The Impact of Geometric Complexity on Neural Collapse in Transfer Learning

Michael Munn*
Google Research
munn@google.com

Benoit Dherin*
Google Research
dherin@google.com

Javier Gonzalvo
Google Research
xavigonzalvo@google.com

Abstract

Many of the recent remarkable advances in computer vision and language models can be attributed to the success of transfer learning via the pre-training of large foundation models. However, a theoretical framework which explains this empirical success is incomplete and remains an active area of research. Flatness of the loss surface and neural collapse have recently emerged as useful pre-training metrics which shed light on the implicit biases underlying pre-training. In this paper, we explore the geometric complexity of a model's learned representations as a fundamental mechanism that relates these two concepts. We show through experiments and theory that mechanisms which affect the geometric complexity of the pre-trained network also influence the neural collapse. Furthermore, we show how this effect of the geometric complexity generalizes to the neural collapse of new classes as well, thus encouraging better performance on downstream tasks, particularly in the few-shot setting.

1 Introduction

Many of the recent remarkable advances in modern machine learning owe their success in part to transfer learning [4, 6, 9, 15, 28, 49, 50, 66]. While there are different approaches to this technique, the standard one involves two stages. In a first stage, called pre-training, ones trains a deep neural network on a general, large-scale dataset in the form of a supervised or unsupervised source task; e.g., ImageNet or CIFAR-100 [14, 33] for image models or the Common Crawl, C4 or LM1B datasets [8, 11, 49] for language models. In the second stage, one then leverages portions of the pre-trained network to use as features map or embeddings that can then be adapted, or fine-tuned, on a more specific target task. Often these target tasks are unknown at the time of pre-training and labeled data may be very scarce, such as in the context of few-shot learning [34, 51, 60]. However, despite these limitations, this approach often results in a fine-tuned model that achieves quite impressive performance substantially better than training on the target task alone and requiring less computational resources [20]. Despite this empirical success, a comprehensive theoretical understanding of the mechanisms underlying this effectiveness of transfer learning is not fully understood and remains an active area of research [63].

One interesting line of research suggests that the effectiveness of transfer learning is due to the implicit biases encoded in pre-trained models [22, 36, 48]. These implicit biases effectively constrain the hypothesis space, guiding the model towards solutions that have a preference for smoother functions [12, 17, 38, 40, 53], simpler geometry in the loss surface [3, 13, 18, 55, 58] or reduced complexity of the internal learned representations [23, 32]. In the same way that these implicit biases have been used to help explain the success of deep learning, recent work has shown that the notion of neural collapse [22, 35, 62] and flatness of the loss surface [36] can also inform the mechanisms behind

*Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2131

transfer learning. This suggests that these preferences during pre-training also act as a form of prior knowledge which is highly transferable to downstream tasks, even with limited task-specific data.

In this paper, we present a novel perspective that further sheds light on these mechanisms and implicit biases hidden within transfer learning. Our approach analyzes the geometric complexity [16, 17] of the internal representations learned by the deep neural networks during pre-training and provides a complementary theoretical framework which unifies previous work examining the role of neural collapse and loss surface flatness in transfer learning [22, 36]. In particular, we show through experiments and theory that the geometric complexity of the pre-trained network directly controls the neural collapse of the pre-trained model and thus its efficacy in transfer learning, particularly in the few-shot setting. We argue that geometric complexity (similarly to flatness of the loss surface and neural collapse) can be used as hidden progress metrics, cf. [2], for transfer learning, serving as an informative proxy toward the transfer power of a pre-trained network.

Our primary contributions are the following:

- We uncover relationships between learning-path flatness, neural network geometric-complexity, and embedding neural-collapse providing a framework to understand how these implicit biases interact (Section 4).
- We show through theory that the geometric complexity (GC) of a neural network controls neural collapse and verify this empirically by showing how mechanisms which regularize the GC in turn put pressure on the neural collapse (Section 4.1 and Fig. 1).
- We demonstrate both theoretically and empirically that pre-trained networks with lower GC promote lower neural collapse on new unseen target classes, and thus enable improved transfer accuracy during fine-tuning (Section 5 and Fig. 4).
- We prove a new generalization bound in terms of geometric complexity (Section 4.3).
- We show that the empirical GC can be accurately and efficiently estimated on a small number of samples, input coordinates, and output features making it computationally tractable compared to other progress metrics in machine learning (Section 4.2 and Fig. 2).

Notation. Throughout, we denote by $\|\cdot\|$ the L2 Euclidean norm and by $\|\cdot\|_F$ the Frobenius norm.

2 Background and Related Work

The implicit biases introduced by an optimization algorithm play a crucial role in deep learning and in the generalization ability of the learned models [43, 44, 55, 64]. They help ensure that the model not only finds a solution with low error but also one with low complexity which generalizes well [26, 67]. Uncovering the mechanisms of these implicit regularizers is crucial for understanding how the model learns, both as a means to improve generalization and as valuable leverage for designing more efficient algorithms and decreasing costly experiment iteration cycles.

Here we focus on three seemingly different themes behind implicit regularization in deep learning and explain how they are related: the loss surface slope measuring the flatness of the learning path [3, 29, 45], the geometric complexity of the learned model function measuring its variability with respect to a dataset [16, 42], and the neural collapse [32, 46] measuring how a neural network efficiently clusters its learned representations of the input class examples in embedding space.

Flatness in parameter space. In parameter space, the learning dynamics is fully characterized by the discrete optimization path θ_t where $t \in \mathbb{N}$. At each step, one can compute the flatness of the learning path as the slope of the tangent space at θ_t to the loss surface $\mathcal{L} = \{(\theta, L(\theta)) : \theta \in \mathbb{R}^n\}$. As shown in [3, Appendix A.2], this slope coincides with the loss-gradient square norm: $\text{slope}(\theta_t) = \|\nabla L(\theta_t)\|^2$. This quantity has been used to bound a generalization gap in [25] and as a beneficial explicit regularizer in [3, 24, 58] indicating that learning curves with lower slope values tend to also have better test performance.

Moreover, many standard optimizers and common training heuristics have been shown to put an implicit pressure on the learning-path flatness, making it an implicit bias of the training procedure [3, 5, 10, 13, 17, 25, 39, 58]. Related to the learning curve slope is the optima sharpness

$$\text{sharpness}(\theta_*) = \frac{1}{n} \operatorname{trace} H(\theta_*)$$

where θ_* is a global minima toward which the learning dynamics converges to $\theta_t \to \theta_*$ and H is the Hessian of the loss. Since the sharpness of an optima corresponds to the mean curvature of the loss surface at that point, flat minima (i.e., minima with low curvature in all directions) can be reached only through learning paths with shallower slopes.

Neural collapse in embedding space. Neural collapse [22, 27, 46] refers to a phenomenon observed in deep learning where the embedding network (i.e., the subnetwork f before the logit layer g in a neural network $h(x) = \mathrm{softmax}(g(f(x)))$ collapses the input points around their respective class means. Furthermore, these class means form a simplex creating an equiangular tight frame (ETF) centered around the global mean with roughly equal distance between its vertices. Intuitively, such a phenomenon is beneficial to generalization since the embeddings of different classes are optimally separated, making the job of the classifier head $\mathrm{softmax}(g(z))$ easier. To measure neural collapse, the authors in [22] introduce the *class-distance normalized variance* (CDNV) as

$$V_f(Q_i, Q_j) := \frac{\mathrm{Var}_f(Q_i) + \mathrm{Var}_f(Q_j)}{2\|\mu_f(Q_i) - \mu_f(Q_j)\|^2}, \tag{1}$$

where $Q_r = q_r(x)dx$ is the input distribution for classes $r \in \{i, j\}$ and where

$$\mu_f(Q_r) = \mathbb{E}_{x\sim Q_r}[f(x)] \quad \text{and} \quad \mathrm{Var}_f(Q_r) = \int \|\mu_f(Q_r) - f(x)\|^2 q_r(x)dx. \tag{2}$$

Neural collapse between two classes happens when their class variance decreases while the distance between their class means increases, causing lower values of their CDNV. Following [22], given a well-balanced input distribution $Q = \frac{1}{k}(Q_1 + \cdots + Q_k)$ with k classes (Q_i being the input distribution for class i), *neural collapse* (NC) during training is characterized by the following limit as the number of training steps t goes to infinity:

$$\lim_{t\to\infty} \mathrm{NC}(f_t, Q) = 0 \quad \text{where} \quad \mathrm{NC}(f, Q) := \mathrm{Avg}_{i\neq j}\left(V_f(Q_i, Q_j)\right), \tag{3}$$

and where f_t is the learned embedding network at step t. Thus, more neural collapse (i.e., more clustering around better separated class-means) happens with lower NC values. Proposition 5 from [22] proves that $\mathrm{NC}(f, Q)$ bounds a generalization gap and lower values of $\mathrm{NC}(Q, f)$ are correlated with better test performance of the neural network f, and [37] shows that explicitly regularizing for neural collapse is beneficial.

Geometric complexity in function space: The *geometric complexity* (GC) of a function $f : \mathbb{R}^d \to \mathbb{R}^k$ w.r.t. a data distribution $Q = q(x)dx$ on $\mathbb{R}^d$ is defined as the expectation of the gradient Frobenius square-norm w.r.t. to the data distribution [17] given by

$$\mathrm{GC}(f, Q) := \mathbb{E}_{x\sim Q}\left[\|\nabla_x f(x)\|_F^2\right].$$

Intuitively, the GC measures the function complexity or variability and is closely related to the Dirichlet energy in geometric analysis [21]. Provided mild Lipschitz smoothness assumptions [42, Proposition 3.4], the GC can be estimated accurately on batches of training data D by its empirical counterpart

$$\widehat{\mathrm{GC}}(f, D) = \frac{1}{|D|}\sum_{x\in D} \|\nabla_x f(x)\|_F^2, \quad \text{where } D \text{ is an i.i.d. sample drawn from } Q. \tag{4}$$

Previous work has explored the relationship between the GC and model generalization. When measured on the full neural network, lower GC values correlate experimentally with higher test accuracy [45] and it has also been used as a beneficial explicit regularizer [30, 59]. In [16, 17], the authors ignore the softmax activation and study the GC measured with respect to the logit network, exploring its connection with various implicit and explicit regularizers and training heuristics. The logit GC has also be used to prove a margin based multi-class generalization bound [42], assuming that the input distribution Q satisfies a mild assumption known as the Poincaré inequality [1].

In fact, both this work and the proof of the generalization bound rely on the Poincaré inequality, which we argue is indeed a natural and mild assumption for the types of data distributions typically encountered in machine learning; see Appendix A.6 for further discussion. For completeness, we recall it here, cf. [19]. A distribution Q satisfies a Poincaré inequality if, for all differentiable functions v defined on $\mathrm{supp}(Q)$, there exists a constant c such that

$$\mathrm{Var}_v(Q) \leq c\mathbb{E}_{x\sim Q}\left[\|\nabla_x v\|_F^2\right] = c\,\mathrm{GC}(v, Q). \tag{5}$$

https://doi.org/10.52202/079017-2131

Note that the same assumption is used in [7] to prove the law of robustness for overparameterized neural networks via isoperimetric inequalities. In this paper, we peel back yet another layer of the network and consider the embedding geometric complexity measured on a feature map (Section 3).

Relationship between flatness, neural collapse, and geometric complexity. The learning-path flatness as measured by its slope influences the geometric complexity of the learned solutions [16, Theorem 5.1]. Namely, for dense layers a regularizing pressure on the slope of the learning path in parameter space transfers to a regularizing pressure on the geometric complexity of the learned solution in function space. A similar result also been shown for special attention layers as well [52].

In this paper, we will see that imposing a regularization pressure on the embedding geometric complexity encourages more neural collapse of the model. This results in the following chain of influences from regularization pressure:

learning path flatness	$\rightsquigarrow$	function geometric complexity	$\rightsquigarrow$	embedding neural collapse
Regularizing Pressure		***Regularizing Pressure***		***Lower CDNV***

3 Problem Formulation

We are interested in understanding the relationship between the geometric complexity (GC) and neural collapse (NC) in the transfer learning setting and explore this relationship in two stages. In the first stage, we examine the general impact of the GC on NC during model training/pre-training. Secondly, we examine how this relationship provides insight into the mechanisms behind transfer learning and the advantageous implicit biases of the pre-training stage. In short, lower GC during pre-training on source classes leads to lower NC for target classes and improved transfer accuracy.

For the first stage of our inquiry (Section 4), we are concerned with a k-classification task. Let $D := \{(x_i, y_i)\}_{i=1}^m$ be a dataset drawn from a distribution Q with $x_i \in \mathbb{R}^d$ and $y_i \in \{e_j \mid j = 1, \ldots, k\}$ where $e_j \in \mathbb{R}^k$ are the canonical basis vectors representing a one-hot encoding of the labels. We aim to learn this task using a neural network denoted by a function $h_\theta : \mathbb{R}^d \to \mathbb{R}^k$ parameterized by θ (though to simplify notation we subsequently drop this dependence on θ).

We can write $h_\theta(x) = \mathrm{softmax}(g(f(x))$ where $g : \mathbb{R}^p \to \mathbb{R}^k$ is a classifier head mapping from the feature space $\mathbb{R}^p$ to the prediction layer $\mathbb{R}^k$ and $f : \mathbb{R}^d \to \mathbb{R}^p$ is a feature mapping from the input space to the penultimate embedding layer of the network before the classification head. The standard way of training h_θ is to find, via some stochastic optimization technique, an optimal parameter configuration θ_* such that $\theta_* = \mathrm{argmin}_\theta \sum_{i=1}^m \ell(h_\theta(x_i), y_i)$ for a given loss function $\ell : \mathbb{R}^k \times \mathbb{R}^k \to [0, \infty)$.

To analyze the role of neural collapse and geometric complexity in this setting, we focus our attention on the feature map $f : \mathbb{R}^d \to \mathbb{R}^p$. The NC as described in Section 2 is defined using this sub-network map f of h and thus we also measure the geometric complexity with respect to this sub-network feature mapping as well. Throughout this paper, we use the term *embedding GC* to refer to the GC of such a feature map f and unless otherwise specified, when we refer to the model GC we mean the embedding GC, not the GC of the logit network as in [16, 17].

Our second stage of inquiry (Section 5) is focused on the role of geometric complexity in transfer learning. For this setting, we assume there is some l-class classification target task we would like to solve and corresponding target distribution $\mathcal{T}$. We aim to learn a classifier $h' : \mathbb{R}^d \to \mathbb{R}^l$ given some training data $D' := \{(x'_i, y'_i)\}_{i=1}^{m'}$ which is potentially very limited in size. In order to find a good solution, we can leverage a pre-trained feature mapping that has been trained on some other source task and source distribution $\mathcal{S}$ where more data is available. For example, by leveraging a feature map f pre-trained as in the above setup, the target task classifier h' can instead be trained on the outputs of our feature map; i.e., $\{f(x'_i), y'_i)\}_{i=1}^{m'}$. In this way, we can write $h' = \mathrm{softmax}(g' \circ f)$ where the parameters of f are fixed and we only need to learn the parameters of $g' : \mathbb{R}^p \to \mathbb{R}^l$. Ideally, provided f is a rich enough feature map trained on a general enough source distribution dataset, then the task for learning g' is much simpler.

4 Geometric complexity and neural collapse

In this section, we explore the general relationship between the embedding GC and neural collapse showing that the geometric complexity can be used to bound neural collapse. Next, we see how the geometric complexity can be efficiently and accurately estimated, both as an approximation of the theoretical GC as well as through a number of sampling techniques for the empirical GC via batch sampling, feature sampling and label sampling. Lastly, following [22], we derive a new generalization bound for neural networks expressed in terms of the embedding geometric complexity; cf. [42].

4.1 Geometric complexity controls neural collapse

Previously, it has been shown [16] that the GC can be controlled implicitly through choice of learning rate and batch size, as well as through standard explicit regularization. Practically, this means that these same beneficial tuning strategies which ensure that models have low GC should also work to keep the NC low as well. The following proposition (which we prove in Appendix A.1) bounds the neural collapse by the geometric complexity provided that the input distribution satisfies a Poincaré inequality also assumed in [42] and [7].

Proposition 4.1. *Suppose that we have a balanced multi-class input-distribution Q with k classes satisfying the Poincaré inequality in* (5) *for some constant c, then the geometric complexity of a network embedding f bounds its neural collapse as measured by* (3)*; namely, we have the following bound*

$$\mathrm{NC}(f, Q) \leq \frac{c \cdot \mathrm{GC}(f, Q)}{k-1} \left(\sum_{i \neq j} \frac{1}{d_{ij}^2} \right), \tag{6}$$

where k is the number of classes, and d_{ij} is the distance between the mean of class i and class j.

We call the RHS of the bound in Eq. (6), excepting the Poincaré constant c, the **geometric collapse**:

$$\frac{\mathrm{GC}(f, Q)}{k-1} \left(\sum_{i \neq j} \frac{1}{d_{ij}^2} \right), \tag{7}$$

where k is the number of classes, and $d_{ij} = \|\mu_f(Q_i) - \mu_f(Q_j)\|$ denotes the Euclidean distance between the mean $\mu_f(Q_i)$ and $\mu_f(Q_j)$ of class i and class j (resp.) in embedding space.

This quantity can be seen as another proxy metric measuring neural collapse. It is made of two main parts decoupling the variances and mean differences present in the neural collapse framework. The variances are consolidated into a single factor through the GC, while the mean differences are averaged across classes in a separate factor. This separation allows the overall intra-class variability to be influenced through the GC term, ensuring sufficient between-class separation through the mechanisms identified in neural collapse.

This bound provides a powerful method to control the overall within-class variability via the L2-norm of the model gradient. As a result, the geometric collapse provides a simplified and more refined approach to managing class separability and variance in deep learning models.

We verify the relationship posed in Eq. (6) through experiments on VGG-13 trained on CIFAR-10 (see Figure 1). Through both implicit and explicit regularizers the pressure on the embedding GC translates to a direct pressure on the neural collapse of the model via the geometric collapse. As already observed in [17], lower levels of GC coincide with higher test accuracy as show in Figure 5 in Appendix A.5 containing the learning curves of that experiment.

In Appendix A.5 we see the same relationship holds across various datasets; e.g., MNIST, Fashion-MNIST, CIFAR-100, and architectures; e.g., ResNet-18, ResNet-50. To avoid possible confounding factors that could arise through batch size and learning rate manipulations, we also directly regularize with the geometric complexity in Appendix A.5.6; the same relations hold in this case too.

4.2 The geometric complexity is a reliable and robust measure

One of the key advantages of the GC as a complexity measure is its computational efficiency.

 https://doi.org/10.52202/079017-2131

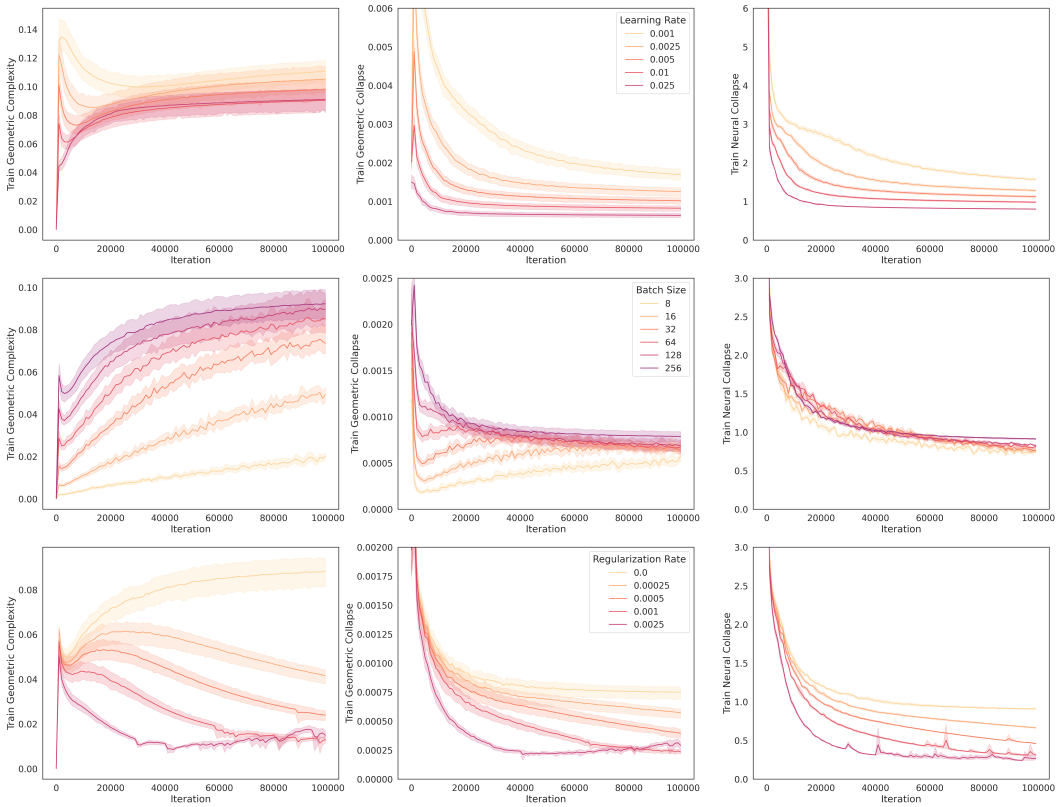

Figure 1: Controlling the neural collapse via the model geometric complexity for VGG-13 trained on CIFAR-10. Lower embedding GC produces lower geometric collapse (Eq. 7) and more neural collapse (i.e., lower NC) for **Top row:** increased learning rates, **Middle row:** decreased batch sizes, and **Bottom row:** increased L2 regularization.

Under mild regularity constraints on the model function the theoretical geometric complexity of a map can be efficiently estimated by its empirical counterpart, as stated in the proposition below, already proven in [42] for logit networks (see Appendix A.2 for a proof). Additionally and crucially, we verify that the empirical GC is a robust and reliable measure with respect to that data sample.

Proposition 4.2. *Let $f : \mathbb{R}^d \to \mathbb{R}^p$ be a map with Lipschitz constant L and let D_X be a sample of m elements drawn independently from an input distribution Q. Then, for any $\delta > 0$, we have with probability $1 - \delta/2$ the following bound*

$$\mathrm{GC}(f, Q) \leq \widehat{\mathrm{GC}}(f, D_X) + L\sqrt{\frac{\log \frac{2}{\delta}}{2m}}. \tag{8}$$

For a function f and data sample D_X as in Proposition 4.2, to measure $\widehat{\mathrm{GC}}(f, D_X)$ requires computing the Frobenius norm of a potentially very large Jacobian matrix, particularly when p represents the embedding dimension of a feature map. Note that,

$$\widehat{\mathrm{GC}}(f, D_X) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{p}\sum_{s=1}^{d}\left(\frac{\partial f^j(x^{(i)})}{\partial x_s}\right)^2. \tag{9}$$

Although the computational complexity of the GC is impervious to increasing complexity of the network architecture, e.g., in terms of parameter count or number of layers, one may run into computation bottlenecks as the sample size m increases or when increasing the dimensionality d (resp. p) of the inputs (resp. outputs). In these scenarios, it is necessary to find efficient means to accurately approximate or sample the Jacobian; cf. [61].

We explore the robustness of the GC when measured via samplings along these three axes; i.e., decreasing the number of examples m in the batch, randomly sampling the full Jacobian matrix which has order $d \times p$, and randomly sampling the number of model outputs p. As shown Figure 2, we find that the value of the sampled $\widehat{\mathrm{GC}}$ remains stable and consistent to its true value through these fairly simple and naive sampling tricks.

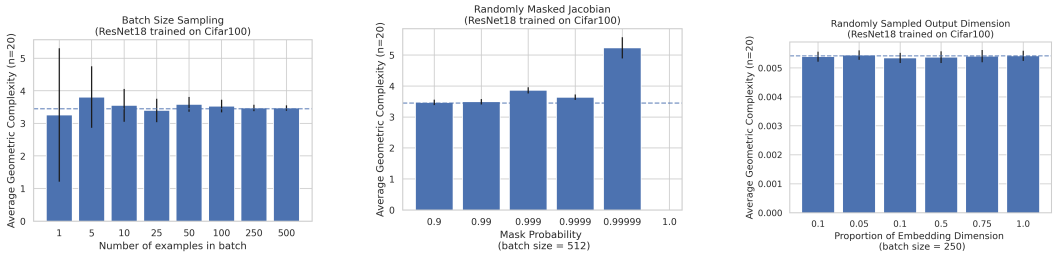

Figure 2: The GC computation is robust and consistent to sampling via **Left:** number of examples in the batch, **Middle:** number of elements in the Jacobian, or **Right:** by sampling the embedding dimension of the model. Here the GC and subnet GC have been computed over 20 trials, plotting the mean and standard deviations for a ResNet-18 model that has been trained to convergence on CIFAR-100. The true value of the GC for each setting is indicated by dotted line.

4.3 A new generalization bound with GC through NC

In [42], the authors derive a margin-based multi-class generalization bound for neural networks which depends on the geometric complexity $\mathrm{GC}(h, Q)$ of the full logit network. In this section, we extend this result. With inspiration from [22, Proposition 5], we show that the geometric complexity measured on the sub-network feature map $\mathrm{GC}(f, Q)$ bounds the classification error of the nearest-mean classifier defined by the feature map f.

As in [22], given a balanced k-class classification problem, let $S = \bigcup_{c\in[k]} S_c$ denote all class samples and define the nearest-mean classifier by $h_{f,S}(x) := \operatorname{argmin}_{c\in[k]} \|f(x) - \mu_f(S_c)\|$ given by the feature map f where $\mu_f(S_c)$ denotes the sample mean of the set S_c under the map f. Define the generalization error

$$\text{Error} := \mathbb{E}_{(x,y)\sim P}\left[\mathbb{I}[h_{f,S}(x) \neq y]\right]$$

where $\mathbb{I}[h_{f,S}(x) \neq y]$ is the indicator function of the error set $[h_{f,S}(x) \neq y]$ and P is the full data distribution from which samples S are drawn.

Proposition 4.3. *Suppose that we have a balanced sample S from a k-class input-distribution $(x, y) \sim P$ with m_c samples per class. Assume further that the induced input distribution $x \sim Q$ satisfies a Poincaré inequality as in* (5) *for some constant c. Then, for any $\delta > 0$, with probability $1 - \delta/2$ we have the following bound for the generalization error*

$$\mathbb{E}\left[\text{Error}\right] \leq 16c\left(\frac{1}{p} + \frac{1}{m_c}\right)\left(\widehat{\mathrm{GC}}(f, S) + L\sqrt{\frac{\log\frac{2}{\delta}}{2m_c k}}\right)\left(\sum_{i\neq j}\frac{1}{d_{ij}^2}\right), \tag{10}$$

where p is the embedding dimension of the feature map f and $d_{ij} = \|\mu_f(S_i) - \mu_f(S_j)\|$. Note, the outer expectation on the left hand side is taken over the samples S used to produce the classifier $h_{f,S}$.

Proof. This follows immediately from [22, Proposition 5] which gives the bound in terms of the neural collapse instead of the geometric complexity. By using Proposition A.1 we can replace the neural collapse with the geometric complexity and using Proposition 4.2 we then replace the theoretical GC with the empirical GC, yielding the additional term with the Poincaré constant. □

In Figure 3, we plot the LHS and RHS of the generalization bound in (10) from Proposition 4.3 for a VGG-13 model trained on CIFAR-10. In this plot, we see that the bound is not vacuous, demonstrating a relatively tight fit. Note that on the RHS, we omitted the term involving the Lipschitz constant L, assuming it is small due to its denominator. Additionally, we estimated a lower bound for the Poincaré constant c by comparing the magnitudes of the RHS and LHS in the inequality (6), based on results in Figure 1. This approximation yielded $c \approx 1000$ in our setup.

https://doi.org/10.52202/079017-2131

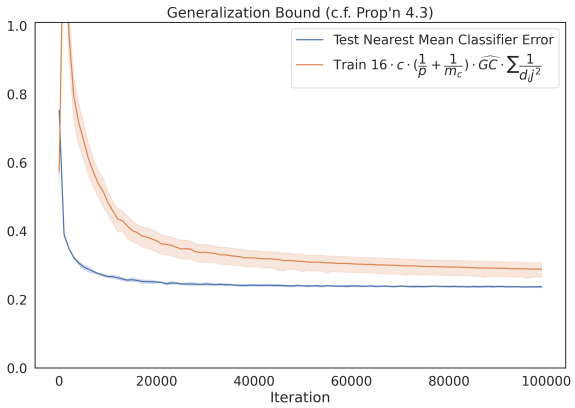

Figure 3: VGG-13 trained on CIFAR-10 with 5 random seeds.

5 The impact of geometric complexity on transfer learning

Many implicit regularization mechanisms in gradient descent exert a pressure on geometric complexity which in turn constrain the neural collapse. In this section, we discuss the affect of this implicit bias in transfer learning. Specifically, we show with theory how this bias during pre-training can help explain the mechanisms behind transfer learning. Namely, that lower GC in a pre-trained embedding network promotes neural collapse on new target classes, simplifying the fine-tuning process.

However, our bound makes it clear that certain compatibility conditions between the pre-training (source) distribution and the fine-tuning (target) distribution are needed, which we argue are satisfied in image foundational models and large language models. At last, we verify empirically that known methods to reduce geometric complexity for the pre-trained embedding result in better performance during fine-tuning on unseen tasks, in agreement with our neural collapse bound for transfer learning.

5.1 Lower pre-trained GC leads to improved fine-tuning

To analyze the impact of GC in the context of transfer learning, we consider the same formal setup as in [22, Proposition 2]. Assume a class c in a set $\mathcal{C}$ of classes is represented by a class-conditional distribution $Q_c(x) = P(x|y = c)$. Both the pre-training/source classes $\widetilde{Q}_1, \ldots, \widetilde{Q}_k$ and the fine-tuning/target classes $Q_1, \ldots, Q_l$ are assumed to be drawn from a distribution over all classes in $\mathcal{C}$. The induced input distribution on all the classes (i.e., the combined source and target input distribution) is denoted Q, while the source-only input distribution is denoted by $\widetilde{Q}$. Let $\mathcal{F}^*$ denote the set of pre-trained feature maps $f : \mathbb{R}^d \to \mathbb{R}^p$ selected by the training procedure (e.g., the set of trained embeddings with different initialization seeds but the same training protocol) and consider

$$\Delta(\mathcal{F}^*) = \inf_{f \in \mathcal{F}^*} \inf_{c \neq c'} \|\mu_f(Q_c) - \mu_f(Q_{c'})\|. \tag{11}$$

With this, we can state the transfer learning bound

Proposition 5.1. *Suppose that the source and target input distributions satisfy a Poincaré inequality with constant $c_{\tilde{Q}}$ and c_Q (resp.). Then, with probability $1 - \delta$, over the selection Q_c and $Q_{c'}$ of target class distributions, we have that the CDNV expectation for two target classes is bounded by the geometric complexity of the embedding network f in the following way*

$$\begin{aligned}
\mathbb{E}_{Q_c \neq Q_{c'}}\left[V_f(Q_c, Q_{c'})\right] \quad \leq \quad & \frac{c_{\tilde{Q}}\,\mathrm{GC}(f, \widetilde{Q})}{k-1}\left(\sum_{i \neq j} \frac{1}{d_{ij}^2}\right) \\
& + \left(8 + \frac{16 c_Q \sup_{c \in C} \mathrm{GC}(\mathcal{F}^*, Q_c)}{\Delta(\mathcal{F}^*)}\right)\left(\frac{\sqrt{2\pi \log(k)}\,\mathfrak{H}(\mathcal{F}^*, \widetilde{Q})}{(k-1)\Delta(\mathcal{F}^*)}\right) \\
& + \left(1 + \frac{4 \sup_{\substack{x \in \mathrm{supp}(Q) \\ f \in \mathcal{F}^*}} \|f(x)\|}{\Delta(\mathcal{F}^*)^2}\right)\left(\frac{2\sqrt{\log \frac{1}{\delta}}\, c_Q \sup_{c \in \mathcal{C}} \mathrm{GC}(\mathcal{F}^*, Q_c)}{\sqrt{k}\,\Delta(\mathcal{F}^*)^2}\right)
\end{aligned}$$

where k is the number of source classes, $\mathrm{GC}(\mathcal{F}^*, Q_c) = \sup_{f\in\mathcal{F}^*} \mathrm{GC}(f, Q_c)$, $d_{ij} = \|\mu_f(Q_{c_i}) - \mu_f(Q_{c_j})\|$, and $\mathfrak{H}(\mathcal{F}^*, \widetilde{Q})$ is a complexity measure for $\mathcal{F}^*$ over $\widetilde{Q}$ (see Appendix A.3).

Proof. This follows immediately from [22, Proposition 2] which proves this bound for the NC and distribution variances. Using our Proposition 4.1, we can replace the NC by the GC and swap the variances by the corresponding geometric complexities via the respective Poincaré inequality. □

Interpretation. In the bound above, the first term depends on the geometric complexity measured over the source distribution $\tilde{Q}$ and decreases as the number of source classes k increases. Thus, we can predict that lower $\mathrm{GC}(f, \tilde{Q})$ encourages lower NC values (i.e. more neural collapse) on the new target classes provided that the second and third term of the bound are also small.

However these other two terms involve both the source and target classes, making the full bound no longer dependent only on the geometric complexity over $\tilde{Q}$.

Moreover these two terms can apriori explode since the infimum over the class-mean distances $\Delta(\mathcal{F}^*)$ no longer only involves source classes (where this distance is bounded away from zero due to neural collapse into an equidistant simplex) but also target classes (for which we no longer have theoretical guarantees).

However, if the source labels are granular enough in the sense the target labels can be represented as combinations of the source labels (as for instance a target label of "dog" can be subsumed as all the breeds of dogs in source classes on ImageNet), then the issues above are mitigated. Formally, we can summarize this granularity compatibility condition between the source and target classes as follows: For each label c of the target class there exists labels $c_1, \ldots, c_k$ in the source classes such that $Q_c = 1/k(Q_{c_1} + \cdots + Q_{c_k})$. Because $\mathrm{GC}(f, \cdot)$ is linear, we have

$$\sup_{c\in\mathcal{C}} \mathrm{GC}(\mathcal{F}^*, Q_c) \leq \sup_{c\in[k]} \mathrm{GC}(\mathcal{F}^*, \tilde{Q}_c)$$

and thus recover dependence only on source classes.

To address the $\Delta(\mathcal{F}^*)$ term, observe that the compatibility condition above implies in terms of class-means that $\mu_f(Q_c) = \frac{1}{k}(\mu_f(Q_{c_1}) + \cdots + \mu_f(Q_{c_k}))$. Because of neural collapse during pre-training, these source class-means form the vertices of a face in the neural-collapse class-mean simplex making the target-class mean $\mu_f(Q_c)$ the barycenter of this face. Since the distance between any two points taken in the vertices of a simplex plus its barycenters is bounded away from zero, so is $\Delta(\mathcal{F}^*)$. Note, albeit intuitive and natural, this granularity condition is hard to verify in practice.

5.2 Improve fine-tuning by controlling pre-trained GC

When the compatibility conditions above are satisfied, Proposition 5.1 indicates that increasing the amount of neural collapse for the target classes can be achieved by lowering the embedding GC during pre-training. We verify this indeed occurs experimentally using the same regularization techniques exploited in Section 4. Furthermore, we verify that these implicit methods of controlling pre-training GC produce feature maps that perform better on fine-tuning tasks on CIFAR-FS with a ResNet-18 in Figure 4 and on mini-ImageNet with VGG-16 in Appendix A.5.5.

6 Limitations and Conclusion

There is a notable limitation when extending our findings to language modeling and, for example, large language models (LLMs). However, this limitation is not specific to our work per-se, but instead it is a limitation of the application of neural collapse to language models in general, as described in the recent work [65]. Namely, language modeling, as conducted via training by token prediction, creates a classification task where the conditions for neural collapse are implausible. The main problem, in addition to an imbalanced vocabulary, is that the embedding dimension for language models is typically far less than the number of classes (i.e., the total vocabulary size), making the neural collapse simplex impossible to exist. Extending the notion of neural collapse to the language models is an open question and an active area of research and beyond the scope of this work.

 https://doi.org/10.52202/079017-2131

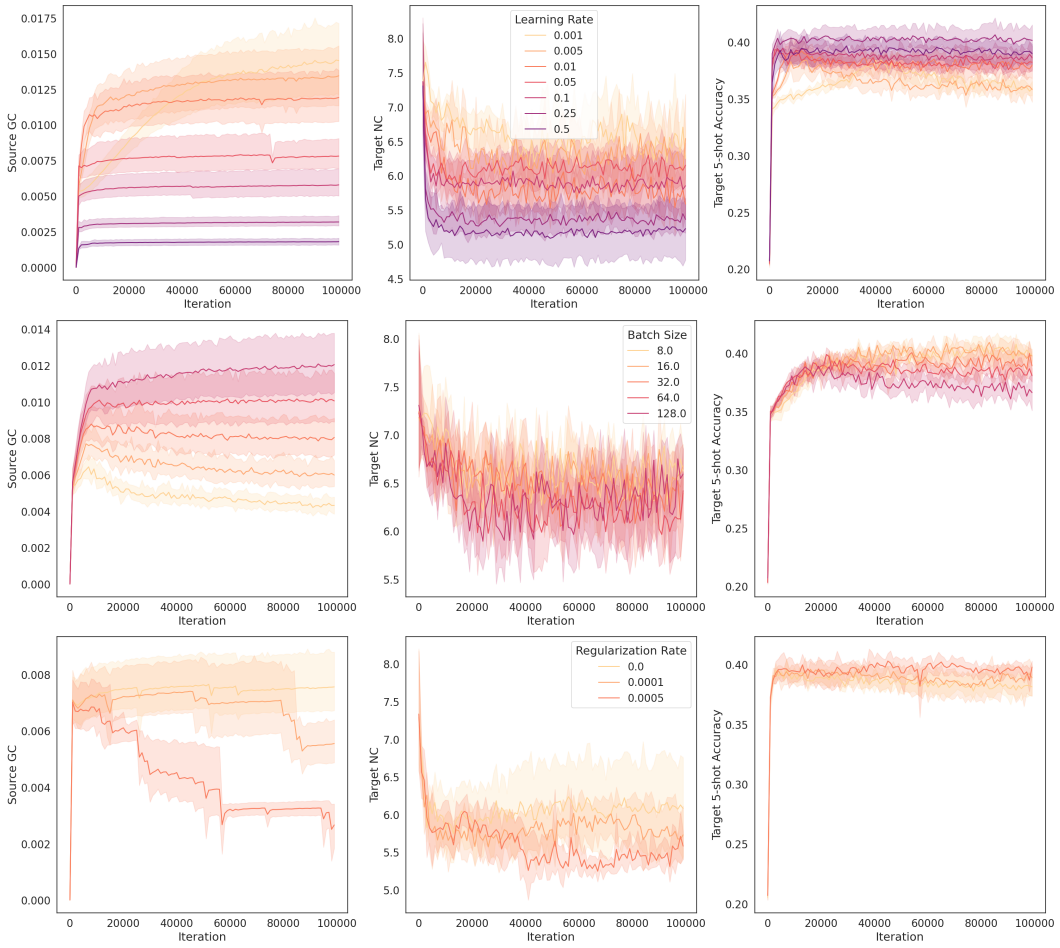

Figure 4: Controlling target neural collapse through source GC on CIFAR-FS with ResNet-18. Lower Source GC produces more neural collapse on target classes (i.e. lower target NC), and higher 5-shot transfer accuracy for **Top row:** increased learning rates, **Middle row:** decreased batch sizes, and **Bottom row:** increased L2 regularization.

Uncovering and understanding the implicit biases that enable successful transfer learning is a critical area of research in modern machine learning. Here we provide a framework connecting three different themes behind implicit regularization: flatness of the loss surface, geometric complexity, and neural collapse. We show that the embedding geometric complexity directly controls the neural collapse during training and, moreover, plays a role in the success of transfer learning.

Extensive experiments on different image classification and fine tuning tasks across different hyperparameters verify our hypothesis. We believe this opens up an intriguing direction of research further exploring the role of geometric collapse and geometric complexity in deep learning and provides valuable insight for designing more efficient and effective techniques for model training and fine-tuning.

Acknowledgments and Disclosure of Funding

We would like to thank Patrick Cole for their support. We would also like to thank Hanna Mazzawi and Peter Bartlett for helpful conversations.

References

[1] D. Bakry, I. Gentil, and M. Ledoux. A simple proof of the poincaré inequality for a large class of probability measures. *Electronic Communications in Probability*, 13:391–401, 2008.

[2] Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

[3] David G.T. Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021.

[4] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings, 2012.

[5] Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Annual Conference Computational Learning Theory*, 2019.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Sébastien Bubeck and Mark Sellke. A universal law of robustness via isoperimetry. *Journal of the ACM*, 70(2):1–18, 2023.

[8] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[9] Rich Caruana. Learning many related tasks at the same time with backpropagation. *Advances in neural information processing systems*, 7, 1994.

[10] Matias Cattaneo, Jason Klusowski, and Boris Shigida. On the implicit bias of adam. *arXiv:2309.00079*, 2023.

[11] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

[12] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204. PMLR, 2015.

[13] Alex Damian, Tengyu Ma, and Jason D. Lee. Label noise SGD provably prefers flat global minimizers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *NeurIPS 2021*, 2021.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] Benoit Dherin, Micheal Munn, and David GT Barrett. The geometric occam's razor implicit in deep learning. *NeurIPS, OPT2021*, 2021.

[17] Benoit Dherin, Michael Munn, Mihaela Rosca, and David Barrett. Why neural networks find simple solutions: the many regularizers of geometric complexity. *Advances in Neural Information Processing Systems*, 35:2333–2349, 2022.

[18] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.

[19] Lawrence C Evans. *Partial differential equations*, volume 19. American Mathematical Society, 2022.

 https://doi.org/10.52202/079017-2131

[20] Abolfazl Farahani, Behrouz Pourshojae, Khaled Rasheed, and Hamid R. Arabnia. A concise review of transfer learning. *CoRR*, abs/2104.02144, 2021. URL `https://arxiv.org/abs/2104.02144`.

[21] Herbert Federer. *Geometric measure theory*. Springer, 2014.

[22] Tomer Galanti, András György, and Marcus Hutter. On the role of neural collapse in transfer learning. In *ICLR*, 2022.

[23] Peifeng Gao, Qianqian Xu, Yibo Yang, Peisong Wen, Huiyang Shao, Zhiyong Yang, Bernard Ghanem, and Qingming Huang. Towards demystifying the generalization behaviors when neural collapse emerges. *arXiv preprint arXiv:2310.08358*, 2023.

[24] Jonas Geiping, Micah Goldblum, Phil Pope, Michael Moeller, and Tom Goldstein. Stochastic training is not necessary for generalization. In *ICLR*, 2022.

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

[26] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31, 2018.

[27] X.Y. Han, Vardan Papyan, and David L. Donoho. Neural collapse under MSE loss: Proximity to and dynamics on the central path. In *ICLR*, 2022.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[29] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

[30] Judy Hoffman, Daniel A. Roberts, and Sho Yaida. Robust learning with jacobian regularization. *arXiv*, 2020. URL `https://arxiv.org/abs/1908.02729`.

[31] Valentin Khrulkov, Artem Babenko, and Ivan Oseledets. Functional space analysis of local gan convergence. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 2021.

[32] Vignesh Kothapalli. Neural collapse: A review on modelling principles and generalization. *arXiv preprint arXiv:2206.04041*, 2022.

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

[34] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10657–10665, 2019.

[35] 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.

[36] Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: Implicit bias matters for language models. In *ICML*, 2023.

[37] Xuantong Liu, Jianfeng Zhang, Tianyang Hu, He Cao, Yuan Yao, and Lujia Pan. Inducing neural collapse in deep long-tailed learning. In *AISTATS*, 2023.

[38] Chao Ma and Lexing Ying. The sobolev regularization effect of stochastic gradient descent. 2021. URL `https://arxiv.org/abs/2105.13462`.

[39] Chao Ma and Lexing Ying. On linear stability of sgd and input-smoothness of neural networks. In *NeurIPS*, 2021.

[40] Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

[41] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[42] Michael Munn, Benoit Dherin, and Javier Gonzalvo. A margin-based multiclass generalization bound via geometric complexity. *ICML, TAGML Workshop*, 2023.

[43] Behnam Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.

[44] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[45] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. In *ICLR*, 2018.

[46] 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), 2020.

[47] Loucas Pillaud-Vivien, Francis Bach, Tony Lelièvre, Alessandro Rudi, and Gabriel Stoltz. Statistical estimation of the poincaré constant and application to sampling multimodal distributions. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2753–2763, 2020.

[48] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[50] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[51] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2016.

[52] Liliang Ren, Yang Liu, Shuohang Wang, Yichong Xu, Chenguang Zhu, and ChengXiang Zhai. Sparse modular activation for efficient sequence modeling. In *NeurIPS*, 2023.

[53] Mihaela Rosca, Theophane Weber, Arthur Gretton, and Shakir Mohamed. A case for new neural network smoothness constraints. In *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, volume 137 of *Proceedings of Machine Learning Research*, 2020.

[54] André Schlichting. Poincaré and log–sobolev inequalities for mixtures. *Entropy*, 21(1):89, 2019.

[55] Keskar Nitish Shirish, Mudigere Dheevatsa, Nocedal Jorge, Smelyanskiy Mikhail, and Tang Ping Tak Peter. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[56] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[58] Samuel L Smith, Benoit Dherin, David G.T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021.

[59] Jure Sokolic, Raja Giryes, Guillermo Sapiro, and Miguel R. D. Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65:4265–4280, 2017.

[60] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[61] Guanhua Wang and Jeffrey A Fessler. Efficient approximation of jacobian matrices involving a non-uniform fast fourier transform (nufft). *IEEE transactions on computational imaging*, 9: 43–54, 2023.

[62] Zijian Wang, Yadan Luo, Liang Zheng, Zi Huang, and Mahsa Baktashmotlagh. How far pre-trained models are from neural collapse on the target dataset informs their transferability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5549–5558, 2023.

 https://doi.org/10.52202/079017-2131

[63] Karl Weiss, Taghi Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3, 05 2016. doi: 10.1186/s40537-016-0043-6.

[64] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017.

[65] Robert Wu and Vardan Papyan. Linguistic collapse: Neural collapse in (large) language models, 2024.

[66] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[67] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115, 2021.

A Appendix

A.1 Proof of Proposition 3.2 (Neural collapse bound)

Proof. Suppose that we have an input probability distribution $Q(x) = q(x)dx$ coming from a data distribution $P(X, Y)$ (i.e. $q(x)dx$ is the marginal probability distribution of $p(x, y)dxdy$) where Y represents the k possible classes. Suppose further that $P(X, Y)$ is a balanced multi-class distribution. In terms of the input distribution $Q(x)$ this means that

$$Q = \frac{1}{k}(Q_1 + \cdots + Q_k)$$

where $Q_i(x) = q_i(x)dx$ is the input distribution of class i. Suppose moreover that $Q(x)$ satisfies the Poincaré inequality in (5), that is,

$$\mathrm{Var}_f(Q) \leq c\mathbb{E}_Q(\|\nabla_x f\|_F^2) = c\,\mathrm{GC}(f, Q)$$

for some constant c. Then the statement we want to prove is that the geometric complexity of f, that is,

$$\mathrm{GC}(f, Q) := \int \|\nabla_x f(x)\|^2 q(x)dx$$

bounds its neural collapse as measured by

$$\mathrm{NC}(f, Q) := \frac{1}{\#\{i \neq j\}} \sum_{i \neq j} \left(\frac{\mathrm{Var}_f(Q_i) + \mathrm{Var}_f(Q_j)}{2\|\mu_f(Q_i) - \mu_f(Q_j)\|^2} \right) \tag{12}$$

More precisely, we want to prove that we have the following inequality:

$$\mathrm{NC}(f, Q) \leq \frac{c \cdot \mathrm{GC}(f, Q)}{k - 1} \left(\sum_{i \neq j} \frac{1}{d_{ij}^2} \right), \tag{13}$$

where $d_{ij} = \|\mu_f(Q_i) - \mu_f(Q_j)\|$ is the distance between the mean of class i and class j. Let us now prove this statement.

First of all, since the geometric complexity respects convex sums of data densities, we have that for a distribution $Q = \frac{1}{k}(Q_1 + \cdots Q_k)$ we can write

$$\mathrm{GC}(f, Q) = \frac{1}{k}(\mathrm{GC}(f, Q_1) + \cdots + \mathrm{GC}(f, Q_k)). \tag{14}$$

In particular, this means that $\mathrm{GC}(f, Q_i) \leq k\,\mathrm{GC}(f, Q)$. Furthermore, the Poincare inequality ensures that $\mathrm{Var}_{Q_i}(f) \leq c\,\mathrm{GC}(f, Q_i)$. Using these two properties and the definition of the neural collapse, we obtain that:

$$\begin{aligned} \mathrm{NC}(f, Q) &= \frac{1}{\#\{i \neq j\}} \sum_{i \neq j} \left(\frac{\mathrm{Var}_f(Q_i) + \mathrm{Var}_f(Q_j)}{2\|\mu_f(Q_i) - \mu_f(Q_j)\|^2} \right) \\ &\leq \frac{c}{\#\{i \neq j\}} \sum_{i \neq j} \frac{\mathrm{GC}(f, Q_i) + \mathrm{GC}(f, Q_j)}{2d_{ij}^2} \\ &\leq \frac{ck\,\mathrm{GC}(f, Q)}{k(k-1)} \left(\sum_{i \neq j} \frac{1}{d_{ij}^2} \right), \end{aligned}$$

which simplifies to the desired result. □

Remark A.1. *Excepting the Poincaré constant, the quantity on the RHS above given by*

$$\frac{\mathrm{GC}(f, Q)}{k - 1} \left(\sum_{i \neq j} \frac{1}{d_{ij}^2} \right) \tag{15}$$

could be taken as an alternative measure of neural collapse. We call this the ***geometric collapse****.*

https://doi.org/10.52202/079017-2131

A.2 Proof of Proposition 4.2 (Estimating the Theoretical GC by $\widehat{\mathrm{GC}}$)

Proof. Let Q be an (input) probability distribution defined over $\mathbb{R}^d$ and let $D = \{x_i\}_{i=1}^m$ of $m \geq 1$ points drawn as i.i.d. samples from Q. Given the map $f : \mathbb{R}^d \to \mathbb{R}^p$, we denote the empirical geometric complexity of f over D by $\widehat{\mathrm{GC}}(f, D)$ which is defined as

$$\widehat{\mathrm{GC}}(f, D) = \frac{1}{m} \sum_{i=1}^{m} \|\nabla_x f(x_i)\|_F^2.$$

We begin by showing that

$$\mathbb{E}_{D \sim Q^m} \left[\widehat{\mathrm{GC}}(f, D) \right] = \mathrm{GC}(f, Q).$$

This follows by direct computation, keeping in mind that Q is a probability distribution and that the points are independently sampled. We can write Q as $Q(x) = q(x)dx$ and note that

$$\begin{aligned}
\mathbb{E}_{D \sim Q^m} \left[\widehat{\mathrm{GC}}(f, D) \right] &= \mathbb{E}_{x_1, \ldots, x_m \sim Q^m} \left[\frac{1}{m} \sum_{i=1}^{m} \|\nabla_x f(x_i)\|_F^2 \right] \\
&= \frac{1}{m} \int_{\mathbb{R}^{m \times d}} \sum_{i=1}^{m} \|\nabla_x f(x_i)\|_F^2 dQ^m(x_1, \ldots, x_m) \\
&= \frac{1}{m} \int_{\mathbb{R}^{m \times d}} \sum_{i=1}^{m} \|\nabla_x f(x_i)\|_F^2 q(x_1) \cdots q(x_m) dx_1 \cdots dx_m \\
&= \frac{1}{m} \sum_{i=1}^{m} \int_{\mathbb{R}^{(m-1) \times d}} \left[\int_{\mathbb{R}^d} \|\nabla_x f(x_i)\|_F^2 q(x_i) dx_i \right] q(x_1) \cdots \widehat{q(x_i)} \cdots q(x_m) dx_1 \cdots \widehat{dx_i} \cdots dx_m \\
&= \frac{1}{m} \sum_{i=1}^{m} \left[\int_{\mathbb{R}^d} \|\nabla_x f(x_i)\|_F^2 q(x_i) dx_i \right] \\
&= \frac{1}{m} \sum_{i=1}^{m} \left[\int_{\mathbb{R}^d} \|\nabla_x f(x_i)\|_F^2 dQ(x_i) \right] \\
&= \frac{1}{m} \sum_{i=1}^{m} \mathrm{GC}(f, Q) \\
&= \mathrm{GC}(f, Q).
\end{aligned}$$

Now suppose we have two separate independent samples D and D' each of of size $m \geq 1$ which differ by exactly one point, say x_i in D and x_i' in D'. By assumption, the map f is L-Lipschitz for some Lipschitz constnt $L > 0$. Therefore, we have

$$\widehat{\mathrm{GC}}(f, D) - \widehat{\mathrm{GC}}(f, D') = \frac{1}{m} \left(\|\nabla_x f(x_i)\|_F^2 - \|\nabla_x f(x_i')\|_F^2 \right) \leq L^2/m,$$

and similarly, $\widehat{\mathrm{GC}}(f, D') - \widehat{\mathrm{GC}}(f, D) \leq L^2/m$. It follows that $|\widehat{\mathrm{GC}}(f, D) - \widehat{\mathrm{GC}}(f, D')| \leq L^2/m$ and by applying McDiarmind's inequality (e.g., [41]), we have that for any $\epsilon > 0$,

$$\mathbb{P} \left[\widehat{\mathrm{GC}}(f, D) - \mathbb{E}_{D \sim \mu^m}[\widehat{\mathrm{GC}}(f, D)] \leq \epsilon \right] \geq 1 - \exp(-2m\epsilon^2/L^2). \tag{16}$$

Since, from the computation above, $\mathbb{E}_{D \sim \mu^m}[\widehat{\mathrm{GC}}(f, D)] = \mathrm{GC}(f, Q)$ and setting $\delta/2 = \exp(-2m\epsilon^2/L^2)$ and substituting for ϵ in (16), we get that for any $\delta > 0$ with probability as least $1 - \delta/2$ the following holds:

$$\mathrm{GC}(f, Q) \leq \widehat{\mathrm{GC}}(f, D) + L \sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

This completes the proof. □

A.3 Definition of the complexity measure $\mathfrak{H}(\mathcal{F}, Q)$ in Proposition 4.1

Let us restate here for the sake of completeness the definition given in Galanti et al. [22]. First consider the Rademacher complexity $R(A)$ of a set $A \subset \mathbb{R}^n$ which is given as the expectation $\mathbb{E}_\epsilon(X)$ of the random variable $X = \sup_{a \in A} \langle \epsilon, a \rangle$ where $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ is a random vector with components uniformly distributed among the two values -1 and 1. Now consider the input distribution Q coming from a multiclass distribution with labels in $c \in \mathcal{C}$ with label input distribution Q_c. We define the complexity measure $\mathfrak{H}(\mathcal{F}, Q)$ for the candidate functions in $\mathcal{F}$ over the source distribution Q as the Rademacher complexity of the following set:

$$\{(\mu_f(Q_c),\, \mathrm{Var}_f(Q_c)) :\ c \in \mathcal{C},\ f \in \mathcal{F})\}\,.$$

A.4 Experiment details

A.4.1 Figure 1: Geometric Complexity Controls Neural Collapse on CIFAR-10.

We trained a VGG-13 neural network on the full CIFAR-10 dataset with the provided architecture [56] and using the standard train/test split. Throughout training, we reported the following metrics measured and averaged over multiple batches of the training dataset: 1) the geometric complexity of the model embedding layer; i.e., the layer on which the neural collapse is measured, 2) the neural collapse as measured by (3) measured on the model embedding layer, and 3) the geometric collapse of the model also measured on the penultimate embedding layer of the model and given by

$$\frac{\mathrm{GC}(f, Q)}{k-1} \left(\sum_{i \neq j} \frac{1}{d_{ij}^2} \right) \tag{17}$$

where Q is the training distribution, $k = 10$ as this was CIFAR-10, and (as in the paper) where $d_{ij} = \|\mu_f(Q_i) - \mu_f(Q_j)\|$ is the distance between the sample means of class i and class j.

These quantities were measured every 1000 steps of a very long pre-training of $100,000$ steps. The optimizer was plain SGD (note we obtained similar results with momentum) without any regularization nor schedule to avoid masking effects. We used random crop and random flip for data augmentation. The results were all averaged over 5 random seeds for the neural network parameter default initialization. The plots have no smoothing applied to the learning curves. **Top row:** We swept over a learning rate range of $\{0.001, 0.0025, 0.005, 0.01, 0.025, 0.1\}$ with a constant batch size of 512. **Middle row:** We swept over a batch size range of $\{8, 16, 32, 64, 128, 256\}$ with a constant learning rate of 0.01. **Bottom row:** We swept over a L2 regularization rate range of $\{0.0, 0.00025, 0.0005, 0.001, 0.0025\}$ with learning rate 0.01 and batch size 256. Each sweep took roughly 10h of training on a single Google Cloud TPU V3 accessed via a Google colab.

In Figure 5, we show the complete learning curves of that experiment.

A.4.2 Figure 3: Tightness of the generalization bound on CIFAR-10

We trained a VGG-13 neural network [56] on the full CIFAR-10 dataset using the provided train/test split. For this model architecture we use an embedding layer dimension $p = 1024$. Note also that the number of examples per class is $m_c = 5000$. We used a constant learning rate of 0.005, a batch size of 512 and trained for 100000 steps reporting metrics every 1000 steps.

Throughout training, we reported the following metrics measured and averaged over multiple batches of the training dataset: 1) the empirical geometric complexity of the model embedding layer; i.e., the layer on which the neural collapse is measured, denoted $\widehat{\mathrm{GC}}$, and 2) the sum of the inverse squares of d_{ij} which denotes the distance between the mean of class i and class j for $i, j \in [k]$ for $i \neq j$ and, here, $k = 10$. These quantities were used to create the blue curve in Figure 3. We also computed the nearest mean classifier error on the test set. Recall, for the feature map f and a sample S the nearest mean classifier is defined as $h_{f,S} := \mathrm{argmin}_{c \in [k]} \|f(x) - \mu_f(S_c)\|$ where, here $k = 10$, and $\mu_f(S_c)$ denotes the sample mean of the set S_c for class c under the feature map f. To create the orange curve in Figure 3 we plot the average error $h_{f,S}(x) \neq y$ over the test set.

https://doi.org/10.52202/079017-2131

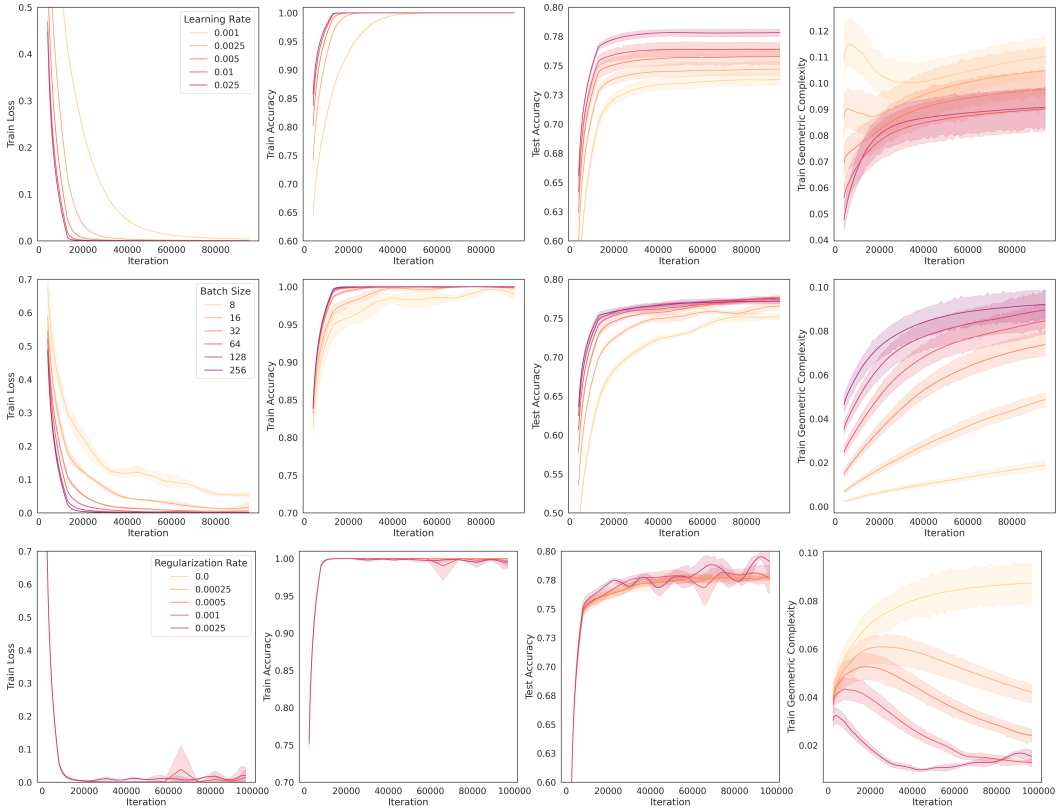

Figure 5: VGG-13 on CIFAR-10. Training has reached terminal phase of training (TPT) with training accuracy equal to 1 (see **first** and **second** column). Lower GC in training correlates with higher test accuracy in all settings (see **third** and **fourth** column).

A.4.3 Figure 4: Controlling target performance through source GC on CIFAR-FS.

We trained a RestNet-18 neural network with width 1 implemented in Flax `https://github.com/google/flax/blob/main/examples/imagenet/models.py` on CIFAR-FS with its initial convolution modified to stride 1 and kernel size of 3 to adapt to CIFAR-FS instead of ImageNet. We used only 10 classes for the source datasets with 600 images per class, which further we split in a 10/90 split for the test and train splits. The metrics were computed on an average of 100 random samples of 5 classes in the remaining classes of CIFAR-FS. We reported 1) the geometric complexity of the embedding layer measured on the source dataset (Source GC), 2) the neural collapse as measured by (3) measured and averaged on the 100 random samples of 5-label samples from the target dataset, 3) the test accuracy obtained by solving the normal equation directly for a ridge regression with only 5 examples per class obtained from the target set. These three quantities were measured every 1000 steps of a very long pre-training of 100_000 steps. The optimizer was plain SGD (note we obtained similar results with momentum) without any regularization nor schedule to avoid masking effects. We used random crop and random flip for data augmentation. The results were all averaged over 5 random seeds for the neural network parameter default initialization. The plots have no smoothing applied to the learning curves. **Top row:** We swept over a learning rate range of $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5\}$ with a constant batch size of 256. **Middle row:** We swept over a batch size range of $\{8, 16, 32, , 64, 128\}$ with a constant learning rate of 0.005. **Bottom row:** We swept over a L2 regularization rate range of $\{0, 0.0001, 0.0005\}$ Each sweep took roughly 10h of training on a single Google Cloud TPU V3 accessed via a Google colab.

A.5 Additional Experiments

A.5.1 Cifar-10 on ResNet-18

In Figure 6, we trained a ResNet-18 neural network on CIFAR-10. The experimental setup is the same as described in Appendix A.4.1. The results were all averaged over 5 random seeds for the neural network parameter default initialization. The plots have no smoothing applied to the learning curves.

Top row: We swept over a learning rate range of $\{0.005, 0.01, 0.025, 0.05, 0.1, 0.2\}$ with a constant batch size of 256. **Middle row:** We swept over a batch size range of $\{8, 16, 32, 64, 128, 256\}$ with a constant learning rate of 0.2. **Bottom row:** We swept over a L2 regularization rate range of $\{0.0, 0.0001, 0.00025, 0.0005, 0.00075\}$ with learning rate 0.02 and batch size 512.

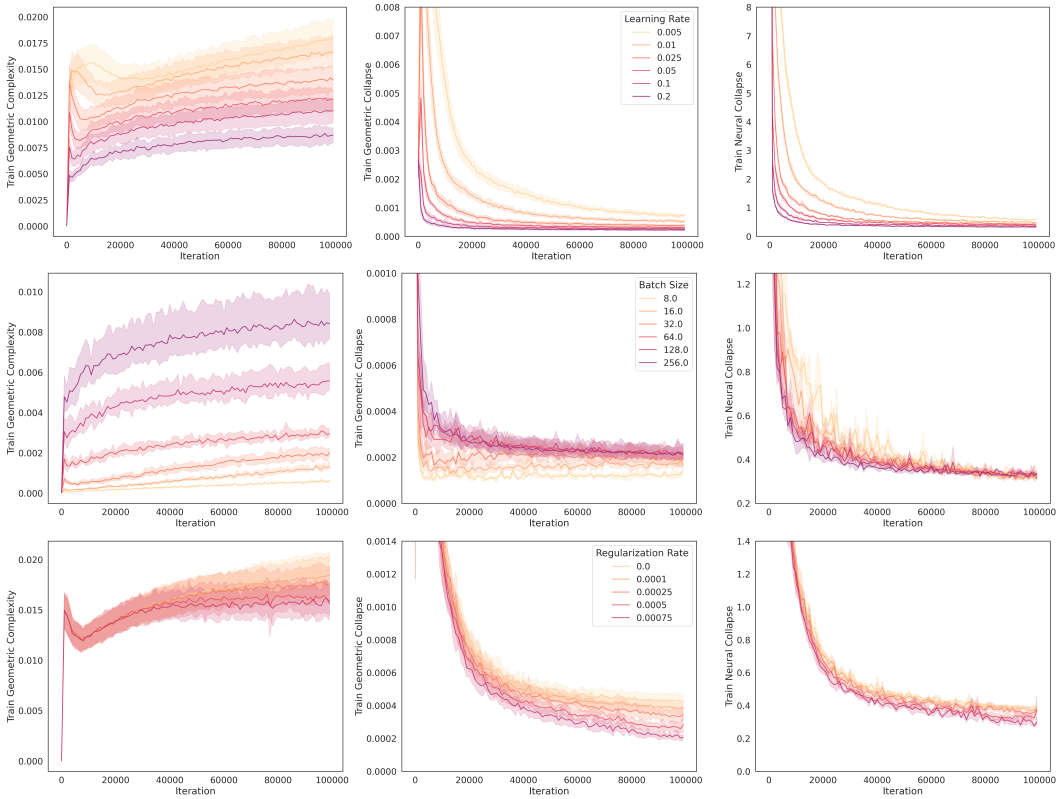

Figure 6: Controlling the neural collapse via the model geometric complexity for ResNet-18 trained on CIFAR-10. Lower GC produces lower geometric collapse and more neural collapse (i.e., lower NC) for **Top row:** increased learning rates, **Middle row:** decreased batch sizes, and **Bottom row:** increased L2 regularization.

A.5.2 MNIST on VGG-11

In Figure 7, we trained a VGG-11 neural network [56] on MNIST. The experimental setup is the same as described in Appendix A.4.1. The results were all averaged over 5 random seeds for the neural network parameter default initialization. The plots have no smoothing applied to the learning curves.

Top row: We swept over a learning rate range of $\{0.0025, 0.005, 0.01, 0.025, 0.05, 0.1\}$ with a constant batch size of 512. **Middle row:** We swept over a batch size range of $\{8, 16, 32, 64, 128, 256, 512, 1024\}$ with a constant learning rate of 0.005. **Bottom row:** We swept over a L2 regularization rate range of $\{0, 0.00001, 0.0001, 0.00025, 0.0005\}$ with learning rate 0.02 and batch size 512.

https://doi.org/10.52202/079017-2131

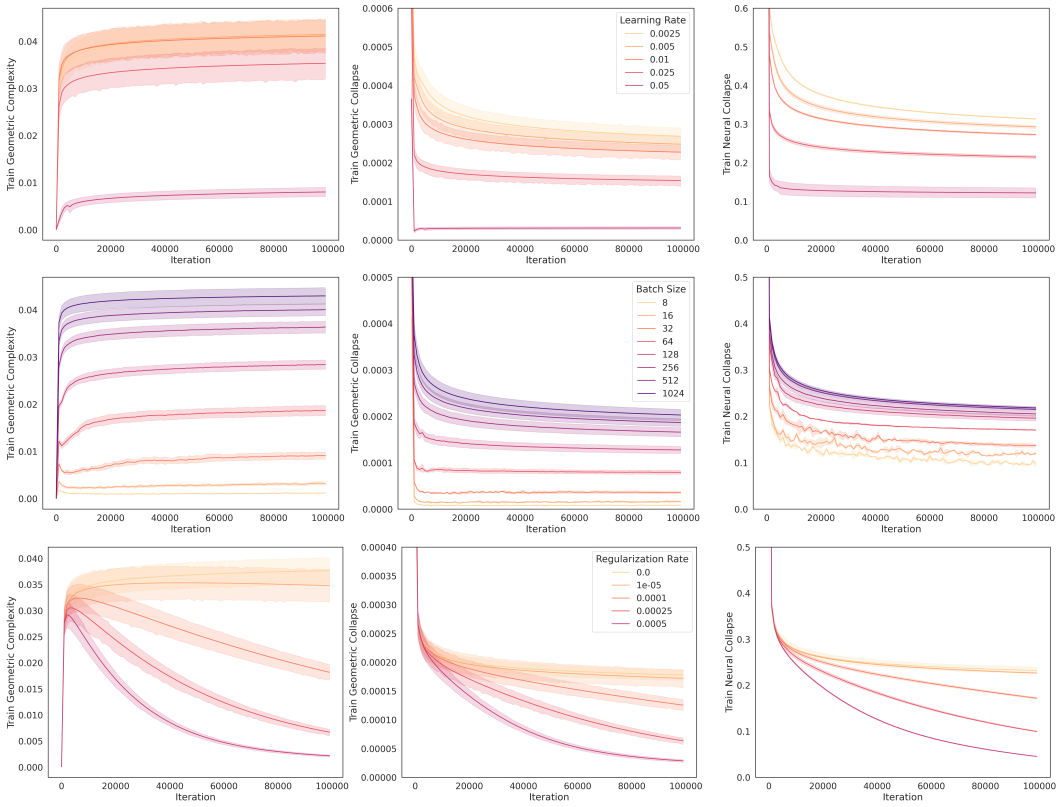

Figure 7: Controlling the neural collapse via the model geometric complexity for VGG-11 trained on MNIST. Lower GC produces lower geometric collapse and more neural collapse (i.e., lower NC) for **Top row:** increased learning rates, **Middle row:** decreased batch sizes, and **Bottom row:** increased L2 regularization.

A.5.3 Fashion-MNIST on VGG-11

In Figure 8, we trained a VGG-11 neural network [56] on Fashion-MNIST. The experimental setup is the same as described in Appendix A.4.1. The results were all averaged over 5 random seeds for the neural network parameter default initialization. The plots have no smoothing applied to the learning curves.

Top row: We swept over a learning rate range of $\{0.005, 0.01, 0.025, 0.05, 0.1, 0.2\}$ with a constant batch size of 256. **Bottom row:** We swept over a L2 regularization rate range of $\{0, 0.00001, 0.0001, 0.00025, 0.0005\}$ with learning rate 0.02 and batch size 512.

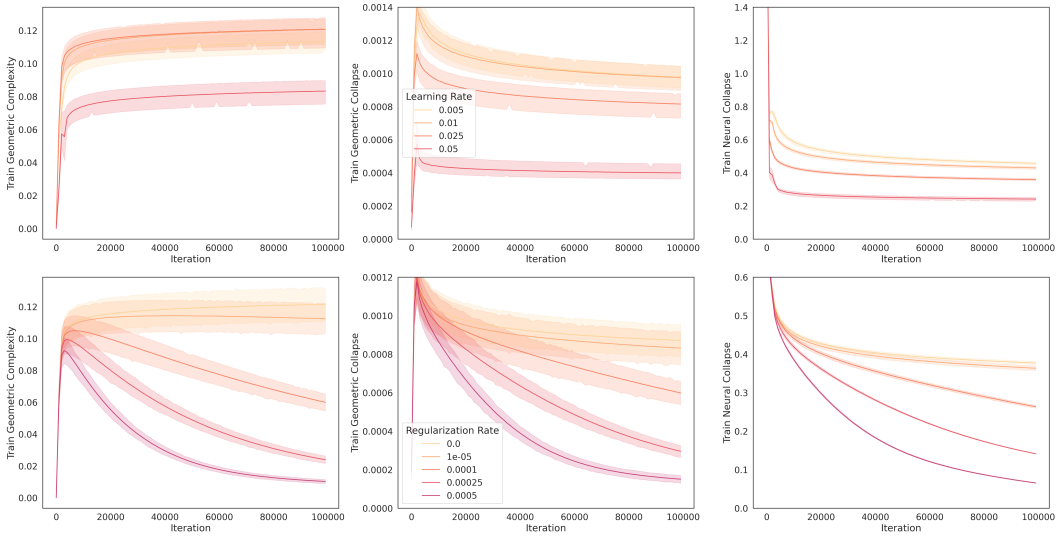

Figure 8: Controlling the neural collapse via the model geometric complexity for VGG-11 trained on Fashion-MNIST. Lower GC produces lower geometric collapse and more neural collapse (i.e., lower NC) for **Top row:** increased learning rates, **Bottom row:** increased L2 regularization.

A.5.4 Cifar-100 on ResNet50

In Figure 9, we trained a ResNet-50 neural network on CIFAR-100. The experimental setup is the same as described in Appendix A.4.1. The results were all averaged over 5 random seeds for the neural network parameter default initialization. The plots have no smoothing applied to the learning curves.

Top row: We swept over a learning rate range of $\{0.005, 0.01, 0.05, 0.1, 0.2\}$ with a constant batch size of 256. **Bottom row:** We swept over a L2 regularization rate range of $\{0.0, 0.0001, 0.00025, 0.0005, 0.00075\}$ with learning rate 0.02 and batch size 256.

https://doi.org/10.52202/079017-2131

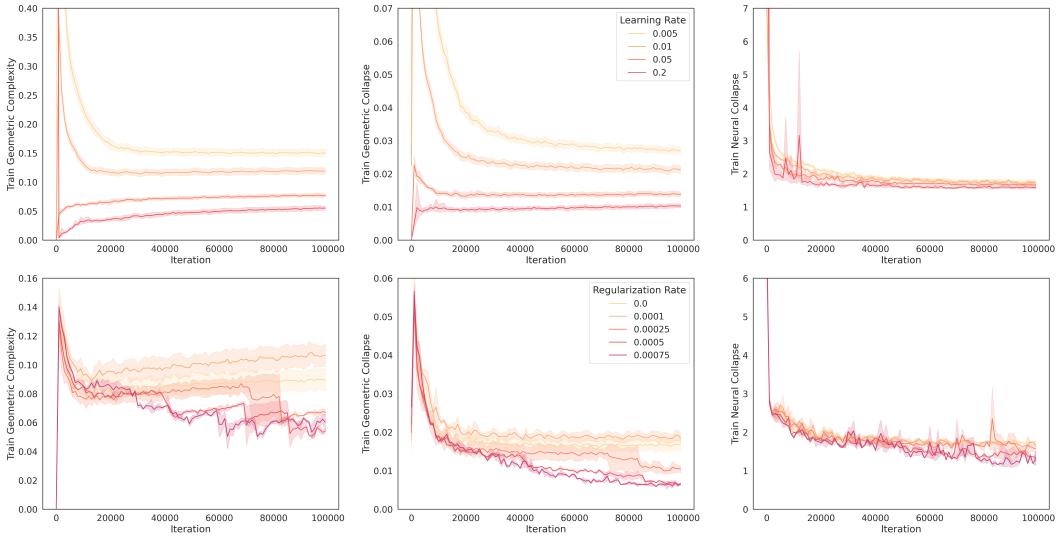

Figure 9: Controlling the neural collapse via the model geometric complexity for ResNet-50 trained on CIFAR-100. Lower GC produces lower geometric collapse and more neural collapse (i.e., lower NC) for **Top row:** increased learning rates, **Bottom row:** increased L2 regularization.

A.5.5 Lower Pre-trained GC Leads to Improved Fine-tuning: mini-ImageNet dataset on VGG architecture

In Figure 10, we trained a VGG16 neural network (as described in [57]) implemented in Flax on Mini-ImageNet (as described in [60]). We used 10 classes with 600 examples per class for training, with a 10/90 train/test split. To evaluate the downstream performance we used 100 downstream tasks consisting of 5 labels for few shot learning randomly chosen from a pool of 20 separate from the training labels. The experimental setup was the same as described in section A.4.3. Namely, we reported 1) the geometric complexity of the embedding layer measured on the source dataset (Source GC), 2) the neural collapse as measured by (3) measured and averaged on the 100 random samples of 5-label samples from the target dataset, 3) the test accuracy obtained by solving the normal equation directly for a ridge regression with only 5 examples per class obtained from the target set. These three quantities were measured every 1000 steps of a very long pre-training of 100,000 steps. The optimizer was plain SGD (note we obtained similar results with momentum) without any regularization nor schedule to avoid masking effects. We used random crop and random flip for data augmentation. The results were all averaged over 5 random seeds for the neural network parameter default initialization. We swept over a learning rate range of $\{0.001, 0.005, 0.01\}$ with a constant batch size of 256. Each sweep took roughly 10h of training on a single Google Cloud TPU V3 accessed via a Google colab.

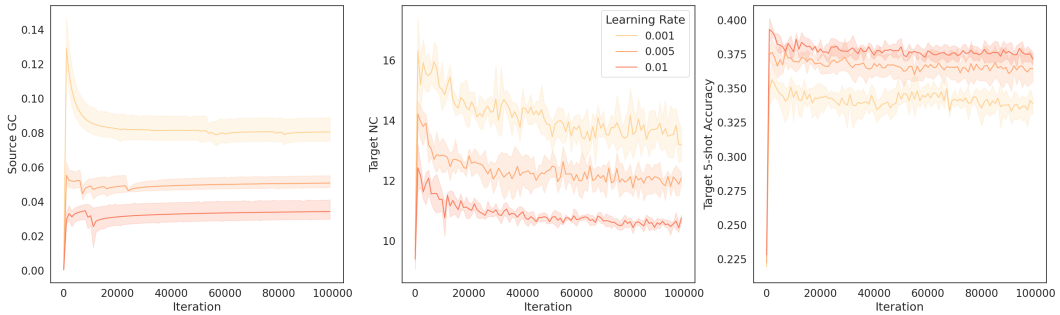

Figure 10: Controlling target NC through source GC on mini-imagenet with VGG-16: Increased learning rates produces lower GC and more neural collapse (i.e. lower NC) resulting in higher 5-shot transfer accuracy.

A.5.6 Direct GC regularization increases neural collapse

To mitigate possible confounding factors due to batch size and learning rate manipulations, we train a VGG-13 model [57] on CIFAR-10 with explicit GC regularization using a fixed learning rate of 0.01 and a batch size of 256. We explore increasing levels of explicit GC regularization with rates $10^{-6}, 10^{-5}, 10^{-4}$ and plot the results in Figure 11. We observe the same predictions as in our experiments which indirectly lower the GC through learning rates, batch sizes and L2 regularization. Namely, lower GC produces lower NC values (i.e., increased neural collapse). Note that in this experiment we regularized w.r.t. the GC computed at the logit layer rather than the embedding GC. This is because the high dimension of the embedding layer makes taking the gradient of the embedding GC prohibitively expensive. Already, regularization with GC taken at the logit layer was possible only for one seed. That being said, explicit regularization by GC computed at the logit layer provides a direct way to control GC computed at the embedding layer.

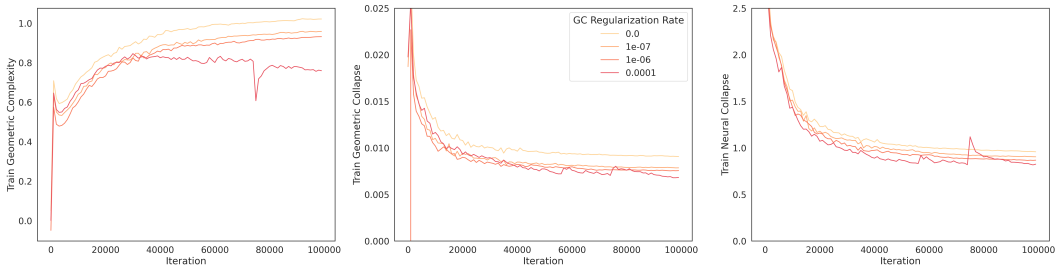

Figure 11: VGG-13 on CIFAR-10 with explicit GC regularization (one seed).

A.6 Discussion on the Poincaré inequality

Recall that a distribution Q satisfies a Poincaré inequality if, for all differentiable functions v defined on the support of Q, there exists a constant $c > 0$ such that

$$\mathrm{Var}_v(Q) \leq c\mathbb{E}_{x\sim Q}\left[\|\nabla_x v\|_F^2\right]. \tag{18}$$

In this section, we argue that the Poincare Inequality (PI) is a mild and natural assumption to make. For instance, the Gaussian distribution, mixtures of Gaussians with equal (or different) variances, mixtures of Gaussians and sub-Gaussians, mixtures of uniform and Gaussian measures; and any log-concave probability measure all satisfy a PI; see, for example, [1, 54]. The same is true for most distributions of importance in probability and statistics; see [47].

The Poincaré constant c itself can take very high values depending on the spread of the distribution. For example, for a standard normal distribution the Poincare constant is 1 while for a multivariate normal distribution the Poincare constant equals the largest eigenvalue of its covariance matrix which can be any positive real number. Intuitively speaking, the Poincaré constant will increase if the space is stretched in any direction and decrease if it is compressed in any direction.

The PI has also been assumed to hold for real life image datasets, where it has been used to help improve GAN convergence, as in [31]. It is also a key assumption to understand the role of over-parameterization in generalization as happens for large neural networks (cf. [7] which frames this as an equivalent isoperimetry condition).

On the contrary, non-PI distributions are considered pathological; they can be constructed for instance by artificially concatenating distributions with fully disjoint support. See [42] for a construction of pathological examples of distributions not satisfying a Poincaré inequality.

https://doi.org/10.52202/079017-2131

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the paper's contributions, assumptions, and limitations. The claims about how geometric complexity influences neural collapse align with the results, reflecting the paper's scope. The abstract clearly indicates the generalizability of the proposal based on the datasets used.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The experiments in this paper focus on image processing and involve a large number of important hyperparameters and extensive runs. The paper clearly states that the generalization effect during transfer learning is tested only on image processing datasets, such as CIFAR-10, CIFAR-100, MNIST, Fashion-MNIST and CIFAR-FS, mini-ImageNet in the transfer learning setting.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [Yes]

   Justification: Yes, the paper provides the full set of assumptions and a complete, correct proof for each theoretical result. Each assumption is clearly stated, and the proofs are meticulously detailed, ensuring the validity and rigor of the theoretical contributions.

   Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: Yes, the paper fully discloses all the information needed to reproduce the main experimental results, ensuring that the main claims and conclusions can be independently verified. This includes detailed descriptions of the methodology, data sources and all relevant parameters and all hyperparameters.

   Guidelines:

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

https://doi.org/10.52202/079017-2131

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: Yes, while the code is not open source, the paper uses open datasets, which are well-known benchmark datasets for image processing, such as CIFAR-10, CIFAR-100, MNIST, Fashion-MNIST and CIFAR-FS and mini-ImageNet for transfer learning. The supplemental material includes detailed instructions to faithfully reproduce the main experimental results.

   Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: Yes, the paper specifies all the training and test details, including data splits, hyperparameters, their selection process, and the type of optimizer, ensuring a comprehensive understanding of the results. The Appendix includes full experiment details for all main paper plots as well as for all plots appearing in the Appendix.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

Answer: [Yes]

Justification: Yes, the paper reports error bars that are suitably and correctly defined, based on multiple runs for each training. It provides the standard deviation for each metric, such as accuracy, geometric complexity, and neural collapse, ensuring a clear understanding of the statistical significance of the experiments. All experiments are repeated over 5 random seeds and error bars are included in the Figures.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: Yes, the paper provides sufficient information on the computer resources required for each experiment, including the type of compute workers, and execution time, ensuring that the experiments can be accurately contextualized for performance.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: Yes, it does.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

https://doi.org/10.52202/079017-2131

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

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

    Answer: [NA]

    Justification: There are no societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: There are no risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [NA]

Justification: No use of third-party code beyond machine learning frameworks like JAX and TensorFlow, and common Python libraries for experimental purposes.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [NA]

    Justification: No new assets released.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: No crowdsourcing or research performed with human subjects.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

https://doi.org/10.52202/079017-2131

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

Answer: [NA]

Justification: No crowdsourcing or research performed with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.