The Prevalence of Neural Collapse in Neural Multivariate Regression

George Andriopoulos[1*] **Zixuan Dong**[2,4*] **Li Guo**[3*] **Zifan Zhao**[3*] **Keith Ross**[1*†]

[1] New York University Abu Dhabi [2] SFSC of AI and DL, NYU Shanghai
[3] New York University Shanghai [4] New York University

Abstract

Recently it has been observed that neural networks exhibit Neural Collapse (NC) during the final stage of training for the classification problem. We empirically show that multivariate regression, as employed in imitation learning and other applications, exhibits Neural Regression Collapse (NRC), a new form of neural collapse: (NRC1) The last-layer feature vectors collapse to the subspace spanned by the n principal components of the feature vectors, where n is the dimension of the targets (for univariate regression, $n = 1$); (NRC2) The last-layer feature vectors also collapse to the subspace spanned by the last-layer weight vectors; (NRC3) The Gram matrix for the weight vectors converges to a specific functional form that depends on the covariance matrix of the targets. After empirically establishing the prevalence of (NRC1)-(NRC3) for a variety of datasets and network architectures, we provide an explanation of these phenomena by modeling the regression task in the context of the Unconstrained Feature Model (UFM), in which the last layer feature vectors are treated as free variables when minimizing the loss function. We show that when the regularization parameters in the UFM model are strictly positive, then (NRC1)-(NRC3) also emerge as solutions in the UFM optimization problem. We also show that if the regularization parameters are equal to zero, then there is no collapse. To our knowledge, this is the first empirical and theoretical study of neural collapse in the context of regression. This extension is significant not only because it broadens the applicability of neural collapse to a new category of problems but also because it suggests that the phenomena of neural collapse could be a universal behavior in deep learning.

1 Introduction

Recently, an insightful phenomenon known as neural collapse (NC) [Papyan et al., 2020] has been empirically observed during the terminal phases of training in classification tasks with balanced data. NC has three principal components: (NC1) The features of samples within each class converge closely around their class mean. (NC2) The averages of the features within each class converge to form the vertices of a simplex equiangular tight frame. This geometric arrangement implies that class means are equidistant and symmetrically distributed. (NC3) The weight vectors of the classifiers in the final layer align with the class means of their respective features. These phenomena not only enhance our understanding of neural network behaviors but also suggest potential simplifications in the architecture and the training of neural networks.

*Equal contribution.
†Corresponding author: keithwross@nyu.edu

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

 https://doi.org/10.52202/079017-4015

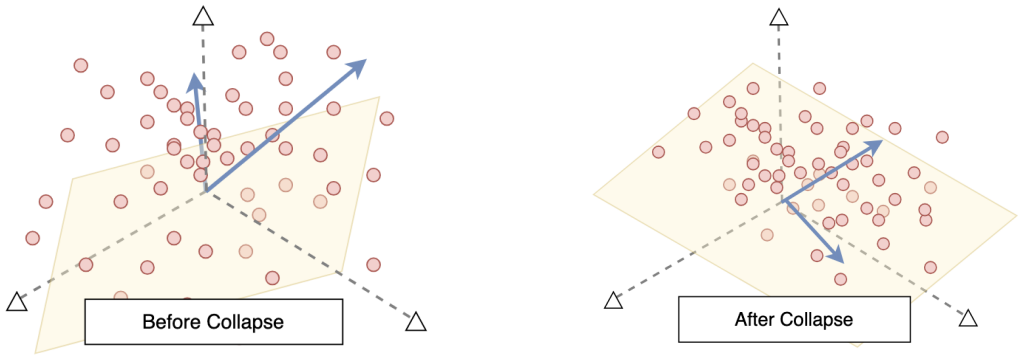

Figure 1: Visualization of the neural regression collapse. The red dots represent the sample features, the blue arrows represent the row vectors of the last layer weight matrix, and the yellow plane represents the plane spanned by the principal components of the sample features. Here the target dimension is $n = 2$. The feature vectors and weight vectors collapse to the same subspace. The angle between the weight vectors takes specific forms governed by the covariance matrix of the targets.

The initial empirical observations of NC have led to the development of theoretical frameworks such as the layered-peeled model [Fang et al., 2021] and the unconstrained feature model (UFM) [Mixon et al., 2020]. These models help explain why NC occurs in classification tasks theoretically. By allowing the optimization to freely adjust last-layer features along with classifier weights, these models provide important insights into the prevalence of neural collapse, showing that maximal class separability is a natural outcome for a variety of loss functions when the data is balanced [Han et al., 2021, Poggio and Liao, 2020, Zhou et al., 2022a,b].

Regression in deep learning is arguably equally important as classification, as it serves for numerous applications across diverse domains. In imitation learning for autonomous driving, regression is employed to predict continuous control actions (such as speed and steering angles) based on observed human driver behavior. Similarly, regression is used in robotics, where the regression model is trained to imitate expert demonstrations. In the financial sector, regression models are extensively used for predictive analytics, such as forecasting stock prices, estimating risk, and predicting market trends. Meteorology also heavily relies on regression models to forecast weather conditions. These models take high-dimensional inputs from various sensors and satellites to predict multiple continuous variables such as temperature, humidity, and wind speed. Moreover, many reinforcement learning algorithms include critical regression components, where regression is employed to predict value functions with the targets being Monte Carlo or bootstrapped returns.

While NC has been extensively studied in classification, to our knowledge, its prevalence and implications in regression remain unexplored. This paper investigates a new form of neural collapse within the context of neural multivariate regression. Analogous to the classification problem, we introduce Neural Regression Collapse (NRC): (NRC1) During training, the last-layer feature vectors collapse to the subspace spanned by the n principal components of the feature vectors, where n is the dimension of the targets (for univariate regression, $n = 1$); (NRC2) The last-layer feature vectors also collapse to the subspace spanned by the weight vectors; (NRC3) The Gram matrix for the weight vectors converges to a specific functional form that depends on the square-root of the covariance matrix of the targets. A visualization of NRC is shown in Figure 1.

Employing six different datasets – including three robotic locomotion datasets, two versions of an autonomous driving dataset, and an age-prediction dataset – and Multi-Layer Perceptron (MLP) and ResNet architectures, we establish the prevalence of NRC1-NRC3. This discovery suggests a universal geometric behavior extending beyond classification into regression models, simplifying our understanding of deep learning more generally.

To help explain these phenomena, we then apply the UFM model to neural multivariate regression with an L2 loss function. In this regression version of the problem, the optimization problem aims to minimize the regularized mean squared error over continuous-valued targets. We show that when the regularization parameters in the UFM model are strictly positive, then (NRC1)-(NRC3) also emerge

as solutions in the UFM optimization problem, thereby providing a mathematical explanation of our empirical observations. Among many observations, we discover empirically and theoretically that when the regression parameters are zero or very small, there is no collapse; and if we increase the parameters a small amount above zero, the (NRC1)-(NRC3) geometric structure emerges.

To the best of our knowledge, this is the first empirical and theoretical study of neural collapse in the context of regression. By demonstrating the prevalence of neural collapse in regression tasks, we reveal that deep learning systems might inherently simplify their internal representations, irrespective of the specific nature of the task, whether it be classification or regression.

2 Related work

Neural collapse (NC) was first identified by Papyan et al. [2020] as a symmetric geometric structure observed in both the last layer features and classification vectors during the terminal phase of training of deep neural networks for classification tasks, particularly evident in balanced datasets. Since then, there has been a surge of research into both theoretical and empirical aspects of NC.

Several studies have investigated NC under different loss functions. For instance, [Han et al., 2021, Poggio and Liao, 2020, Zhou et al., 2022a] have observed and studied neural collapse under the Mean Squared Error (MSE) loss, while papers such as [Zhou et al., 2022b, Guo et al., 2024] have demonstrated that label smoothing loss and focal loss also lead to neural collapse. In addition to the last layer, some papers [He and Su, 2023, Rangamani et al., 2023] have also examined the occurrence of the NC properties within intermediate layers. Furthermore, beyond the balanced case, researchers have investigated the neural collapse phenomena in imbalanced scenarios. [Fang et al., 2021] identified a phenomenon called minority collapse for training on imbalanced data, while [Hong and Ling, 2023, Thrampoulidis et al., 2022, Dang et al., 2023] offer more precise characterizations of the geometric structure under imbalanced conditions.

To facilitate the theoretical exploration of the neural collapse phenomena, [Fang et al., 2021, Mixon et al., 2020] considered the unconstrained feature model (UFM). The UFM simplifies a deep neural network into an optimization problem by treating the last layer features as free variables to optimize over. This simplification is motivated by the rationale of the universal approximation theorem [Hornik et al., 1989], asserting that sufficiently over-parameterized neural networks can be highly expressive and can accurately approximate arbitrary smooth functions. Leveraging the UFM, studies such as [Zhu et al., 2021, Zhou et al., 2022a, Thrampoulidis et al., 2022, Tirer and Bruna, 2022, Tirer et al., 2023, Ergen and Pilanci, 2021, Wojtowytsch et al., 2020] have investigated models with different loss functions and regularization techniques. These studies have revealed that the global minima of the empirical risk function under UFMs align with the characterization of neural collapse observed by [Papyan et al., 2020]. Beyond the UFM, some work [Tirer and Bruna, 2022, Súkeník et al., 2024] has extended the model to explore deep constrained feature models with multiple layers, aiming to investigate neural collapse properties beyond the last layer.

In addition to its theoretical implications, NC serves as a valuable tool for gaining deeper insights into DNN models and various regularization techniques [Guo et al., 2024, Fisher et al., 2024]. It provides crucial insights into the generalization and transfer learning capabilities of neural networks [Hui et al., 2022, Kothapalli, 2022, Galanti et al., 2021], inspiring the design of enhanced model architectures for diverse applications. These include scenarios with imbalanced data [Yang et al., 2022, Kim and Kim] and contexts involving online continuous learning [Seo et al., 2024].

Despite extensive research on the neural collapse phenomena and its implications in classification, to the best of our knowledge, there has been no investigation into similar issues regarding neural regression models. Perhaps the paper closest to the current work is [Zhou et al., 2022a], which applies the UFM model to the balanced classification problem with MSE loss. Although focused on classification, [Zhou et al., 2022a] derive some important results which apply to regression as well as to classification. Our UFM analysis leverages this related paper, particularly their Lemma B.1.

3 Prevalence of neural regression collapse

We consider the multivariate regression problem with M training examples $\{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, M\}$, where each input $\mathbf{x}_i$ belongs to $\mathbb{R}^D$ and each target vector $\mathbf{y}_i$ belongs to $\mathbb{R}^n$. For the regression

 https://doi.org/10.52202/079017-4015

task, the deep neural network (DNN) takes as input an example $\mathbf{x} \in \mathbb{R}^D$ and produces an output $\mathbf{y} = f(\mathbf{x}) \in \mathbb{R}^n$. For most DNNs, including those used in this paper, this mapping takes the form $f_{\theta,\mathbf{W},\mathbf{b}}(\mathbf{x}) = \mathbf{W}\mathbf{h}_\theta(\mathbf{x}) + \mathbf{b}$, where $\mathbf{h}_\theta(\cdot) : \mathbb{R}^D \to \mathbb{R}^d$ is the non-linear feature extractor consisting of several nonlinear layers, $\mathbf{W}$ is a $n \times d$ matrix representing the final linear layer in the model, and $\mathbf{b} \in \mathbb{R}^n$ is the bias vector. For most neural regression tasks, $n << d$, that is the dimension of the target space is much smaller than the dimension of the feature space. For univariate regression, $n = 1$. The parameters θ, $\mathbf{W}$, and $\mathbf{b}$ are all trainable.

We train the DNN using gradient descent to minimize the regularized L2 loss:

$$\min_{\theta,\mathbf{W},\mathbf{b}} \frac{1}{2M} \sum_{i=1}^{M} ||f_{\theta,\mathbf{W},\mathbf{b}}(\mathbf{x}_i) - \mathbf{y}_i||_2^2 + \frac{\lambda_\theta}{2}||\theta||_2^2 + \frac{\lambda_\mathbf{W}}{2}||\mathbf{W}||_F^2,$$

where $||\cdot||_2$ and $||\cdot||_F$ denote the L_2-norm and the Frobenius norm, respectively. As commonly done in practice, in our experiments we set all the regularization parameters to the same value, which we refer to as the weight-decay parameter λ_{WD}, that is, we set $\lambda_\theta = \lambda_\mathbf{W} = \lambda_{WD}$.

3.1 Definition of neural regression collapse

In order to define Neural Regression Collapse (NRC), let $\mathbf{\Sigma}$ denote the $n \times n$ covariance matrix corresponding to the targets $\{\mathbf{y}_i, i = 1, \ldots, M\}$: $\mathbf{\Sigma} = M^{-1}(\mathbf{Y} - \bar{\mathbf{Y}})(\mathbf{Y} - \bar{\mathbf{Y}})^T$, where $\mathbf{Y} = [\mathbf{y}_1 \cdots \mathbf{y}_M]$, $\bar{\mathbf{Y}} = [\bar{\mathbf{y}} \cdots \bar{\mathbf{y}}]$, and $\bar{\mathbf{y}} = M^{-1}\sum_{i=1}^{M} \mathbf{y}_i$. Throughout this paper, we make the natural assumption that $\mathbf{Y}$ and $\mathbf{\Sigma}$ have full rank. Thus $\mathbf{\Sigma}$ is positive definite. Let $\lambda_{\text{min}} > 0$ denote the minimum eigenvalue of $\mathbf{\Sigma}$.

Denote $\mathbf{H} := [\mathbf{h}_1 \cdots \mathbf{h}_M]$, where $\mathbf{h}_i$ is the feature vector associated with input $\mathbf{x}_i$, that is, $\mathbf{h}_i := \mathbf{h}_\theta(\mathbf{x}_i)$. Further denote the normalized feature vector $\widetilde{\mathbf{h}}_i := \mathbf{h}_i \cdot ||\mathbf{h}_i||^{-1}$. Of course, $\mathbf{W}$, $\mathbf{H}$, and $\mathbf{b}$ are changing throughout the course of training. For any $p \times q$ matrix $\mathbf{C}$ and any p-dimensional vector $\mathbf{v}$, let $proj(\mathbf{v}|\mathbf{C})$ denote the projection of $\mathbf{v}$ onto the subspace spanned by the columns of $\mathbf{C}$. Let $\mathbf{H}_{PCA_n}$ be the $d \times n$ matrix with the columns consisting of the n principal components of $\mathbf{H}$.

We say that *Neural Regression Collapse (NRC)* emerges during training if the following three phenomena occur:

- NRC1 = $\frac{1}{M}\sum_{i=1}^{M} \left|\left|\widetilde{\mathbf{h}}_i - proj(\widetilde{\mathbf{h}}_i|\mathbf{H}_{PCA_n})\right|\right|_2^2 \to 0.$
- NRC2 = $\frac{1}{M}\sum_{i=1}^{M} \left|\left|\widetilde{\mathbf{h}}_i - proj(\widetilde{\mathbf{h}}_i|\mathbf{W}^T)\right|\right|_2^2 \to 0.$
- There exists a constant $\gamma \in (0, \lambda_{\min})$ such that:

$$\text{NRC3} = \left|\left| \frac{\mathbf{W}\mathbf{W}^T}{||\mathbf{W}\mathbf{W}^T||_F} - \frac{\mathbf{\Sigma}^{1/2} - \gamma^{1/2}\mathbf{I}_n}{||\mathbf{\Sigma}^{1/2} - \gamma^{1/2}\mathbf{I}_n||_F} \right|\right|_F^2 \to 0.$$

NRC1 $\to 0$ indicates that there is *feature-vector collapse*, that is, the d-dimensional feature vectors $\mathbf{h}_i$, $i = 1, \ldots, M$, collapse to a much lower n-dimensional subspace spanned by their n principal components. In many applications, $n = 1$, in which case the feature vectors are collapsing to a line in the original d-dimensional space. NRC2 $\to 0$ indicates that there is a form of *self duality*, that is, the feature vectors also collapse to the n-dimensional space spanned by the rows of $\mathbf{W}$. NRC3 $\to 0$ indicates that the last-layer weights have a *specific structure* within the collapsed subspace. In particular, it gives detailed information about the norms of the row vectors in $\mathbf{W}$ and the angles between those row vectors. NRC3 $\to 0$ indicates that angles between the rows in $\mathbf{W}$ are influenced by $\mathbf{\Sigma}^{1/2}$. If the targets are uncorrelated so that $\mathbf{\Sigma}$ and $\mathbf{\Sigma}^{1/2}$ are diagonal, then NRC3 $\to 0$ implies that the rows in $\mathbf{W}$ will be orthogonal. NRC3 $\to 0$ also implies a specific structure for the feature vectors, as discussed in Section 4.

3.2 Experimental validation of neural regression collapse

In this section, we validate the emergence of NRC1-NRC3 during training across various datasets and deep neural network (DNN) architectures.

Datasets. The empirical experiments in this section are based on the following datasets:

- The **Swimmer**, **Reacher**, and **Hopper datasets** are based on MoJoCo [Todorov et al., 2012, Brockman et al., 2016, Towers et al., 2023], a physics engine that simulates diverse continuous multi-joint robot controls and has been a canonical benchmark for deep reinforcement learning research. In our experiments, we use publicly available expert datasets (see appendix A.1). Each dataset comprises raw robotic states as inputs ($\mathbf{x}_i$'s) and robotic actions as targets ($\mathbf{y}_i$'s). In order to put these expert datasets in an imitation learning context, we reduced the size of the dataset by keeping only a small portion of the episodes.
- The **CARLA dataset** originates from the CARLA Simulator, an open-source project designed to support the development of autonomous driving systems. We utilize a dataset Codevilla et al. [2018] sourced from expert-driven offline simulations. During these simulations, images ($\mathbf{x}_i$'s) from cameras mounted on the virtual vehicle and corresponding expert driver actions as targets ($\mathbf{y}_i$'s) are recorded as human drives in the simulated environment. We consider two dataset versions: a 2D version with speed and steering angle, and a 1D version with only the speed.
- The **UTKFace dataset** [Zhang et al., 2017] is widely used in computer vision to study age estimation from facial images of humans. This dataset consists of about 25,000 facial images spanning a wide target range of ages, races, and genders.

Table 1 summarizes the six datasets, with the dimensions of the target vectors $\mathbf{y}$ ranging from one to three. The table also includes the minimum eigenvalue of the associated covariance matrix $\mathbf{\Sigma}$ and the Pearson correlation values between the i-th and j-th target components for $i \neq j$. When $n = 1$, there is no correlation value; when $n = 2$, there is one correlation value between the two target components; and when $n = 3$, there are three correlation values among the three target components. From the table, we observe that the target components in CARLA 2D and Reacher are nearly uncorrelated, whereas those in Hopper and Swimmer exhibit stronger correlations.

Table 1: Overview of datasets employed in our neural regression collapse analysis.

Dataset	**Data Size**	**Input Type**	**Target Dimension** n	**Target Correlation**	$\lambda_{\min}$
Swimmer	1,000	raw state	2	-0.244	0.276
Reacher	1,000	raw state	2	-0.00933	0.0097
Hopper	10,000	raw state	3	[-0.215, -0.090, 0.059]	0.215
Carla 1D	600,000	RGB image	1	NA	208.63
Carla 2D	600,000	RGB image	2	-0.0055	0.156
UTKface	25,000	RGB image	1	NA	1428

Experiment Settings. For the Swimmer, Reacher, and Hopper datasets, we employed a four-layer MLP (with the last layer being the linear layer) as the policy network for the prediction task. Each layer consisted of 256 nodes, aligning with the conventional model architecture in most reinforcement learning research [Tarasov et al., 2022]. For the CARLA and UTKFace datasets, we employed ResNet18 and ResNet34 He et al. [2016], respectively. To focus on behaviors associated with neural collapse and minimize the influence of other factors, we applied standard preprocessing without data augmentation.

All experimental results are averaged over at least 2 random seeds and variance is displayed by a shaded area. The choices of weight decay employed during training varied depending on the dataset. Also, the number of epochs required for training depends on both the dataset and the degree of weight decay. In particular, we used a large number of epochs when using very small weight decay values. appendix A provides the full experimental setup.

Empirical Results. Figure 2 presents the experimental results for the six datasets mentioned above. The results show that the training and testing errors decrease as training progresses, as expected. The converging coefficient of determination (R^2) also indicates that model performance becomes stable. Most importantly, the figure confirms the presence of NRC1-NRC3 across all six datasets. This indicates that neural collapse is not only prevalent in classification but also often occurs in multivariate regression.

 https://doi.org/10.52202/079017-4015

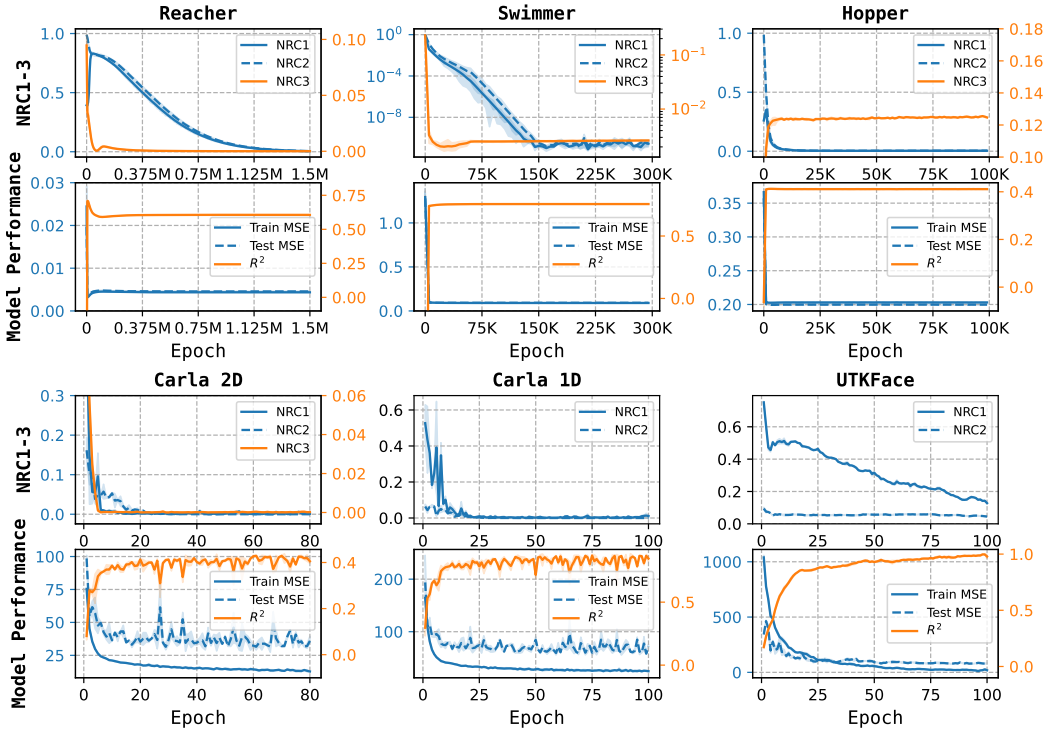

Figure 2: Prevalence of NRC1-NRC3 in the six datasets. Train/Test MSE and the coefficient of determination (R^2) are also shown.

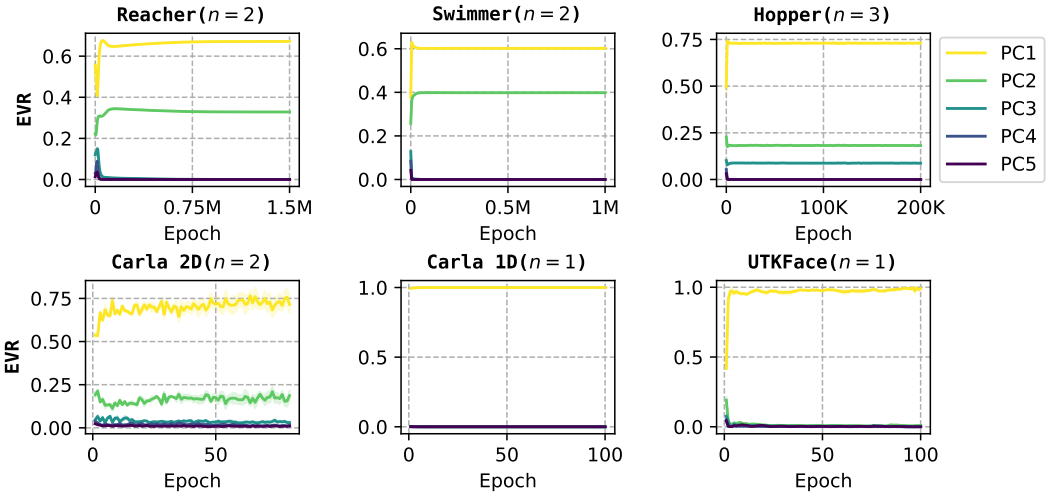

Figure 3: Explained Variance Ratio (EVR) for the first 5 principal components (PC).

We also experimentally analyze the explained variance ratio (EVR) of principal components to further verify the collapse to the subspace spanned by the first n components. In Figure 3, we investigate the EVR of the first 5 principal components of $\mathbf{H}$ during the training process. For all datasets, there is significant variance for all of the first n components after a short period of training; for other components, there is very low or even no variance. This also supports that a perfect collapse occurs in the subspace spanned by the first n principal components.

Our definition of NRC3 involves finding a scaling factor γ for which the property holds. Figure 4 illustrates the values of NRC3 as a function of γ for $\mathbf{W}$ obtained after training. We observe that each

dataset exhibits a unique minimum value of γ. More details about computing NRC3 can be found in appendix A.3.

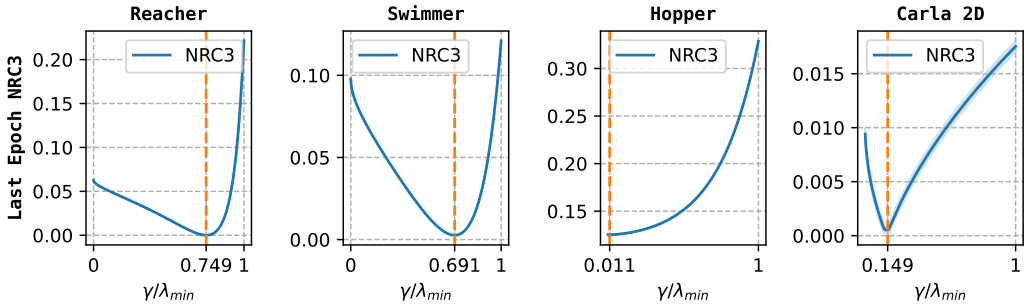

Figure 4: The optimal value of γ for NRC3.

Figure 5 investigates neural regression collapse for small values of the weight-decay parameter λ_{WD}. (appendix A.4 contains results on all 6 datasets.) We see that when $\lambda_{WD} = 0$, there is no neural regression collapse. However, if we increase λ_{WD} by a small amount, collapse emerges for all three metrics. Thus we can conclude that the geometric structure NRC1-3 that emerges during training is due to regularization, albeit the regularization can be very small. In the next section, we will introduce a mathematical model that helps explain why there is no collapse when $\lambda_{WD} = 0$ and why it quickly emerges as λ_{WD} is increased above zero.

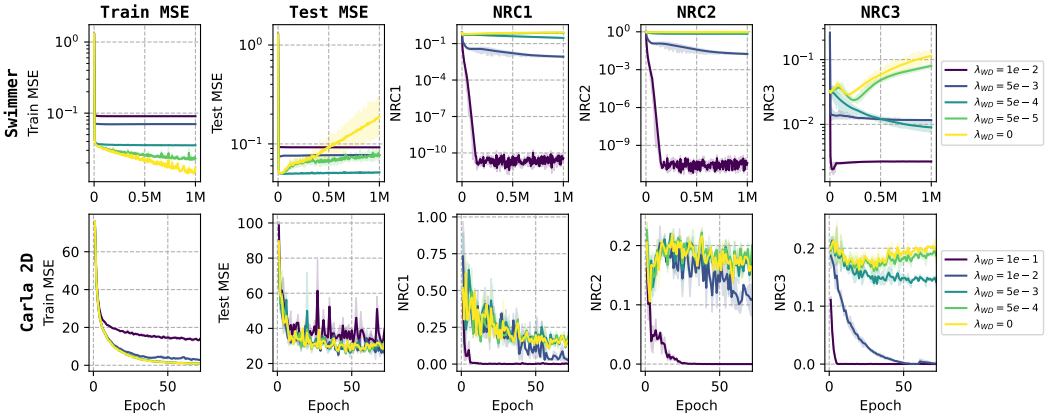

Figure 5: Phase change in neural collapse for small weight-decay values

4 Unconstrained feature model

As discussed in the related work section, the UFM model has been extensively used to help explain the prevalence of neural collapse in the classification problem. In this section, we explore whether the UFM model can also help explain neural collapse in neural multivariate regression.

Specifically, we consider minimizing $\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b})$, where

$$\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b}) = \frac{1}{2M}||\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y}||_F^2 + \frac{\lambda_{\mathbf{H}}}{2M}||\mathbf{H}||_F^2 + \frac{\lambda_{\mathbf{W}}}{2}||\mathbf{W}||_F^2, \tag{1}$$

where $\mathbf{1}_M^T := [1 \cdots 1]$ and $\lambda_{\mathbf{H}}$, $\lambda_{\mathbf{W}}$ are non-negative regularization constants.

The optimization problem studied here bears some resemblance to the standard linear multivariate regression problem. If we view the features $\mathbf{h}_i$, $i = 1, \ldots, M$, as the inputs to linear regression, then $\hat{\mathbf{y}}_i := \mathbf{W}\mathbf{h}_i + \mathbf{b}$ is the predicted output, and $||\mathbf{y}_i - \hat{\mathbf{y}}_i||_2^2$ is the squared error. In standard linear regression, the $\mathbf{h}_i$'s are fixed inputs. In the UFM model, however, not only are we optimizing over the weights $\mathbf{W}$ and biases $\mathbf{b}$ but also over all the "inputs" $\mathbf{H}$.

https://doi.org/10.52202/079017-4015

For the case of classification, regularization is needed in the UFM model to prevent the norms of $\mathbf{H}$ and/or $\mathbf{W}$ from going to infinity in the optimal solutions. In contrast, in the UFM regression model, the norms in the optimal solutions will be finite even without regularization. However, as regularization is typically used in neural regression problems to prevent overfitting, it is useful to include regularization in the UFM regression model as well.

4.1 Regularized loss function

Throughout this subsection, we assume that both $\lambda_{\mathbf{W}}$ and $\lambda_{\mathbf{H}}$ are strictly positive. We shall consider the $\lambda_{\mathbf{W}} = \lambda_{\mathbf{H}} = 0$ case subsequently. We also make a number of assumptions in order to not get distracted by less important sub-cases. Throughout we assume $n \leq d$, that is, the dimension of the targets is not greater than the dimension of the feature space. As stated in a previous subsection, for problems of practical interest, we have $n << d$. Recall that $\mathbf{\Sigma}$ is the covariance matrix of the target data. Since $\mathbf{\Sigma}$ is a covariance matrix and is assumed to have full rank, it is also positive definite. It therefore has a positive definite square root, which we denote by $\mathbf{\Sigma}^{1/2}$. Let $\lambda_{\max} := \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n := \lambda_{\min} > 0$ denote the n eigenvalues of $\mathbf{\Sigma}$. We further define the $n \times n$ matrix

$$\mathbf{A} := \mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n, \tag{2}$$

where $c := \lambda_{\mathbf{W}}\lambda_{\mathbf{H}}$. Also for any $p \times q$ matrix $\mathbf{C}$ with columns $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_q$, we denote $[\mathbf{C}]_j$ to be the $p \times q$ matrix whose first j columns are identical to those in $\mathbf{C}$ and whose last $q - j$ columns are all zero vectors, i.e., $[\mathbf{C}]_j = [\mathbf{c}_1\ \mathbf{c}_2 \cdots \mathbf{c}_j\ \mathbf{0} \cdots \mathbf{0}]$. All proofs are provided in the appendix.

Theorem 4.1. *Any global minimum* $(\mathbf{W}, \mathbf{H}, \mathbf{b})$ *for (1) takes the following form: If* $0 < c < \lambda_{\max}$*, then for any semi-orthogonal matrix* $\mathbf{R}$*,*

$$\mathbf{W} = \left(\frac{\lambda_{\mathbf{H}}}{\lambda_{\mathbf{W}}}\right)^{1/4} [\mathbf{A}^{1/2}]_{j*}\mathbf{R}, \qquad \mathbf{H} = \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}\mathbf{W}^T[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}), \qquad \mathbf{b} = \bar{\mathbf{y}}, \tag{3}$$

where $j* := \max\{j : \lambda_j \geq c\}$*. If* $c > \lambda_{\max}$*, then* $(\mathbf{W}, \mathbf{H}, \mathbf{b}) = (\mathbf{0}, \mathbf{0}, \bar{\mathbf{y}})$*. Furthermore, if* $(\mathbf{W}, \mathbf{H}, \mathbf{b})$ *is a critical point but not a global minimum, then it is a strict saddle point.*

Theorem 4.1 has numerous implications, which we elaborate on below.

4.2 One-dimensional univariate case

In this subsection, we highlight the important special case $n = 1$, which often arises in practice (such as with Carla 1D and the UTKface datasets). When $n = 1$, $\mathbf{\Sigma}$ is simply the scalar σ^2, which is the variance of the one-dimensional targets over the M samples. Also, $\mathbf{W}$ is a row vector, which we denote by $\mathbf{w}$. Theorem 4.1, for $n = 1$ provides the following insights:

1. Depending on whether $0 < c < \sigma^2$ or not, the global minimum takes on strikingly different forms. In the case, $c > \sigma^2$, corresponding to very large regularization parameters, the optimization problem ignores the MSE and entirely focuses on minimizing the norms $||\mathbf{H}||_F^2$ and $||\mathbf{w}||_2^2$, giving $||\mathbf{H}||_F^2 = 0$, $||\mathbf{w}||_2^2 = 0$.
2. When $0 < c < \sigma^2$, the optimal solution takes a more natural and interesting form: For any unit vector $\mathbf{e} \in \mathbb{R}^d$, the solution $(\mathbf{H}, \mathbf{w}, b)$ given by

$$\mathbf{w}^T = \sqrt{\lambda_{\mathbf{H}}\left(\frac{\sigma}{c^{1/2}} - 1\right)}\mathbf{e}, \qquad \mathbf{H} = \frac{\sqrt{\lambda_{\mathbf{W}}}}{\sqrt{\lambda_{\mathbf{H}}}\sigma}\mathbf{w}^T(\mathbf{Y} - \bar{\mathbf{Y}}), \qquad b = \bar{y}, \tag{4}$$

   is a global minimum. Thus, all vectors $\mathbf{w}$ on the sphere given by $||\mathbf{w}||_2^2 = \lambda_{\mathbf{H}}(\frac{\sigma}{c^{1/2}} - 1)$ are optimal solutions. Furthermore, $\mathbf{h}_i$, $i = 1, \ldots, M$, are all in the one-dimensional subspace spanned by $\mathbf{w}$. Thus the optimal solution of the UFM model provides a theoretical explanation for NRC1-NRC2. (NRC3 is not meaningful for the one-dimensional case.) Note that the $\mathbf{h}_i$'s have a global zero mean and the norm of $\mathbf{h}_i$ is proportional to $|y_i - \bar{y}|$.

4.3 General n-dimensional multivariate case

In most cases of practical interest, we will have $c < \lambda_{\min}$, so that $[\mathbf{A}^{1/2}]_{j*} = \mathbf{A}^{1/2}$ in Theorem 4.1.

Corollary 4.2. *Suppose* $0 < c < \lambda_{min}$. *Then the global minima given by* (3) *have the following properties:*

(i) All of the d*-dimensional feature vectors* $\mathbf{h}_i$, $i = 1, \ldots, M$, *lie in the* n*-dimensional subspace spanned by the* n *rows of* $\mathbf{W}$. *(ii)* $\mathbf{W}\mathbf{W}^T = \sqrt{\frac{\lambda_\mathbf{H}}{\lambda_\mathbf{W}}}\left[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n\right]$, *(iii)* $\lambda_\mathbf{H}||\mathbf{H}||_F^2 = M\lambda_\mathbf{W}||\mathbf{W}||_F^2$, *(iv)* $\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b}) = nc/2 + \sqrt{c}||\mathbf{A}^{1/2}||_F^2$, *(v)* $\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y} = -\sqrt{c}[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}})$.

From Theorem 4.1 and Corollary 4.2, we make the following observations:

1. Most importantly, the global minima in the UFM solution match the empirical properties (NRC1)-(NRC3) observed in Section 3. In particular, the theory precisely predicts NRC3, with $\gamma = c$. This confirms that the UFM model is an appropriate model for neural regression.
2. Unlike the one-dimensional case, the feature vectors are no longer colinear with any of the rows of $\mathbf{W}$. Moreover, after rotation and projection (determined by the semi-orthogonal matrix $\mathbf{R}$), the angles between the target vectors in $\mathbf{Y} - \bar{\mathbf{Y}}$ do not in general align with the angles between the feature vectors in $\mathbf{H}$. However, if the target components are uncorrelated, so that $\mathbf{\Sigma}$ is diagonal, then $\mathbf{A}$ is also diagonal and there is alignment between $\mathbf{H}$ and $\mathbf{Y} - \bar{\mathbf{Y}}$.

Theorem 4.1 also provides insight into the "strong regularization" case of $c > \lambda_{\text{min}}$. In this case, the rows of $\mathbf{W}$ and the feature vectors $\mathbf{H}$ in the global minima belong to a subspace that has dimension even smaller than n, specifically, to dimension $j* < n$. To gain some insight, assume that the target components are uncorrelated so that $\mathbf{\Sigma}$ is diagonal and $\lambda_j = \sigma_j^2$, i.e., σ_j^2 is the variance of the j-th target component. Then for a target component for which $c > \sigma_j^2$, the corresponding row in $\mathbf{W}$ will be zero and the component prediction will be $\hat{\mathbf{y}}_i^{(j)} = \bar{\mathbf{y}}^{(j)}$ for all examples $i = 1, \ldots, M$. For more details, we refer the reader to Section D.1 in the appendix.

4.4 Removing regularization

In the previous theorem and corollary, we assumed the presence or L2 regularization for $\mathbf{W}$ and $\mathbf{H}$, that is, we assumed $\lambda_\mathbf{W} > 0$ and $\lambda_\mathbf{H} > 0$. Now we explore the structure of the solutions to the UFM when $\lambda_\mathbf{W} = \lambda_\mathbf{H} = 0$. In this case, the UFM model is modeling the real problem with λ_{WD} equal to or close to zero. The loss function becomes:

$$L(\mathbf{W}, \mathbf{H}) = \frac{1}{2M}||\mathbf{W}\mathbf{H} - \mathbf{Y}||_F^2. \tag{5}$$

For this case, we do not need bias since we can obtain zero loss without it.

Theorem 4.3. *The solution* $(\mathbf{W}, \mathbf{H})$ *is a global minimum if and only if* $\mathbf{W}$ *is any* $n \times d$ *full rank matrix and*

$$\mathbf{H} = \mathbf{W}^+\mathbf{Y} + (\mathbf{I}_d - \mathbf{W}^+\mathbf{W})\mathbf{Z}, \tag{6}$$

where $\mathbf{W}^+$ *is the pseudo-inverse of* $\mathbf{W}$ *and* $\mathbf{Z}$ *is any* $d \times M$ *matrix. Consequently, when there is no regularization, for each full-rank* $\mathbf{W}$ *there is an infinite number of global minima* $(\mathbf{W}, \mathbf{H})$ *that do not collapse to any subspace of* $\mathbb{R}^d$.

From Theorem 4.3, when there is no regularization, the feature vectors do not collapse. Moreover, any full rank $\mathbf{W}$ provides an optimal solution. For example, for $n = 2$, the two rows of $\mathbf{W}$ can have any angle between them except angle 0 and angle 180. This is very different from the results we have for $\lambda_\mathbf{H}, \lambda_\mathbf{W} > 0$, in which case $\mathbf{W}$ depends on the covariance matrix $\mathbf{\Sigma}$. Note that if we set $\lambda_\mathbf{H} = \lambda_\mathbf{W}$ and let $\lambda_\mathbf{H} \to 0$, then the limit of $\mathbf{W}$ still depends on $\mathbf{\Sigma}$. Thus there is a major discontinuity in the solution when $\lambda_\mathbf{H}, \lambda_\mathbf{W}$ goes to zero. We also observed this phase shift in the experiments (see Figure 5). We can therefore conclude that neural regression collapse is not an intrinsic property of neural regression alone. The geometric structure of neural regression collapse is due to the inclusion of regularization in the loss function.

4.5 Empirical results with UFM assumptions

We also provide empirical results for the case when we train with the same form of regularization as assumed by the UFM model. Specifically, we turn off weight decay and add an L2 penalty on the last-layer features $\mathbf{h}_i$, $i = 1, \ldots, M$, and on the layer linear weights $\mathbf{W}$. Additionally, we omit the

 https://doi.org/10.52202/079017-4015

ReLU activation function in the penultimate layer, allowing the feature representation produced by the feature extractor to take any value, thus reflecting the UFM model. For these empirical results, when evaluating NRC3, rather than searching for γ as in the definition of NRC3, we use the exact value of γ given by Theorem 4.1, that is, $\gamma = \lambda_{\mathbf{W}}\lambda_{\mathbf{H}} = c$.

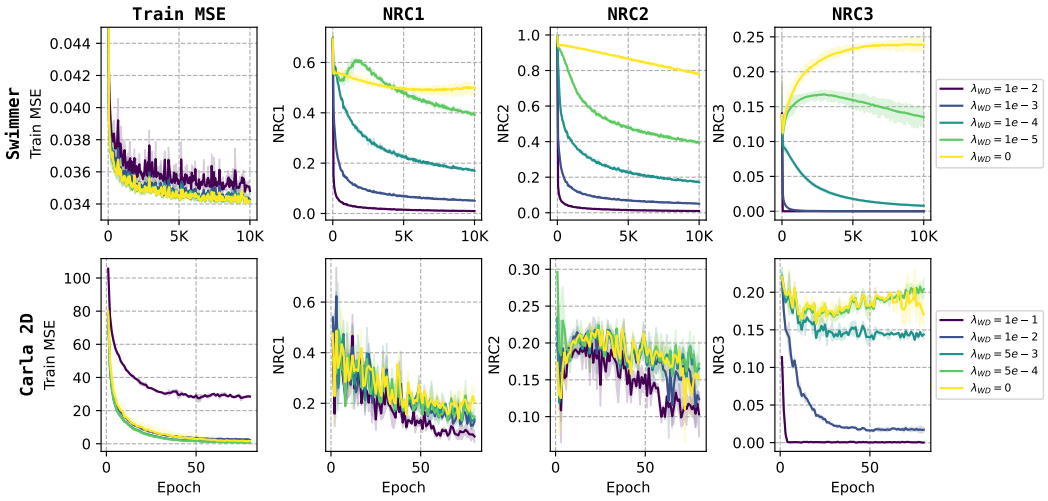

Figure 6: Empirical results with UFM assumption where L2 regularization on $\mathbf{H}$ and $\mathbf{W}$ are used instead of weight decay.

Figure 6 illustrates training MSE and NRC metrics for varying values of c. (For simplicity, we only considered the case where $\lambda_{\mathbf{W}} = \lambda_{\mathbf{H}}$. Appendix B contains results on remaining datasets.) As we are considering a different model and loss function for these empirical experiments, convergence occurs more quickly and so we train for a smaller number of epochs. We can conclude that the UFM theory not only accurately predicts the behavior of the standard L2 regularization approach with weight-decay for all parameters (Figure 5), but also accurately predicts the behavior when regularization follows the UFM assumptions (Figure 6).

5 Conclusion

We provided strong evidence, both empirically and theoretically, of the existence of neural collapse for multivariate regression. This extension is significant not only because it broadens the applicability of neural collapse to a new category of problems but also because it suggests that the phenomena of neural collapse could be a universal behavior in deep learning. However, it is worth acknowledging that while we have gained a better understanding of the model behavior of deep regression models in the terminal phase of training, we have not addressed the connection between neural regression collapse and model generalization. This crucial aspect remains an important topic for future research.

Acknowledgments and Disclosure of Funding

This work is submitted in part by the NYU Abu Dhabi Center for Artificial Intelligence and Robotics, funded by Tamkeen under the Research Institute Award CG010.

This work is partially supported by Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning at NYU Shanghai. Experimental computation was supported in part through the NYU IT High-Performance Computing resources and services.

References

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *International Conference on Robotics and Automation (ICRA)*, 2018.

Hien Dang, Tan Nguyen, Tho Tran, Hung Tran, and Nhat Ho. Neural collapse in deep linear network: From balanced to imbalanced data. *arXiv preprint arXiv:2301.00437*, 2023.

Tolga Ergen and Mert Pilanci. Revealing the structure of deep neural networks via convex duality. In *International Conference on Machine Learning*, pages 3004–3014. PMLR, 2021.

Cong Fang, Hangfeng He, Qi Long, and Weijie J Su. Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proceedings of the National Academy of Sciences*, 118(43):e2103091118, 2021.

Quinn Fisher, Haoming Meng, and Vardan Papyan. Pushing boundaries: Mixup's influence on neural collapse. *arXiv preprint arXiv:2402.06171*, 2024.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Tomer Galanti, András György, and Marcus Hutter. On the role of neural collapse in transfer learning. *arXiv preprint arXiv:2112.15121*, 2021.

Quentin Gallouédec, Edward Beeching, Clément Romac, and Emmanuel Dellandréa. Jack of All Trades, Master of Some, a Multi-Purpose Transformer Agent. *arXiv preprint arXiv:2402.09844*, 2024. URL `https://arxiv.org/abs/2402.09844`.

Li Guo, Keith Ross, Zifan Zhao, Andriopoulos George, Shuyang Ling, Yufeng Xu, and Zixuan Dong. Cross entropy versus label smoothing: A neural collapse perspective. *arXiv preprint arXiv:2402.03979*, 2024.

XY Han, Vardan Papyan, and David L Donoho. Neural collapse under MSE loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*, 2021.

Hangfeng He and Weijie J Su. A law of data separation in deep learning. *Proceedings of the National Academy of Sciences*, 120(36):e2221704120, 2023.

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.

Wanli Hong and Shuyang Ling. Neural collapse for unconstrained feature model under cross-entropy loss with imbalanced data. *arXiv preprint arXiv:2309.09725*, 2023.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Like Hui, Mikhail Belkin, and Preetum Nakkiran. Limitations of neural collapse for understanding generalization in deep learning. *arXiv preprint arXiv:2202.08384*, 2022.

 https://doi.org/10.52202/079017-4015

Hoyong Kim and Kangil Kim. Fixed non-negative orthogonal classifier: Inducing zero-mean neural collapse with fea-ture dimension separation.

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

Dustin G Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *arXiv preprint arXiv:2011.11619*, 2020.

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

Tomaso Poggio and Qianli Liao. Explicit regularization and implicit bias in deep network classifiers trained with the square loss. *arXiv preprint arXiv:2101.00072*, 2020.

Akshay Rangamani, Marius Lindegaard, Tomer Galanti, and Tomaso A Poggio. Feature learning in deep classifiers through intermediate neural collapse. In *International Conference on Machine Learning*, pages 28729–28745. PMLR, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Minhyuk Seo, Hyunseo Koh, Wonje Jeung, Minjae Lee, San Kim, Hankook Lee, Sungjun Cho, Sungik Choi, Hyunwoo Kim, and Jonghyun Choi. Learning equi-angular representations for online continual learning. *arXiv preprint arXiv:2404.01628*, 2024.

Peter Súkeník, Marco Mondelli, and Christoph H Lampert. Deep neural collapse is provably optimal for the deep unconstrained features model. *Advances in Neural Information Processing Systems*, 36, 2024.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. CORL: Research-oriented deep offline reinforcement learning library. In *3rd Offline RL Workshop: Offline RL as a "Launchpad"*, 2022. URL `https://openreview.net/forum?id=SyAS49bBcv`.

Christos Thrampoulidis, Ganesh Ramachandra Kini, Vala Vakilian, and Tina Behnia. Imbalance trouble: Revisiting neural-collapse geometry. *Advances in Neural Information Processing Systems*, 35:27225–27238, 2022.

Tom Tirer and Joan Bruna. Extended unconstrained features model for exploring deep neural collapse. In *International Conference on Machine Learning*, pages 21478–21505. PMLR, 2022.

Tom Tirer, Haoxiang Huang, and Jonathan Niles-Weed. Perturbation analysis of neural collapse. In *International Conference on Machine Learning*, pages 34301–34329. PMLR, 2023.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023. URL `https://zenodo.org/record/8127025`.

Stephan Wojtowytsch et al. On the emergence of simplex symmetry in the final and penultimate layers of neural network classifiers. *arXiv preprint arXiv:2012.05420*, 2020.

Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? *Advances in neural information processing systems*, 35:37991–38002, 2022.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. On the optimization landscape of neural collapse under MSE loss: Global optimality with unconstrained features. In *International Conference on Machine Learning*, pages 27179–27202. PMLR, 2022a.

Jinxin Zhou, Chong You, Xiao Li, Kangning Liu, Sheng Liu, Qing Qu, and Zhihui Zhu. Are all losses created equal: A neural collapse perspective. *Advances in Neural Information Processing Systems*, 35:31697–31710, 2022b.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834, 2021.

 https://doi.org/10.52202/079017-4015

A Experimental details for Section 3.2

A.1 MuJoCo

For Reacher and Swimmer environments, the datasets come from an open-source repository [Gallouédec et al., 2024] and contain expert data collected by a policy trained by PPO [Schulman et al., 2017]. The hopper dataset is part of the D4RL datasets [Fu et al., 2020], a well-acknowledged benchmark for offline reinforcement learning research. Table 2 summarizes all model hyperparameters and experimental settings used in section 3.2. In all experiments, we train the models long enough so that the model weights converge. We provide more details below about the MuJoCo datasets employed and some hyperparameter settings depending on each dataset.

Table 2: Hyperparameter settings for experiments with weight decay on MuJoCo datasets.

	Hyperparameter	**Value**
Model Architecture	Number of hidden layers	3
	Hidden layer dimension	256
	Activation function	ReLU
	Number of linear projection layer ($\mathbf{W}$)	1
Training	Epochs	1.5e6, Reacher 1e6, Swimmer 2e5, Hopper
	Batch size	256
	Optimizer	SGD
	Learning rate	1e-2
	Weight decay	1.5e-3, Reacher 1e-2, Swimmer 1e-2, Hopper
	Seeds	0, 1, 2
	Compute resources	Intel(R) Xeon(R) Platinum 8268 CPU
	Number of CPU compute workers	4
	Requested compute memory	16 GB
	Approximate average execution time	16 hours

MuJoCo environment descriptions We use expert data obtained from Gallouédec et al. [2024] and Fu et al. [2020] for the Reacher, Swimmer, and Hopper environments. Reacher is a robot arm with two joints; the goal of this environment is to control the tip of this arm to reach a randomly generated target point in a 2-dimensional plane. Swimmer is a linear-chain-like robot with three different body parts connected by two rotors; the goal of Swimmer is to move forward on a 2-dimensional plane as fast as possible. Similarly, Hopper is a 2-dimensional one-legged robot with four body parts, and the goal is to hop forward as fast as possible. All three simulated robots are controlled by applying torques on the joints connecting the body parts. Those torques are therefore the actions. In creating the datasets, online reinforcement learning was used to find expert policies [Gallouédec et al., 2024, Fu et al., 2020]. To generate the offline expert datasets, the expert policy is then applied to the environment to generate episodes consisting of states $\mathbf{x}_i$ and actions (that is, targets) $\mathbf{y}_i$. The state $\mathbf{x}_i$ includes robot positions, and angle, velocity, and angular velocity of all robot joints, and the targets $\mathbf{y}_i$ include the torques on joints.

Low data regime Using regularized regression to train a neural network with expert state-action data is often referred to as *imitation learning*. In this paper, we follow the common practice of using relatively small MLP architectures for the MuJoCo environments [Tarasov et al., 2022]. In imitation learning, it is desirable to learn a good policy with as little expert data as possible. We therefore train the models with subsets of the expert data in the datasets for each of the three environments. Specifically, we use 20, 1, and 10 episodes (complete expert demonstrations) for Reacher, Swimmer, and Hopper, respectively. This corresponds to 1,000, 1,000, and 10,000 data points for the three environments, respectively. For each environment, we also take a subset of the full validation (test) dataset and keep the number of data 20% of training data size. As we are using fewer full expert

demonstrations for Swimmer, we increase the weight decay value to further mitigate overfitting in this case.

A.2 CARLA and UTKface

The Carla dataset is collected by recording surroundings via automotive cameras, while a human driver operates a vehicle in a simulative urban environment [Codevilla et al., 2018]. The recorded images are states $\mathbf{x}_i$ of the vehicle and the expert control from the driver, which includes speed and steering angles, serves as actions $\mathbf{y}_i \in [0, 85] \times [-1, 1]$ in the dataset. A well-trained model on this dataset is expected to drive the vehicle safely in the virtual environment. The UTKface dataset consists of full-face photographs $\mathbf{x}_i$ of humans whose ages range from 1 to 116 [Zhang et al., 2017]. The goal of this dataset is to accurately predict the age $\mathbf{y}_i$ of the person in each photo.

In both cases, ResNet network [He et al., 2016] is employed as the model backbone to extract image features. And the full dataset is used for training both models, as learning a good feature extractor from visual inputs requires a large number of images [He et al., 2016, Sun et al., 2017]. To adapt ResNet architecture, a native of classification tasks, to regression tasks, we replace the last layer classifier with a fully connected layer to map learned features to the continuous targets. Depending on the task complexity, we select ResNet18 for Carla and ResNet34 for UTKface. The experimental setup for CARLA 1D/2D and UTKface datasets are summarized in Table 3.

Table 3: Hyperparameters of ResNet for Carla and UTKface datasets.

	Hyperparameter	**Value**
Architecture	Backbone of hidden layers	ResNet18, Carla ResNet34, UTKface
	Last layer hidden dim	512
	Final layer activation function	ReLU
Training	Epochs	100
	Batch size	512
	Optimizer	SGD
	Momentum	0.9
	Learning rate	0.001
	Multistep_gamma	0.1
	Seeds	0, 1
	Compute resources	NVIDIA A100 8358 80GB
	Number of compute workers	8
	Requested compute memory	200 GB
	Approximate average execution time	42 hours

A.3 Computing NRC3

For univariate regression, note that $\mathbf{W}\mathbf{W}^T = ||\mathbf{w}||_2^2$, and $\mathbf{\Sigma}^{1/2} - \gamma^{1/2}\mathbf{I}_n = \sigma - \gamma^{1/2}$, where $\mathbf{w}$ is a vector of the final linear layer of the model; σ is the standard deviation of the one-dimensional targets. Thus, NRC3 is trivially zero. Alternatively, to align with the theory in Section 4.2, one may define one-dimensional NRC3 as:

$$\text{NRC3} = \left| ||\mathbf{w}||_2^2 - \gamma_2(\sigma - \gamma_1^{1/2}) \right|^2 \to 0,$$

for some $\gamma_1 \in (0, \lambda_{\min})$ and $\gamma_2 > 0$. However, this is also trivially true, e.g. for any $\gamma_1 \in (0, \sigma^2)$ (note that $\lambda_{\min} = \sigma^2$) and $\gamma_2 = ||\mathbf{w}||_2^2(\sigma - \gamma_1^{1/2})^{-1}$ after parameters $\mathbf{w}$ become stable. Therefore, we found NRC3 for univariate regression to be not as meaningful, and therefore omitted the corresponding plots.

For multivariate regression, we run all experiments long enough in order to ensure that the training has entered the terminal phase of training as measured by R^2 (see Figure 2). After training, we extract the $\mathbf{W}$ matrix and identify γ that minimizes NRC3 for that specific $\mathbf{W}$. This γ was then used to compute the NRC3 metric for all $\mathbf{W}$ matrices during training, resulting in the NRC3 curves shown in Figure 2. Figure 4 visualizes NRC3 as a function of γ for the final trained $\mathbf{W}$.

 https://doi.org/10.52202/079017-4015

In Appendix F, we show that under a condition that is satisfied if λ_{WD} is reasonably large, a non-normalized version of NRC3, see (32), is convex and it has a unique minimum. Since we employ relatively large weight decay for experiments in Figure 2, the condition of Theorem F.1 is satisfied, and thus Figure 4 displays a unique optimal γ for all datasets.

A.4 Results for small weight decay

Figure 7 and Figure 8 include results on studying small weight decay values for all datasets. When weight decay approaches zero, NRC1-3 typically become larger, compared with NRC1-3 obtained with larger weight decay values.

Particularly, when there is no weight decay, we observe that NRC1-3 has a strong tendency to converge (There is a relatively small amount of collapse since gradient descent tends to seek solutions with small norms.), while the test MSE increases on small MuJoCo datasets. Theorem 4.3 provides some insight: when there is no regularization, there is an infinite number of non-collapsed optimal solutions under UFM; whereas Theorem 4.1 shows that when there is regularization, all solutions are collapsed. When there is regularization, we are seeking a small norm optimal solution, which leads to NRC1-3.

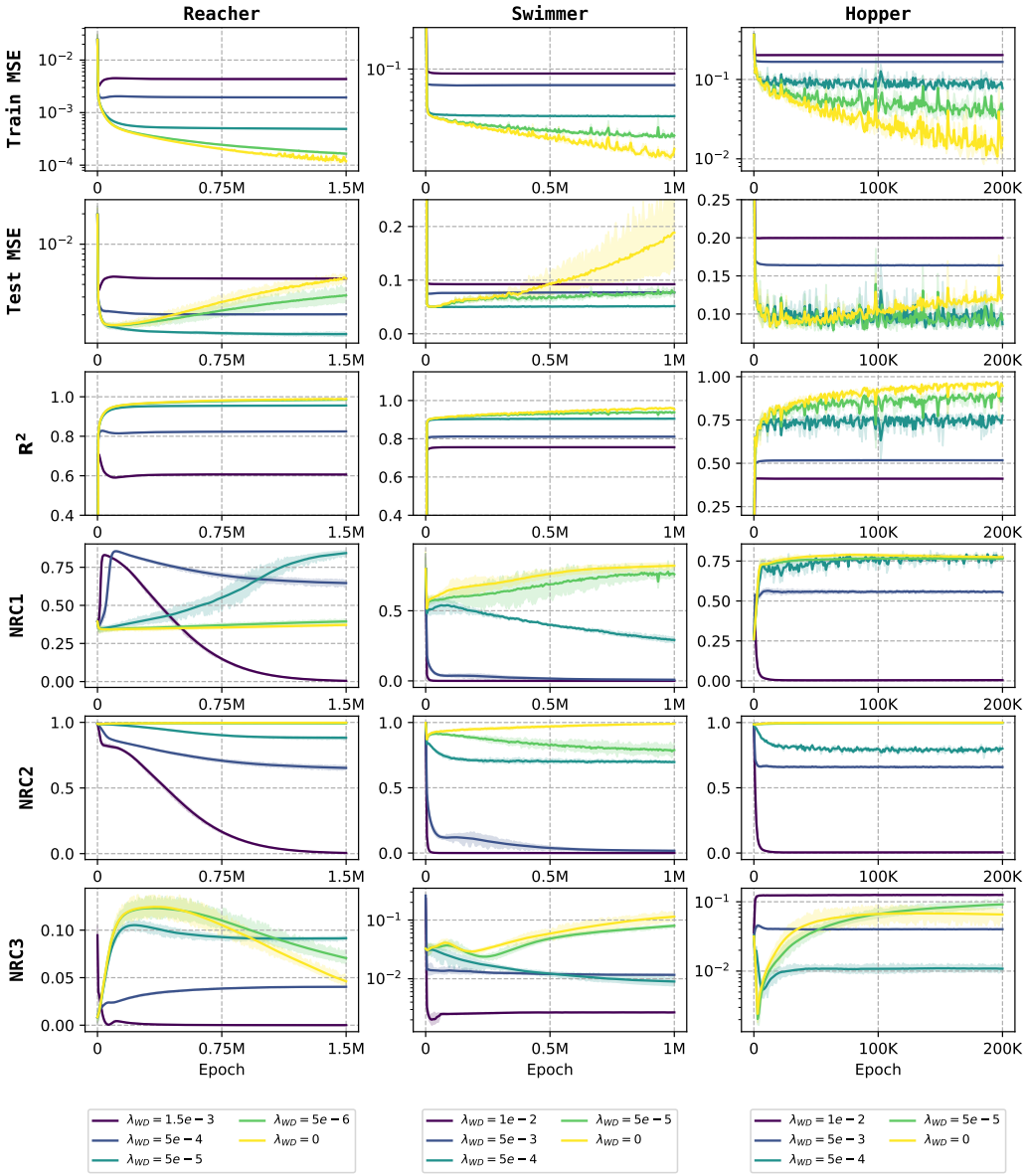

Figure 7: Train/Test MSE, R^2, and NRC1-3 under different weight decays for MuJoCo datasets.

https://doi.org/10.52202/079017-4015

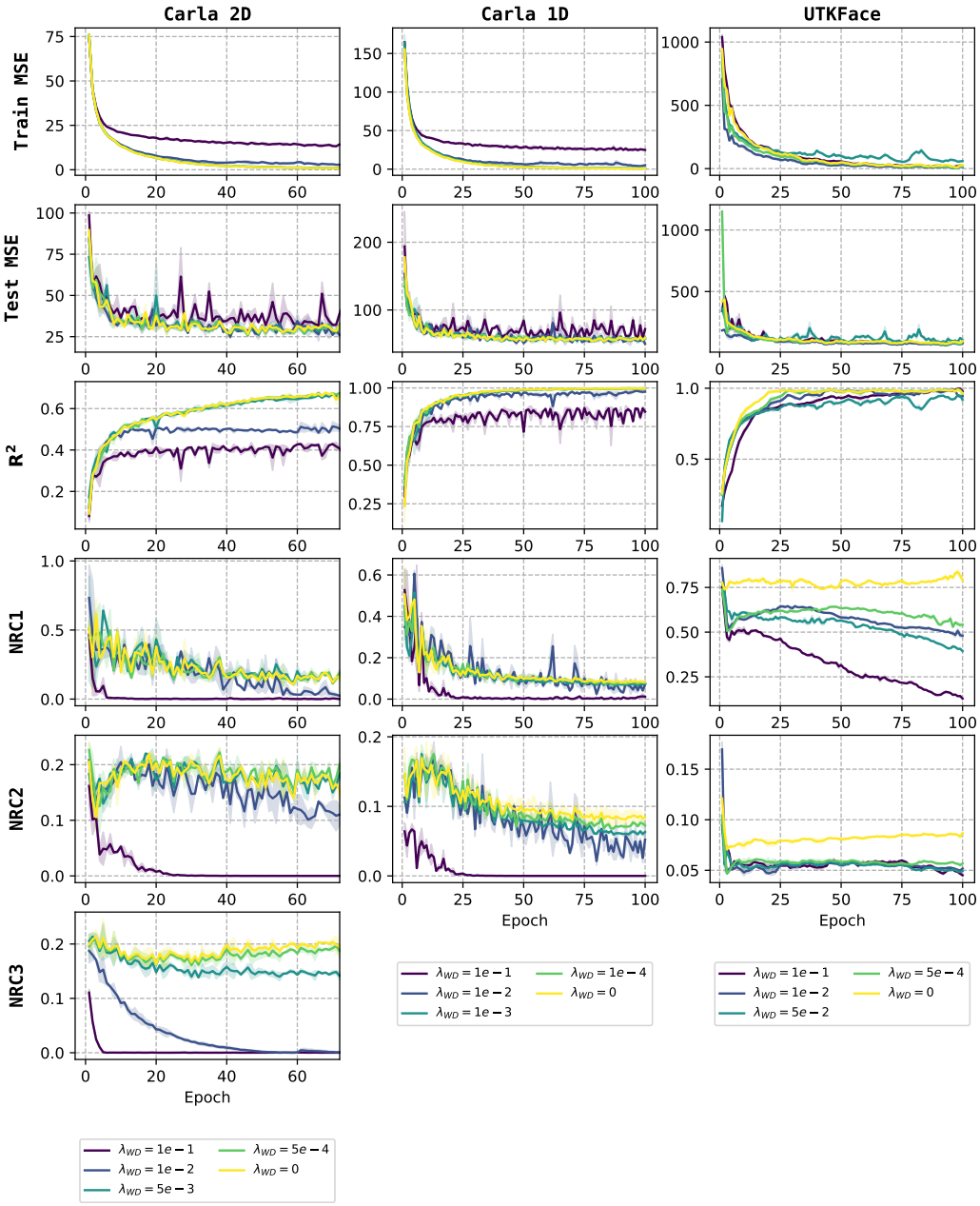

Figure 8: Train/Test MSE, R^2, and NRC1-3 under different weight decays for Carla and UTKFace datasets.

B Additional experimental results

In this section, we delve into additional experiments aimed at further exploring the phenomena of neural regression collapse.

Complete experiments under UFM assumption As studied in Section 4.5, we run experiments that align with the UFM assumption and verify the theoretical NRC1-3. In addition to the L2 regularization on both $\mathbf{H}$ and $\mathbf{W}$, the model thus is empowered with more expressive learned feature $\mathbf{H}$, e.g. removing ReLU in the penultimate layer to allow negative feature values and incorporating more training data. Complete results for all six datasets are shown in Figure 9. As we can see, NRC1-3 do not converge to very low values as is when regularization is stronger. This confirms our NRC theory in Section 4 and also corresponds to the results observed in Figure 7 and Figure 8 where we apply weight decay to all model parameters in practice.

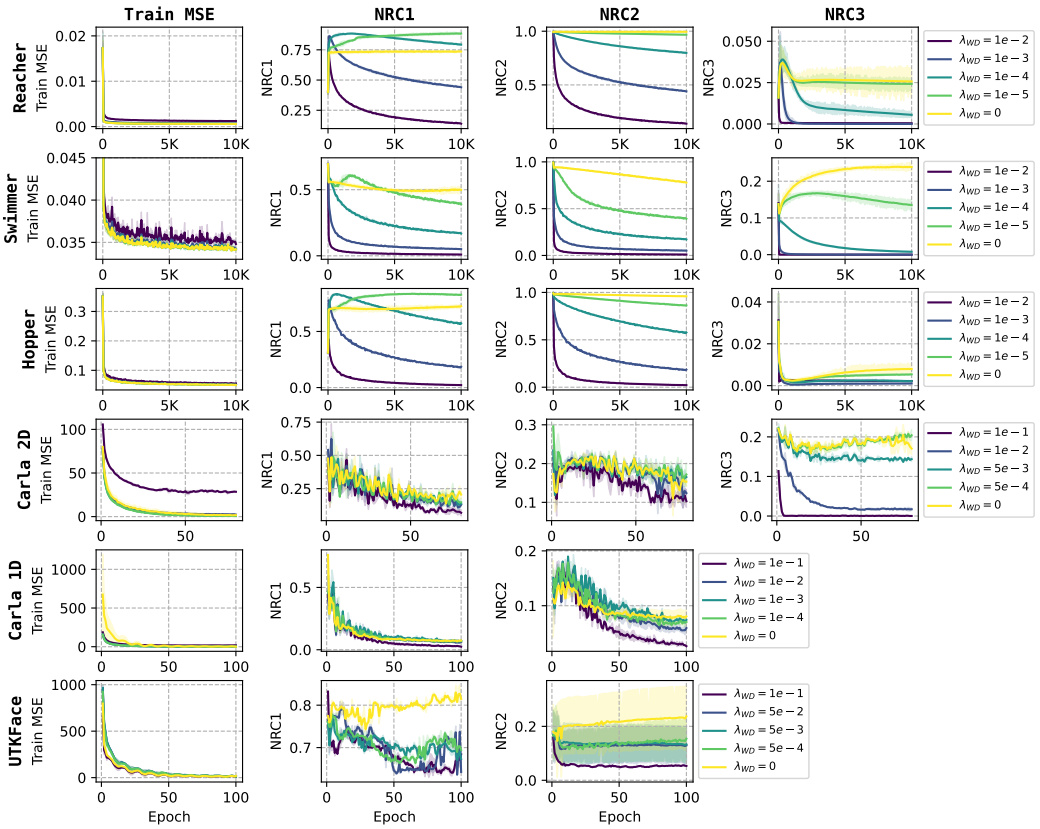

Figure 9: Empirical results with UFM assumption where L2 regularization on $\mathbf{H}$ and $\mathbf{W}$ are used instead of weight decay for all six datasets.

Norms of H and W As demonstrated in Corollary 4.2(iii), the norms of the last layer weight matrix and the feature matrix depend on the ratio of regularization parameters $\lambda_{\mathbf{H}}/\lambda_{\mathbf{W}}$. In Figure 10, we empirically demonstrate how the regularization parameters impact the norms of the last layer weight matrix and the feature matrix. Specifically, we fixed $\lambda_{\mathbf{W}} = 0.01$ and varied the value of $\lambda_{\mathbf{H}}$. We observe that with increasing $\lambda_{\mathbf{H}}$, the feature norm monotonically decreases, and the norms of the rows of the weight matrix monotonically increase, which is consistent with our theoretical result.

Connection to whitening In statistical analysis, whitening (or sphering) refers to a common preprocessing step to transform random variables to orthogonality. A whitening transformation (or sphering transformation) is a linear transformation that transforms a set of vectors of random variables with a known covariance matrix into a new set of vectors of random variables such that the components of the transformed vectors are uncorrelated and have unit variances. The transformation

https://doi.org/10.52202/079017-4015

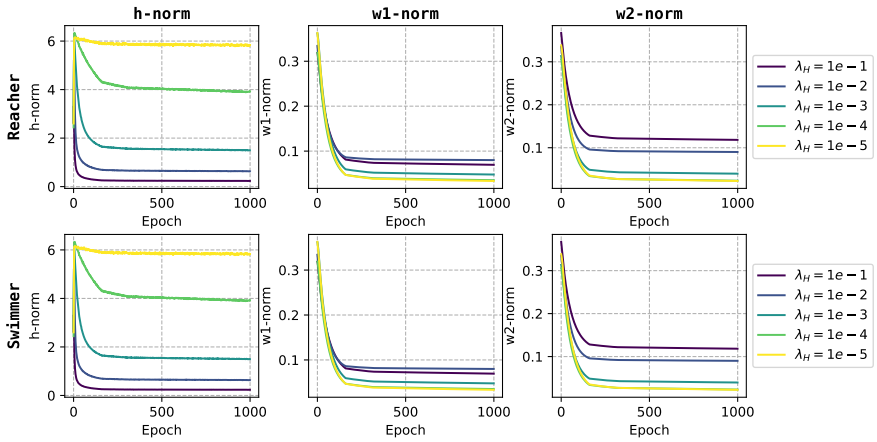

Figure 10: Comparison of the norms of $\mathbf{W}$ and $\mathbf{H}$ with fixed $\lambda_{\mathbf{W}}$ and varying $\lambda_{\mathbf{H}}$. The columns from left to right represent the model's average feature norm and the norms for $\mathbf{w}_1$ and $\mathbf{w}_2$, respectively.

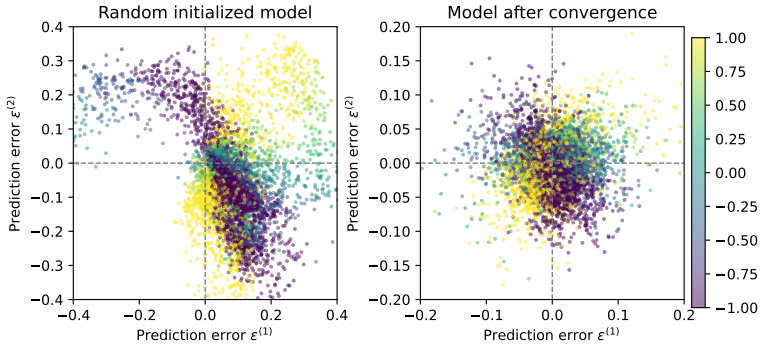

Figure 11: Residual errors $\varepsilon^{(2)}$ versus $\varepsilon^{(1)}$ for both the randomly initialized model and the trained model after convergence on the Reacher dataset. The color of the points indicates the ratio $z^{(2)}/z^{(1)}$.

is called "whitening" because it changes the input vector to white noise. Due to rotational freedom, there are infinitely many possible whitening methods, and consequently there is a diverse range of whitening procedures in use such as PCA whitening, Cholesky whitening, and Mahalonobis or zero-phase component analysis (ZCA) whitening among others. In the latter, the matrix used for the procedure of whitening is $\mathbf{W}^{ZCA} = [\mathbf{\Sigma}^{1/2}]^{-1}$, where $\mathbf{\Sigma}$ is the covariance matrix of the original data. Interestingly, $\mathbf{W}^{ZCA}$ is obtained as the whitening transformation that minimizes the MSE between the original and the whitened data, which is appealing since in many applications, it is desirable to remove correlations with minimal additional adjustment, with the aim that the transformed data remains as similar as possible to the original data.

From Corollary 4.2(v), we get the residual error of the regression model, $\mathbf{E} \in \mathbb{R}^{n \times M}$, which can be formulated as

$$\mathbf{E} = -\sqrt{c}[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}).$$

The residual error matrix is proportional to the ZCA-whitened centered target matrix. If we denote the ZCA-whitened target matrix by

$$\mathbf{Z}^{ZCA} = [\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}),$$

we have that

$$M^{-1}\mathbf{Z}^{ZCA}(\mathbf{Z}^{ZCA})^T = \mathbf{I}_n.$$

As a consequence, the residual error matrix can be represented as $\mathbf{E} = -\sqrt{c}\mathbf{Z}^{ZCA}$, from which a significant implication arises. After the model converges, the residual error matrix will be mean zero white noise with sample covariance matrix given by $M^{-1}\mathbf{E}\mathbf{E}^T = c\mathbf{I}_n$, i.e., the errors are uncorrelated across the n target dimensions and each has variance equal to c.

For any given sample, upon examining individual dimensions, it becomes apparent that the j-th dimension of the residual error, $\varepsilon^{(j)}$, is proportional to the j-th dimension of the standardized target variable $z^{(j)}$. We trained a 4-layer MLP model on the Reacher dataset for which the target variable is 2-dimensional to validate the above-mentioned properties. We created scatter plots of the residual error $\varepsilon^{(2)}$ versus $\varepsilon^{(1)}$ for both the randomly initialized model and the trained model after convergence. Figure 11 illustrates these scatter plots, with the color of the samples indicating $z^{(2)}/z^{(1)}$. As observed from the plot, after the model converges, the residual errors reduce to white noise. Additionally, from the right plot (for the model after convergence), we observe that the plot exhibits a circular pattern where the color of the samples gradually changes as they move from one quadrant to another. This indicates the consistency between $\varepsilon^{(2)}/\varepsilon^{(1)}$ and $z^{(2)}/z^{(1)}$, which is consistent with the result in Corollary 4.2(v).

 https://doi.org/10.52202/079017-4015

C Supplementary lemmas

Let us recall the form of the objective:

$$\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b}) = \frac{1}{2M}||\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y}||_F^2 + \frac{\lambda_{\mathbf{H}}}{2M}||\mathbf{H}||_F^2 + \frac{\lambda_{\mathbf{W}}}{2}||\mathbf{W}||_F^2, \tag{7}$$

where $\mathbf{1}_M^T = [1 \cdots 1]$ and $\lambda_{\mathbf{H}}, \lambda_{\mathbf{W}} > 0$ regularization constants.

In Lemma C.1, we demonstrate that if $(\mathbf{H}, \mathbf{W}, \mathbf{b})$ is critical for (7), then $\mathbf{W}$ can be written as a closed-form function of $\mathbf{H}$ and the residual error. In an analogous way, $\mathbf{H}$ can be written as a closed-form function of $\mathbf{W}$ and the residual error. Furthermore, $\mathbf{b} = \bar{\mathbf{y}}$, where $\bar{\mathbf{y}}$ is the mean of the targets. In addition, we provide the identity that connects the matrix norms of the two, see (iii) below.

Lemma C.1. *i) If* $(\mathbf{H}, \mathbf{W}, \mathbf{b})$ *is a critical point of* (7)*, then*

$$\begin{aligned}
\mathbf{H} &= -\lambda_{\mathbf{H}}^{-1}\mathbf{W}^T(\mathbf{W}\mathbf{H} + \bar{\mathbf{Y}} - \mathbf{Y}),\\
\mathbf{W} &= -\frac{\lambda_{\mathbf{W}}^{-1}}{M}(\mathbf{W}\mathbf{H} + \bar{\mathbf{Y}} - \mathbf{Y})\mathbf{H}^T,\\
\mathbf{b} &= \bar{\mathbf{y}}.
\end{aligned}$$

ii) If $(\mathbf{H}, \mathbf{W}, \mathbf{b})$ *is a critical point of* (7)*, for fixed* $(\mathbf{H}, \mathbf{W})$*,* $\mathbf{b} = \bar{\mathbf{y}}$ *minimizes* $\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b})$.
iii) $\lambda_{\mathbf{H}}||\mathbf{H}||_F^2 = M\lambda_{\mathbf{W}}||\mathbf{W}||_F^2$.

Proof. i) To prove the first part of the lemma, we will proceed by equating to zero the gradients w.r.t. the variables of the optimization objective $\mathcal{L}$. Those can be written in the form of a matrix in the following way:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{H}} = \mathbf{W}^T\frac{1}{M}(\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y}) + \frac{\lambda_{\mathbf{H}}}{M}\mathbf{H}, \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \frac{1}{M}(\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y})\mathbf{H}^T + \lambda_{\mathbf{W}}\mathbf{W}, \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}} = \frac{1}{M}(\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y})\mathbf{1}_M. \tag{10}$$

We set $\frac{\partial \mathcal{L}}{\partial \mathbf{b}} = \mathbf{0}$ in (10) and observe that

$$\mathbf{b} = \frac{1}{M}(\mathbf{Y} - \mathbf{W}\mathbf{H})\mathbf{1}_M = \frac{\mathbf{Y}\mathbf{1}_M}{M} - \mathbf{W}\frac{\mathbf{H}\mathbf{1}_M}{M} = \bar{\mathbf{y}} - \mathbf{W}\bar{\mathbf{h}}, \tag{11}$$

recalling that $\bar{\mathbf{y}} = M^{-1}\sum_{i=1}^M \mathbf{y}_i$ and $\bar{\mathbf{h}} = M^{-1}\sum_{i=1}^M \mathbf{h}_i$.

We set $\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \mathbf{0}$ in (9) and observe that

$$\lambda_{\mathbf{W}}\mathbf{W} = -\frac{1}{M}(\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y})\mathbf{H}^T. \tag{12}$$

We set $\frac{\partial \mathcal{L}}{\partial \mathbf{H}} = \mathbf{0}$ in (8) and observe that

$$\lambda_{\mathbf{H}}\mathbf{H} = -\mathbf{W}^T(\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y}), \tag{13}$$

$$\lambda_{\mathbf{H}}\mathbf{h}_i = -\mathbf{W}^T(\mathbf{W}\mathbf{h}_i + \mathbf{b} - \mathbf{y}_i), \qquad \forall i = 1, .., M, \tag{14}$$

$$\lambda_{\mathbf{H}}\bar{\mathbf{h}} = -\mathbf{W}^T(\mathbf{W}\bar{\mathbf{h}} + \mathbf{b} - \bar{\mathbf{y}}). \tag{15}$$

We derived (15) by summing both sides of (14) over i, and subsequently dividing them by M. Substituting $\mathbf{b}$ for $\bar{\mathbf{y}} - \mathbf{W}\bar{\mathbf{h}}$, see (11), we get

$$\bar{\mathbf{h}} = \mathbf{0}, \qquad \mathbf{b} = \bar{\mathbf{y}}. \tag{16}$$

Thus, combining (12), (13), and (16) completes the first part of the proof of i).

ii) If $(\mathbf{H}, \mathbf{W}, \mathbf{b})$ is a critical point of (7), noting that for fixed $(\mathbf{H}, \mathbf{W})$, the objective $\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b})$ is convex w.r.t. $\mathbf{b}$, readily yields that $\mathbf{b} = \bar{\mathbf{y}}$ minimizes $\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b})$.

iii) Let us now verify that $\lambda_{\mathbf{H}}||\mathbf{H}||_F^2 = M\lambda_{\mathbf{W}}||\mathbf{W}||_F^2$. If $(\mathbf{H}, \mathbf{W}, \mathbf{b})$ is a critical point, then

$$\frac{\partial \mathcal{L}}{\partial \mathbf{H}}\mathbf{H}^T - \mathbf{W}^T\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = 0. \tag{17}$$

Recalling the first-order gradients of $\mathcal{L}$ w.r.t. $\mathbf{H}$ and $\mathbf{W}$ respectively, see (8) and (9), and substituting those in (17), implies

$$\left[\mathbf{W}^T\frac{1}{M}(\mathbf{W}\mathbf{H} + \bar{\mathbf{Y}} - \mathbf{Y}) + \frac{\lambda_{\mathbf{H}}}{M}\mathbf{H}\right]\mathbf{H}^T = \mathbf{W}^T\left[\frac{1}{M}(\mathbf{W}\mathbf{H} + \bar{\mathbf{Y}} - \mathbf{Y})\mathbf{H}^T + \lambda_{\mathbf{W}}\mathbf{W}\right],$$

which gives

$$\frac{\lambda_{\mathbf{H}}}{M}\mathbf{H}\mathbf{H}^T = \lambda_{\mathbf{W}}\mathbf{W}^T\mathbf{W}.$$

By definition,

$$\frac{\lambda_{\mathbf{H}}}{M}||\mathbf{H}||_F^2 = \frac{\lambda_{\mathbf{H}}}{M}\text{tr}(\mathbf{H}^T\mathbf{H}) = \frac{\lambda_{\mathbf{H}}}{M}\text{tr}(\mathbf{H}\mathbf{H}^T) = \lambda_{\mathbf{W}}\text{tr}(\mathbf{W}^T\mathbf{W}) = \lambda_{\mathbf{W}}||\mathbf{W}||_F^2,$$

and this establishes the assertion $\lambda_{\mathbf{H}}||\mathbf{H}||_F^2 = M\lambda_{\mathbf{W}}||\mathbf{W}||_F^2$.

□

Next, we touch upon various implications of Theorem 4.1 in the case when $0 < c < \lambda_{\min}$, so that $[\mathbf{A}^{1/2}]_{j*} = \mathbf{A}^{1/2}$, where

$$\mathbf{A} = \mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n. \tag{18}$$

For convenience, let us break the form of a global minimum $(\mathbf{H}, \mathbf{W}, \mathbf{b})$, see (3), into three parts.

$$\mathbf{W} = \left(\frac{\lambda_{\mathbf{H}}}{\lambda_{\mathbf{W}}}\right)^{1/4}\mathbf{A}^{1/2}\mathbf{R}, \tag{19}$$

$$\mathbf{H} = \left(\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}\right)^{1/4}\mathbf{R}^T\mathbf{A}^{1/2}[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}), \tag{20}$$

$$\mathbf{b} = \bar{\mathbf{y}}, \tag{21}$$

where $\mathbf{R} \in \mathbb{R}^{n\times d}$ is semi-orthogonal, i.e., $\mathbf{R}\mathbf{R}^T = \mathbf{I}_n$. In the following lemma, we demonstrate that the residual error is a rescaled version of the centered targets, the value of the loss function at the global minimum can be computed directly by invoking the value of c and the matrix norm of $\mathbf{A}^{1/2}$.

Lemma C.2. *Suppose $0 < c < \lambda_{\min}$. For a global minimum $(\mathbf{H}, \mathbf{W}, \mathbf{b})$ of (7), we have that the residual error takes the following form:*

$$\mathbf{W}\mathbf{H} + \bar{\mathbf{Y}} - \mathbf{Y} = -\sqrt{c}[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}),$$

Moreover, the value of the loss function $\mathcal{L}$ at the global minimum is calculated as

$$\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b}) = \frac{nc}{2} + \sqrt{c}||(\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n)^{\frac{1}{2}}||_F^2.$$

Proof. By (19)-(21), for the first assertion,

$$\begin{aligned}\mathbf{W}\mathbf{H} + \bar{\mathbf{Y}} - \mathbf{Y} &= \mathbf{A}^{1/2}\mathbf{R}\mathbf{R}^T\mathbf{A}^{1/2}[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}) - (\mathbf{Y} - \bar{\mathbf{Y}}) \\ &= \left[\mathbf{A}[\mathbf{\Sigma}^{1/2}]^{-1} - \mathbf{I}_n\right](\mathbf{Y} - \bar{\mathbf{Y}}) \\ &= -\sqrt{c}[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}),\end{aligned}$$

noting, for the first equality, that $\mathbf{R}\mathbf{R}^T = \mathbf{I}_n$, $\mathbf{A}^{1/2}\mathbf{A}^{1/2} = \mathbf{A}$. For the third equality, see (18). Therefore,

$$\frac{1}{M}(\mathbf{W}\mathbf{H} + \bar{\mathbf{Y}} - \mathbf{Y})(\mathbf{W}\mathbf{H} + \bar{\mathbf{Y}} - \mathbf{Y})^T = c[\mathbf{\Sigma}^{1/2}]^{-1}\mathbf{\Sigma}[\mathbf{\Sigma}^{1/2}]^{-1} = c\mathbf{I}_n. \tag{22}$$

Using Lemma C.1(iii) and (22), we deduce

$$\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b}) = \frac{nc}{2} + \lambda_{\mathbf{W}}||\mathbf{W}||_F^2 = \frac{nc}{2} + \lambda_{\mathbf{W}}\sqrt{\frac{\lambda_{\mathbf{H}}}{\lambda_{\mathbf{W}}}}\text{tr}(\mathbf{A}) = \frac{nc}{2} + \sqrt{c}||\mathbf{A}^{1/2}||_F^2,$$

which completes the proof of the lemma.

□

The proof of Corollary 4.2 directly follows:

 https://doi.org/10.52202/079017-4015

C.1 Proof of corollary 4.2

(i) is derived in the proof of Lemma C.1, see (14) and (16). It is also easy to derive them from the form of $\mathbf{H}$ as given in (3). (ii) follows by the form $\mathbf{W}$, see (3). By Lemma C.1(iii) and Lemma C.2, (iii)-(v) follow immediately.

D Proof of theorem 4.1

The proof of Theorem 4.1 leverages [Zhou et al., 2022a, Lemma B.1]. For clarity, we now restate their lemma in our notation.

Lemma D.1. *[Zhou et al., 2022a, Lemma B.1] For n, d, M with $d \geq n$, and $\tilde{\mathbf{Y}} := \mathbf{Y} - \bar{\mathbf{Y}} \in \mathbb{R}^{n\times M}$ with SVD given by $\tilde{\mathbf{Y}} = U\tilde{\mathbf{\Sigma}}V^T = \sum_{i=1}^n \sigma_i u_i v_i^T$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ are the singular values, the following problem*

$$\min_{\mathbf{H}\in\mathbb{R}^{d\times M}, \mathbf{W}\in\mathbb{R}^{n\times d}} \mathcal{L}(\mathbf{H}, \mathbf{W}, \bar{\mathbf{y}})$$

is a strict saddle function with no spurious local minima, in the sense that
i) Any local minimum $(\mathbf{H}, \mathbf{W}, \bar{\mathbf{y}})$ of (7) is a global minimum of (7), with the following form

$$\mathbf{WH} = U[\tilde{\mathbf{\Sigma}} - \sqrt{M\lambda_{\mathbf{W}}\lambda_{\mathbf{H}}}\mathbf{I}_n]_+ V^T.$$

Correspondingly, the minimal objective value of (7) is

$$\mathcal{L}(\mathbf{H}, \mathbf{W}, \bar{\mathbf{y}}) = \frac{1}{2}\sum_{i=1}^n (\eta_i - \sigma_i)^2 + \sqrt{M\lambda_{\mathbf{W}}\lambda_{\mathbf{H}}}\sum_{i=1}^n \eta_i,$$

where $\eta_i := \eta_i(\lambda_{\mathbf{H}}, \lambda_{\mathbf{W}})$ is the i-th diagonal entry of $[\tilde{\mathbf{\Sigma}} - \sqrt{M\lambda_{\mathbf{W}}\lambda_{\mathbf{H}}}\mathbf{I}_n]_+$.
ii) Any critical point $(\mathbf{H}, \mathbf{W}, \bar{\mathbf{y}})$ that is not a local minimum is a strict saddle point with negative curvature, i.e., the Hessian at this critical point has at least one negative eigenvalue.

Let $\tilde{\mathbf{Y}} = \mathbf{Y} - \bar{\mathbf{Y}} = U\tilde{\mathbf{\Sigma}}V^T = \sum_{i=1}^n \sigma_i u_i v_i^T$, denote the compact SVD of $\tilde{\mathbf{Y}} \in \mathbb{R}^{n\times M}$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$ are the singular values, and $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{n\times n}$ is diagonal, containing the aforementioned singular values. Furthermore, $U \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{M\times n}$ are orthogonal and semi-orthogonal respectively, i.e., $UU^T = U^TU = \mathbf{I}_n$ and $V^TV = \mathbf{I}_n$ respectively. For the proof, recall the value of $c = \lambda_{\mathbf{W}}\lambda_{\mathbf{H}}$.

Proof of Theorem 4.1. Let $(\mathbf{H}, \mathbf{W}, \bar{\mathbf{y}})$ be a global minimum of (1). By Lemma D.1, $(\mathbf{H}, \mathbf{W}, \bar{\mathbf{y}})$ has the following form:

$$\mathbf{WH} = U[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n]_+ V^T. \tag{23}$$

In light of Lemma C.1 and the identity $\lambda_{\mathbf{H}}||\mathbf{H}||_F^2 = M\lambda_{\mathbf{W}}||\mathbf{W}||_F^2$, from (23), we have that

$$\mathbf{W} = \left(\frac{\lambda_{\mathbf{H}}}{M\lambda_{\mathbf{W}}}\right)^{1/4} U[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n]_+^{\frac{1}{2}}\mathbf{R}, \tag{24}$$

$$\mathbf{H} = \left(\frac{M\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}\right)^{1/4} \mathbf{R}^T[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n]_+^{\frac{1}{2}} V^T, \tag{25}$$

for all $\mathbf{R} \in \mathbb{R}^{n\times d}$ such that $\mathbf{R}\mathbf{R}^T = \mathbf{I}_n$. Furthermore, using the SVD of $\tilde{\mathbf{Y}} = U\tilde{\mathbf{\Sigma}}V^T$,

$$\mathbf{\Sigma} = \frac{\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T}{M} = U\frac{\tilde{\mathbf{\Sigma}}}{\sqrt{M}}V^TV\frac{\tilde{\mathbf{\Sigma}}}{\sqrt{M}}U^T = U\left[\frac{\tilde{\mathbf{\Sigma}}}{\sqrt{M}}\right]^2 U^T,$$

which deduces $\mathbf{\Sigma}^{1/2} = U\frac{\tilde{\mathbf{\Sigma}}}{\sqrt{M}}U^T$. Since $U^T = U^{-1}$, this further yields

$$\sqrt{M}[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n] = U\left[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n\right]U^{-1}, \tag{26}$$

which implies that the matrices $\sqrt{M}[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n]$ and $\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n$ are similar. As a result, they have the same eigenvalues. The $n \times n$ matrix on the left-hand side of (26) has eigenvalues given by

$\sqrt{M\lambda_i} - \sqrt{Mc}$, $i = 1, ..., n$, where λ_i is the i-th eigenvalue (in descending order) of $\mathbf{\Sigma}$ whereas $\sigma_i - \sqrt{Mc}$, $i = 1, ..., n$ are the (ordered) diagonal elements of the $n \times n$ matrix on the right-hand side of (26). So,

$$\sqrt{\lambda_i} = \frac{\sigma_i}{\sqrt{M}}, \qquad \text{for all } i = 1, ..., n. \tag{27}$$

Case I: If $0 < c < \lambda_{\text{min}}$, then by (27), it is the case that $\sigma_i > \sqrt{Mc}$, $\forall i$, and thus $[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n]_+^{\frac{1}{2}} = [\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n]^{\frac{1}{2}}$. By (26),

$$U[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n]^{\frac{1}{2}} = M^{1/4}[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n]^{\frac{1}{2}}U, \tag{28}$$

and thus (24) becomes

$$\mathbf{W} = \left(\frac{\lambda_{\mathbf{H}}}{\lambda_{\mathbf{W}}}\right)^{1/4} [\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n]^{\frac{1}{2}}\tilde{\mathbf{R}},$$

for $\tilde{\mathbf{R}} := U\mathbf{R} \in \mathbb{R}^{n \times d}$ semi-orthogonal. The first assertion of the theorem follows by recalling the definition of $\mathbf{A} = \left[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n\right]$.

For the second assertion of the theorem, it remains to observe that

$$[\mathbf{\Sigma}^{1/2}]^{-1}\tilde{\mathbf{Y}} = \sqrt{M}U\tilde{\mathbf{\Sigma}}^{-1}U^TU\tilde{\mathbf{\Sigma}}V^T = \sqrt{M}UV^T.$$

So, $V^T = M^{-1/2}U^T[\mathbf{\Sigma}^{1/2}]^{-1}\tilde{\mathbf{Y}}$, and from (25), we get

$$\begin{aligned}
\mathbf{H} &= \left(\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}\right)^{1/4} \mathbf{R}^T M^{-1/4}[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n]^{\frac{1}{2}}U^T[\mathbf{\Sigma}^{1/2}]^{-1}\tilde{\mathbf{Y}} \\
&= \left(\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}\right)^{1/4} \tilde{\mathbf{R}}^T[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n]^{\frac{1}{2}}[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}) \\
&= \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}\mathbf{W}^T[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}),
\end{aligned}$$

where the second equality follows from (28).

Case II: If $c > \lambda_{\text{max}}$, then by (27), it is the case that $\sigma_i < \sqrt{Mc}$, $\forall i$, and thus $[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}]_+^{\frac{1}{2}} = \mathbf{0}$. By (24) and (25), $(\mathbf{H}, \mathbf{W}, \bar{\mathbf{y}}) = (\mathbf{0}, \mathbf{0}, \bar{\mathbf{y}})$ as desired.

Case III: If $\lambda_{\text{min}} < c < \lambda_{\text{max}}$, by (27), it is the case that

$$[\sigma_i - \sqrt{Mc}]_+ = \begin{cases} \sigma_i - \sqrt{Mc}, & \text{if } i \leq j* \\ 0, & \text{otherwise,} \end{cases}$$

where $j* = \max\{j : \lambda_j \geq c\}$, and thus $[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}]_+^{\frac{1}{2}} = [\tilde{\mathbf{\Sigma}} - \sqrt{Mc}]_{j*}^{\frac{1}{2}}$. By (26),

$$U\left[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n\right]_{j*}^{1/2} = M^{1/4}\left[\left[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n\right]^{\frac{1}{2}}\right]_{j*}U, \tag{29}$$

and thus (24) becomes

$$\mathbf{W} = \left(\frac{\lambda_{\mathbf{H}}}{\lambda_{\mathbf{W}}}\right)^{1/4} \left[\left[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n\right]^{\frac{1}{2}}\right]_{j*} \tilde{\mathbf{R}},$$

for $\tilde{\mathbf{R}} := U\mathbf{R} \in \mathbb{R}^{n \times d}$ semi-orthogonal. The first assertion of the theorem follows by recalling the definition of $\mathbf{A} = \left[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n\right]$.

For the second assertion of the theorem, it remains to observe that

$$[\mathbf{\Sigma}^{1/2}]^{-1}\tilde{\mathbf{Y}} = \sqrt{M}U\tilde{\mathbf{\Sigma}}^{-1}U^TU\tilde{\mathbf{\Sigma}}V^T = \sqrt{M}UV^T.$$

So, $V^T = M^{-1/2}U^T[\mathbf{\Sigma}^{1/2}]^{-1}\tilde{\mathbf{Y}}$, and from (25), we get

$$\begin{aligned}
\mathbf{H} &= \left(\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}\right)^{1/4} \mathbf{R}^T M^{-1/4}\left[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n\right]_{j*}^{\frac{1}{2}} U^T[\mathbf{\Sigma}^{1/2}]^{-1}\tilde{\mathbf{Y}} \\
&= \left(\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}\right)^{1/4} \tilde{\mathbf{R}}^T\left[\left[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n\right]^{\frac{1}{2}}\right]_{j*}[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}) \\
&= \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}\mathbf{W}^T[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}),
\end{aligned}$$

 https://doi.org/10.52202/079017-4015

where the second equality follows from (29).

□

D.1 Examples for theorem 4.1 (uncorrelated target components)

In this subsection, we examine closely the case when $n = 3$ and the target components are uncorrelated. This simplifies considerably the problem as now the covariance matrix $\mathbf{\Sigma}$ is a diagonal matrix with entries given (in order) by σ_j^2, where σ_j^2 denotes the variance of the j-th target component, for $j = 1, 2, 3$. The unique positive definite and symmetric matrix $\mathbf{A}^{1/2}$, see (18), is given by

$$\mathbf{A}^{1/2} = \begin{bmatrix} (\sigma_1 - \sqrt{c})^{\frac{1}{2}} & 0 & 0 \\ 0 & (\sigma_2 - \sqrt{c})^{\frac{1}{2}} & 0 \\ 0 & 0 & (\sigma_3 - \sqrt{c})^{\frac{1}{2}} \end{bmatrix}. \tag{30}$$

Without loss of generality, assume that $\sigma_{\text{max}} := \sigma_1 \geq \sigma_2 \geq \sigma_3 := \sigma_{\text{min}} > 0$.

- If $0 < c < \sigma_{\text{min}}^2 = \sigma_3^2$, by Theorem 4.1, $j* = 3$, and therefore any global minimum $(\mathbf{H}, \mathbf{W}, \mathbf{b})$ of (7) takes the following form:

$$\mathbf{W} = \left(\frac{\lambda_{\mathbf{H}}}{\lambda_{\mathbf{W}}}\right)^{1/4} \mathbf{A}^{1/2}\mathbf{R}, \qquad \mathbf{H} = \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}\mathbf{W}^T[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}), \qquad \mathbf{b} = \bar{\mathbf{y}},$$

  for any semi-orthogonal matrix $\mathbf{R} \in \mathbb{R}^{3\times d}$. The form of $\mathbf{A}^{1/2}$, see (30), readily yields

$$\mathbf{w}_j^T = \sqrt{\lambda_{\mathbf{H}}\left(\frac{\sigma_j}{c^{1/2}} - 1\right)}\mathbf{e}_j, \qquad j = 1, 2, 3,$$

  c.f., (4), where $\{\mathbf{e}_j : j = 1, 2, 3\}$ is any collection of vectors lying in $\mathbb{R}^d$ such that $\mathbf{e}_j$ is a unit vector, for all $j = 1, 2, 3$, and $\mathbf{e}_k$ is orthogonal to $\mathbf{e}_{k'}$, for all $k \neq k'$.
  To interpret the landscape of global minima in the case when the target components are uncorrelated, the UFM "forces" the angle between the weight matrix rows to be $\pi/2$ (fixes the weight matrix rows to be orthogonal). Then, the configuration of the bw_j's is exactly as in the 1-dimensional target case, that is those are restricted to lie on spheres of certain radiuses. The feature vector $\mathbf{h}_i$ that corresponds to the i-th training example is then on the 3-dimensional subspace spanned by $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$.
- If $\sigma_{\text{min}}^2 < c < \sigma_{\text{max}}^2$, by Theorem 4.1, $j* = 1$ or $j* = 2$. We analyze the latter, in which case $c < \sigma_1^2$, $c < \sigma_2^2$ but $c > \sigma_3^2$. By Theorem 4.1, any global minimum $(\mathbf{H}, \mathbf{W}, \mathbf{b})$ of (7) takes the form below:

$$\mathbf{W} = \left(\frac{\lambda_{\mathbf{H}}}{\lambda_{\mathbf{W}}}\right)^{1/4} [\mathbf{A}^{1/2}]_{2*}\mathbf{R}, \qquad \mathbf{H} = \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}\mathbf{W}^T[\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{Y} - \bar{\mathbf{Y}}), \qquad \mathbf{b} = \bar{\mathbf{y}},$$

  for any semi-orthogonal matrix $\mathbf{R} \in \mathbb{R}^{3\times d}$. The form of $[\mathbf{A}^{1/2}]_{2*}$, see (30), readily yields

$$\begin{aligned} \mathbf{w}_j^T &= \sqrt{\lambda_{\mathbf{H}}\left(\frac{\sigma_j}{c^{1/2}} - 1\right)}\mathbf{e}_j, \qquad j = 1, 2, \\ \mathbf{w}_3^T &= \mathbf{0}, \end{aligned}$$

  c.f., (4), where $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{R}^d$ unit vectors orthogonal to each other. It is also worth mentioning that

$$\begin{aligned} \mathbf{h}_i &= \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}\left[\mathbf{w}_1^T\ \mathbf{w}_2^T\ \mathbf{0}\right][\mathbf{\Sigma}^{1/2}]^{-1}(\mathbf{y}_i - \bar{\mathbf{y}}) \\ &= \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}\left[\frac{(\mathbf{y}_i^{(1)} - \bar{\mathbf{y}}^{(1)})}{\sigma_1}\mathbf{w}_1^T + \frac{(\mathbf{y}_i^{(2)} - \bar{\mathbf{y}}^{(2)})}{\sigma_2}\mathbf{w}_2^T\right], \end{aligned}$$

  for all $i = 1, ..., M$. In other words, for fixed i, the feature vector $\mathbf{h}_i$ lies on span$\{\mathbf{w}_1^T, \mathbf{w}_2^T\}$. Moreover, the previous formula indicates that the inner product between $\mathbf{h}_i$ and $\mathbf{w}_j^T$ is proportional to the j-th standardized target component.

The analysis of the case $j* = 1$, i.e., $c < \sigma_1^2$ but $c > \sigma_2^2$, $c > \sigma_3^2$ is analogous, therefore we just record the form of the $\mathbf{w}_j$'s and $\mathbf{h}_i$'s, omitting any further details:

$$\mathbf{w}_1^T = \sqrt{\lambda_{\mathbf{H}} \left(\frac{\sigma_1}{c^{1/2}} - 1 \right)} \mathbf{e}, \qquad \mathbf{w}_2^T = \mathbf{w}_3^T = \mathbf{0},$$

$$\mathbf{h}_i = \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}} \frac{(\mathbf{y}_i^{(1)} - \bar{\mathbf{y}}^{(1)})}{\sigma_1} \mathbf{w}_1^T,$$

for all $i = 1, ..., M$. In other words, for fixed i, the feature vector $\mathbf{h}_i$ is colinear with $\mathbf{w}_1^T$.

- If $c > \sigma_{\text{max}}^2 = \sigma_1^2$, by Theorem 4.1, $(\mathbf{H}, \mathbf{W}, \mathbf{b}) = (\mathbf{0}, \mathbf{0}, \bar{\mathbf{y}})$.

E Proof of theorem 4.3 (no regularization)

We first show

$$\min_{\mathbf{W},\mathbf{H}} L(\mathbf{W}, \mathbf{H}) = 0$$

Clearly $L(\mathbf{W}, \mathbf{H}) \geq 0$ for all $\mathbf{W}$ and $\mathbf{H}$. Now consider any fixed $n \times d$ matrix $\mathbf{W}$ with rank$(\mathbf{W}) = n$. Since $\mathbf{W}$ has rank n, $\{\mathbf{W}\mathbf{h} : \mathbf{h} \in \mathbb{R}^d\} = \mathbb{R}^n$. Thus there exists $\mathbf{h}_i \in \mathbb{R}^d$ such that $\mathbf{W}\mathbf{h}_i = \mathbf{y}_i$ for all $i = 1, \ldots, M$. Let $\mathbf{H} = [\mathbf{h}_1 \cdots \mathbf{h}_M]$. For this $\mathbf{W}$ and $\mathbf{H}$ we have $L(\mathbf{W}, \mathbf{H}) = 0$. Thus $\min_{\mathbf{W},\mathbf{H}} L(\mathbf{W}, \mathbf{H}) = 0$.

Now let $\mathbf{W}$ be any $n \times d$ matrix with full rank n, and consider the set of $\mathbf{H}$ that satisfy $L(\mathbf{W}, \mathbf{H}) = 0$ w.r.t. this $\mathbf{W}$. This is a standard least squares problem for which the solution is well known:

$$\mathbf{H} = \mathbf{W}^+\mathbf{Y} + (\mathbf{I}_d - \mathbf{W}^+\mathbf{W})\mathbf{Z} \tag{31}$$

where $\mathbf{W}^+$ is the pseudo-inverse of $\mathbf{W}$ and $\mathbf{Z}$ is any $d \times M$ matrix.

To complete the proof, we need to show that if rank$(\mathbf{W}) < n$, then $\mathbf{W}$ cannot be part of an optimal solution for $L(\mathbf{W}, \mathbf{H})$. Suppose rank$(\mathbf{W}) < n$. $\mathbf{W}\mathbf{h} = \mathbf{y}$ only has a solution if $\mathbf{y}$ lies in the column space of $\mathbf{W}$. Thus $L(\mathbf{H}, \mathbf{W}) = 0$ only if $\mathbf{y}_1, \ldots, \mathbf{y}_M$ all lie in the column space of $\mathbf{W}$. Since rank$(\mathbf{Y}) = n$ and rank$(\mathbf{W}) < n$, there will be at least one $\mathbf{y}_i$ that is not in column space of $\mathbf{W}$. Thus for this $\mathbf{y}_i$ there is no $\mathbf{h}_i$ such that $\mathbf{W}\mathbf{h}_i = \mathbf{y}_i$. Thus for this $\mathbf{y}_i$ we will have $(\mathbf{y}_i - \mathbf{W}\mathbf{h}_i)^2 > 0$, implying $\mathbf{W}$ cannot be part of an optimal solution $\mathbf{W}, \mathbf{H}$.

Finally, we note from (6) that a column $\mathbf{h}$ in $\mathbf{H}$ is the sum of two vectors, with the first vector lying in the row space of $\mathbf{W}$ and the second vector lying in the (orthogonal) null space of $\mathbf{W}$. Since $\mathbf{W}$ has full rank, this implies that the columns of $\mathbf{H}$ can span all of $\mathbb{R}^d$.

F Proof of the uniqueness of γ

Mathematically, we can show that, under a condition that is satisfied if λ_{WD} is reasonably large,

$$\text{NRC3}(\gamma) := \left|\left| \mathbf{W}\mathbf{W}^T - \mathbf{\Sigma}^{1/2} + \gamma^{1/2}\mathbf{I}_n \right|\right|_F^2, \tag{32}$$

as given in the definition of NRC3, without normalizing the Gram matrix of $\mathbf{W}$ and $\mathbf{\Sigma}^{1/2} - \gamma^{1/2}\mathbf{I}_n$, is convex and it has a unique minimum.

Theorem F.1. *If*

$$\text{tr}(\mathbf{\Sigma}^{1/2}) > \text{tr}(\mathbf{W}\mathbf{W}^T),$$

NRC3(γ)*, as given in* (32)*, is minimized at*

$$\gamma^* := \left[\frac{\text{tr}(\mathbf{\Sigma}^{1/2}) - \text{tr}(\mathbf{W}\mathbf{W}^T)}{n} \right]^2.$$

Moreover,

$$\text{NRC3}(\gamma^*) = \left|\left| \left(\mathbf{W}\mathbf{W}^T - \frac{\text{tr}(\mathbf{W}\mathbf{W}^T)}{n}\mathbf{I}_n \right) - \left(\mathbf{\Sigma}^{1/2} - \frac{\text{tr}(\mathbf{\Sigma}^{1/2})}{n}\mathbf{I}_n \right) \right|\right|_F^2.$$

https://doi.org/10.52202/079017-4015

Proof. Expanding the squared matrix norm appearing in the definition of NRC3(γ), it is necessary and sufficient to minimize

$$f(\gamma) := 2\gamma^{1/2}\left[\text{tr}(\mathbf{W}\mathbf{W}^T) - \text{tr}(\mathbf{\Sigma}^{1/2})\right] + n\gamma,$$

which has first and second derivatives given by

$$f'(\gamma) = \frac{\text{tr}(\mathbf{W}\mathbf{W}^T) - \text{tr}(\mathbf{\Sigma}^{1/2})}{\gamma^{1/2}} + n,$$
$$f''(\gamma) = \frac{\text{tr}(\mathbf{\Sigma}^{1/2}) - \text{tr}(\mathbf{W}\mathbf{W}^T)}{2\gamma^{3/2}}.$$

Since $\text{tr}(\mathbf{\Sigma}^{1/2}) > \text{tr}(\mathbf{W}\mathbf{W}^T)$, by the 2nd derivative test, f, and consequently NRC3(γ), is convex with unique minimum achieved at

$$\gamma^* := \left[\frac{\text{tr}(\mathbf{\Sigma}^{1/2}) - \text{tr}(\mathbf{W}\mathbf{W}^T)}{n}\right]^2.$$

□

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: In the abstract, we claim to empirically demonstrate the phenomenon of Neural Regression Collapse as discussed in Section 3.2, and then we provide a theoretical explanation from the perspective of the Unconstrained Feature Model, see Section 4 and Appendix B,C,D. Thus, we show that the phenomena of neural collapse could be a universal behavior in deep learning both empirically and theoretically.

   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: In Section 5, we point out that our explanation of neural regression collapse doesn't have implications for model generalization.

   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 the 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**

https://doi.org/10.52202/079017-4015

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

Answer: [Yes]

Justification: In Section 4 and Appendix B,C,D, E, we provide full set of assumptions and proofs.

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 the 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: In Sections 3.2 and Appendix A, we provide a comprehensive description of the experimental setup, including detailed information on the datasets used, model architectures, and hyperparameter settings. This detailed disclosure ensures that other researchers can reliably reproduce the experimental results and validate the claims made in the paper.

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).

(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: [Yes]

   Justification: We upload the code with environment in the supplemental materials. The datasets used are all open-source datasets.

   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: In Sections 3.2 and Appendix A, we provide a comprehensive description of the experimental setup, including detailed information on the datasets used, model architectures, and hyperparameter settings. More details can be found in the code in the supplemental materials.

   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: [No]

https://doi.org/10.52202/079017-4015

Justification: All the results are reported in terms of learning curves, and each figure includes many plots, so error bars are not reported. But we do run the experiments multiple times and observe very similar performance in terms of NRC, testing loss, etc.

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: In Appendix A, we provide the details for computation resources.

   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: We are sure that the research presented in this paper adheres to the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - 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: This paper mainly focuses on showing and understanding the neural regression collapse phenomena observed in practical neural networks and unconstrained feature models. No potential negative societal impact is expected of this work.

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]

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: [Yes]

Justification: All the owner of assets are properly cited in the reference and main body of our paper.

Guidelines:

 https://doi.org/10.52202/079017-4015

- 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]

    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]

    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**

    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]

    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.

https://doi.org/10.52202/079017-4015