a lower bound for $\|\boldsymbol{x}_a(\boldsymbol{w})\|$, enabling us to apply Lemma 5.

Lemma 6 (Lemma 11 of [9])

$$\|\boldsymbol{x}_a(\boldsymbol{w})\| \ge \sqrt{\mu} n_a(\boldsymbol{w}) x_{\min}, \tag{22}$$

where $n_a(\boldsymbol{w}) = |\{i \in [n] \mid \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle > 0\}|$.

Proof Let $\mathcal{I}_a(\boldsymbol{w}) = \{i \in [n] \mid \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle > 0\}$. Then

$$\begin{aligned}
\|\boldsymbol{x}_a(\boldsymbol{w})\| &= \|\sum_{i \in \mathcal{I}_a(\boldsymbol{w})} y_i \boldsymbol{x}_i\| \\
&= \sqrt{\sum_{i \in \mathcal{I}_a(\boldsymbol{w})} \|y_i \boldsymbol{x}_i\|^2 + \sum_{i,j \in \mathcal{I}_a(\boldsymbol{w}), i \neq j} \langle y_i \boldsymbol{x}_i, y_j \boldsymbol{x}_j \rangle} \\
&= \sqrt{\sum_{i \in \mathcal{I}_a(\boldsymbol{w})} y_i^2 \|\boldsymbol{x}_i\|^2 + \sum_{i,j \in \mathcal{I}_a(\boldsymbol{w}), i \neq j} \|\boldsymbol{x}_i\| \|\boldsymbol{x}_j\| \langle \frac{y_i \boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}, \frac{y_j \boldsymbol{x}_j}{\|\boldsymbol{x}_j\|} \rangle} \\
&\geq \sqrt{\sum_{i \in \mathcal{I}_a(\boldsymbol{w})} \|\boldsymbol{x}_i\|^2 + \sum_{i,j \in \mathcal{I}_a(\boldsymbol{w}), i \neq j} \mu \|\boldsymbol{x}_i\| \|\boldsymbol{x}_j\|} \\
&= \sqrt{n_a(\boldsymbol{w}) x_{\min}^2 + \mu n_a(\boldsymbol{w})(n_a(\boldsymbol{w}) - 1) x_{\min}^2} \\
&= \sqrt{n_a(\boldsymbol{w}) + \mu n_a(\boldsymbol{w})(n_a(\boldsymbol{w}) - 1)} x_{\min} \\
&= \sqrt{\mu n_a(\boldsymbol{w})^2 + (1 - \mu) n_a(\boldsymbol{w})} x_{\min} \\
&\geq \sqrt{\mu} n_a(\boldsymbol{w}) x_{\min}
\end{aligned}$$

■

Proof of Claim 1. It is trivial that $\boldsymbol{w}_j \in \mathcal{S}_{dead}$ does not change $\boldsymbol{x}_a(\boldsymbol{w}_j)(= 0)$. Suppose that $j \in \mathcal{N}_+$ and $\boldsymbol{w}_j \notin \mathcal{S}_{dead}$. From Lemma 5, we obtain

$$\frac{d}{dt} y_i \langle \frac{\boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}, \frac{\boldsymbol{w}_j}{\|\boldsymbol{w}_j\|} \rangle |_{\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle = 0} = \langle \frac{y_i \boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}, \frac{\boldsymbol{x}_a(\boldsymbol{w}_j)}{\|\boldsymbol{x}_a(\boldsymbol{w}_j)\|} \rangle \|\boldsymbol{x}_a(\boldsymbol{w}_j)\|. \tag{23}$$

By orthogonal separability, $\langle \frac{y_i \boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}, \frac{\boldsymbol{x}_a(\boldsymbol{w}_j)}{\|\boldsymbol{x}_a(\boldsymbol{w}_j)\|} \rangle \geq \mu$. Hence, by Lemma 6,

$$\frac{d}{dt} y_i \langle \frac{\boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}, \frac{\boldsymbol{w}_j}{\|\boldsymbol{w}_j\|} \rangle |_{\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle = 0} \geq \mu^{3/2} n_a(\boldsymbol{w}) x_{\min} > 0 \tag{24}$$

This implies two things. First, if $y_i = +1$ and $\boldsymbol{x}_i$ is activated, then it cannot be deactivated, i.e., cannot cross the boundary $\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle = 0$, since $\frac{d}{dt} \langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle > 0$ at the boundary. Similarly, if $y_i = -1$ and $\boldsymbol{x}_i$ is deactivated, then it cannot be activated since $\frac{d}{dt} \langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle < 0$ at the boundary. Hence, $\boldsymbol{x}_a(\boldsymbol{w}_j)$ can change its value only when a positively labeled datapoint becomes activated or a negatively labeled datapoint becomes deactivated. This implies that the $\boldsymbol{x}_a(\boldsymbol{w}_j)$ changes its value at most n times. The case $j \in \mathcal{N}_-$ can be proved in a similar manner. ■

Next, we present the second important observation.

Claim 2. $\boldsymbol{x}_a(\boldsymbol{w}_j)$ will change its value within a finite amount of time.

To prove this claim, we need several technical lemmas. The following lemma formalizes the non-degenerate initialization (Assumption 3).

Lemma 7 (Formalization of nondegenerate initialization) *There exists $\zeta > 0$ s.t.*

$$\max_{j \in \mathcal{N}_+} \cos(\boldsymbol{w}_j(0), \boldsymbol{x}_-) \leq \sqrt{1 - \zeta}, \quad \max_{j \in \mathcal{N}_-} \cos(\boldsymbol{w}_j(0), \boldsymbol{x}_+) \leq \sqrt{1 - \zeta} \tag{25}$$

The following lemma is an easy consequence of our previous observation $\boldsymbol{x}_a(\boldsymbol{w}) \in \operatorname{Int} \mathcal{S}_+$.

Lemma 8 (Lemma 9 of [9]) *Let $\boldsymbol{x}_1$ be a positive linear combination of $y_i\boldsymbol{x}_i$s, and let $\boldsymbol{x}_2 \in \mathcal{S}_+^c \cap \mathcal{S}_-^c$. Then there exist $\xi > 0$ s.t.*

$$\sup |\cos(\boldsymbol{x}_1, \boldsymbol{x}_2)| \leq \sqrt{1-\xi}. \tag{26}$$

Proof of Claim 2. We focus on the case when $j \in \mathcal{N}_+$. By ***Claim 1***, $\boldsymbol{x}_a(\boldsymbol{w}_j)$ does not change when $\boldsymbol{w}_j \in \mathcal{S}_+ \cup \mathcal{S}_{dead}$. Hence, it suffices to consider the following two cases: $\boldsymbol{w}_j \in \mathcal{S}_-$ or $\boldsymbol{w}_j \in \mathcal{S}_+^c \cap \mathcal{S}_-^c \cap \mathcal{S}_{dead}^c$.

We first consider the case $\boldsymbol{w}_j \in \mathcal{S}_-$. Then $\boldsymbol{x}_a(\boldsymbol{w}_j) = -\boldsymbol{x}_-$ by the definition of $\mathcal{S}_-$. Let $t_1 > 2/(\zeta\sqrt{\mu}x_{\min})$, and assume that $\boldsymbol{x}_a(\boldsymbol{w}_j(t_0)) = \boldsymbol{x}_a(\boldsymbol{w}_j(t_0+t_1))$. Then we choose $\boldsymbol{x}_r = \boldsymbol{x}_-$ and use Lemma 5, which yields

$$\frac{d}{dt}\cos(\boldsymbol{w}_j(t), \boldsymbol{x}_-) = -(1-\cos^2(\boldsymbol{w}_j(t), \boldsymbol{x}_-))\|\boldsymbol{x}_-\| \tag{27}$$

This formula, together with Lemma 6 and Lemma 7, shows that for any $t \in [t_0, t_0+t_1]$,

$$\frac{d}{dt}\cos(\boldsymbol{w}_j(t), \boldsymbol{x}_-) \leq -\zeta n_a(\boldsymbol{w}_j(t))\|\boldsymbol{x}_-\| \leq -\zeta\sqrt{\mu}x_{\min}. \tag{28}$$

Now, by the fundamental theorem of calculus,

$$\begin{aligned}
\cos(\boldsymbol{w}_j(t_0+t_1), \boldsymbol{x}_-) &= \cos(\boldsymbol{w}_j(t_0), \boldsymbol{x}_-) + \int_{t_0}^{t_0+t_1} \frac{d}{dt}\cos(\boldsymbol{w}_j(t_0+\tau), \boldsymbol{x}_-)d\tau \\
&\leq \cos(\boldsymbol{w}_j(t_0), \boldsymbol{x}_-) + \int_{t_0}^{t_0+t_1} -\zeta\sqrt{\mu}x_{\min}d\tau \\
&< \cos(\boldsymbol{w}_j(t_0), \boldsymbol{x}_-) - 2 \\
&\leq -1
\end{aligned}$$

This is a contradiction, so we can deduce that $\boldsymbol{x}_a(\boldsymbol{w}_j)$ will change within a finite amount of time t_1.

Now suppose that $\boldsymbol{w}_j \in \mathcal{S}_+^c \cap \mathcal{S}_-^c \cap \mathcal{S}_{dead}^c$. Let $t_2 > 2/(\xi\sqrt{\mu}x_{\min})$, and assume that $\boldsymbol{x}_a(\boldsymbol{w}_j(t_0)) = \boldsymbol{x}_a(\boldsymbol{w}_j(t_0+t_2))$. Then for $t \in [t_0, t_0+t_2]$, we can use Lemma 5 with $\boldsymbol{x}_r = \boldsymbol{x}_a(\boldsymbol{w}_j(t_0))$ since $\boldsymbol{x}_a(\boldsymbol{w}_j)$ is fixed in that time interval. Lemma 5 yields

$$\frac{d}{dt}\cos(\boldsymbol{w}_j(t), \boldsymbol{x}_a(\boldsymbol{w}_j(t_0))) = (1-\cos^2(\boldsymbol{w}_j(t), \boldsymbol{x}_a(\boldsymbol{w}_j(t_0))))\|\boldsymbol{x}_a(\boldsymbol{w}_j(t_0))\| \tag{29}$$

Similarly, Lemma 6 and Lemma 8 implies that

$$\frac{d}{dt}\cos(\boldsymbol{w}_j(t), \boldsymbol{x}_a(\boldsymbol{w}_j(t_0))) \geq \xi\sqrt{\mu}x_{\min}, \quad \forall t \in [t_0, t_0+t_2] \tag{30}$$

Then FTC yields contradiction in a similar manner, so we can deduce that $\boldsymbol{x}_a(\boldsymbol{w}_j)$ will change within a finite amount of time t_2. The proof of the case $j \in \mathcal{N}_-$ is is similar and hence omitted for brevity. ■

Now, the combination of **Claim 1** and **Claim 2** directly implies the following theorem.

Theorem 9 (Early Neural Alignment) *Let* $T = \frac{2n}{\sqrt{\mu} x_{\min}} \times \frac{1}{\min\{\zeta, \xi\}}$. *Then following statements hold :*

- *If* $j \in \mathcal{N}_+$, *then either* $\boldsymbol{w}_j(T) \in \mathcal{S}_+$ *or* $\boldsymbol{w}_j(T) \in \mathcal{S}_{dead}$, *and remains permanently in one of those cones. Moreover, if* $\max_{i \in \mathcal{I}_+} \langle \boldsymbol{w}_j(0), \boldsymbol{x}_i \rangle > 0$, *then* $\boldsymbol{w}_j(T) \in \mathcal{S}_+$.
- *If* $j \in \mathcal{N}_-$, *then either* $\boldsymbol{w}_j(T) \in \mathcal{S}_-$ *or* $\boldsymbol{w}_j(T) \in \mathcal{S}_{dead}$, *and remains permanently in one of those cones. Moreover, if* $\max_{i \in \mathcal{I}_-} \langle \boldsymbol{w}_j(0), \boldsymbol{x}_i \rangle > 0$, *then* $\boldsymbol{w}_j(T) \in \mathcal{S}_-$.

D.3. Asymptotic Behavior

Using Theorem 9, we now show that the entire network can be decomposed into two linear networks, each of which exclusively activates positively or negatively labeled data. If $\boldsymbol{w}_j \in \mathcal{S}_{dead}$, there is nothing we need to do since that neuron won't activate any data. Therefore, we may assume that $\boldsymbol{w}_j \in \mathcal{S}_+$ if $j \in \mathcal{N}_+$, and $\boldsymbol{w}_j \in \mathcal{S}_-$ if $j \in \mathcal{N}_-$. The proof is analogous if there exists a dead neuron.

Define $\boldsymbol{W}_\pm = [\boldsymbol{w}_j]_{j \in \mathcal{N}_\pm}$, $\boldsymbol{v}_\pm = [v_j]_{j \in \mathcal{N}_\pm}^\top$. Without loss of generality, we assume that $\mathcal{N}_+ = \{1, \cdots, k\}, \mathcal{N}_- = \{k+1, \cdots, h\}$ for some $1 \le k \le h$. From Appendix D.2, we deduce that

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{v}, \boldsymbol{W}) &= \sum_{j=1}^{h} \sum_{i=1}^{n} -y_i v_j \sigma(\boldsymbol{w}_j^\top \boldsymbol{x}_i) \\
&= \sum_{j=1}^{k} \sum_{i=1}^{n} -y_i v_j \sigma(\boldsymbol{w}_j^\top \boldsymbol{x}_i) + \sum_{j=k+1}^{h} \sum_{i=1}^{n} -y_i v_j \sigma(\boldsymbol{w}_j^\top \boldsymbol{x}_i) \\
&= -\sum_{j=1}^{k} \sum_{i \in \mathcal{I}_+} v_j \boldsymbol{w}_j^\top \boldsymbol{x}_i + \sum_{j=k+1}^{h} \sum_{i \in \mathcal{I}_-} v_j \boldsymbol{w}_j^\top \boldsymbol{x}_i \qquad (\because \text{Theorem 9}) \\
&= -\sum_{i \in \mathcal{I}_+} \boldsymbol{v}_+^\top \boldsymbol{W}_+^\top \boldsymbol{x}_i + \sum_{i \in \mathcal{I}_-} \boldsymbol{v}_-^\top \boldsymbol{W}_-^\top \boldsymbol{x}_i \\
&= -\boldsymbol{v}_+^\top \boldsymbol{W}_+^\top \boldsymbol{x}_+ - \boldsymbol{v}_-^\top \boldsymbol{W}_-^\top (-\boldsymbol{x}_-)
\end{aligned}
$$

In other words, the dynamics of the neurons in $\{\boldsymbol{w}_j, v_j\}_{j \in \mathcal{N}_+}$ are completely decoupled from those in $\{\boldsymbol{w}_j, v_j\}_{j \in \mathcal{N}_-}$, and the dynamics of each set of neurons can be fully described by analyzing the asymptotic behavior of two-layer linear networks trained by linear loss.

Proof of Theorem 4. We consider only the positive class, as the proof for the negative class is analogous. We also omit the subscript $\cdot_+$ to simplify the notation. Namely, we are considering loss of the following form :

$$
\hat{\mathcal{L}}(\boldsymbol{v}, \boldsymbol{W}) = -\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x}. \tag{31}
$$

If we train $\boldsymbol{v}, \boldsymbol{W}$ by gradient flow, then

$$
\frac{d}{dt} \boldsymbol{v} = \boldsymbol{W}^\top \boldsymbol{x}, \quad \frac{d}{dt} \boldsymbol{W} = \boldsymbol{x} \boldsymbol{v}^\top \tag{32}
$$

and hence

$$\frac{d}{dt}\|\boldsymbol{v}\| = \frac{\boldsymbol{v}^\top \frac{d\boldsymbol{v}}{dt}}{\|\boldsymbol{v}\|} = \frac{\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x}}{\|\boldsymbol{v}\|}, \tag{33}$$

$$\frac{d}{dt}\|\boldsymbol{W}\| = \frac{\mathrm{tr}(\boldsymbol{W}^\top \frac{d\boldsymbol{W}}{dt})}{\|\boldsymbol{W}\|} = \frac{\mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{x}\boldsymbol{v}^\top)}{\|\boldsymbol{W}\|} = \frac{\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x}}{\|\boldsymbol{W}\|}. \tag{34}$$

Hence,

$$\frac{d}{dt}\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} = \frac{\boldsymbol{W}^\top \boldsymbol{x}}{\|\boldsymbol{v}\|} - \frac{(\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x})\boldsymbol{v}}{\|\boldsymbol{v}\|^3}, \quad \frac{d}{dt}\frac{\boldsymbol{W}}{\|\boldsymbol{W}\|} = \frac{\boldsymbol{x}\boldsymbol{v}^\top}{\|\boldsymbol{W}\|} - \frac{(\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x})\boldsymbol{W}}{\|\boldsymbol{W}\|^3}. \tag{35}$$

Now, we consider the scalar $c = \frac{\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x}}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|} = \frac{\boldsymbol{x}^\top \boldsymbol{W}\boldsymbol{v}}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|}$ and its time derivative dc/dt.

$$\begin{aligned}
\frac{d}{dt}c =& \boldsymbol{x}^\top \left(\frac{d}{dt}\frac{\boldsymbol{W}}{\|\boldsymbol{W}\|}\right)\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} + \boldsymbol{x}^\top \frac{\boldsymbol{W}}{\|\boldsymbol{W}\|}\left(\frac{d}{dt}\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\right) \\
=& \left[\frac{\boldsymbol{x}^\top \boldsymbol{x}\boldsymbol{v}^\top \boldsymbol{v}}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|} - \frac{\boldsymbol{x}^\top(\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x})\boldsymbol{W}\boldsymbol{v}}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|^3}\right] + \left[\frac{\boldsymbol{x}^\top \boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{x}}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|} - \frac{\boldsymbol{x}^\top \boldsymbol{W}(\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x})\boldsymbol{v}}{\|\boldsymbol{v}\|^3\|\boldsymbol{W}\|}\right] \\
=& \frac{1}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|^3}\left[\|\boldsymbol{x}\|^2\|\boldsymbol{v}\|^2\|\boldsymbol{W}\|^2 - (\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x})^2\right] \\
&+ \frac{1}{\|\boldsymbol{v}\|^3\|\boldsymbol{W}\|}\left[\|\boldsymbol{v}\|^2\|\boldsymbol{W}^\top \boldsymbol{x}\|^2 - (\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x})^2\right] \\
\geq& 0
\end{aligned}$$

by Cauchy-Schwarz inequality. Moreover, the final equality holds only when $\boldsymbol{v} \parallel \boldsymbol{W}^\top \boldsymbol{x}$ and $\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x} = \|\boldsymbol{v}\|\|\boldsymbol{W}\|\|\boldsymbol{x}\|$, which implies $\boldsymbol{W} \parallel \boldsymbol{x}\boldsymbol{v}^T$. Hence, LaSalle principle yields the one-rank collapse of $\boldsymbol{W}$:

$$\left\|\frac{\boldsymbol{W}}{\|\boldsymbol{W}\|} - \frac{\boldsymbol{x}\boldsymbol{v}^\top}{\|\boldsymbol{x}\|\|\boldsymbol{v}\|}\right\| \to 0. \tag{36}$$

Now, it suffices to show that $\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}, \frac{\boldsymbol{W}}{\|\boldsymbol{W}\|}$ actually have limits. To do this, we consider the quantity

$$\boldsymbol{a}(t) = \int_0^t \boldsymbol{v}(s)ds. \tag{37}$$

Then we have

$$\boldsymbol{W}(t) = \boldsymbol{W}(0) + \boldsymbol{x}\boldsymbol{a}(t)^\top \to \dot{\boldsymbol{v}} = \boldsymbol{W}(0)^\top \boldsymbol{x} + \|\boldsymbol{x}\|^2 \boldsymbol{a}. \tag{38}$$

Since $\dot{\boldsymbol{a}} = \boldsymbol{v}$, we have the following ODE about $\boldsymbol{a}$:

$$\ddot{\boldsymbol{a}} - \|\boldsymbol{x}\|^2 \boldsymbol{a} = \boldsymbol{W}(0)^\top \boldsymbol{x}, \quad \boldsymbol{a}(0) = 0, \quad \dot{\boldsymbol{a}}(0) = \boldsymbol{v}(0). \tag{39}$$

The general solution is

$$\boldsymbol{a}(t) = -\frac{\boldsymbol{W}(0)^\top \boldsymbol{x}}{\|\boldsymbol{x}\|^2} + e^{\|\boldsymbol{x}\|t}\boldsymbol{u}_1 + e^{-\|\boldsymbol{x}\|t}\boldsymbol{u}_2 \tag{40}$$

where

$$\boldsymbol{u}_1 = \frac{1}{2}\left(\frac{\boldsymbol{W}(0)^\top \boldsymbol{x}}{\|\boldsymbol{x}\|^2} + \frac{\boldsymbol{v}(0)}{\|\boldsymbol{x}\|}\right), \quad \boldsymbol{u}_2 = \frac{1}{2}\left(\frac{\boldsymbol{W}(0)^\top \boldsymbol{x}}{\|\boldsymbol{x}\|^2} - \frac{\boldsymbol{v}(0)}{\|\boldsymbol{x}\|}\right) \tag{41}$$

and hence

$$\boldsymbol{v}(t) = \|\boldsymbol{x}\| e^{\|\boldsymbol{x}\|t} \boldsymbol{u}_1 - \|\boldsymbol{x}\| e^{-\|\boldsymbol{x}\|t} \boldsymbol{u}_2 \tag{42}$$

This implies that $\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}$ has a limit $\frac{\boldsymbol{u}_1}{\|\boldsymbol{u}_1\|}$, and the one-rank collapse also suggests that $\frac{\boldsymbol{W}}{\|\boldsymbol{W}\|}$ also has a limit.

Now, we finally prove Theorem 4. Notice that $\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sigma\left(\begin{bmatrix} \boldsymbol{W}_+^\top \boldsymbol{x}_i \\ \boldsymbol{W}_-^\top \boldsymbol{x}_i \end{bmatrix}\right)$, so

$$i \in \mathcal{I}_+ \to \boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \begin{bmatrix} \boldsymbol{W}_+^\top \boldsymbol{x}_i \\ \boldsymbol{0} \end{bmatrix}, \quad i \in \mathcal{I}_- \to \boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{W}_-^\top \boldsymbol{x}_i \end{bmatrix}$$

Hence, each feature mapping will eventually lie on $\operatorname{span}\left(\begin{bmatrix} \bar{\boldsymbol{v}}_+/\|\bar{\boldsymbol{v}}_+\| \\ \boldsymbol{0} \end{bmatrix}\right)$ or $\operatorname{span}\left(\begin{bmatrix} \boldsymbol{0} \\ \bar{\boldsymbol{v}}_-/\|\bar{\boldsymbol{v}}_-\| \end{bmatrix}\right)$, according to its label. This proves the directional collapse of features. Note that $s_\pm(t)$ is determined by Equation (42). It is obvious that two vectors above are orthogonal. Finally, from $\mathcal{L}(\boldsymbol{v}, \boldsymbol{W}) = -\boldsymbol{v}_+^\top \boldsymbol{W}_+^\top \boldsymbol{x}_+ + \boldsymbol{v}_-^\top \boldsymbol{W}_-^\top \boldsymbol{x}_-$ and Equation (42), we obtain $\boldsymbol{v}(t) = s_+(t)\boldsymbol{u}_{1,+}(t) - s_-(t)\boldsymbol{u}_{1,-}(t)$. ■

D.4. The Effect of Weight Normalization in Asymptotic Behavior

In this subsection, we show that the directional collapse described in Theorem 4 can also occur under weight normalization. Assume that we decompose each v_i and $\boldsymbol{w}_i$ by magnitude and direction as Salimans and Kingma [13] does, but fix their magnitudes. Then our loss function becomes

$$\mathcal{L}(\boldsymbol{v}, \boldsymbol{W}) = -\sum_{j=1}^{h} \frac{v_j}{|v_j|} \frac{\boldsymbol{w}_j^\top}{\|\boldsymbol{w}_j\|} \boldsymbol{x}_a(\boldsymbol{w}_j) \tag{43}$$

This function is 0-homogeneous, so $|v_j|$, $\|\boldsymbol{w}_j\|$ actually does not change by Euler's homogeneous function theorem. Moreover, by calculation,

$$\frac{d}{dt}\frac{\boldsymbol{w}_j}{\|\boldsymbol{w}_j\|} = \frac{1}{|\boldsymbol{v}_j|\|\boldsymbol{w}_j\|} \times \operatorname{sgn}(v_j)\left(\boldsymbol{I} - \frac{\boldsymbol{w}_j \boldsymbol{w}_j^\top}{\|\boldsymbol{w}_j\|^2}\right) \boldsymbol{x}_a(\boldsymbol{w}_j) \tag{44}$$

under a balanced initialization, which is a constant multiple of that of unnormalized GF. Hence, we can directly use the result of Appendix D.2 to prove neural alignment.

After a neural alignment, we consider the GF dynamics of

$$\hat{\mathcal{L}}(\boldsymbol{v}, \boldsymbol{W}) = -\frac{\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x}}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|}, \tag{45}$$

where the subscript $\pm$ is omitted. In this case, $\|\boldsymbol{v}\|, \|\boldsymbol{W}\|$ is constant again, and

$$\frac{d}{dt}\boldsymbol{v} = \frac{1}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|}\left(\boldsymbol{I} - \frac{\boldsymbol{v}\boldsymbol{v}^\top}{\|\boldsymbol{v}\|^2}\right)\boldsymbol{W}^\top \boldsymbol{x} = \frac{1}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|}\boldsymbol{W}^\top \boldsymbol{x} - \frac{\boldsymbol{v}^\top \boldsymbol{W}^\top \boldsymbol{x}}{\|\boldsymbol{v}\|^3\|\boldsymbol{W}\|}\boldsymbol{v}, \tag{46}$$

$$\frac{d}{dt}\boldsymbol{W} = \frac{1}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|}\boldsymbol{x}\boldsymbol{v}^\top - \frac{\boldsymbol{v}^\top\boldsymbol{W}^\top\boldsymbol{x}}{\|\boldsymbol{v}\|\|\boldsymbol{W}\|^3}\boldsymbol{W}. \tag{47}$$

Then by a similar way to Appendix D.3, one can prove that $c = \boldsymbol{v}^\top\boldsymbol{W}^\top\boldsymbol{x}$ is an increasing function and $dc/dt = 0$ only when $\boldsymbol{W} \parallel \boldsymbol{x}\boldsymbol{v}^\top$. Then LaSalle principle yields the one-rank collapse :

$$\frac{\boldsymbol{W}}{\|\boldsymbol{W}\|} \to \frac{\boldsymbol{x}\boldsymbol{v}^\top}{\|\boldsymbol{x}\|\|\boldsymbol{v}\|} \tag{48}$$

In this case, we don't have closed form solution of $\boldsymbol{v}$ and $\boldsymbol{W}$. However, since $\hat{\mathcal{L}}$ is semialgebraic in $\{\boldsymbol{v}, \boldsymbol{W} \mid \boldsymbol{v} \neq \boldsymbol{0}, \boldsymbol{W} \neq \boldsymbol{0}\}$, the trajectory is finite, and hence both $\boldsymbol{v}$ and $\boldsymbol{W}$ converge to a single point [1]. Since $\|\boldsymbol{v}\|, \|\boldsymbol{W}\|$ remains constant, there exist limits $\lim_{t\to\infty} s_\pm(t)$, i.e., the severe asymmetry described in Section 5.2 does not appear.

Appendix E. Additional Experiment Result for Unbounded-Below Loss Family

In this section, we provide supplementary results complementing those in Section 6. We compare the evolution of NC1 and NC2 for neural networks trained with linear loss, logistic loss, and the unbounded loss family defined in Equation (3). As discussed in Section 6, the unbounded loss family is more stable than the linear loss, even without weight normalization, while maintaining nearly identical performance (Figure 3).

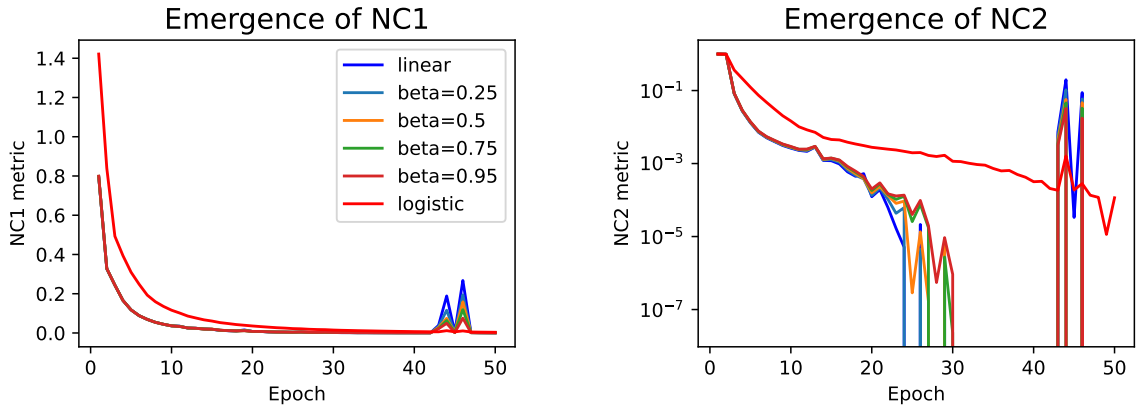

Figure 3: Emergence of NC1 and NC2 under different losses.