# ON THE ROLE OF NEURAL COLLAPSE IN TRANSFER LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We study the ability of foundation models for classification to learn representations that are transferable to new, unseen classes. Recent results in the literature show that representations learned by a single classifier over many classes are competitive on few-shot learning problems with representations learned by special-purpose algorithms designed for such problems. In this paper we provide an explanation for this behavior based on the recently observed phenomenon that the features learned by overparameterized neural classifiers show an interesting clustering property, called neural collapse. We demonstrate both theoretically and empirically that neural collapse generalizes to new samples from the training classes, and – more importantly – to new classes as well, allowing foundation models to provide feature maps that work well in transfer learning and, specifically, in the few-shot setting.

## 1 INTRODUCTION

In a variety of machine learning applications, we have access to a limited amount of data from the task that we would like to solve, as labeled data is oftentimes scarce and/or expensive. In such scenarios, training directly on the available data is unlikely to produce a hypothesis that generalizes well on new, unseen test samples. A prominent solution to this problem is to apply transfer learning (see, e.g., Caruana, 1995; Bengio, 2012; Yosinski et al., 2014). In transfer learning, we are typically given a large-scale source task (e.g., ImageNet ILSVRC, Russakovsky et al., 2015) and a target task from which we encounter only a limited amount of data. While there are multiple approaches to transfer knowledge between tasks, a popular approach suggests to train a large neural network on a source classification task with a wide range of classes (such as ResNet50, He et al., 2016, MobileNet, Howard et al., 2017 or the VGG network, Simonyan & Zisserman, 2014), and then to train a relatively smaller network (e.g., a linear classifier or a shallow MLP) on top of the penultimate layer of the pretrained network, using the data available in the target task.

Due to the effectiveness of this approach, transfer learning has become a central element in the machine learning toolbox. For instance, using pretrained feature maps is common practice in a variety of applications, including fine-grained classification (Chen et al., 2019a; Huang & Li, 2020; Yang et al., 2018), object detection, (Redmon et al., 2016; Ren et al., 2015; He et al., 2017), semantic segmentation (Long et al., 2015; Chen et al., 2018), or medical imaging (Chen et al., 2019b). In fact, due to the cumulative success of transfer learning, large pretrained models that can be effectively adapted to a wide variety of tasks (Brown et al., 2020; Ramesh et al., 2021) have recently been characterized as foundation models (Bommasani et al., 2021), emphasizing their central role in solving various learning tasks.

Typically, a foundation model is pretrained on a source task at a time when the concrete description of the target task (or target tasks) is not – or only partially – available to the practitioner. Therefore, the ability to precondition or design the training regime of the foundation model to match the target task is limited. As a result, there might be a domain shift between the source and target tasks, such as a different amount or set of target classes, or a different number of available samples. Hence, intuitively, a foundation model should be fairly generic and applicable in a wide range of problems.

On the other hand, when some specifics of the target tasks are known, often special-purpose algorithms are designed to utilize this information. Such an example is the problem of few-shot learning, when it is known in advance that the target problems come with a very small training set (Vinyals

et al., 2016; Ravi & Larochelle, 2017; Finn et al., 2017; Lee et al., 2019). While these specialized algorithms have significantly improved the state of the art, the recent work of Tian et al. (2020); Dhillon et al. (2020) demonstrated that state-of-the-art performance is achievable on few-shot learning benchmarks based on foundation models.

Despite the wide range of applications and success of transfer learning and foundation models in particular, only relatively little is known theoretically why transfer learning is possible between two tasks in the setting mentioned above.

**Contributions.** In this paper we present a new perspective on transfer learning with foundation models, based on the recently discovered phenomenon of neural collapse (Papyan et al., 2020). Informally, neural collapse identifies training dynamics of deep networks for standard classification tasks, where the features (the output of the penultimate layer) associated with training samples belonging to the same class concentrate around their class feature mean. We demonstrate that this phenomenon generalizes to new data points and new classes (e.g., the target classes), when the model is trained on a large set of classes with many samples for each class. In addition, we show that in the presence of neural collapse, training a linear classifier on top of the learned penultimate layer can be done using only few samples. We verify these findings both theoretically and via experiments. In particular, our results provide a compelling explanation for the empirical success of foundation models, as observed, e.g., by Tian et al. (2020); Dhillon et al. (2020).

The rest of the paper is organized as follows: Some additional related work is discussed in Section 1.1. The problem setting, in particular, foundation models, are introduced in Section 2. Neural collapse is described in Section 3. Our theoretical analysis is presented in Section 4: generalization to unseen examples is considered in Section 4.1, to new classes in Section 4.2, while the effect of neural collapse on the classifier performance is discussed in Section 4.3. Experiments are presented in Section 5, and conclusions are drawn in Section 6. Proofs and additional experiments are relegated to the appendix.

### 1.1 Other Related Work

Various publications, such as (Baxter, 2000; Maurer et al., 2016; Pentina & Lampert, 2014; Galanti et al., 2016; Khodak et al., 2019; Du et al., 2021), suggested different frameworks to theoretically study various multi-task learning settings. However, all of these works consider learning settings in which the learning algorithm is provided with a sequence of learning problems, and therefore, are unable to characterize the case where only one source task is available.

In contrast, we propose a theoretical framework in which the learning algorithm is provided with one classification task and is required to learn a representation of the data that can be adapted to solve new unseen tasks using few samples. This kind of modeling is aligned with the common practice of transfer learning using deep neural networks.

While in unsupervised domain adaptation (Ben-david et al., 2006; Mansour, 2009) we typically have access to a single source and a single target task, we do not have access to labels from the former, making the problem ill-posed by nature and shaping the algorithms to behave differently than those that are provided with labeled samples from the target task. In particular, even though in transfer learning we typically train the feature map on the source dataset, independently of the target task, in domain adaptation this would significantly limit the algorithm's ability to solve the target task.

## 2 Problem Setup

We consider the problem of training foundation models, that is, general feature representations which are useful on a wide range of learning tasks, and are trained on some auxiliary task. To model this problem, we assume that the final target task we want to solve is a $k$-class classification problem $T$ (the *target* problem), coming from an unknown distribution $\mathcal{D}$ over such problems, and the auxiliary task where the feature representation is learned on an $l$-class classification problem, called the *source* problem. Formally, the target class is defined by a distribution $P$ over samples $(x, y) \in \mathcal{X} \times \mathcal{Y}_k$, where $\mathcal{X} \subset \mathbb{R}^d$ is the instance space, and $\mathcal{Y}_k$ is a label space with cardinality $k$. To simplify the presentation, we use one-hot encoding for the label space, that is, the labels are represented by the unit vectors in $\mathbb{R}^k$, and $\mathcal{Y}_k = \{e_i : i = 1, \ldots, k\}$ where $e_i \in \mathbb{R}^k$ is the $i$th standard

unit vector in $\mathbb{R}^k$; with a slight abuse of notation, sometimes we will also write $y = i$ instead of $y = e_i$. For a pair $(x, y)$ with distribution $P$, we denote by $P_i$ the class conditional distribution of $x$ given $y = i$ (i.e., $P_i(\cdot) = \mathbb{P}[x \in \cdot \mid y = i]$).

A classifier $h : \mathcal{X} \to \mathbb{R}^k$ assigns a *soft* label to an input point $x \in \mathcal{X}$, and its performance on the target task $T$ is measured by the risk

$$L_T(h) = \mathbb{E}_{(x,y)\sim P}[\ell(h(x), y)], \tag{1}$$

where $\ell : \mathbb{R}^k \times \mathcal{Y}_k \to [0, \infty)$ is a non-negative loss function (e.g., zero-one loss or cross-entropy loss).

Our goal is to learn a classifier $h$ from some training data $S = \{(x_i, y_i)\}_{i=1}^n$ of $n$ independent and identically distributed (i.i.d.) samples drawn from $P$. However, when $n$ is small and the classification problem is complicated, this might not be an easy problem. To facilitate finding a good solution, we aim to find a classifier of the form $h = g \circ f$, where $f : \mathbb{R}^d \to \mathbb{R}^p$ is a feature map from a family of functions $\mathcal{F} \subset \{f' : \mathbb{R}^k \to \mathbb{R}^p\}$ and $g \in \mathcal{G} = \{g' : \mathbb{R}^p \to \mathbb{R}^k\}$ is a classifier used on the feature space $\mathbb{R}^p$. The idea is that the feature map $f$ is learned on some other problem where more data is available, and then $g$ is trained to solve the hopefully simpler classification problem of finding $y$ based on $f(x)$, instead of $x$. That is, $g$ is actually a function of the modified training data $\{(f(x_i), y_i)\}_{i=1}^n$; to emphasize this dependence, we denote the trained classifier by $g_{f,S}$.

We assume that the auxiliary (source) task helping to find $f$ is an $l$-class classification problem over the same sample space $\mathcal{X}$, given by a distribution $\tilde{P}$, and here we are interested in finding a classifier $\tilde{h} : \mathcal{X} \to \mathbb{R}^l$ of the form $\tilde{h} = \tilde{g} \circ f$ where $\tilde{g} \in \tilde{\mathcal{G}} \subset \{g' : \mathbb{R}^p \to \mathbb{R}^l\}$ is a classifier over the feature space $f(\mathcal{X}) = \{f(x) : x \in \mathcal{X}\}$. Given a training data set $\tilde{S} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^m$, both components of the trained classifier, denoted by $f_{\tilde{S}}$ and $\tilde{g}_{\tilde{S}}$ are trained on $\tilde{S}$, with the goal of minimizing the risk in the source task, given by

$$L_S(\tilde{h}_{\tilde{S}}) = \mathbb{E}_{(x,y)\sim\tilde{P}}[\ell(\tilde{g}_{\tilde{S}}(f_{\tilde{S}}(x)), y)]$$

(note that with a slight abuse of notation we used the same loss function as in (1), although they operate over spaces of different dimensions). A standard assumption is that $\tilde{S}$ is drawn i.i.d. form $\tilde{P}$; however, in this work instead we will use hierarchical schemes where first classes are sampled according to their marginals in $\tilde{P}$, and then for any selected class $c$, training samples $\tilde{S}_c$ are chosen from the corresponding class-conditional distribution (or just class conditional, for short) $\tilde{P}_c$; the specific assumptions on the distribution of $\tilde{S}$ will be given when needed. Since $f, \tilde{g}$ and $\tilde{h}$ always depend on the source data only, to simplify the notation we will often drop the $\tilde{S}$ subscript whenever this does not cause any confusion.

In a typical setting $\tilde{h}$ is a deep neural network, $f$ is the representation in the last internal layer of the network (i.e., the penultimate *embedding* layer), and $\tilde{g}$, the last layer of the network, is a linear map; similarly $g$ in the target problem is often taken to be linear. The learned feature map $f$ is called a *foundation model* (Bommasani et al., 2021) when it can be effectively used in a wide variety of tasks. In particular, its effectiveness can be measure by its expected performance over target tasks:

$$L_{\mathcal{D}}(f) = \mathbb{E}_{T\sim\mathcal{D}}\mathbb{E}_{S\sim P^n}[L_T(g_S \circ f)]$$

(recall that $P$ is the distribution associated with task $T$).

Notice that while the feature map $f$ is evaluated on the distribution of target tasks determined by $\mathcal{D}$, the training of $f$ in a foundation model, as described above, is fully agnostic of this target. In contrast, several transfer learning methods (such as few-shot learning algorithms) have been developed which optimize $f$ not only as a function of its training data $\tilde{S}$, but also based on some properties of $\mathcal{D}$, such as the number of classes and the number of samples per class in $S$. Perhaps surprisingly, recent studies (Tian et al., 2020; Dhillon et al., 2020) demonstrate that the target-agnostic training of foundation models (or some slight variation of them, such as transductive learning) are competitve with such special purpose algorithm. In the rest of the paper we analyze this phenomenon, and provide an explanation through the recent concept of neural collapse.

**Notation.** For an integer $a$, $[a] = \{1, \ldots, a\}$. For any real vector $z$, $\|z\|$ denotes its Euclidean norm. For a given set $A = \{a_1, \ldots, a_n\} \subset B$ and a function $u \in \mathcal{U} \subset \{u' : B \to \mathbb{R}\}$, we define

$u(A) = (u(a_1), \ldots, u(a_n))$ and $\mathcal{U}(A) = \{u(A) : u \in \mathcal{U}\}$. Let $Q$ be a distribution over $\mathcal{X} \subset \mathbb{R}^d$ and $u : \mathcal{X} \to \mathbb{R}^p$. We denote by $\mu_u(Q) = \mathbb{E}_{x\sim Q}[u(x)]$ and by $\mathrm{Var}_u(Q) = \mathbb{E}_{x\sim Q}[\|u(x) - \mu_u(Q)\|^2]$ the mean and variance of $u(x)$ for $x \sim Q$. For $A$ above, we denote by $\mathrm{Avg}_{i=1}^n[a_i] = \mathrm{Avg}\, A = \frac{1}{n}\sum_{i=1}^n a_i$ the average of $A$. We denote by $\mathrm{sign} : \mathbb{R} \to \{\pm 1\}$ the sign function.

## 3 Neural Collapse

Neural collapse (NC) is a recently discovered phenomenon in deep learning (Papyan et al., 2020): it has been observed that during the training of deep networks for standard classification tasks, the features (the output of the penultimate, a.k.a. embedding layer) associated with training samples belonging to the same class concentrate around the mean feature value for the same class. More concretely, Papyan et al. (2020) observed essentially that the ratio of the within-class variances and the distances of the class means converge to zero. They also noticed that asymptotically the class-means (centered at their global mean) are not only linearly separable, but are actually maximally distant and located on a sphere centered at the origin up to scaling (they form a simplex equiangular tight frame), and furthermore, that the behavior of the last-layer classifier (operating on the features) converges to that of the nearest-class-mean decision rule. Han et al. (2021) provided further empirical evidence regarding the presence of neural collapse in the terminal stages of training for both MSE- and cross-entropy-loss minimization.

On the theoretical side, already Papyan et al. (2020) showed the emergence of neural collapse for a Gaussian mixture model and linear score-based classifiers. Poggio & Liao (2020b;a); Mixon et al. (2020) theoretically investigated whether neural collapse occurs when training neural networks under MSE-loss minimization with different settings of regularization. Furthermore, Zhu et al. (2021) showed that in certain cases, any global optimum of the classical cross-entropy loss with weight decay in the unconstrained features model satisfies neural collapse.

Papyan et al. (2020) defined neural collapse via identifying that the within-class variance normalized by the intra-class-covariance tends to zero towards the end of the training (see Appendix B.2 for details). In this paper we work with a slightly different variation of within-class variation collapse, which will be later connected to the clusterability of the sample feature vectors. For a feature map $f$ and two (class-conditional) distributions $Q_1, Q_2$ over $\mathcal{X}$, we define their *class-distance normalized variance* (CDNV) to be

$$V_f(Q_1, Q_2) = \frac{\mathrm{Var}_f(Q_1) + \mathrm{Var}_f(Q_2)}{2\|\mu_f(Q_1) - \mu_f(Q_2)\|^2}.$$

The above definition can be extended to finite sets $S_1, S_2 \subset \mathcal{X}$ by defining $V_f(S_1, S_2) = V_f(U[S_1], U[S_2])$, where $U[S_1]$ and $U[S_2]$ denote the uniform distribution over $S_1$ and $S_2$, respectively. Our version of *neural collapse* (at training) asserts that $\lim_{t\to\infty} \mathrm{Avg}_{i\neq j\in[l]} \left[V_{f_t}(\tilde{S}_i, \tilde{S}_j)\right] = 0$. Intuitively, when decreasing the variance of the features of a given class in comparison to its distance from another class, we expect to be able to classify the features into the classes with a higher accuracy. In later sections, we also consider slight variations of neural collapse, requiring that some version of the CDNV (e.g., in expectation over target distributions) converges to zero towards the end of the training.

## 4 Neural Collapse on Unseen Data

In this section we theoretically analyze neural collapse in the setting of Section 2, with particular attention of variance collapse on data not used in the training of the classifier. First, in Section 4.1 we show that if neural collapse happens on the training set $\tilde{S}$, then it generalizes to unseen samples from the same task under mild, natural assumptions. Next we show, in Section 4.2, that one can also expect neural collapse to happen over new classes when the classes in the source and target tasks are selected randomly from the same class distributions. The significance of these results is that they show that the feature representations learned during training are immediately useful in other tasks; we show in Section 4.3 how this leads to low classification error in few-shot learning.

### 4.1 GENERALIZATION TO NEW SAMPLES FROM THE SAME CLASS

In the following proposition, proved in Appendix C, we provide a generalization bound on the CDNV, $V_f(\tilde{P}_i, \tilde{P}_j)$, between two source class-conditional distributions $\tilde{P}_i$ and $\tilde{P}_j$ in terms of its empirical counterpart, $V_f(\tilde{S}_i, \tilde{S}_j)$, and certain generalization gap terms bounding the difference between expectations and empirical averages for $f$ and its variants. Assume that for all $c \in [l]$, $\tilde{S}_c$ is a set of $m_c$ i.i.d. samples drawn from $\tilde{P}_c$, and for any $\delta \in (0,1)$, let $\epsilon_{f,1}(\tilde{P}_c, m_c, \delta)$ and $\epsilon_{f,2}(\tilde{P}_c, m_c, \delta)$ be the smallest positive values such that with probability at least $1-\delta$,

$$\left\|\mathbb{E}_{x\sim\tilde{P}_c}[f(x)] - \mathrm{Avg}_{x\in\tilde{S}_c}[f(x)]\right\| \leq \epsilon_{f,1}(\tilde{P}_c, m_c, \delta)$$
$$\left|\mathbb{E}_{x\sim\tilde{P}_c}[\|f(x)\|^2] - \mathrm{Avg}_{x\in\tilde{S}_c}[\|f(x)\|^2]\right| \leq \epsilon_{f,2}(\tilde{P}_c, m_c, \delta),$$

respectively. Typically, these quantities can be upper bounded using Rademacher complexities related to the class $\mathcal{F}$, which typically scale as $\mathcal{O}(1/\sqrt{m_c})$ (when fixing $\delta$ and $c$) and logarithmically with respect to $1/\delta$. However, these quantities are potentially smaller as they depend on the training process by which $f$ is selected from $\mathcal{F}$. For simplicity, we assume that $\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\| \geq C_0$ with probability 1 for some constant $C_0 > 0$.

**Proposition 1.** *Fix two source classes, $i$ and $j$ with distributions $\tilde{P}_i$ and $\tilde{P}_j$, and let $\delta \in (0,1)$. Let $\tilde{S}_c \sim \tilde{P}_c^{m_c}$ for $c = i, j$. Then, for large enough $m_i$ and $m_j$, with probability at least $1-\delta$ over the selection of $\tilde{S}$, we have*

$$V_f(\tilde{P}_i, \tilde{P}_j) \leq V_f(\tilde{S}_i, \tilde{S}_j) + 3(\mathrm{Var}_f(\tilde{P}_i) + \mathrm{Var}_f(\tilde{P}_j)) \cdot \frac{\epsilon_{f,1}(\tilde{P}_i, m_i, \delta/12)^2 + \epsilon_{f,1}(\tilde{P}_j, m_j, \delta/12)^2}{\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^4}$$
$$+ \mathrm{Avg}_{c\in\{i,j\}}\left[\frac{\|\mu_f(\tilde{S}_c)\| \cdot \epsilon_{f,1}(\tilde{P}_c, m_c, \delta/12)}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2} + \frac{\epsilon_{f,2}(\tilde{P}_c, m_c, \delta/12)}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2}\right]$$

The bound in the proposition has two parts. The first one is the empirical CDNV, $V_f(\tilde{S}_i, \tilde{S}_j)$, between the two datasets $\tilde{S}_i \sim \tilde{P}_i^{m_i}$ and $\tilde{S}_j \sim \tilde{P}_j^{m_j}$. This term is assumed to be small by the neural collapse phenomenon during training. The second part contains several terms proportional to the generalization gaps $\epsilon_{f,1}(\tilde{P}_c, m_c, \delta/12)$ and $\epsilon_{f,2}(\tilde{P}_c, m_c, \delta/12)$; as discussed above, typically we expect these terms to scale as $\mathcal{O}(1/\sqrt{m_c})$ (when fixing $\delta$) and to scale as $\mathcal{O}(\log(l/\delta))$ (when fixing $m_c$). Therefore, assuming we have a large enough training sets for a pair of source classes $i, j$, if the CDNV $V_f(\tilde{S}_i, \tilde{S}_j)$ of $f$ over the training samples is small, then we expect it to be small for new, unseen samples as well. That is, if neural collapse emerges in the training data of two source classes, we should also expect it to emerge in unseen samples of the same classes.

### 4.2 NEURAL COLLAPSE GENERALIZES TO NEW CLASSES

In the previous section we showed that if the CDNV is minimized by the learning algorithm on the training data, we expect it to be small for unseen source samples. As a next step, we show that if neural collapse emerges in the set of source classes, we can also expect it to emerge in new, unseen target classes.

To analyze this scenario, we essentially treat class-distributions as data points on which the feature map $f$ is trained (in a noisy manner, depending on the actual samples), and apply standard techniques to derive generalization bounds to new data points, which, in this case, are class-conditional distributions. Accordingly, we assume that both the source and the target classes come from a distribution over $\mathcal{D}_\mathcal{C}$ over a set of classes $\mathcal{C}$. Each class is represented by its class-conditional distribution, hence, with a slight abuse of notation, we say that the class-conditional distributions $\tilde{\mathcal{P}} = \{\tilde{P}_i\}_{i=1}^l$ in the source tasks are selected according to $\mathcal{D}_\mathcal{C}(\tilde{P}_1, \ldots, \tilde{P}_l \mid \tilde{P}_i \neq \tilde{P}_j \text{ for all } i \neq j \in [l])$. Below we show that if neural collapse emerges in the source classes, it also emerges in unseen source classes $c \neq c'$ with class-conditional distributions $P_c, P_{c'}$ sampled from $\mathcal{D}_\mathcal{C}(P_c, P_{c'} \mid P_c \neq P_{c'})$ in the sense that $V_f(P_c, P_{c'})$ is expected to be small. Our bound below applies a uniform convergence argument based on Gaussian complexities. The Gaussian complexity of a given set $Y \subset \mathbb{R}^n$ is defined as $G(Y) = \mathbb{E}_\gamma \sup_{y\in Y}[\langle\gamma, y\rangle]$, where $\gamma = (\gamma_1, \ldots, \gamma_n)$ is a vector of independent standard normal random variables. For a given feature map $f \in \mathcal{F}$, we also define a mapping

$H_f : \mathcal{C} \to \mathbb{R}^{p+1}$ as $H_f(P_c) = (\mu_f(P_c), \mathrm{Var}_f(P_c))$, and for a class of functions $\mathcal{F}^*$, we define $H_{\mathcal{F}^*}(\tilde{\mathcal{P}}) = \{(H_f(\tilde{P}_c))_{c=1}^l : f \in \mathcal{F}^*\}$.

**Proposition 2.** *Let $\mathcal{F}^* \subset \mathcal{F}$ be any finite set of functions and $\Delta(\mathcal{F}^*) = \inf_{f\in\mathcal{F}^*} \inf_{P_c \neq P_{c'} \in \mathcal{C}} \|\mu_f(P_c) - \mu_f(P_{c'})\|$. Then, under the settings of this section, if $P_c, P_{c'}$ are sampled from the distribution $\mathcal{D}_{\mathcal{C}}(P_c, P_{c'} \mid P_c \neq P_{c'})$,*

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}^*}\left[\mathbb{E}_{P_c\neq P_{c'}}[V_f(P_c,P_{c'})] - \mathrm{Avg}_{i\neq j}\left[V_f(\tilde{P}_i,\tilde{P}_j)\right]\right]\right] \leq \left(6 + \frac{24 \max_{f\in\mathcal{F}^*, P'\in\mathcal{C}} \mathrm{Var}_f(P')}{\Delta(\mathcal{F}^*)}\right) \cdot \frac{\mathbb{E}[G(H_{\mathcal{F}^*}(\tilde{\mathcal{P}}))]}{l \cdot \Delta(\mathcal{F}^*)^2}.$$

The above proposition, proved in Appendix D, provides an upper bound on the discrepancy between the expected value of $V_f(P_c, P_{c'})$ for two class-conditional distributions $P_c \neq P_{c'}$ and its averaged value across a set of source distributions $\tilde{P}_1, \ldots, \tilde{P}_l$ that were sampled from $\mathcal{D}_{\mathcal{C}}$. In addition, we treat $\mathcal{F}^*$ as a set of candidate functions from which the learning algorithm selects its candidates. Similarly to Theorem 2 of Maurer & Pontil (2019), this set is assumed to be finite only for technical (measurability) reasons. To extend the proposition to the case where $\mathcal{F}^*$ is infinite, we can simply replace $\sup_{f\in\mathcal{F}^*}$ with $\sup_{\mathcal{F}_0\subset\mathcal{F}^*,\ |\mathcal{F}_0|<\infty} \sup_{f\in\mathcal{F}_0}$.

In case of neural collapse, $\mathrm{Avg}_{i\neq j}\left[V_f(\tilde{P}_i, \tilde{P}_j)\right]$ is small, and hence for the feature map $f$ learned on the source tasks, $V_f(P_c, P_{c'})$ is small for two new classes $c$ and $c'$ if the main term in the bound, $\frac{\mathbb{E}[G(H_{\mathcal{F}^*}(\tilde{\mathcal{P}}))]}{l\cdot\Delta(\mathcal{F}^*)^3}$ is small (assuming the feature variances remain bounded for any class). According to Golowich et al. (2017); Bartlett et al. (2017), the Gaussian complexity term for neural networks scales with $\sqrt{l}$ (also, if the learning algorithm does not explore the whole function space, so the bound can be sharpened to include the complexity of the set of functions which can actually be learned from a given initialization, see, e.g., Allen-Zhu et al., 2019), so for the bound to go to zero as the number of source classes $l$ increases is to ensure that $\Delta(\mathcal{F}^*)$ is not too small (in particular, is not zero). While there is no way to guarantee this (e.g., if two class-conditional distributions are too close, $\Delta(\mathcal{F}^*)$ can be very small), we can say that if the feature map $f$ keeps the classes apart, the bound may become reasonably small. If the feature dimension $p$ is large and $f$ is similar to some random map, this assumption typically holds (e.g., if $f$ is not a constant function, which can be expected with a proper training method). Even when the number of classes is very large, the bound can be reasonable: if the corresponding feature-means are more-or-less uniformly distributed in a bounded set, and $l$ is a constant fraction of all the classes, $\Delta(\mathcal{F}^*) = \Theta(pl^{-2/p})$, and hence combining with the dependence of the Gaussian complexity on $l$, the bound becomes $O(l^{-1/2+6/p})$, which goes to zero as $l$ increases. The above considerations also suggest that having a large feature dimension $p$ is useful to enhance the generalization of neural collapse to new classes.

### 4.3 CDNV and Classification Error

In this section we study the relationship between the value of the CDNV and the classification performance it induces. For simplicity, consider first a two-class classification problem with distributions $P_1$ and $P_2$ for the two classes, and the oracle classifier $h^*(x) = \arg\min_{i\in\{1,2\}} \|f(x) - \mu_f(x)\|$ (note that $h^*$ is a linear classifier). Then if $x \sim P_1$, defining the projection of $f(x)$ onto the line connecting $\mu_f(P_1)$ and $\mu_f(P_2)$ by $z$, by Chebyshev's inequality, $\Pr[h^*(x) \neq 1] \leq \frac{4\mathbb{E}\left[\|z-\mu_f(P_1)\|^2\right]}{\|\mu_f(P_2)-\mu_f(P_1)\|^2}$ Then, since

$$\mathbb{E}[\|z - \mu_f(P_1)\|^2] \leq \mathrm{Var}_f(P_1) \tag{2}$$

by symmetry we obtain that if $x$ is equally likely to come from $P_1$ and $P_2$ with true label $c$, then the error probability of $h^*$ can be bounded as $\Pr[h^*(x) \neq c] \leq 4V_f(P_1, P_2)$. Thus, if $V_f(P_1, P_2)$ is small, the error probability is small. On the other hand, the inequality (2) can be quite pessimistic if the variance of $x$ does not concentrate in the direction connecting the class means. In fact, if $x$ is spherically symmetric, we have $\mathbb{E}[\|z - \mu_f(P_1)\|^2] = \mathrm{Var}_f(P_1)/p$, and hence the error probability can be bounded by $4V_f(P_1, P_2)/p$, which is a huge reduction if $p$ is large.

In practice, we do not know the class means, but given $n$ i.i.d. training samples $S_c$ from each class $c$, $\mu_f(S_c)$ is a good estimate of $\mu_f(P_c)$, with variance $\mathrm{Var}_f(P_c)/n$. Taking this into account, the error probability of the classifier $h(x) = \arg\min \|x - \mu_f(S_c)\|$ can easily be shown to be bounded by $16V_f(P_1, P_2)(1 + 2/n)$, or, in case of a spherically symmetric distributions within each class by

| Method | Architecture | Mini-ImageNet | CIFAR-FS | FC-100 |
|---|---|---|---|---|
| Matching Networks (Vinyals et al., 2016) | 64-64-64-64 | $55.31 \pm 0.73$ | - | - |
| LSTM Meta-Learner (Ravi & Larochelle, 2017) | 64-64-64-64 | $60.60 \pm 0.71$ | - | - |
| MAML (Finn et al., 2017) | 32-32-32-32 | $63.11 \pm 0.92$ | $71.5 \pm 1.0$ | - |
| Prototypical Networks (Snell et al., 2017) | 64-64-64-64 | $68.20 \pm 0.66^{\dagger}$ | $72.0 \pm 0.6$ | $48.6 \pm 0.6$ |
| Relation Networks (Sung et al., 2018) | 64-96-128-256 | $65.32 \pm 0.7$ | $69.3 \pm 0.8$ | - |
| SNAIL (Mishra et al., 2018) | ResNet-12 | $68.88 \pm 0.92$ | - | - |
| TADAM (Oreshkin et al., 2018) | ResNet-12 | $76.7 \pm 0.3$ | - | $56.1 \pm 0.4$ |
| AdaResNet (Munkhdalai et al., 2018) | ResNet-12 | $71.94 \pm 0.57$ | - | - |
| Dynamics Few-Shot (Gidaris & Komodakis, 2018) | 64-64-128-128 | $73.00 \pm 0.64$ | - | - |
| Activation to Parameter (Qiao et al., 2018) | WRN-28-10 | $73.74 \pm 0.19^{\dagger}$ | - | - |
| R2D2 (Bertinetto et al., 2019) | 96-192-384-512 | $68.8 \pm 0.1$ | $79.4 \pm 0.1$ | - |
| SARN (Hui et al., 2019) | ResNet-101 | $66.16 \pm 0.51$ | - | - |
| Shot-Free (Ravichandran et al., 2019) | ResNet-12 | $77.64 \pm n/a$ | $84.7 \pm n/a$ | - |
| TEWAM Qiao et al. (2019) | ResNet-12 | $75.90 \pm n/a$ | $81.3 \pm n/a$ | - |
| TPN (Liu et al., 2019) | ResNet-12 | $75.64 \pm n/a$ | - | - |
| LEO (Rusu et al., 2019) | WRN-28-10 | $77.59 \pm 0.12^{\dagger}$ | - | - |
| MTL (Sun et al., 2019) | ResNet-12 | $57.6 \pm 0.9$ | - | $57.6 \pm 0.9$ |
| OptNet-RR (Lee et al., 2019) | ResNet-12 | $77.88 \pm 0.46$ | $84.3 \pm 0.5$ | $55.3 \pm 0.6$ |
| MetaOptNet (Lee et al., 2019) | ResNet-12 | $78.63 \pm 0.46$ | $84.2 \pm 0.5$ | $55.5 \pm 0.6$ |
| Transductive Fine-Tuning (Dhillon et al., 2020) | WRN-28-10 | $78.40 \pm 0.52$ | $85.79 \pm 0.50$ | $57.57 \pm 0.55$ |
| Distill (Tian et al., 2020) | WRN-28-10 | $82.14 \pm 0.43$ | $86.9 \pm 0.5$ | $60.9 \pm 0.6$ |
| Ours | WRN-28-4 | $71.75 \pm 0.39$ | $82.06 \pm 0.94$ | $56.68 \pm 0.51$ |

Table 1: **Comparison to prior work on MiniImageNet, CIFAR-FS, and FC-100 on 5-shot 5-class classification.** The notation a-b-c-d denotes a 4-layer convolutional network with a, b, c, and d filters in each layer. $^{\dagger}$Results obtained when the meta-learning algorithm was trained on both train and validation classes. We took the data from (Dhillon et al., 2020) and (Tian et al., 2020).

$16V_f(P_1, P_2)(1 + 2/n)/p$. Therefore, in case of neural collapse (i.e., small $V_f(P_1, P_2)$), the error probability is small, explaining the success of foundation models in the low-data regime (such as in few-shot learning problems) in the presence of neural collapse.

## 5 Experiments

In this section we experimentally analyze the neural collapse phenomenon and how it generalizes to new data points and new classes. We use reasonably good (but typically not state of the art) classifiers to demonstrate that, in addition to the neural collapse observed in training time by Papyan et al. (2020), it is also observable on test data from the same classes, as well as on data from new classes, as predicted by our theoretical considerations in the previous section. We also show that, as expected intuitively, neural collapse is strongly correlated with accuracy in few-shot learning scenarios. The experiments are conducted over multiple datasets and multiple architectures (with good, but not necessarily state-of-the-art prediction performance), providing strong empirical evidence that neural collapse provides a compelling explanation for the good performance of foundation models in few-shot learning tasks.

### 5.1 Setup

**Method.** Following our theoretical setup, we first train a neural network classifier $\tilde{h} = \tilde{g} \circ f$ on a source task. Then we evaluate the few-shot performance of $f$ on target classification tasks by training a new classifier $h = g \circ f$. More specifically, $\tilde{h}$ is trained by minimizing the cross-entropy loss between the logits of the network and the one-hot encodings of the labels. The training is conducted using SGD with learning rate $\eta$ and momentum $0.9$ with batch size $64$. Here, $\tilde{g}$ is the top linear layer of the neural network and $f$ is the mapping implemented by all other layers. At the second stage, given a target few-shot classification task with training data $S = \{(x_i, y_i)\}_{i=1}^n$, we train a new top layer $g$ as a solution of ridge regression acting on the dataset $\{(f(x_i), y_i)\}_{i=1}^n$ with regularization $\lambda_n = \alpha\sqrt{n}$. Thus, $g$ is a linear transformation with the weight matrix

$$w_{S,f} = (f(X)^\top f(X) + \lambda_n I)^{-1} f(X)^\top Y, \tag{3}$$

where $f(X)$ is the $n \times d$ data matrix for the ridge regression problem containing the feature vectors $\{f(x_i)\}$ (i.e., $f(X)^\top = [f(x_1), \ldots, f(x_n)]$) and $Y$ is the $n \times k$ label matrix (i.e.,

$Y^\top = [y_1, \ldots, y_n]$) where $X \in \mathbb{R}^{n \times d}$, $Y \in \mathbb{R}^{n \times k}$ and $f_\theta(X) \in \mathbb{R}^{n \times p}$. In the experiments we sample 5-class classification tasks randomly from the target data set, with 5 training samples for each class (thus, altogether $n = 25$ above), and measure the performance on 100 random samples (non-overlapping with the training samples) from each class. We report the resulting accuracies averaged over 100 randomly chosen tasks.

**Datasets.** We consider four different datasets: (i) Mini-ImageNet (Vinyals et al., 2016), (ii) CIFAR-FS (Bertinetto et al., 2019), (iii) FC-100 (Oreshkin et al., 2018), and EMNIST (balanced) (Cohen et al., 2017). For a complete description of the datasets, see Appendix A.

**Architectures and hyperparameter.** We experimented with two types of architectures for $\tilde{h}$: wide ResNets (Zagoruyko & Komodakis, 2016) and vanilla convolutional networks of the same structure without the residual connections. The networks are denoted by WRN-$N$-$M$ and Conv-$N$-$M$, where $N$ is the depth and $M$ is the width factor. See appendix A for a complete description of the architectures. We used the following default hyperparameters: $\eta = 2^{-4}$, batch size $64$ and $\alpha = 1$ in setting the ridge regression regularization parameter $\lambda_n = \alpha/\sqrt{n}$. In the Figs. 2-3 in the appendix we provide experiments showing that the results are consistent for different $\eta$. More details on the architectures are given in Appendix A.

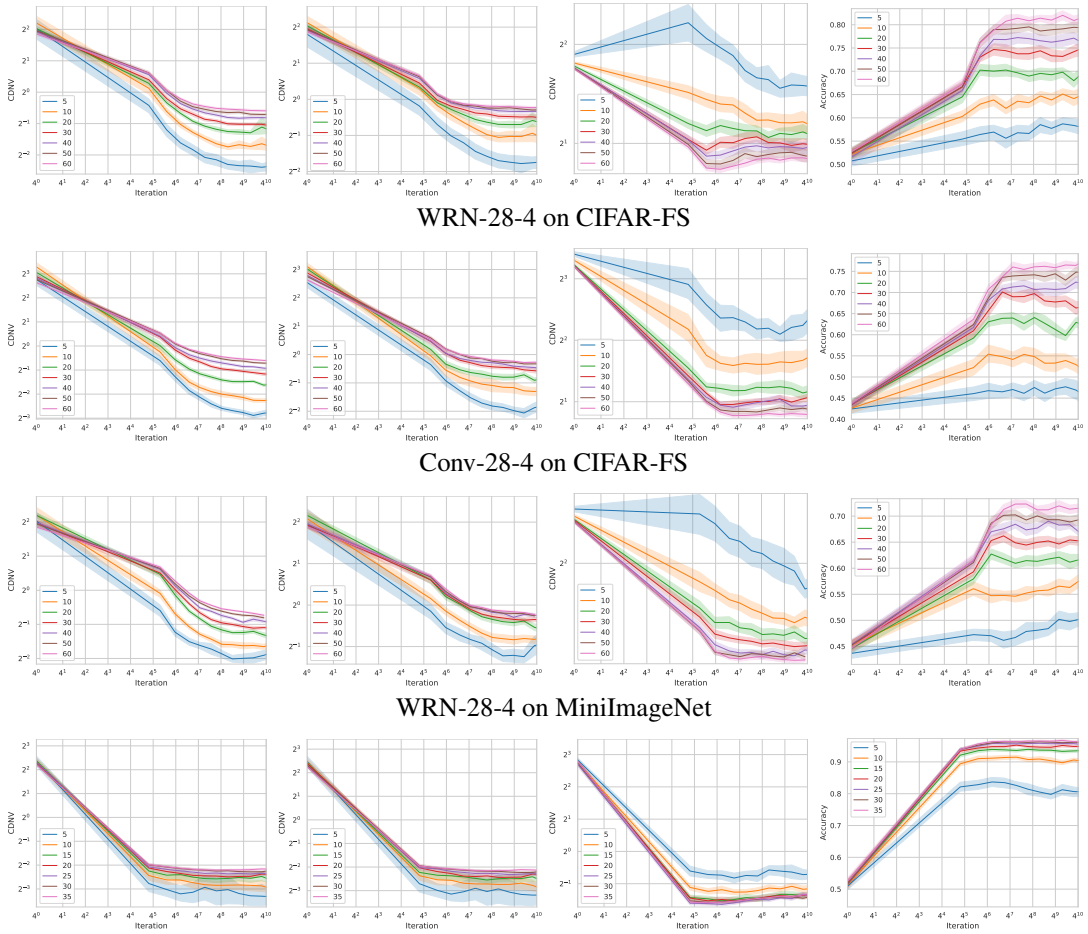
### 5.2 Results

While the ridge regression method to train the few-shot classifier $g$ may seem simplistic, it provides reasonably good performance, and hence it is suitable for studying the success of recent transfer/transductive learning methods (Dhillon et al., 2020; Tian et al., 2020) for few-shot learning. To demonstrate this, we compared the performance of our simplistic method to several few-shot learning algorithms on MiniImageNet, CIFAR-FS and FC-100, summarized in Table 1. On each dataset, we report the performance of our method when averaged across 20 random initializations and the last 10 epochs after 200 training epochs. As can be seen, the method we study in this paper is competitive with the rest of the literature on the three benchmarks; indeed, since we did not apply sophisticated initialization, scheduling schemes, or distillation procedures, the performance we observe is lower.

As our main experiment, we validate the theoretical assessments we made in Section 4. We argued that in the presence of neural collapse on the training classes, the trained feature map can be used for training a classifier with a small number of samples on new unseen classes. The argument asserts that if we observe neural collapse on a large set of source classes, then we expect to have neural collapse on new unseen classes as well, when assuming that the classes are selected in an i.i.d. manner. In this section we demonstrate that neural collapse generalizes to new samples from the same classes, and also to new classes, and we show that it is correlated with good few-shot performance.

To validate the above, we trained classifiers $\tilde{h}$ with a varying number of $l$ randomly selected source classes. For each run, we plot in Figure 1 the CDNV as a function of the epoch for the training and test datasets of the source classes and over the test samples of the target classes. In addition, we plot the few-shot accuracy of ridge regression using the learned feature map $f$. In each plot we report averaged values over 20 different initializations along with 95% confidence intervals.

As it can be seen in Figure 1, the value of the CDNV decreases over the course of training on the training and test datasets of the source classes, showing that neural collapse generalizes to new samples from the training classes. Since the classification tasks with fewer number of source classes are easier to learn, the CDNV tends to be larger when training with a wider set of source classes. In contrast, we observe that when increasing the number of source classes, the presence of neural collapse in the target classes strengthens. This is in alignment with our theoretical expectations (the more "training" classes, the better generalization to new classes), and the few-shot performance also consistently improves when the overall number of source classes is increased. To validate the generality of our results, this phenomena is demonstrated in several settings. Specifically, in rows 1-2 of Figure 1 we report the results of WRN-28-4 and Conv-28-4, respectively, on CIFAR-FS, in row 3 we report the results of WRN-28-4 on MiniImageNet, and in row 4 we report the results of Conv-16-2 on EMNIST. We chose to experiment with a smaller architecture on EMNIST since it is an easier dataset. As can be seen, the values of the CDNV on the target classes are relatively large, in comparison with the ones corresponding to the source, except for the results on EMNIST. However, since the dimension of $f$ is $p = 256$, by the analysis in Section 4.3 these values still indicate




**(a)** train CDNV **(b)** test CDNV **(c)** target CDNV **(d)** target accuracy

Figure 1: **Within-class variation collapse.** In **(a)** we plot the CDNV on the source training data, in **(b)** we plot the CDNV over the source test data and in **(c)** the CDNV over the target classes (log-log scale). In **(d)** we plot the target accuracy rate (lin-log scale). In each experiment we trained the model using SGD on a set of $l$ source classes (see legend).

for reasonable few-shot learning performance. These results consistently validate our theoretical findings, that is, that neural collapse generalizes to new source samples, it emerges for new classes, and its presence immediately facilitates good performance in few-shot learning.

## 6 Conclusions

Employing foundation models for transfer learning is a successful approach for dealing with overfitting in the low-data regime. However, the reasons for this success are not clear. In this paper we presented a new perspective on this problem by connecting it to the newly discovered phenomenon of neural collapse. We showed that the within-class variance collapse tends to emerge in the test data associated with the classes encountered at train time and, more importantly, in new unseen classes when the new classes are drawn from the same distribution as the training classes. In addition, we showed that when neural collapse emerges in the new classes, then it requires very few samples to train a linear classifier on top of the learned feature representation that accurately predicts the new classes. These results provide a justification to the recent successes of transfer learning in few-shot tasks, as observed by Tian et al., 2020; Dhillon et al., 2020.

## REFERENCES

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6241–6250, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Jonathan Baxter. A model of inductive bias learning. *J. Artif. Int. Res.*, 12(1):149–198, March 2000.

Shai Ben-david, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 137–144, 2006.

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pp. 17–36, Bellevue, Washington, USA, 02 Jul 2012. PMLR.

Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Rich Caruana. Learning many related tasks at the same time with backpropagation. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1995.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV (7)*, volume 11211 of *Lecture Notes in Computer Science*, pp. 833–851. Springer, 2018.

Sihong Chen, Kai Ma, and Yefeng Zheng. Med3d: Transfer learning for 3d medical image analysis. *CoRR*, abs/1904.00625, 2019b.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations*, 2020.

Simon Shaolei Du, Wei Hu, Sham M. Kakade, Jason D. Lee, and Qi Lei. Few-shot learning via learning the representation, provably. In *International Conference on Learning Representations*, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 06–11 Aug 2017.

Tomer Galanti, Lior Wolf, and Tamir Hazan. A theoretical framework for deep transfer learning. *Information and Inference: A Journal of the IMA*, 5:159–209, 2016.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *Information and Inference: A Journal of the IMA*, 9, 12 2017. doi: 10.1093/imaiai/iaz007.

X. Y. Han, Vardan Papyan, and David L. Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

Zixuan Huang and Yin Li. Interpretable and accurate fine-grained recognition via region grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8662–8672, 2020.

Binyuan Hui, Pengfei Zhu, Qinghua Hu, and Qilong Wang. Self-attention relation network for few-shot learning. In *2019 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 198–203, 2019.

Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 05 2012.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sungju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations*, 2019.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015. doi: 10.1109/CVPR.2015.7298965.

Yishay Mansour. Learning and domain adaptation. In *Algorithmic Learning Theory, 20th International Conference, ALT*, pp. 4–6, 2009.

Andreas Maurer and M. Pontil. Uniform concentration and symmetrization for weak interactions. In *COLT*, 2019.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *J. Mach. Learn. Res.*, 17(1):2853–2884, January 2016.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Dustin G. Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features, 2020.

Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3664–3673. PMLR, 2018.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

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

Anastasia Pentina and Christoph Lampert. A pac-bayesian bound for lifelong learning. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 991–999, Bejing, China, 22–24 Jun 2014. PMLR.

T. Poggio and Q. Liao. Implicit dynamic regularization in deep networks. Technical report, Center for Brains, Minds and Machines (CBMM), 2020a.

Tomaso Poggio and Qianli Liao. Explicit regularization and implicit bias in deep network classifiers trained with the square loss, 2020b.

Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L. Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021.

S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 331–339, 2019.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: A good embedding is all you need? In *CProceedings of the European Conference on Computer Vision (ECCV)*, volume 12359 of *Lecture Notes in Computer Science*, pp. 266–282. Springer, 2020.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Sergey Zagoruyko and N. Komodakis. Wide residual networks. *ArXiv*, abs/1605.07146, 2016.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features, 2021.

# A EXPERIMENTAL DETAILS

**Datasets.** Throughout the experiments, we consider four different datasets: (i) Mini-ImageNet (Vinyals et al., 2016) and (ii) CIFAR-FS (Bertinetto et al., 2019), (iii) FC-100 (Oreshkin et al., 2018) and EMNIST (balanced) (Cohen et al., 2017). Each dataset is split into meta-train, meta-validation and meta-test classes; we select the data for the source classes from the meta-training, and use similarly the meta-test data for the target tasks (we do not use the meta-validation classes). Each one of the class splits is also partitioned into train and test samples; we use these for training and evaluating our models. The Mini-ImageNet dataset contains 100 classes randomly chosen from ImageNet ILSVRC-2012 (Russakovsky et al., 2015) with 600 images of size $84 \times 84$ pixels per class. It is split into 64 meta-training classes, 16 meta-validation classes and 20 classes for meta-testing. CIFAR-FS and FC-100 are two derivatives of the CIFAR-100 dataset (Krizhevsky, 2012). CIFAR-FS consists of a random split of the CIFAR-100 classes into 64 classes for meta-training, 14 for meta-validation and 20 for meta-testing. FC-100 contains 100 classes which are grouped into 20 superclasses. These classes are partitioned into 60 meta-train classes from 12 superclasses, 20 classes from 4 superclasses for meta-validation, and 20 meta-test classes from 4 superclasses. We also consider the EMNIST dataset, which is an extension of the original MNIST dataset. We randomly split its classes into 35 source classes and 12 target classes (which are: 2, 10, 11, 12, 13, 16, 18, 22, 25, 33, 34, 44). We use random cropping augmentations at training time across all of the experiments.

**Architectures** We experimented with two types of architectures for $\tilde{h}$: wide ResNets (Zagoruyko & Komodakis, 2016) and vanilla convolutional networks. Wide ResNets start with a convolutional layer (with kernels $3 \times 3$ and 16 output channels), followed by three groups of layers. Each group of layers includes a convolutional layer, followed by a sequence of $N$ residual layers. Each residual layer in the $i$'th group contains two convolutional layers using $3 \times 3$ kernels and output $2^{i+3}M$ channels. Each convolutional layer is followed by a ReLU activations and batch normalization post activations. The network's penultimate layer is a mean pooling activation with kernels of size $4 \times 4$, returning an output of dimension $256$ and followed by a linear layer. The vanilla convolutional networks have the same architectures as the wide ResNets, except that we omit the residual connections. The networks are denoted by WRN-$N$-$M$ and Conv-$N$-$M$, respectively.

# B ADDITIONAL EXPERIMENTS

In this section we report additional experiments to provide further insights.

## B.1 VARYING HYPERPARAMETERS

We start with presenting experimental results to validate the robustness of our findings when varying different hyperparameters. In the first experiment we repeated the experiments in rows 1 and 4 of Figure 1 with learning rates $\eta = 2^{-2i-2}$ for $i = 1, 2, 3, 4$. We also report the train and test accuracy rates on the source data. The results are reported in Figure 2 and Figure 3. The results are consistent with the observations we had in Figure 1. Interestingly, when using smaller learning rates, the CDNV over the source train and test data tends to be smaller, while the CDNV on the target classes tends to be larger. We attribute this observation to overfitting to the source classes and indeed we observe that the few-shot performance is generally worse in those cases.

## B.2 CLASS-COVARIANCE NORMALIZED VARIANCE

In this section we consider the formal definition of Papyan et al. (2020) for neural collapse (their first definition), which uses a normalized variance definition somewhat different from CDNV we used in our analysis. For completeness, we present some experiments which demonstrate that our findings based on CDNV also apply to their normalization.

To give the formal definition, consider a training set $\tilde{S} = \bigcup_{c=1}^{l} \tilde{S}_c = \bigcup_{c=1}^{l} \{(\tilde{x}_{cj}, c)\}_{j=1}^{m_c}$ for an $l$-class classification problem, where $\tilde{S}_c$ is a collection of $m_c$ samples from class $c$. Assume that a neural network is trained to classify the samples in $\tilde{S}$, where after $t$ steps of the training we obtain the classifier $\tilde{h}_t = \tilde{g}_t \circ f_t$, where, as before, $\tilde{g}_t$ denotes the linear map of the last layer of the network, and $f_t$ is the feature map.

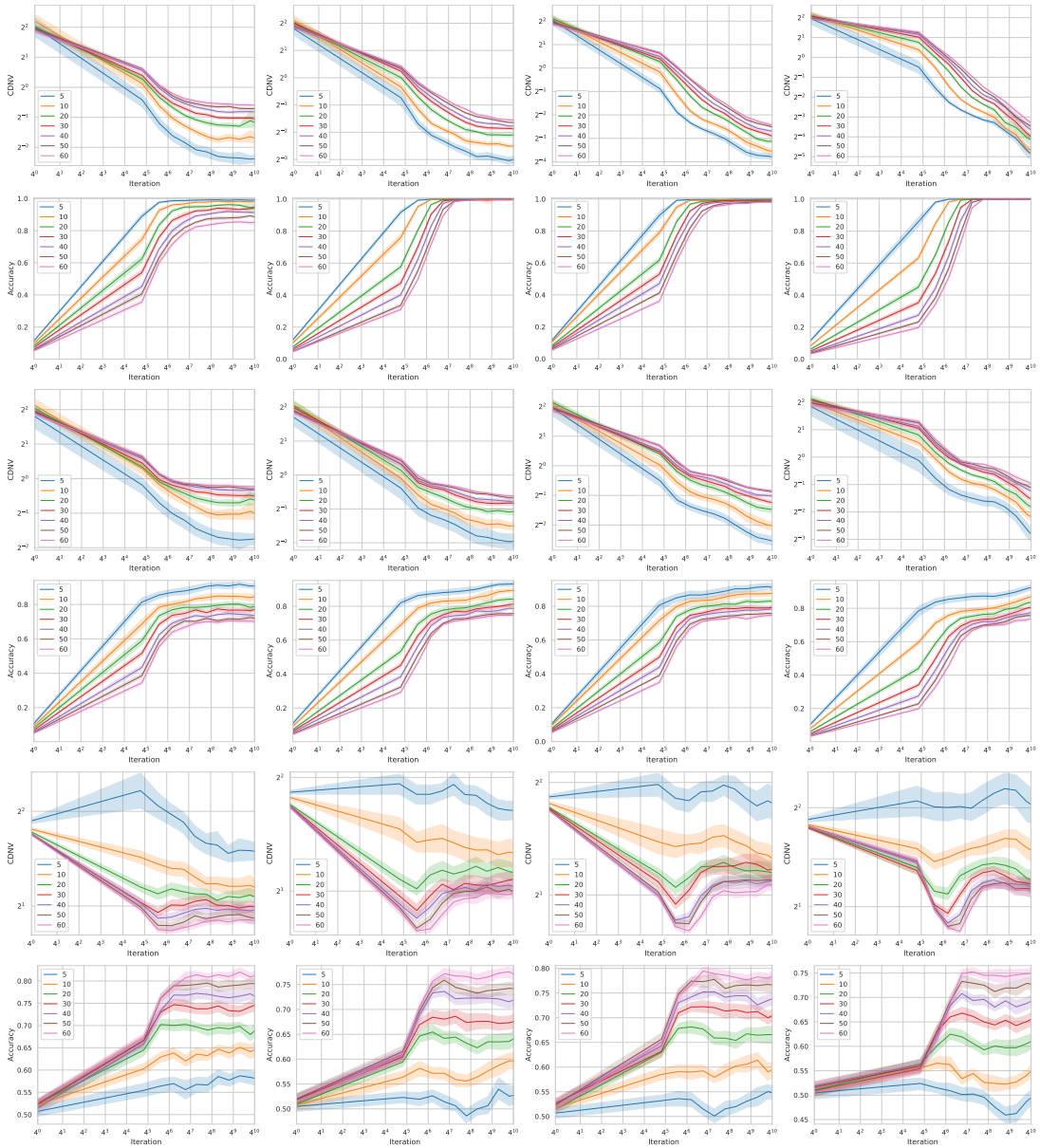



Figure 2: **Averaged class variance and accuracy rates when varying the number of source classes on CIFAR100 FS split.** In **(rows 1-2)** We plot the CDNV and accuracy rates on the source training data (resp.), in **(row 3-4)** on the source test data and in **(row 5-6)** on the target test data. In each experiment we trained a WRN-28-4 with SGD on a set of $l = 5, 10, 20 \ldots, 60$ source classes (see legend). The $i$'th column stands for learning rate $\eta = 2^{-2i-2}$.

**Class-Covariance Normalized Variance.** Papyan et al. (2020) define the *class-covariance normalized variance* (CCNV) to define neural collapse: For a given feature map $f$, let $\tilde{\mu}_c^f = \mu_f(\tilde{S}_c)$ denote the mean feature value of class $c$, and let $\tilde{\mu}_G^f = \mathrm{Avg}_{c=1}^{l}[\tilde{\mu}_c^f]$ denote the global feature mean. The CCNV is defined to be $\mathrm{Tr}\left(\Sigma_W^f {\Sigma_B^f}^{\dagger}\right)$, where $\Sigma_W^f$ and $\Sigma_B^f$ are the intra- and inter-class covariance matrices $\Sigma_W^f = \mathrm{Avg}_{\substack{c\in[l] \\ j\in[m_c]}}\left[(f(\tilde{x}_{cj}) - \tilde{\mu}_c^f)\cdot(f(\tilde{x}_{cj}) - \tilde{\mu}_c^f)^\top\right]$ and $\Sigma_B^f = \mathrm{Avg}_{c\in[l]}\left[(\tilde{\mu}_c^f - \tilde{\mu}_G^f)\cdot(\tilde{\mu}_c^f - \tilde{\mu}_G^f)^\top\right]$, where $A^\dagger$ stands for the Moore-Penrose inverse of a (square)

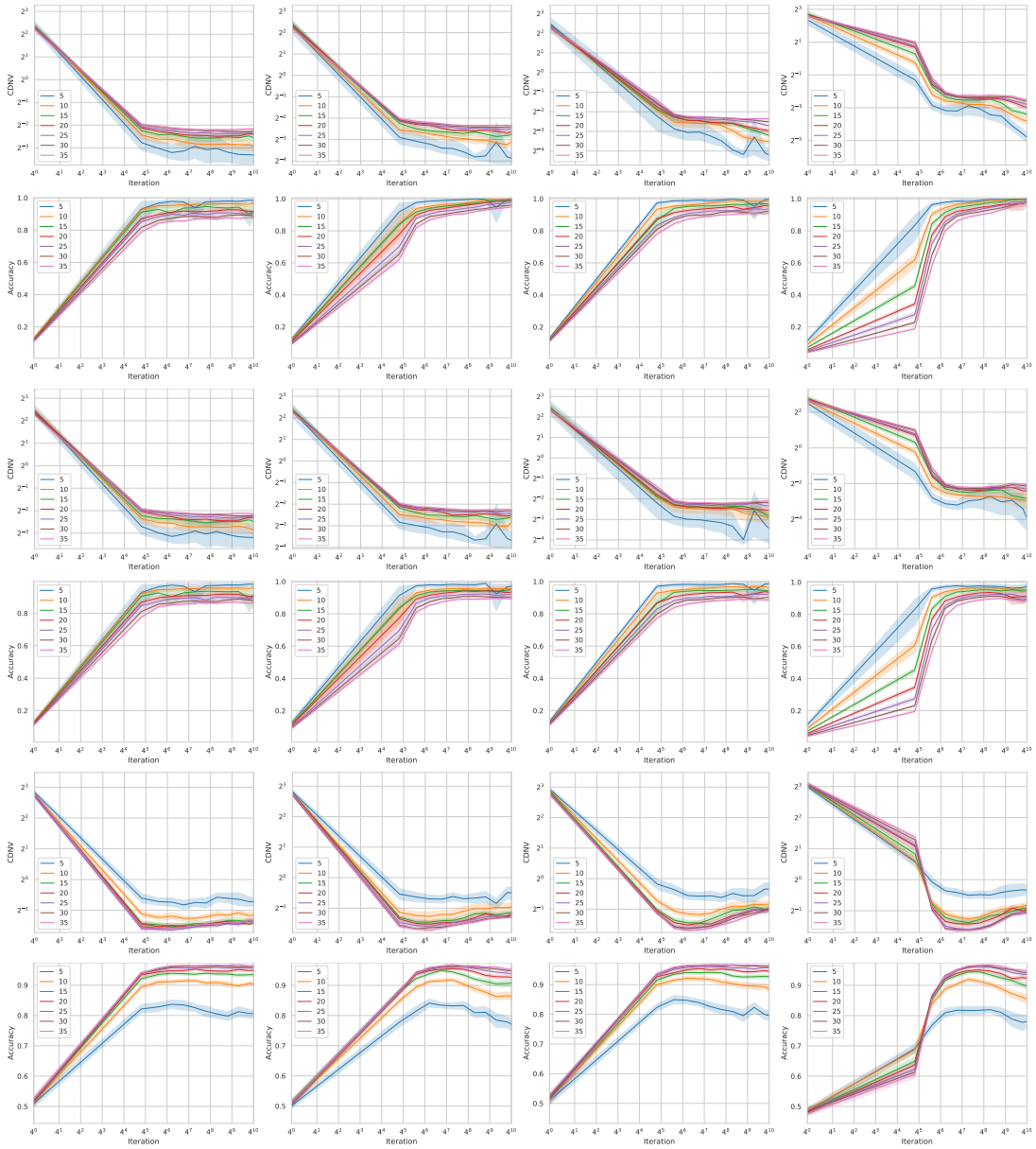



Figure 3: **Averaged class variance and accuracy rates when varying the number of source classes on EMNIST.** In **(rows 1-2)** We plot the CDNV and accuracy rates on the source training data (resp.), in **(row 3-4)** on the source test data and in **(row 5-6)** on the target test data. In each experiment we trained a WRN-28-4 with SGD on a set of $l = 5, 10, 20 \ldots, 60$ source classes (see legend). The $i$'th column stands for the results when training with learning rate $\eta = 2^{-2i-2}$.

matrix $A$. According to Papyan et al. (2020), (their first version of) neural collapse happens if $\lim_{t\to\infty} \mathrm{Tr}\left(\Sigma_W^{f_t} {\Sigma_B^{f_t}}^{\dagger}\right) = 0$.

In Figures 4 and 5 we plot the CCNV as a function of training iterations on the source training data, source test data and target classes. As can be seen, the value of the CCNV decreases on the train and test data of the source classes. In addition, the CCNV on the target classes decreases when training with a larger set of source classes, which is due to better generalization. In contrast, the CCNV on the source classes increases when training with a larger set of source classes, since it increases the complexity of the optimization problem. These results are qualitatively very similar to the once

presented for CDNV, showing that both definitions of neural collapse behaves essentially the same way. The corresponding train/test/few-shot accuracy rates are provided in Figures 2-3.

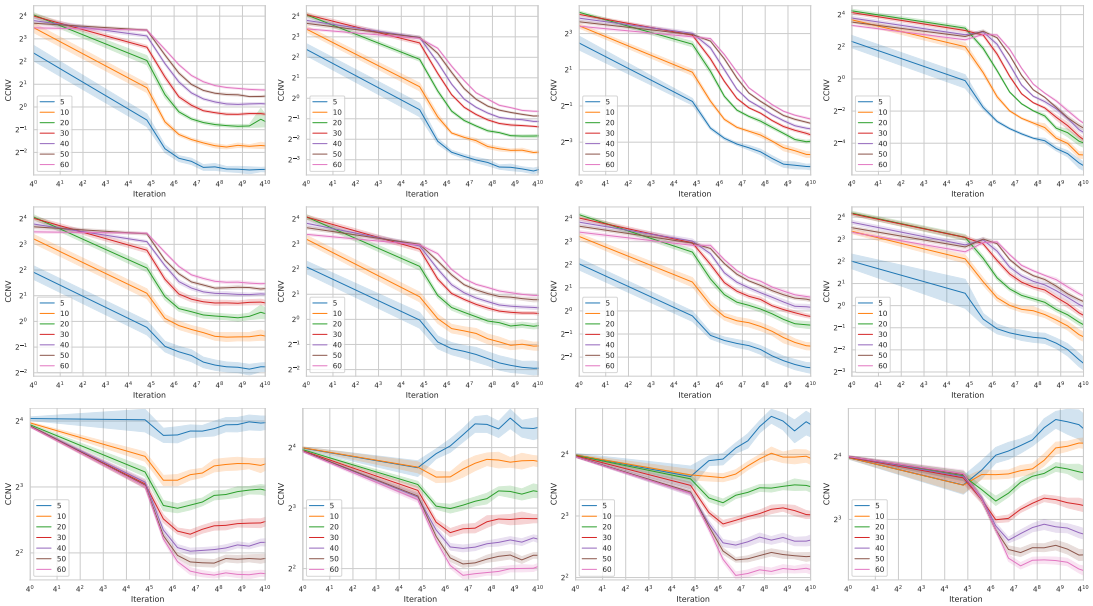



Figure 4: **Within-class variation collapse on CIFAR-FS.** In **(rows 1)** We plot the CCNV on the source training data, in **(row 2)** on the source test data and in **(row 3)** on the target test data. In each experiment we trained a WRN-28-4 using SGD on a set of $l \in \{5, 10, 20, 30, 40, 50, 60\}$ source classes (see legend). The $i$'th column stands for the results when training with learning rate $\mu = 2^{-2i-2}$.

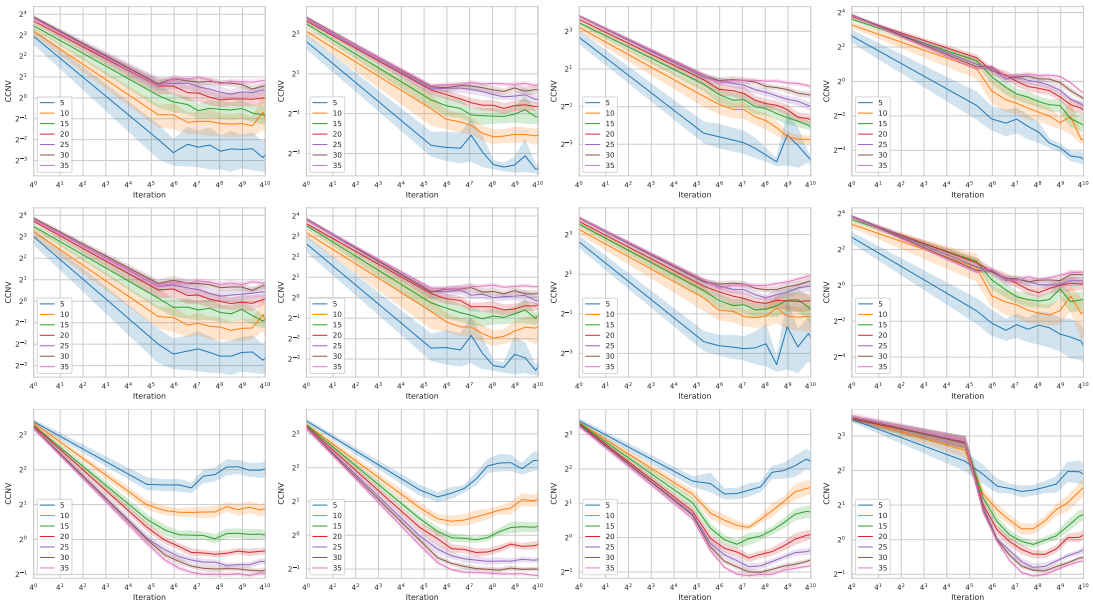



Figure 5: **Within-class variation collapse on EMNIST.** In **(rows 1)** We plot the CCNV on the source training data, in **(row 2)** on the source test data and in **(row 3)** on the target test data. In each experiment we trained a WRN-28-4 using SGD on a set of $l \in \{5, 10, 15, 20, 25, 30, 35\}$ source classes (see legend). The $i$'th column stands for learning rate $\mu = 2^{-2i-2}$.

## C PROOF OF PROPOSITION 1

**Lemma 1.** *Let $\tilde{P}_c$ be a class distribution, $\delta \in (0,1)$ and $\tilde{S}_c \sim \tilde{P}_c^{m_c}$ be a dataset of samples. Let $f$ be the output of the learning algorithm. Then, with probability at least $1-\delta$, we have*

$$\mathrm{Var}_f(\tilde{P}_c) \leq \mathrm{Var}_f(\tilde{S}_c) + \epsilon_{f,2}(\tilde{P}_c, m_c, \delta/2) + 2\,\|\tilde{\mu}_c\| \cdot \epsilon_{f,1}(\tilde{P}_c, m_c, \delta/2) \tag{4}$$

*Proof.* Recall that

$$\mathrm{Var}_f(\tilde{P}_c) = \mathbb{E}_{x\sim\tilde{P}_c}[\|f(x)\|^2] - \|\mathbb{E}_{x\sim\tilde{P}_c}[f(x)]\|^2 \,. \tag{5}$$

In addition, for any distribution $\tilde{P}_c$, with probability at least $1-\delta/2$ over the selection of $\tilde{S}_c$, we have

$$\mathbb{E}_{x\sim\tilde{P}_c}[\|f(x)\|^2] \leq \mathrm{Avg}_{j=1}^{m_c}[\|f(\tilde{x}_{cj})\|^2] + \epsilon_{f,2}(\tilde{P}_c, m_c, \delta/2) \,. \tag{6}$$

By the reversed triangle inequality,

$$\left| \|\mathbb{E}_{x\sim\tilde{P}_c}[f(x)]\| - \|\,\mathrm{Avg}_{j=1}^{m_c}[f(\tilde{x}_{cj})]\| \right| \leq \left\| \mathbb{E}_{x\sim\tilde{P}_c}[f(x)] - \mathrm{Avg}_{j=1}^{m_c}[f(\tilde{x}_{cj})] \right\| \,. \tag{7}$$

Hence, for any distribution $\tilde{P}_c$, with probability at least $1-\delta/2$ over the selection of $\tilde{S}_c$, we have:

$$\|\mathbb{E}_{x\sim\tilde{P}_c}[f(x)]\| \geq \|\,\mathrm{Avg}_{j=1}^{m_c}[f(\tilde{x}_{cj})]\| - \epsilon_{f,1}(\tilde{P}_c, m_c, \delta/2) \tag{8}$$

In particular, we have:

$$\left\|\mathbb{E}_{x\sim\tilde{P}_c}[f(x)]\right\|^2 \geq \left\|\mathrm{Avg}_{j=1}^{m_c}[f(\tilde{x}_{cj})]\right\|^2 - 2\left\|\mathrm{Avg}_{j=1}^{m_c}[f(\tilde{x}_{cj})]\right\| \cdot \epsilon_{f,1}(\tilde{P}_c, m_c, \delta/2) \tag{9}$$

By union bound over Equations (6) with (9), we obtain that with probability at least $1-\delta$, we have:

$$\begin{aligned} \mathrm{Var}_f(\tilde{P}_c) \leq& \mathrm{Var}_f(\tilde{S}_c) + \epsilon_{f,2}(\tilde{P}_c, m_c, \delta/2) + 2\left\|\mathrm{Avg}_{j=1}^{m_c}[f(\tilde{x}_{cj})]\right\| \cdot \epsilon_{f,1}(\tilde{P}_c, m_c, \delta/2) \\ =& \mathrm{Var}_f(\tilde{S}_c) + \epsilon_{f,2}(\tilde{P}_c, m_c, \delta/2) + 2\,\|\tilde{\mu}_c\| \cdot \epsilon_{f,1}(\tilde{P}_c, m_c, \delta/2) \end{aligned} \tag{10}$$

□

**Lemma 2.** *Let $\tilde{P}_i$ and $\tilde{P}_j$ be two class distributions, $\delta \in (0,1)$ and $\tilde{S}_c \sim \tilde{P}_c^{m_c}$ (for $c = i, j$). Let $\Delta = (\mu_f(\tilde{S}_i) - \mu_f(\tilde{P}_i)) + (\mu_f(\tilde{P}_j) - \mu_f(\tilde{S}_j))$. Then, for large enough $m_i, m_j$, with probability at least $1-\delta$, we have:*

$$\|\Delta\| \cdot \|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\| \leq \frac{9\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2}{20} \tag{11}$$

*Proof.* By simple algebraic manipulations, we can write:

$$\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2 = \|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^2 - 2\Delta^\top \cdot (\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)) + \|\Delta\|^2 \tag{12}$$

In particular,

$$\left| \|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2 - \|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^2 \right| \leq 2\|\Delta\| \cdot \|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\| + \|\Delta\|^2 \tag{13}$$

By the triangle inequality:

$$\|\Delta\| \leq \|\mu_f(\tilde{S}_i) - \mu_f(\tilde{P}_i)\| + \|\mu_f(\tilde{S}_j) - \mu_f(\tilde{P}_j)\| \tag{14}$$

We notice that with probability at least $1-\delta/2$, we have:

$$\|\mu_f(\tilde{S}_c) - \mu_f(\tilde{P}_c)\| \leq \epsilon_{f,1}(\tilde{P}_c, m_c, \delta/2) \tag{15}$$

In addition, for any $c = 1, 2$ and $\delta \in (0,1)$, we have,

$$\lim_{m_c\to\infty} \epsilon_{f,1}(\tilde{P}_c, m_c, \delta/2) = 0 \tag{16}$$

Therefore, for any $\delta \in (0,1)$, for large enough $m_i, m_j$, we can ensure that $\|\Delta\| \cdot \|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|$ is as small as we wish and $\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2$ is as close to $\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^2 > C_0^2$ as we wish, which gives the desired inequality. □

**Proposition 1.** *Fix two source classes, $i$ and $j$ with distributions $\tilde{P}_i$ and $\tilde{P}_j$, and let $\delta \in (0,1)$. Let $\tilde{S}_c \sim \tilde{P}_c^{m_c}$ for $c = i, j$. Then, for large enough $m_i$ and $m_j$, with probability at least $1-\delta$ over the selection of $\tilde{S}$, we have*

$$V_f(\tilde{P}_i, \tilde{P}_j) \leq V_f(\tilde{S}_i, \tilde{S}_j) + 3(\mathrm{Var}_f(\tilde{P}_i) + \mathrm{Var}_f(\tilde{P}_j)) \cdot \frac{\epsilon_{f,1}(\tilde{P}_i, m_i, \delta/12)^2 + \epsilon_{f,1}(\tilde{P}_j, m_j, \delta/12)^2}{\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^4}$$
$$+ \mathrm{Avg}_{c\in\{i,j\}} \left[ \frac{\|\mu_f(\tilde{S}_c)\| \cdot \epsilon_{f,1}(\tilde{P}_c, m_c, \delta/12)}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2} + \frac{\epsilon_{f,2}(\tilde{P}_c, m_c, \delta/12)}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2} \right]$$

*Proof.* By Equation (13), we have:

$$\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^2 \geq \|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2 - 2\|\Delta\| \cdot \|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\| \tag{17}$$

By Lem. 2, for large enough $m_i, m_j$, with probability at least $1-\delta/3$, we have:

$$2\|\Delta\| \cdot \|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\| \leq \frac{9\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2}{10}, \tag{18}$$

where $\Delta$ is defined in Lem. 2. Hence, by taking $a = \|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2$ and $u = 2\|\Delta\| \cdot \|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^2$, since $(a-u)^{-1} \leq a^{-1} + 10a^{-1}u^2$ for every $a > 0$ and $u < 9a/10$, we obtain that:

$$\begin{aligned}
\frac{1}{\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^2} \leq& \frac{1}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2 - 2\|\Delta\| \cdot \|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|} \\
\leq& \frac{1}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2} + \frac{10\|\Delta\|^2}{4\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^4} \\
\leq& \frac{1}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2} + \frac{3\|\Delta\|^2}{\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^4}
\end{aligned} \tag{19}$$

In addition, by Lem. 1, with probability at least $1-\delta/3$, we have:

$$\mathrm{Var}_{\tilde{P}_i}(f) \leq \mathrm{Var}_f(\tilde{S}_i) + \epsilon_{f,2}(\tilde{P}_i, m_i, \delta/6) + 2\left\|\mu_f(\tilde{S}_i)\right\| \cdot \epsilon_{f,1}(\tilde{P}_i, m_i, \delta/6) \tag{20}$$

Therefore, by applying union bound, with probability at least $1-\delta$, Equations (19) and (20) hold for $c = i, j$. Hence, we can combine all of these derivations altogether to obtain the following inequality:

$$\begin{aligned}
\frac{\mathrm{Var}_f(\tilde{P}_i)}{\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^2} \leq& \frac{\mathrm{Var}_f(\tilde{P}_i)}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2} + \frac{3\mathrm{Var}_f(\tilde{P}_i) \cdot \|\Delta\|^2}{\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^4} \\
\leq& \frac{\mathrm{Var}_f(\tilde{S}_i)}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2} + \frac{\epsilon_{f,2}(\tilde{P}_i, m_i, \delta/6)}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2} \\
&+ \frac{2\left\|\mu_f(\tilde{S}_i)\right\| \cdot \epsilon_{f,1}(\tilde{P}_i, m_i, \delta/6)}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2} + \frac{3\mathrm{Var}_{\tilde{P}_i}(f) \cdot \|\Delta\|^2}{\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^4}
\end{aligned} \tag{21}$$

We recall that $(a+b)^2 \leq 2a^2 + 2b^2$. Hence,

$$\|\Delta\|^2 \leq 2\epsilon_{f,1}(\tilde{P}_i, m_i, \delta/6)^2 + 2\epsilon_{f,1}(\tilde{P}_j, m_j, \delta/6)^2 \tag{22}$$

Therefore, for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have:

$$\begin{aligned}
\frac{\mathrm{Var}_f(\tilde{P}_i)}{\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^2} \leq& V_f(\tilde{S}_i, \tilde{S}_j) + 6\mathrm{Var}_f(\tilde{P}_i) \cdot \frac{\epsilon_{f,1}(\tilde{P}_i, m_i, \delta/6)^2 + \epsilon_{f,1}(\tilde{P}_j, m_j, \delta/6)^2}{\|\mu_f(\tilde{P}_i) - \mu_f(\tilde{P}_j)\|^4} \\
&+ \frac{2\left\|\mu_f(\tilde{S}_i)\right\| \cdot \epsilon_{f,1}(\tilde{P}_i, m_i, \delta/6)}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2} + \frac{\epsilon_{f,2}(\tilde{P}_i, m_i, \delta/6)}{\|\mu_f(\tilde{S}_i) - \mu_f(\tilde{S}_j)\|^2},
\end{aligned} \tag{23}$$

Hence, by union bound, with probability at least $1-\delta$, we have:

$$\begin{aligned}
V_f(\tilde{P}_i,\tilde{P}_j) =& \frac{\mathrm{Var}_f(\tilde{P}_i)}{2\|\mu_f(\tilde{P}_i)-\mu_f(\tilde{P}_j)\|^2} + \frac{\mathrm{Var}_f(\tilde{P}_j)}{2\|\mu_f(\tilde{P}_i)-\mu_f(\tilde{P}_j)\|^2} \\
\leq& V_f(\tilde{S}_i,\tilde{S}_j) + 3(\mathrm{Var}_f(\tilde{P}_i)+\mathrm{Var}_f(\tilde{P}_j)) \cdot \frac{\epsilon_{f,1}(\tilde{P}_i,m_i,\delta/12)^2+\epsilon_{f,1}(\tilde{P}_j,m_j,\delta/12)^2}{\|\mu_f(\tilde{P}_i)-\mu_f(\tilde{P}_j)\|^4} \\
&+ \frac{\left\|\mu_f(\tilde{S}_i)\right\| \cdot \epsilon_{f,1}(\tilde{P}_i,m_i,\delta/12)}{\|\mu_f(\tilde{S}_i)-\mu_f(\tilde{S}_j)\|^2} + \frac{\epsilon_{f,2}(\tilde{P}_i,m_i,\delta/12)}{\|\mu_f(\tilde{S}_i)-\mu_f(\tilde{S}_j)\|^2} \\
&+ \frac{\left\|\mu_f(\tilde{S}_j)\right\| \cdot \epsilon_{f,1}(\tilde{P}_j,m_j,\delta/12)}{\|\mu_f(\tilde{S}_i)-\mu_f(\tilde{S}_j)\|^2} + \frac{\epsilon_{f,2}(\tilde{P}_j,m_j,\delta/12)}{\|\mu_f(\tilde{S}_i)-\mu_f(\tilde{S}_j)\|^2}
\end{aligned} \tag{24}$$

□

## D PROOF OF PROPOSITIONS 2

**Proposition 2.** *Let $\mathcal{F}^* \subset \mathcal{F}$ be any finite set of functions and $\Delta(\mathcal{F}^*) = \inf_{f\in\mathcal{F}^*}\inf_{P_c\neq P_{c'}\in\mathcal{C}}\|\mu_f(P_c)-\mu_f(P_{c'})\|$. Then, under the settings of this section, if $P_c, P_{c'}$ are sampled from the distribution $\mathcal{D}_{\mathcal{C}}(P_c, P_{c'} \mid P_c \neq P_{c'})$,*

$$\mathbb{E}\Big[\sup_{f\in\mathcal{F}^*}\Big[\mathbb{E}_{P_c\neq P_{c'}}[V_f(P_c,P_{c'})] - \mathrm{Avg}_{i\neq j}\Big[V_f(\tilde{P}_i,\tilde{P}_j)\Big]\Big]\Big] \leq \left(6+\frac{24\max_{f\in\mathcal{F}^*,P'\in\mathcal{C}}\mathrm{Var}_f(P')}{\Delta(\mathcal{F}^*)}\right)\cdot\frac{\mathbb{E}[G(H_{\mathcal{F}^*}(\tilde{\mathcal{P}}))]}{l\cdot\Delta(\mathcal{F}^*)^2}.$$

To prove this proposition, we apply Theorem 2 of Maurer & Pontil (2019). For clarity we restate their Definition 1 with our terminology.

**Definition 1** (Maurer & Pontil, 2019)**.** *Let $\mathcal{U}$ be any set and $A : \mathcal{U}^l \to \mathbb{R}$. For $\mathbf{z} = (z_1,\dots,z_l)$, $j \in \{1,\dots,l\}$ and $z, z' \in \mathcal{U}$, we define the $j$-th partial difference operator on a function $A$ as:*

$$D^j_{z_j,z'_j}A(\mathbf{z}) = A(\dots,z_{j-1},z_j,z_{j+1}\dots) - A(\dots,z_{j-1},z'_j,z_{j+1},\dots) \tag{25}$$

*In addition, we define the Lipschitzness of $A$ as follows:*

$$M_{Lip}(A) = \max_{j\in[l]}\max_{\substack{\mathbf{z}\in\mathcal{U}^l\\ z_j\neq z'_j\in\mathcal{U}}}\frac{D^j_{z_j,z'_j}A(\mathbf{z})}{\|z_j-z'_j\|} \tag{26}$$

*and the second-order Lipschitzness of $A$ as:*

$$J_{Lip}(A) = l\cdot\max_{r\neq j}\sup_{\substack{\mathbf{z}\in\mathcal{U}^l\\ y_r\neq y'_r\in\mathcal{U}\\ z_j\neq z'_j\in\mathcal{U}}}\frac{D^r_{y_r,y'_r}D^j_{z_j,z'_j}A(\mathbf{z})}{\|z_j-z'_j\|} \tag{27}$$

*Proof.* Let $\mathcal{U} = H_{\mathcal{F}^*}(\mathcal{C})$, $\mathbf{z} = (u_i,v_i)_{i=1}^l \in \mathcal{U}^l$ and $A$ be an aggregation function that is defined as follows:

$$A(\mathbf{z}) = \mathrm{Avg}_{i\neq j}\left[\frac{v_i+v_j}{\|u_i-u_j\|^2}\right] \tag{28}$$

By Theorem 2 in (Maurer & Pontil, 2019),

$$\begin{aligned}
&\mathbb{E}\left[\sup_{f\in\mathcal{F}^*}\left[\mathbb{E}\left[\mathrm{Avg}_{i\neq j}V_f(P_i,P_j)\right] - \mathrm{Avg}_{i\neq j}V_f(\tilde{P}_i,\tilde{P}_j)\right]\right] \\
\leq& \sqrt{2\pi}\left(2M_{Lip}(A)+J_{Lip}(A)\right)\cdot\mathbb{E}[G(H_{\mathcal{F}^*}(\tilde{\mathcal{P}}))],
\end{aligned} \tag{29}$$

where $P_1,\dots,P_l \sim \mathcal{D}_{\mathcal{C}}(P_1,\dots,P_l \mid \forall i\neq j\in[l] : P_i\neq P_j)$. By the linearity of expectation, $\mathbb{E}\left[\mathrm{Avg}_{i\neq j}V_f(P_i,P_j)\right] = \mathbb{E}_{P_c\neq P_{c'}}\left[V_f(P_c,P_{c'})\right]$.

We would like to bound the terms $M_{Lip}(A)$ and $J_{Lip}(A)$ of the aggregation function $A$ when applied to the space $\mathcal{U}^l$ with $\mathcal{U} = H_{\mathcal{F}^*}(\mathcal{C})$. Let $j \in [l]$ be some index and $\mathbf{z}' = (z_i)_{i=1}^l = (u'_i, v'_i)_{i=1}^l$ be a vector, such that, $z'_i = z_i$ for all $i \neq j$ and $z'_j \neq z_j$. Then,

$$\begin{aligned}\frac{D^j_{z_j,z'_j}A(\mathbf{z})}{\|z_j - z'_j\|} =& \frac{A(\mathbf{z}) - A(\mathbf{z}')}{\|(v_j, u_j) - (v'_j, u'_j)\|} \\ =& \frac{1}{l(l-l)} \sum_{i:\ i\neq j} \left[ \frac{\frac{v_i+v_j}{\|u_i-u_j\|^2} - \frac{v_i+v'_j}{\|u_i-u'_j\|^2}}{\|(v_j, u_j) - (v'_j, u'_j)\|} \right]\end{aligned} \tag{30}$$

We denote $a = \|u_i - u_j\|$ and $b = \|u_i - u'_j\|$ and $c = \|z_j - z'_j\|$. By simple algebraic manipulations:

$$\begin{aligned}\frac{\frac{v_i+v_j}{\|u_i-u_j\|^2} - \frac{v_i+v'_j}{\|u_i-u'_j\|^2}}{\|(v_j, u_j) - (v'_j, u'_j)\|} =& \frac{(v_i+v_j)\frac{(b-a)(b+a)}{a^2b^2} + \frac{v_j-v'_j}{b^2}}{c} \\ \leq& \frac{(v_i+v_j)\frac{|b-a|\cdot(b+a)}{a^2b^2} + \frac{|v_j-v'_j|}{b^2}}{c}\end{aligned} \tag{31}$$

We notice that: $|v_j - v'_j| \leq c$ and also by the reversed triangle inequality, $|b - a| \leq c$. Hence, since $z_i, z'_i \in \mathcal{U}$ for all $i \in [l]$, we have:

$$\begin{aligned}\frac{\frac{v_i+v_j}{\|u_i-u_j\|^2} - \frac{v_i+v'_j}{\|u_i-u'_j\|^2}}{\|(v_j, u_j) - (v'_j, u'_j)\|} \leq& (v_i+v_j)\cdot\left(\frac{1}{ab^2} + \frac{1}{a^2b}\right) + \frac{1}{b^2} \\ \leq& \frac{4\max_{f\in\mathcal{F}^*}\max_{P_c\in\mathcal{C}}\mathrm{Var}_f(P_c)}{\Delta(\mathcal{F}^*)^3} + \frac{1}{\Delta(\mathcal{F}^*)^2}\end{aligned} \tag{32}$$

Therefore, we have:

$$M_{Lip}(A) \leq \frac{4\max_{f\in\mathcal{F}^*}\max_{P_c\in\mathcal{C}}\mathrm{Var}_f(P_c)}{l\cdot\Delta(\mathcal{F}^*)^3} + \frac{1}{l\cdot\Delta(\mathcal{F}^*)^2} \tag{33}$$

Next, we would like to upper bound the second degree Lipschitzness of the function. We note that:

$$\begin{aligned}\frac{D^r_{z_r,z'_r}D^j_{z_j,z'_j}A(\mathbf{z})}{\|z_j - z'_j\|} =& \frac{1}{l(l-1)}\left[\frac{\frac{v_r+v_j}{\|u_r-u_j\|^2} - \frac{v_r+v'_j}{\|u_r-u'_j\|^2}}{\|z_j - z'_j\|} - \frac{\frac{v'_r+v_j}{\|u'_r-u_j\|^2} - \frac{v'_r+v'_j}{\|u'_r-u'_j\|^2}}{\|z_j - z'_j\|}\right] \\ \leq& \frac{1}{l(l-1)}\left[\left|\frac{\frac{v_r+v_j}{\|u_r-u_j\|^2} - \frac{v_r+v'_j}{\|u_r-u'_j\|^2}}{\|z_j - z'_j\|}\right| + \left|\frac{\frac{v'_r+v_j}{\|u'_r-u_j\|^2} - \frac{v'_r+v'_j}{\|u'_r-u'_j\|^2}}{\|z_j - z'_j\|}\right|\right]\end{aligned} \tag{34}$$

By Equation 32, we have:

$$J_{Lip}(A) \leq \frac{1}{l-1}\left(\frac{8\max_{f\in\mathcal{F}^*}\max_{P_c\in\mathcal{C}}\mathrm{Var}_f(P_c)}{\Delta(\mathcal{F}^*)^3} + \frac{2}{\Delta(\mathcal{F}^*)^2}\right) \tag{35}$$

By Equations 33 and 35, we have:

$$\sqrt{2\pi}(2M_{Lip}(A) + J_{Lip}) \leq \left(\frac{24\max_{f\in\mathcal{F}^*}\max_{P_c\in\mathcal{C}}\mathrm{Var}_f(P_c)}{l\cdot\Delta(\mathcal{F}^*)^3} + \frac{6}{l\cdot\Delta(\mathcal{F}^*)^2}\right) \tag{36}$$

By combining the above inequality with Equation 29, we obtain the desired bound. □