On the Robustness of Neural Collapse and the Neural Collapse of Robustness

Jingtong Su
Center for Data Science
New York University
js12196@nyu.edu

Ya Shi Zhang
Department of Pure Mathematics
and Mathematical Statistics
University of Cambridge
ysz23@cam.ac.uk

Nikolaos Tsilivis
Center for Data Science
New York University
nt2231@nyu.edu

Julia Kempe
Center for Data Science and
Courant Institute of Mathematical Sciences
New York University

Abstract

Neural Collapse refers to the curious phenomenon in the end of training of a neural network, where feature vectors and classification weights converge to a very simple geometrical arrangement (a simplex). While it has been observed empirically in various cases and has been theoretically motivated, its connection with crucial properties of neural networks, like their generalization and robustness, remains unclear. In this work, we study the stability properties of these simplices. We find that the simplex structure disappears under small adversarial attacks, and that perturbed examples "leap" between simplex vertices. We further analyze the geometry of networks that are optimized to be robust against adversarial perturbations of the input, and find that Neural Collapse is a pervasive phenomenon in these cases as well, with clean and perturbed representations forming aligned simplices, and giving rise to a robust simple nearest-neighbor classifier. By studying the propagation of the amount of collapse inside the network, we identify novel properties of both robust and non-robust machine learning models, and show that earlier, unlike later layers maintain reliable simplices on perturbed data.

1 Introduction

Reinforcing arguments about the simplicity of neural networks found by stochastic gradient descent in classification settings, Papyan et al. [2020] made the surprising empirical observation that both the feature representations in the penultimate layer (grouped by their corresponding class) and the weights of the final layer form a *simplex equiangular tight frame* (ETF) with C vertices, where C is the number of classes. Curiously, such a geometric arrangement becomes more pronounced well-beyond the point of (effectively) zero loss on the training data, motivating the common tendency of practitioners to optimize a network for as long as the computational budget allows. The collection of these empirical phenomena was termed *Neural Collapse*.

While the results of [Papyan et al., 2020] fueled much research in the field, many questions remain regarding the connection of Neural Collapse with properties like generalization and robustness of Neural Networks. In particular with regards to *adversarial robustness*, the ability of a model to withstand adversarial modifications of the input without effective drops in performance, it has been originally claimed that the instantiation of Neural Collapse has positive effect on defending against adversarial attacks [Papyan et al., 2020, Han et al., 2022]. However, this seems to at least superficially

Preprint.

contradict the fact that neural networks are not a priori adversarially robust [Szegedy et al., 2014, Carlini and Wagner, 2017].

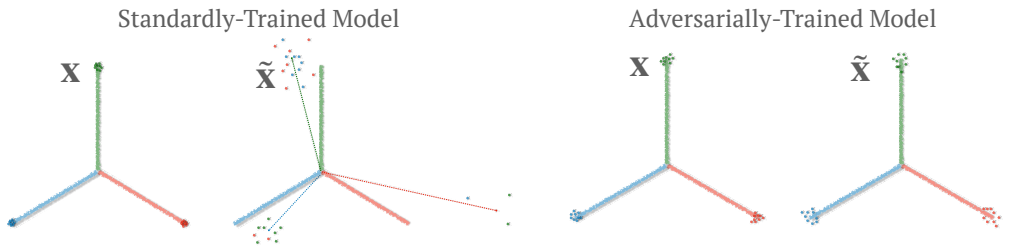

Figure 1: Visualisation of our findings. Sticks represent class-means. Small dots correspond to the representation of an individual datum, and the color represents the ground-truth label. **Left to Right:** clean representations with standardly-trained (ST) networks; perturbed representations with ST networks; clean representations with adversarially-trained (AT) networks; perturbed representations with AT networks. With ST nets, the adversarial perturbations push the representation to "leap" towards another cluster with slight angular deviation. AT makes the simplex resilient to such adversarial attacks, with higher and intra-class variance.

In this paper, we thoroughly study the stability properties of the simplices under adversarial attacks and then investigate whether Neural Collapse happens and whether it is necessary for adversarially robust models. In particular, our contributions and findings, partially illustrated in Figure 1, are:

- **Is NC robust?** We initiate the study of the neural collapse phenomenon in the context of adversarial robustness, both for standarly trained networks under adversarial attacks and for adversarially trained robust networks to investigate the stability and prevalence of the NC phenomenon. Our work exposes considerable additional fundamental, and we think, surprising, geometrical structure:
- **No!** For standardly trained networks we find that small, imperceptible adversarial perturbations of the training data remove any simplicial structure at the representation layer: neither variance collapse nor simplex representations appear under standard metrics. Further analysis through class-targeted attacks that preserve class-balance shows a "cluster-leaping" phenomenon: representations of adversarially perturbed data jump to the (angular) vicinity of the original class means.
- **Yes for AT networks! Two identical simplices emerge.** Adversarially trained, robust, networks exhibit a simplex structure both on original clean and adversarially perturbed data, albeit of higher variance. These two simplices turn out to be the same. We find that the simple nearest-neighbor classifiers extracted from such models also exhibit robustness.
- **Early layers are more robust.** Analyzing NC metrics in the representations of the inner layers, we observe that initial layers exhibit a higher degree of collapse on adversarial data. The resulting simplices, when used for Nearest Neighbour clustering, give surprisingly robust classifiers. This phenomenon disappears in later layers.

2 Background

Papyan et al. [2020] demonstrated the prevalence of NC on networks optimized by SGD, by tracing the following quantities (please refer to Appendix B.2 for exact definitions):

(NC1) Variability collapse: For all classes, the within-class variation of the last layer representations collapses to zero.

(NC2, Equiangular): Class-Means converge to equal, maximal pairwise angles.

(NC2, Equinorm): Class-Means converge to equal length.

(NC3) Convergence to self-duality: The linear classifier and the class-means converge to each other (after rescaling).

(NC4) Simplification to Nearest Class Center (NCC) classifier: The prediction of the network is equivalent to that of the NCC classifier formed by the (non-centered) class-means.

We adopt the natural extensions of the above metrics for adversarially trained models, and further study some new quantities, relevant to our analysis, which are defined and explained in Appendix B.2.

3 Experiments

In this section, we present our main experimental results measuring neural collapse in standardly (ST) (with SGD) and adversarially trained (AT) models [Madry et al., 2018]. We consider image classification tasks on CIFAR-10 and CIFAR-100 and we train two large convolutional networks, a standard VGG and a Pre-Activation ResNet18, from random initializations. We launch 3 independent runs and report the mean and standard deviation throughout our paper. Further results for varying choices of hyperparameters can be found in the Appendix.

Remark: When collecting feature representations for adversarially perturbed data, we always compute the *current epoch's* perturbations.

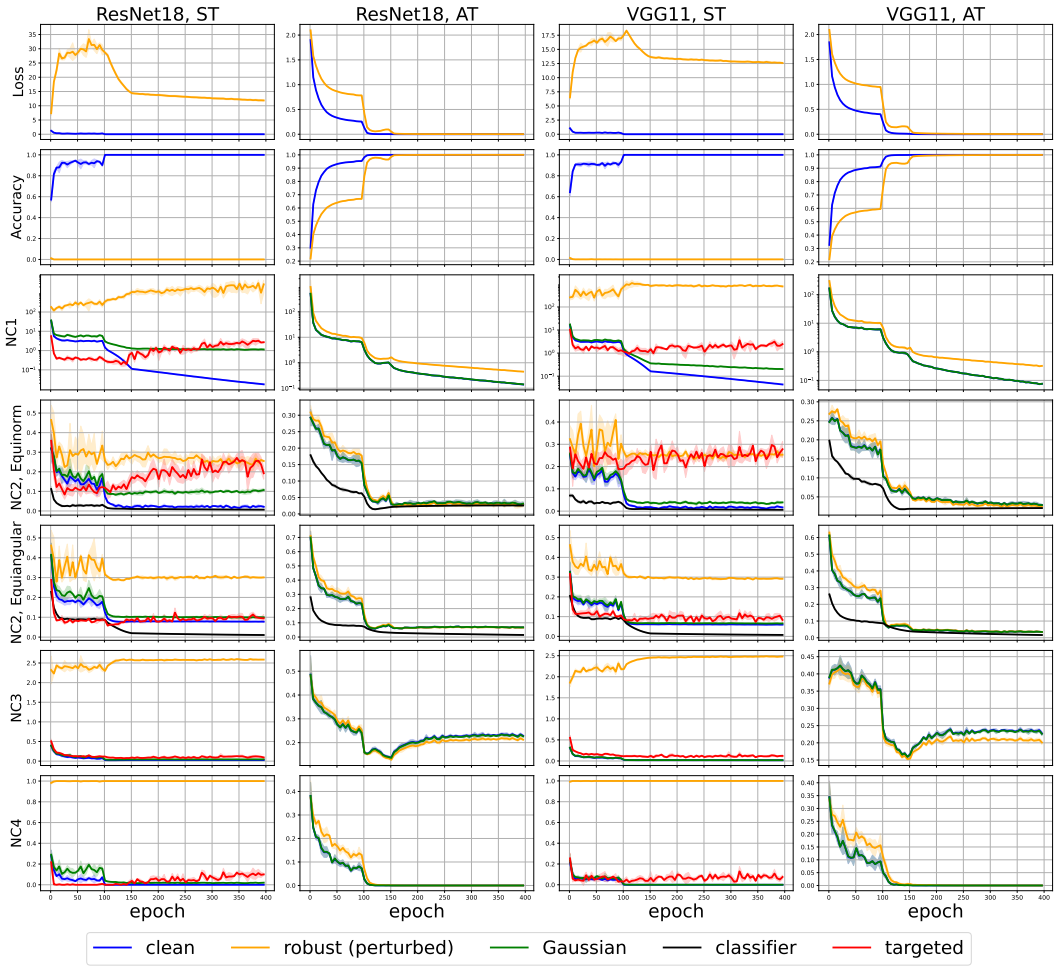

Figure 2: Accuracy, Loss, and NC evolution for standardly (ST) and adversarially (AT) trained VGG and ResNet. For AT models, clean and Guassian curves coincide. Setting: CIFAR-10, ℓ_∞ adversary.

3.1 Standardly trained neural nets

The first and third column of Figure 2 show the evolution of the NC quantities as described in Section 2 for standardly trained models. We use both adversarially perturbed and Gaussian reference data to study the stability of the original simplices. As expected, NC metrics converge on the clean training

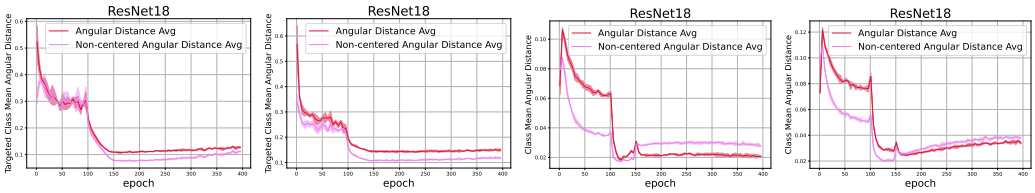

Figure 3: Angular distance. *Left and Inner Left:* Average between targeted attack class-means and clean class-means on **ST** network. *Inner Right and Right:* Average between perturbed class-means and clean class-means on **AT** network. Setting: CIFAR-10, ℓ_∞ adversary.

data. Neural Collapse is slightly attenuated on Gaussian reference data, but disappears strikingly for adversarially perturbed data, suggesting that the simplex formed by clean training data is robust against random perturbations, but fragile to adversarial attacks. The results certainly corroborate the conclusion that the representation class-means of perturbed points with ground-truth label c do not form any geometrically-meaningful structure at all.

Figure 3 (left) shows Simplex Similarity and non-centered Angular Distance of the simplices formed by targeted adversarial examples and by clean examples as described in Appendix B.2. These results give us a full glimpse of how standardly trained networks are non-robust and fail under adversarial attacks: adversarial perturbations break the simplex ETF by "leaping" the representation from one class-mean to another, forming a norm-imbalanced less concentrated structure around the original simplex.

3.2 Neural Collapse during Adversarial Training

We train neural nets adversarially to full convergence with perfect clean and robust training accuracy and measure NC metrics for clean and perturbed (epoch-wise) training data in Figure 2 (columns 2 and 4). Interestingly, we find that Neural Collapse *qualitatively* occurs in this setting as well, both for clean and perturbed data, and two simplices emerge. Notice, however, that the extent of variability collapse (NC1) on the perturbed points is smaller than on the "clean" data or the Gaussian noise benchmark, indicating that clean examples are more concentrated around the vertices. To understand the relative positioning of the two simplices, we investigate the Simplex Similarity and Angular Distance between non-centered class-means in Figure 3 (right). The vanishing distance suggests these two simplices are exactly the same. These results suggest that Adversarial Training nudges the network to learn simple representational structures (namely, a simplex ETF) not only on clean examples but also on perturbed examples to achieve robustness against adversarial perturbations. Equivalently, the simplices induced by robust networks are *not fragile* anymore, but *resilient*. Note also that NC4 results imply that there is a simple nearest-neighbor classifier that is robust against adversarial perturbations generated from the network.

Curiously, this is not the case for all training algorithms that produce robust models. In particular, a state-of-the-art algorithm that aims to balance clean and robust accuracy, TRADES [Zhang et al., 2019], shows fundamentally different behavior (see Figure 4 in the Appendix). Even though both terms of the loss (see Equation 4) are driven to zero, we do not observe Neural Collapse; the amount of collapse is roughly one order of magnitude larger than for vanilla AT, and the feature representations do not approach the ETF formation, even well past the onset of the terminal phase. *We view this as evidence that the prevalence of Neural Collapse is not necessary for robust classification.*

3.3 Layerwise Analysis

Furthermore, the Appendix contains our detailed layerwise analysis on both ST and AT models. We observe that initial layers exhibit a higher degree of collapse on adversarial data, while NCC classifiers defined on intermediate layers show surprising robustness, even if the whole model fails to do so.

4 Conclusion

Neural Collapse is an interesting phenomenon displayed by Neural Networks used in classification tasks. We empirically studied and quantified the sensitivity of this geometric arrangement to input perturbations, and, further, displayed that Neural Collapse can appear (but not always does!) in Neural Networks trained to be robust. We conclude that Neural Collapse is prevalent in many deep learning settings, including adversarially trained networks, though it does not seem to be necessary for robustness.

Acknowledgement

This work was supported by the National Science Foundation under NSF Award 1922658.

References

Ido Ben-Shaul and Shai Dekel. Nearest class-center simplification through intermediate layers. *CoRR*, abs/2201.08924, 2022.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society, 2017.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, 1995.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Weinan E and Stephan Wojtowytsch. On the emergence of simplex symmetry in the final and penultimate layers of neural network classifiers. In Joan Bruna, Jan Hesthaven, and Lenka Zdeborova, editors, *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, volume 145 of *Proceedings of Machine Learning Research*, pages 270–290. PMLR, 16–19 Aug 2022.

Cong Fang, Hangfeng He, Qi Long, and Weijie J. Su. Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proceedings of the National Academy of Sciences of the United States of America*, 118, 2021.

Tomer Galanti, András György, and Marcus Hutter. On the role of neural collapse in transfer learning. *CoRR*, abs/2112.15121, 2021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

X.Y. Han, Vardan Papyan, and David L. Donoho. Neural collapse under MSE loss: Proximity to and dynamics on the central path. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=w1UbdvWH_R3.

Hangfeng He and Weijie J. Su. A law of data separation in deep learning. 2022.

Like Hui, Mikhail Belkin, and Preetum Nakkiran. Limitations of neural collapse for understanding generalization in deep learning. *CoRR*, abs/2202.08384, 2022.

Wenlong Ji, Yiping Lu, Yiliang Zhang, Zhun Deng, and Weijie J Su. An unconstrained layer-peeled perspective on neural collapse. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WZ3yjh8coDg.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, 2017.

Xiao Li, Sheng Liu, Jinxin Zhou, Xinyu Lu, Carlos Fernandez-Granda, Zhihui Zhu, and Qing Qu. Principled and efficient transfer learning of deep models via neural collapse. *arXiv preprint arXiv:2212.12206*, 2022.

Yan Li, Ethan X. Fang, Huan Xu, and Tuo Zhao. Implicit bias of gradient descent based adversarial training on separable data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Bochen Lv and Zhanxing Zhu. Implicit bias of adversarial training for deep neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*, 2018.

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

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6614.

Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Ramesh Karri, Ozgur Sinanoglu, Ahmad-Reza Sadeghi, and Xun Yi, editors, *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 506–519. ACM, 2017.

Vardan Papyan, XY 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.

Akshay Rangamani, Marius Lindegaard, Tomer Galanti, and Tomaso Poggio. Feature learning in deep classifiers through intermediate neural collapse. Technical report, Center for Brains, Minds and Machines (CBMM), 2023.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8093–8104. PMLR, 2020.

Robert E. Schapire, Yoav Freund, Peter Barlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In Douglas H. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997*, pages 322–330. Morgan Kaufmann, 1997.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3353–3364, 2019.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.*, 19:70:1–70:57, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

Tom Tirer and Joan Bruna. Extended unconstrained features model for exploring deep neural collapse. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 21478–21505. PMLR, 17–23 Jul 2022.

Tom Tirer, Haoxiang Huang, and Jonathan Niles-Weed. Perturbation analysis of neural collapse. *arXiv preprint arXiv:2210.16658*, 2022.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *J. Mach. Learn. Res.*, 10:1485–1510, 2009.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019.

Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. On the optimization landscape of neural collapse under MSE loss: Global optimality with unconstrained features. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 27179–27202. PMLR, 17–23 Jul 2022.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

Appendix

A Relevant Work

Neural Collapse & Geometric properties of Optimization in Deep Learning. The term Neural Collapse was coined by Papyan et al. [2020] to describe phenomena about the feature representations of the last layer and the classification weights of a deep neural network at *convergence*. It collectively refers to the onset of variability collapse of within-class representations (NC1), the formation of two simplices (NC2) - one from the class-mean representations and another from the classification weights - that are actually dual (NC3), and, finally, the underlying simplicity of the prediction rule of the network, which becomes nothing but a simple nearest-neighbor classifier (NC4) (see Section B for formal definitions). [Papyan et al., 2020], using ideas from Information Theory, showed that the formation of a simplex is optimal in the presence of vanishing within-class variability. Mixon et al. [2022] introduced the *unconstrained features* model (independently proposed by [Fang et al., 2021] as the Layer-Peeled model), a model where the feature representations are considered as free optimization variables, and showed that a global optimizer of this problem (for the MSE loss) exhibits Neural Collapse. Many derivative works have proven Neural Collapse modifying this model, by either considering other loss functions or trying to incorporate more deep learning elements into it [Fang et al., 2021, Zhu et al., 2021, Ji et al., 2022, E and Wojtowytsch, 2022, Zhou et al., 2022, Tirer and Bruna, 2022, Han et al., 2022]. The notion of maximum separability dates back to Support Vector Machines [Cortes and Vapnik, 1995], while the bias of gradient-based optimization algorithms towards such solutions has been used to explain the success of boosting methods [Schapire et al., 1997], and, more recently, to motivate the generalization properties of neural networks [Neyshabur et al., 2015, Soudry et al., 2018, Lyu and Li, 2020]. The connection between Neural Collapse and generalization of neural networks (on in-distribution and transfer-learning tasks) has been explored in [Galanti et al., 2021, Hui et al., 2022]. Finally, the propagation of Neural Collapse inside the network has been studied by [Ben-Shaul and Dekel, 2022, He and Su, 2022, Hui et al., 2022, Li et al., 2022, Tirer et al., 2022, Rangamani et al., 2023].

Adversarial Examples & Robustness. Neural Networks are famously susceptible to adversarial perturbations of their inputs, even of very small magnitude [Szegedy et al., 2014]. Most of the attacks that drive the performance of networks to zero are gradient-based [Goodfellow et al., 2015, Carlini and Wagner, 2017]. These perturbations are surprisingly consistent between different architectures and hyperparameters, they are in many cases transferable between models [Papernot et al., 2017], and they can also be made universal (one perturbation for all inputs) [Moosavi-Dezfooli et al., 2017]. For training robust models, one can resort to algorithms from robust optimization [Xu et al., 2009, Goodfellow et al., 2015, Madry et al., 2018]. In particular, the most effective algorithm used in deep learning is called *Adversarial Training* [Madry et al., 2018]. During adversarial training one alternates steps of generating adversarial examples and training on this data instead of the original one. Several variations of this approach have been proposed in the literature (e.g. Zhang et al. [2019], Shafahi et al. [2019], Wong et al. [2020]), modifying either the attack used for data generation or the loss used to measure mistakes. However, models produced by this algorithm, despite being relatively robust, still fall behind in terms of absolute performance [Croce et al., 2021], while there are still many unresolved conceptual questions about adversarial training [Rice et al., 2020]. In terms of the geometrical properties of the solutions, [Li et al., Lv and Zhu, 2022] showed that in some cases (either in the presence of separable data or/and homogeneous networks) adversarial training converges to a solution that maximally separates the *adversarial* points.

B Methodology

In this section, we proceed with formal definitions of Neural Collapse (NC), adversarial attacks, and Adversarial Training (AT), together with the variants we study in this paper.

B.1 Notation

Let $\mathcal{X}$ be an input space, and $\mathcal{Y}$ be an output space, with $|\mathcal{Y}| = C$. Denote by $\mathcal{S}$ a given class-balanced dataset that consists of C classes and n data points per class. Let $f : \mathcal{X} \to \mathcal{Y}$ be a neural network, with its final linear layer denoted as $\mathbf{W}$. For each class c, the corresponding classifier is denoted as $\mathbf{w}_c$,

and the bias is called b_c. Denote the representation of the i-th sample within class c as $\mathbf{h}_{i,c} \in \mathbb{R}^p$, and the union of such representations $H(\mathcal{S})$. We define the global-mean vector $\boldsymbol{\mu}_G \in \mathbb{R}^p$, and class-mean vector $\boldsymbol{\mu}_c \in \mathbb{R}^p$ associated with $\mathcal{S}$ as $\boldsymbol{\mu}_G \triangleq \frac{1}{nC}\sum_{i,c}\mathbf{h}_{i,c}$ and $\boldsymbol{\mu}_c \triangleq \frac{1}{n}\sum_i \mathbf{h}_{i,c}, c = 1, \ldots, C$. For brevity, we refer in the text to the globally-centered class-means, $\{\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\}_{c=1}^C$, as just *class-means*, since these vectors are constituents of the simplex. We denote $\tilde{\boldsymbol{\mu}}_c = (\boldsymbol{\mu}_c - \boldsymbol{\mu}_G)/\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|_2$ the normalized class-means. Unless otherwise specified, the term "representation" refers to the penultimate layer of the network.

B.2 Neural Collapse Concepts

[Papyan et al., 2020] demonstrate the prevalence of NC on networks optimized by SGD, by tracing the following quantities[1]. Throughout our paper, we closely follow Papyan et al. [2020] and Han et al. [2022] on formalization of NC1-NC4. Before proceeding to the NC concepts, we introduce

Simplex ETF: A *standard simplex ETF* composed of C points is a set of points in $\mathbb{R}^C$, each point belonging to a column of

$$\sqrt{\frac{C}{C-1}}(\boldsymbol{I} - \frac{1}{C}\mathbf{1}_C\mathbf{1}_C^\top),$$

where $\boldsymbol{I} \in \mathbb{R}^{C\times C}$ is the identity matrix and $\mathbf{1}_C = [1 \quad \cdots \quad 1]^\top \in \mathbb{R}^C$ is the all-ones vector. In our discussion, a *simplex* can be thought of as a standard simplex ETF up to partial rotations, reflections, and rescaling.

Between-class and within-class covariance: Using terminology developed in Section B, we define between-class covariance $\Sigma_B \in \mathbb{R}^{p\times p}$ as

$$\Sigma_B \triangleq \mathrm{AVG}_c(\boldsymbol{\mu}_c - \boldsymbol{\mu}_G)(\boldsymbol{\mu}_c - \boldsymbol{\mu}_G)^\top$$

and $\Sigma_W \in \mathbb{R}^{p\times p}$ as

$$\Sigma_W \triangleq \mathrm{AVG}_{i,c}(\mathbf{h}_{i,c} - \boldsymbol{\mu}_c)(\mathbf{h}_{i,c} - \boldsymbol{\mu}_c)^\top.$$

Next, we start the introduction of NC concepts. We use the following exact definitions proposed by Han et al. [2022]:

(NC1) Variability Collapse:

$$\Sigma_B^\dagger \Sigma_W \to \mathbf{0},$$

where † denotes the Moore-Penrose inverse. The NC1 curve corresponds to $\mathrm{Tr}(\Sigma_\mathrm{B}^\dagger\Sigma_\mathrm{W})$.

(NC2) Convergence to Simplex ETF:

$$\langle \tilde{\boldsymbol{\mu}}_c, \tilde{\boldsymbol{\mu}}_{c'}\rangle \to -\frac{1}{C-1} \quad \forall\, c \neq c'$$

$$|\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|_2 - \|\boldsymbol{\mu}_{c'} - \boldsymbol{\mu}_G\|_2| \to 0 \quad \forall\, c, c'$$

The NC2 Equinorm curve corresponds to the variation of $||\boldsymbol{\mu}_c - \boldsymbol{\mu}_G||_2$ across all labels c, the standard deviation of these c quantities: $\mathrm{std}(||\boldsymbol{\mu}_c - \boldsymbol{\mu}_G||_2)$. The NC2 Equiangular curve corresponds to $\mathrm{AVG}_{c\neq c'}\ \mathrm{abs}(\langle \tilde{\boldsymbol{\mu}}_c, \tilde{\boldsymbol{\mu}}_{c'}\rangle + \frac{1}{C-1})$, where abs is the absolute value operator.

(NC3) Convergence to self-duality:

$$\frac{\mathbf{w}_c}{||\mathbf{w}_c||_2} - \frac{\boldsymbol{\mu}_c - \boldsymbol{\mu}_G}{||\boldsymbol{\mu}_c - \boldsymbol{\mu}_G||_2} \to 0 \quad \forall\, c.$$

The NC3 curve corresponds to

$$\sqrt{\sum_c ||\frac{\mathbf{w}_c}{||\mathbf{w}_c||_2} - \frac{\boldsymbol{\mu}_c - \boldsymbol{\mu}_G}{||\boldsymbol{\mu}_c - \boldsymbol{\mu}_G||_2}||_2^2}.$$

(NC4) Simplification to NCC classifier:

$$\arg\max_{c'} \langle \mathbf{w}_{c'}, \mathbf{h}\rangle + b_{c'} \to \arg\min_{c'} \|\mathbf{h} - \boldsymbol{\mu}_{c'}\|_2 \quad \forall\, \mathbf{h} \in H(\mathcal{S}).$$

The NC4 curve corresponds to the mismatch ratio of these two quantities.

[1] In particular, Neural Collapse becomes more evident in the so-called *Terminal Phase of Training*, the phase beyond the point of (effectively) zero training loss.

In our experiments, we calculate the NC statistics with the code provided by Han et al. [2022][2].

Furthermore, in this work, we compare representations of original and perturbed data, which imposes ambiguity on which class-mean vectors $\boldsymbol{\mu}_c, \boldsymbol{\mu}_G$ to use (from $\mathcal{S}$ or $\mathcal{S}'$). In the spirit of the original definitions, for NC1-4 we will use the class means induced by the dataset $\mathcal{S}'$, even if different from the training set. NC4 studies the predictive power of the NCC classifier on $\mathcal{S}'$ by comparing it to the network classification output, which at TPT is equivalent to the ground truth label. For study of reference data $\mathcal{S}'$ outside the TPT, we introduce two quantities, also applicable to any intermediate layer:

NCC-Network Matching Rate: measures the rate at which the NCC classifier defined in NC4 trained on $\mathcal{S}$ coincides with the output of the network on dataset $\mathcal{S}'$. Note that we use $\boldsymbol{\mu}_c$ calculated by $\mathcal{S}$.

$$\arg\max_{c'} \langle \mathbf{w}_{c'}, \mathbf{h} \rangle + b_{c'} \stackrel{?}{=} \arg\min_{c'} \|\mathbf{h} - \boldsymbol{\mu}_{c'}\|_2, h \in H(\mathcal{S}').$$

NCC Accuracy: measures the accuracy on dataset $\mathcal{S}'$ of the NCC classifier defined in NC4 trained on $\mathcal{S}$. Note that we use $\boldsymbol{\mu}_c$ calculated by $\mathcal{S}$. c_h denotes the ground-truth label of the input.

$$c_h \stackrel{?}{=} \arg\min_{c'} \|\mathbf{h} - \boldsymbol{\mu}_{c'}\|_2, h \in H(\mathcal{S}').$$

Note that when $\mathcal{S} = \mathcal{S}'$, both NCC-Network Matching Rate and NCC Accuracy stem from (NC4). We also introduce the following measures to quantify the proximity of two simplices over C-classes:

Simplex Similarity: We define the similarity measure between two C-class simplices with normalized class means $\tilde{\boldsymbol{\mu}}_c, \tilde{\boldsymbol{\mu}}_c^{'}$ as

$$\mathrm{AVG}_c \arccos \langle \tilde{\boldsymbol{\mu}}_c, \tilde{\boldsymbol{\mu}}_c' \rangle.$$

Non-centered Angular Distance: Similarly, given two simplices, without taking the global mean $\boldsymbol{\mu}_G$ and $\boldsymbol{\mu}_G'$ into account, we can calculate the angular distance with non-centered class-means directly:

$$\mathrm{AVG}_c \arccos \langle \frac{\boldsymbol{\mu}_c}{||\boldsymbol{\mu}_c||_2}, \frac{\boldsymbol{\mu}_c'}{||\boldsymbol{\mu}_c'||_2} \rangle.$$

Note that the similarity and angular distance between a simplex and itself is zero.

B.3 Gradient-Based Adversarial Attack, Adversarial Training (AT), and TRADES

Given a deep neural network f with parameters θ, a clean example $(\mathbf{x}, y)$ and cross-entropy loss $\mathcal{L}(\cdot, \cdot)$, the *untargeted* adversarial perturbation is crafted by running multiple steps of projected gradient descent (PGD) to maximize the CE loss [Kurakin et al., 2017, Madry et al., 2018] (in what follows, we focus on ℓ_∞ adversary with ℓ_2 deferred to the appendix):

$$\mathbf{x}^{k+1} = \Pi_{\mathcal{B}^\epsilon_{\mathbf{x}^0}} \left(\mathbf{x}^k + \alpha \cdot \mathrm{sign}(\nabla_{\mathbf{x}^k} \mathcal{L}(f(\mathbf{x}^k), y) \right), \tag{1}$$

where $\mathbf{x}^0 = \mathbf{x}$ is the original example, α is the step size, $\tilde{\mathbf{x}} = \mathbf{x}^N$ is the final adversarial example, and Π is the projection on the valid ϵ-constraint set, $\mathcal{B}^\epsilon_{\mathbf{x}}$, of the data. $\mathcal{B}^\epsilon_{\mathbf{x}}$ is usually taken as either an ℓ_∞ or ℓ_2 ball centered in $\mathbf{x}^0$. Further, to control the predicted label of $\tilde{\mathbf{x}}$, a variant called *targeted attack* minimizes the CE loss w.r.t. a target label $y_t \neq y$:

$$\mathbf{x}^{k+1} = \Pi_{\mathcal{B}^\epsilon_{\mathbf{x}^0}} \left(\mathbf{x}^k - \alpha \cdot \mathrm{sign}(\nabla_{\mathbf{x}^k} \mathcal{L}(f(\mathbf{x}^k), y_t) \right). \tag{2}$$

With a standardly-trained network, both these methods can effectively reduce the accuracy to 0%. To combat this phenomenon, robust optimization algorithms have been proposed. The most representative methodology, *adversarial training* [Madry et al., 2018], generates $\tilde{\mathbf{x}}$ *on-the-fly* with Equation (1) for each epoch from $\mathbf{x}$, and takes the model-gradient update on $\tilde{\mathbf{x}}$ only.

An alternative robust training variant, TRADES [Zhang et al., 2019], is of particular interest as it aims to address both robustness and clean accuracy. Thus the gradient steps of TRADES directly involve both $\mathbf{x}$ and $\bar{\mathbf{x}}$, where $\bar{\mathbf{x}}$ is also obtained by PGD, but under the KL-divergence loss:

$$\mathbf{x}^{k+1} = \Pi_{\mathcal{B}^\epsilon_{\mathbf{x}^0}} \left(\mathbf{x}^k + \alpha \cdot \mathrm{sign}(\nabla_{\mathbf{x}^k} \mathcal{L}_{KL}(f(\mathbf{x}), f(\mathbf{x}^k))) \right). \tag{3}$$

[2] https://colab.research.google.com/github/neuralcollapse/neuralcollapse/blob/main/neuralcollapse.ipynb

The total TRADES loss is a summation of the CE loss on the clean data and a KL-divergence (KLD) loss between the predicted probability of $\mathbf{x}$ and $\bar{\mathbf{x}}$ with a regularization constant β:

$$\mathcal{L}_{CE}(f(\mathbf{x}), y) + \beta \cdot \mathcal{L}_{KL}(f(\mathbf{x}), f(\bar{\mathbf{x}})). \tag{4}$$

C Experimental Details

Code. For ℓ_∞ and ℓ_2 PGD attacks with ST and AT, we used the code from Rice et al. [2020] [3]. For TRADES, we adopted the original implementation[4]. We have attached the code for reproducing NC results with ST, AT, and TRADES within a zip file.

Plotting. Throughout our paper, we plot all quantities per 5 epochs in all figures.

Layerwise NC. We study the layerwise NC1, NC2 and NC4 quantities for both PreActResNet18 (ResNet18) and VGG11. With ResNet18, which consists of one convolutional layer, four residual blocks, and the final linear layer, we use the features after every block for the first five blocks (one convolutional layer and four residual blocks) as representations. With VGG, which consists of eight convolutional blocks (convolutional layer + batch-normalization + max-pooling) and the final linear layer, we use the features after each convolutional block as representations. We apply average-pooling subsampling on representations that are too large for feasible computation of NC1's pseudo-inverse.

D TRADES Results on CIFAR-10 with ℓ_∞ adversary

For CIFAR-10's results with TRADES, we have produced Figure 4, which depicts the evolution of loss, accuracy and all of the NC metrics under the standard ℓ_∞ adversary. Note that we plot the KLD-loss here to showcase optimization convergence, to avoid the effect of the regularization constant β.

E Complementary Results on CIFAR-10, ℓ_2 adversary.

Here we complement our main text with robust network experiments on CIFAR-10 for ℓ_2 adversarial perturbations.

Figure 5 illustrates NC results of Adversarial Training and TRADES training with the ℓ_2 adversary. All plots are consistent with our findings in the main text: Adversarial Training alters Neural Collapse such that the clean representation simplex overlaps with the perturbed representation simplex, whereas TRADES does not lead to any simplex ETF.

F Complementary Results on CIFAR-100

In this section, we reproduce our experiments on CIFAR-100. We illustrate results with (ℓ_∞, ℓ_2) adversaries and obtain the same conclusions as those on CIFAR-10. This suggests the universality of the intrinsic adversarial perturbation dynamics that we have detailed in the main text.

F.1 CIFAR-100 ℓ_∞ Standard and Adversarial Training Results

All results are summarized within Figure 6. Similar to the main text, we plot the untargeted attack illustration in Figure 7. Notably, on CIFAR-100 with ST, adversarial perturbations also push the representation to leap toward the predicted class's simplex cluster with very small angular deviation.

F.2 CIFAR-100 ℓ_∞ TRADES Results

For CIFAR-100 ℓ_∞ trained with TRADES, Figure 8 depicts the results, and we observe that no simplex exists, consistent with previous results.

[3] https://github.com/locuslab/robust_overfitting

[4] https://github.com/yaodongyu/TRADES

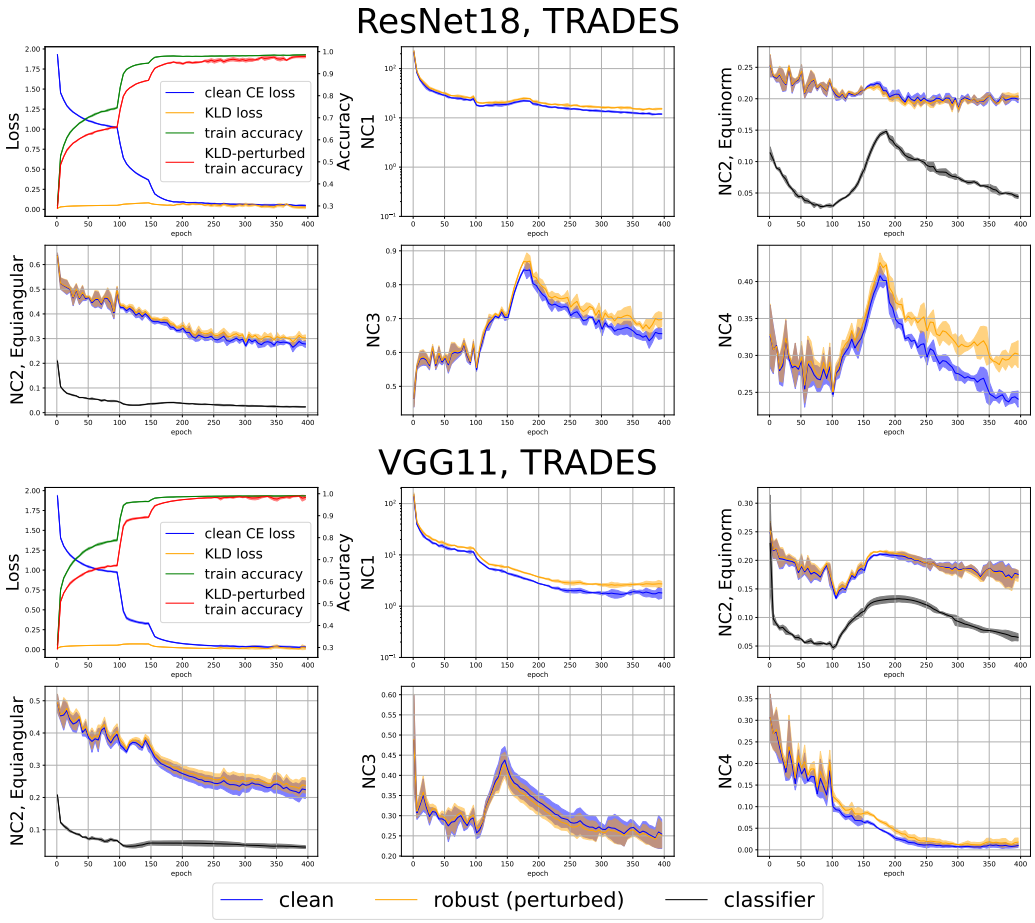

Figure 4: Accuracy, Loss and NC evolution with TRADES trained networks. *Upper:* ResNet18; *Lower:* VGG11. No simplices are formed with TRADES training. Setting: CIFAR-10, ℓ_∞ adversary.

F.3 CIFAR-100 ℓ_2 AT and TRADES Results

These results are shown in Figure 9. All observations are consistent with previous results.

F.4 CIFAR-100 Simplex Similarity Results

The Simplex Similarity and non-centered Angular Distance of the simplices formed by targeted adversarial and clean examples with ST, and the simplices generated by clean and perturbed examples with AT, are depicted in Figure 10. The result is the same as the one for CIFAR-10 in the main text, Figure 3.

G Layerwise Results

While originally variability collapse and simplex formation were observed for the last layer representations, follow-up studies extended the analysis to the intermediate layers of the neural network. In particular, He and Su [2022] found that the amount of variability collapse measured at different layers (at convergence) decreases smoothly as a function of the index of the layer. Further, Hui et al. [2022] coined the term Cascading Neural Collapse to describe the phenomenon of cascading variability collapse; starting from the end of the network, the collapse of one layer seemed to be signaling the collapse of the previous layers (albeit to a lesser extent). Here, we replicate this study of the intermediate layer computations, while also studying the representations of the perturbed points (both in standard and adversarial training). In particular, we collect the input of either convolutional

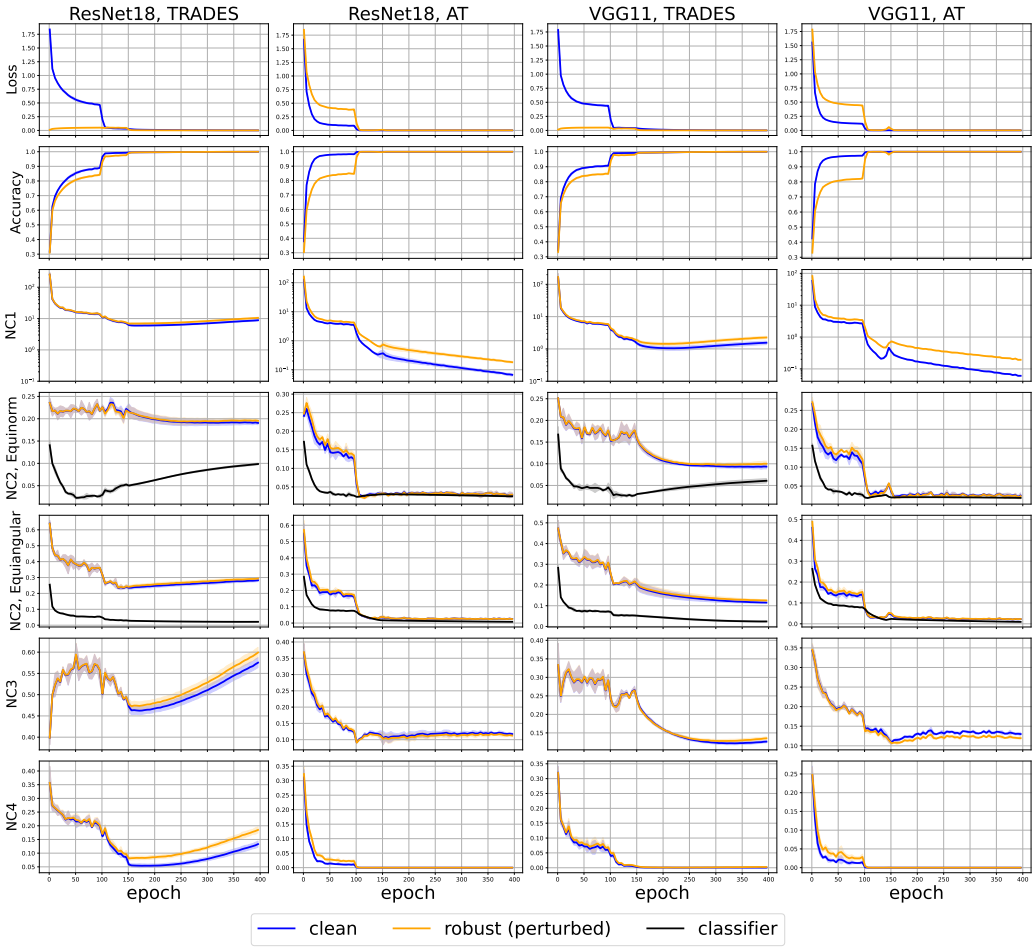

Figure 5: Accuracy, Loss and NC evolution with adversarially trained and TRADES trained networks. Setting: CIFAR-10, ℓ_2 adversary.

or linear layers of the network *at convergence*, order them by depth index, and compute the NC quantities of Section B. The results are presented in Figure 11.

Both for ST and AT models, we reproduce the power law behavior observed in [He and Su, 2022] for clean data; the feature variability collapses progressively, and, interestingly, undergoes a slower decrease in the case of adversarial training. The adversarial data representations for ST models, however, while failing to collapse at the final layer (as already established in Figure 2), exhibit the same amount of "clustering" as those of the original data for the earlier layers. This hints that from the viewpoint of the earlier layers, clean and adversarial data are indistinguishable. And, this, is indeed the case! Looking at the first and third column of Figure 12, we observe that the simple classifier formed by the centers of the early layers is quite robust ($\sim 40\%$) to these adversarial examples (both train and test). Curiously, this robustness is higher than the one of the simple classifiers defined by layers of an adversarially trained model (although the two numbers are not directly comparable). This is, undeniably, a peculiar phenomenon of standardly trained models that is worth more exploration; could it be that the lesser variability exhibited in the earlier layers is actually beneficial for robustness or is it just the stability of the feature space that makes prediction more robust?

In Figure 13 and Figure 14, we perform the same computations on CIFAR-100. We arrive at the same conclusions.

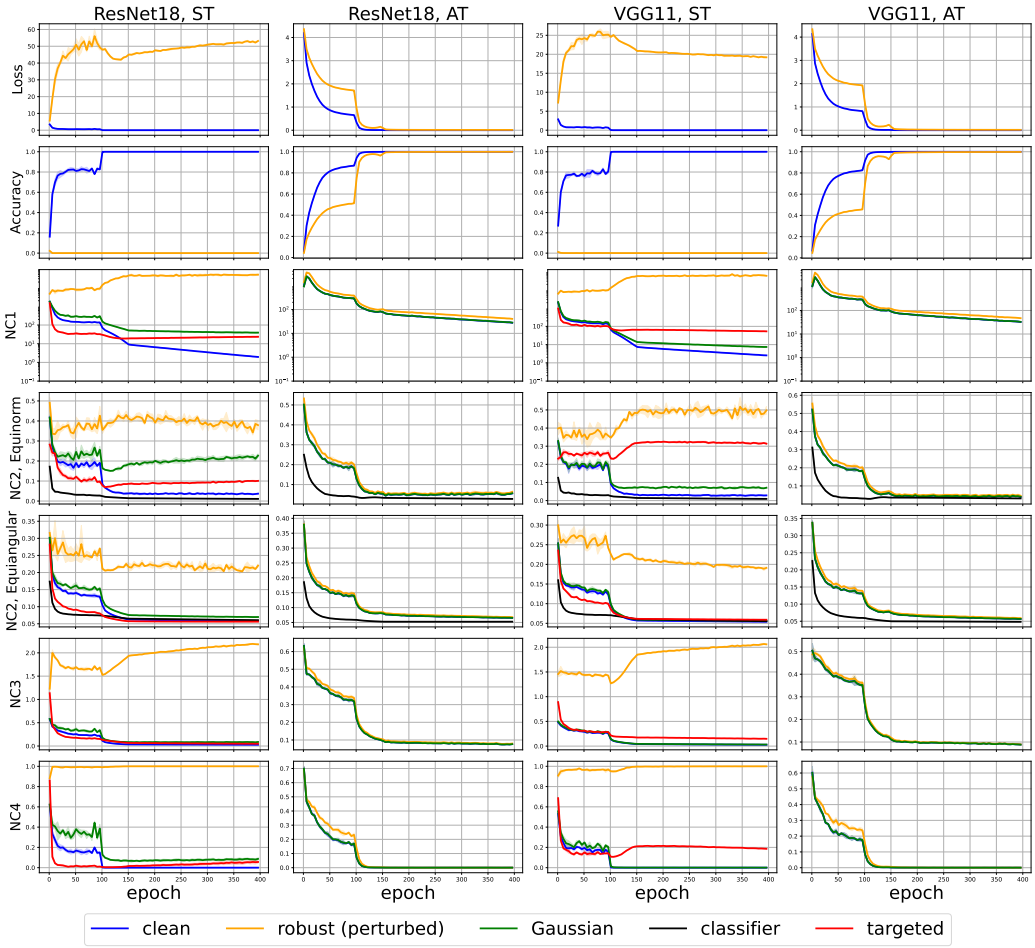

Figure 6: Accuracy, Loss and NC evolution with standardly trained networks. Setting: CIFAR-100, ℓ_∞ adversary.

H Small Epsilon Results

Here, we illustrate how AT indeed progressively induces more robust NC metrics and simplex ETFs with respect to the perturbation radius ϵ. Figure 15 shows the NC metrics over $8/255-$perturbed data. Conversely, using an ST model, the NC metrics when evaluating on $(2/255, 4/255, 8/255)-$perturbed data also increases monotonically with adversarial strength. This is illustrated in Figure 16.[5]

[5]For small radius AT and small radius adversarial attack for ST, we scale the PGD step size α linearly with ϵ to ensure PGD to work properly.

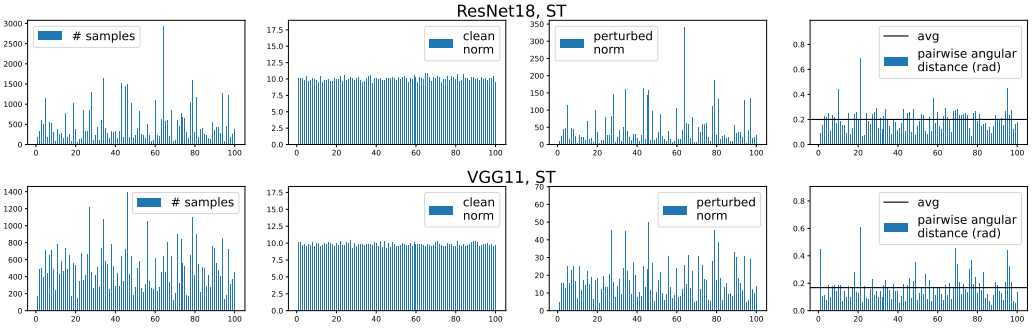

Figure 7: Illustration of untargeted adversarial attacks on standardly trained, converged, models that correspond to one random seed. (CIFAR-100, ℓ_∞). *Left:* Number of examples with a certain predicted label. *Inner Left:* The norms of clean class-means. *Inner Right:* The norms of predicted class-means with perturbed data. *Right:* Angular distance between clean and predicted class-mean with perturbed data. *Upper:* ResNet18; *Lower:* VGG11. For 100 classes, the between-class angular distance is $\arccos\left(-\frac{1}{99}\right) = 1.58$ rad $= 90.58$ degrees, while 0.2 rad is only 11.4 degrees.

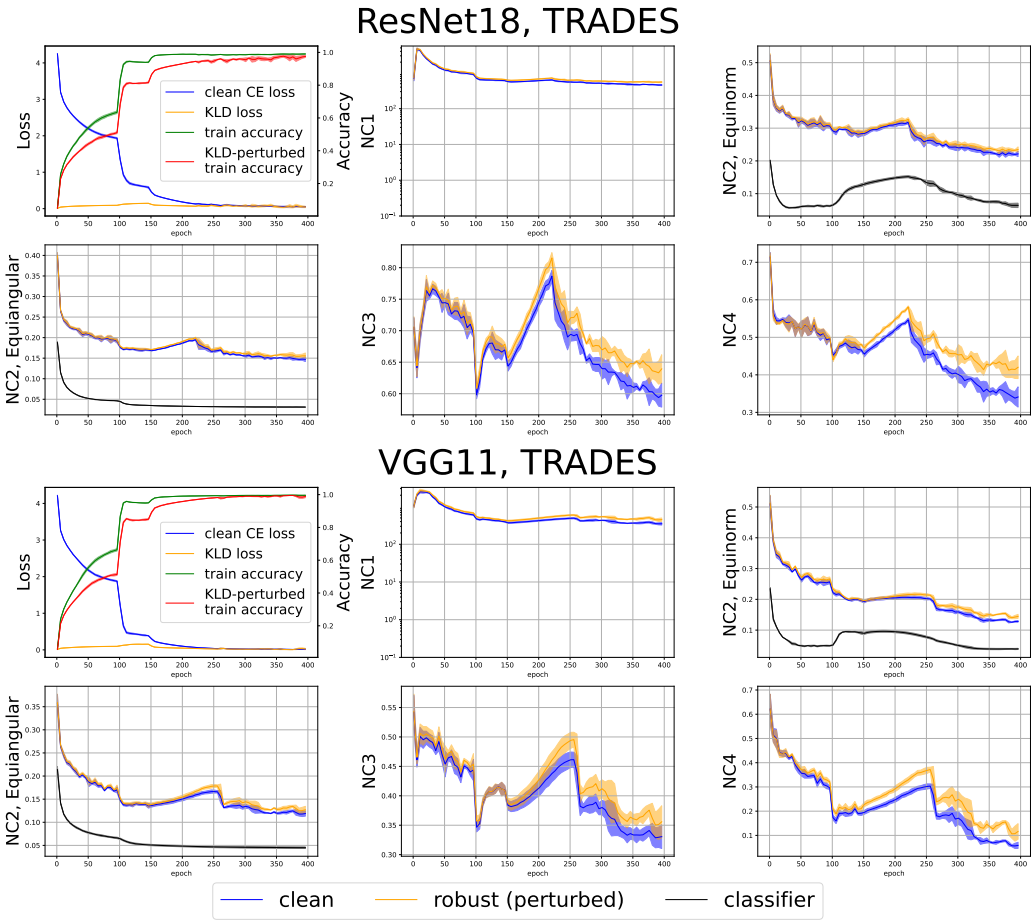

Figure 8: Accuracy, Loss and NC evolution with TRADES trained networks. *Upper:* ResNet18; *Lower:* VGG11. Results indicate AT boosts Neural Collapse so that it also happens on adversarially-perturbed data. Setting: CIFAR-100, ℓ_∞ adversary.

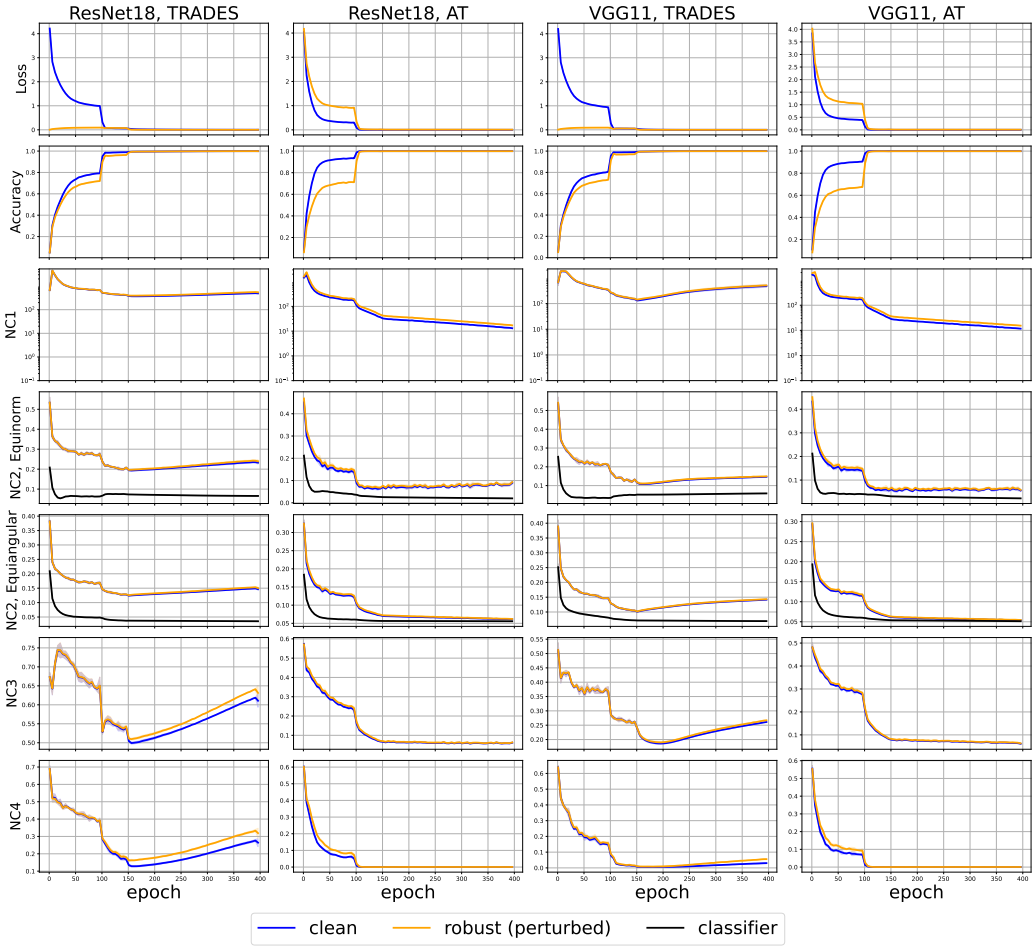

Figure 9: Accuracy, Loss and NC evolution with ℓ_2 robust models on CIFAR-100. Setting: CIFAR-100, ℓ_2 adversary.

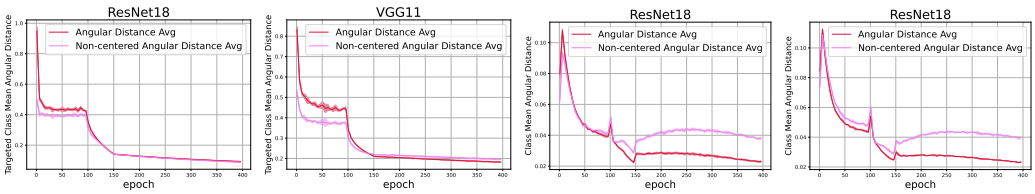

Figure 10: Angular distance. *Left and Inner Left:* Average between targeted attack class-means and clean class-means on **ST** network. *Inner Right and Right:* Average between perturbed class-means and clean class-means on **AT** network. Setting: CIFAR-100, ℓ_∞ adversary.

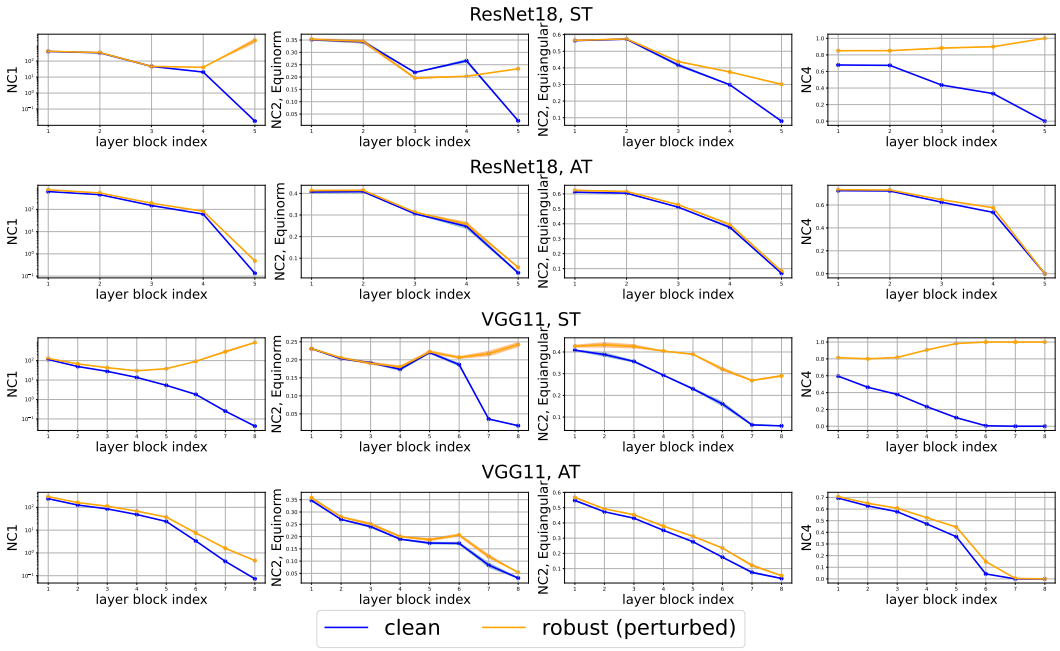

Figure 11: Layerwise evolution of NC1, NC2 and NC4 for ST and AT networks. NC metrics for perturbed data tend to undergo some amount of clustering in the earlier layers. For AT, collapse undergoes a slower decrease through layers than for ST. Setting: CIFAR-10, ℓ_∞ adversary.

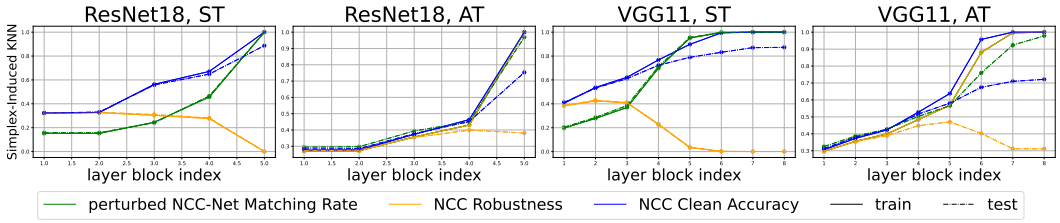

Figure 12: Layerwise NCC classifier. We measure the performance of the NCC classifier obtained from (training) class means on both train and test data. NCC Robustness refers to NCC Accuracy on perturbed data. Note that, on training data, the NCC Robustness and the perturbed NCC-Net Matching Rate curves overlap. Early layers give a surprisingly robust NCC classifier (NCC Robustness) for both train and test data. Setting: CIFAR-10, ℓ_∞ adversary.

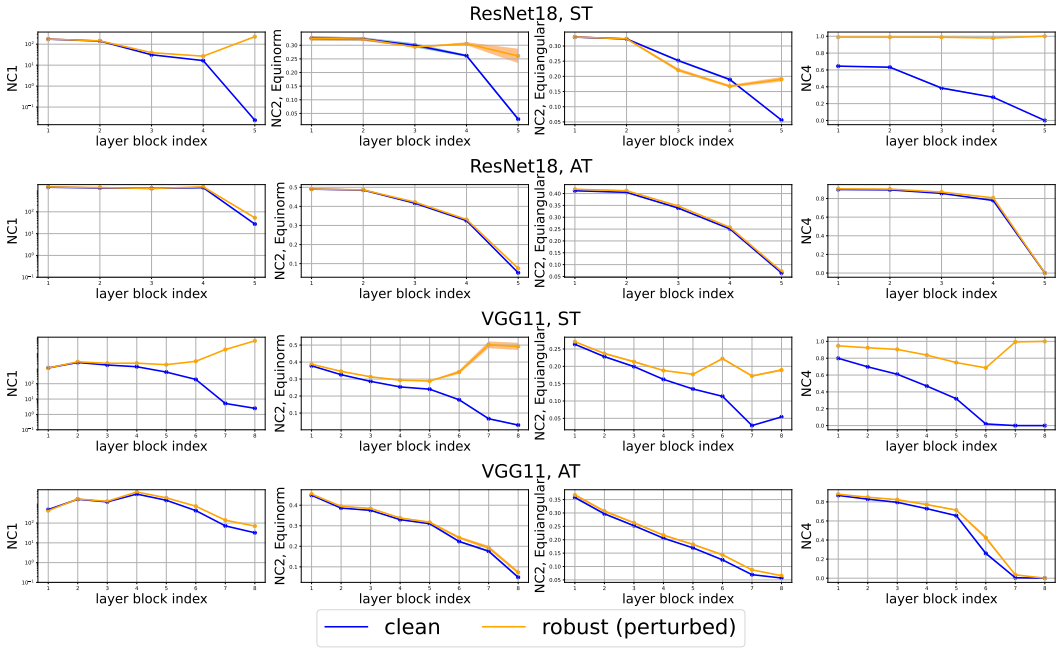

Figure 13: Layerwise evolution of NC1, NC2 and NC4 for ST and AT networks. NC metrics for perturbed data tend to undergo some amount of clustering in the earlier layers. For AT, collapse undergoes a slower decrease through layers than for ST. Setting: CIFAR-100, ℓ_∞ adversary.

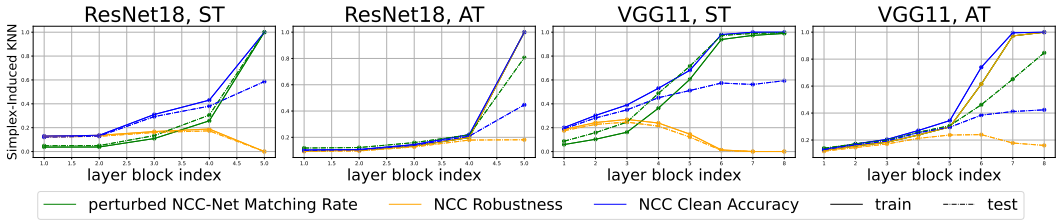

Figure 14: Layerwise NCC classifier. We measure the performance of the NCC classifier obtained from (training) class means on both train and test data. NCC Robustness refers to NCC Accuracy on perturbed data. Note that, on training data, the NCC Robustness and the perturbed NCC-Net Matching Rate curves overlap. Early layers give a surprisingly robust NCC classifier (NCC Robustness) for both train and test data. Setting: CIFAR-100, ℓ_∞ adversary.

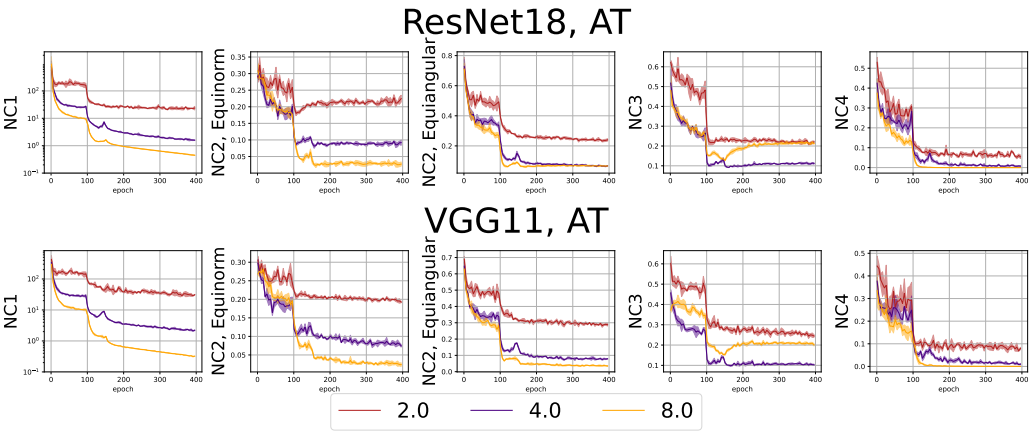

Figure 15: Progressive Loss and NC evolution, AT with varying strength. The color indicates the epsilon used for **training**. Setting: CIFAR-10, ℓ_∞ adversary.

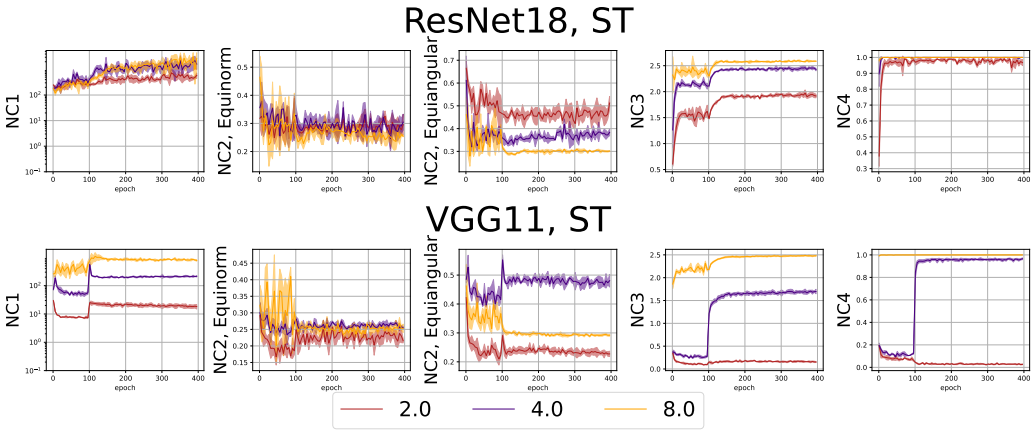

Figure 16: Progressive Loss and NC evolution, ST with varying attacking strength. The color indicates the epsilon used for **evaluation**. Setting: CIFAR-10, ℓ_∞ adversary.