NEURAL MULTIVARIATE REGRESSION: QUALITATIVE INSIGHTS FROM THE UNCONSTRAINED FEATURE MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

The Unconstrained Feature Model (UFM) is a mathematical framework that enables closed-form approximations for minimal training loss and related performance measures in deep neural networks (DNNs). This paper leverages the UFM to provide qualitative insights into neural multivariate regression, a critical task in imitation learning, robotics, and reinforcement learning. Specifically, we address two key questions: (1) How do multi-task models compare to multiple single-task models in terms of training performance? (2) Can whitening and normalizing regression targets improve training performance? The UFM theory predicts that multi-task models achieve strictly smaller training MSE than multiple single-task models when the same or stronger regularization is applied to the latter, and our empirical results confirm these findings. Regarding whitening and normalizing regression targets, the UFM theory predicts that they reduce training MSE when the average variance across the target dimensions is less than one, and our empirical results once again confirm these findings. These findings highlight the UFM as a powerful framework for deriving actionable insights into DNN design and data pre-processing strategies.

1 INTRODUCTION

Deep neural networks (DNNs) have become a cornerstone of modern machine learning, enabling transformative advancements in fields such as computer vision, natural language processing, and robotics. These models, often comprising millions to trillions of parameters, are trained by minimizing regularized loss functions over a high-dimensional parameter space. However, the non-convexity and high dimensionality of these loss functions make it practically impossible to derive closed-form expressions for their minimum value or for performance metrics, such as cross-entropy loss or mean-squared error (MSE), evaluated at a global minimum. This limitation has impeded theoretical exploration and limited our understanding of how design choices influence DNN behavior.

A recent breakthrough in this domain is the Unconstrained Feature Model (UFM) and the closely related Layer-Peeled Model (Fang et al., 2021; Mixon et al., 2022). A typical DNN consists of a nonlinear feature extractor followed by a final linear layer. Inspired by the universal approximation theorem (Cybenko, 1989; Hornik et al., 1989), the UFM assumes that the feature extractor can map any set of training examples to any desired set of feature vectors. The UFM framework further simplifies the problem by replacing L2-regularization of the feature extractor parameters with L2-regularization of the feature vectors themselves. These assumptions lead to a new optimization problem that, while still non-convex, is mathematically tractable. In particular, the UFM allows for the derivation of closed-form expressions for minimal training loss, providing a powerful theoretical tool to analyze DNN behavior.

In this paper, we explore whether the closed-form expressions derived from the UFM can yield qualitative insights into the behavior of DNNs. We focus specifically on neural multivariate regression, where the DNN predicts a vector of targets. This task is central to several important applications, including imitation learning in autonomous driving and robotics, as well as deep reinforcement learning.

For multivariate regression with MSE loss, the UFM decomposes the training MSE into two distinct components: a term linear in the regularization constant; a term that depends on the eigenvalues of the sample covariance matrix of the target data. This decomposition highlights how both regularization and data structure influence training performance.

Building on these basic UFM results, we investigate two fundamental questions in multivariate regression:

1. **Single multi-task model versus multiple single-task models:** Multivariate regression can be approached by training a single multi-task model that predicts all targets simultaneously or by training multiple single-task models, each dedicated to a specific target type. Multi-task models are significantly more efficient in terms of computation and memory. Under the UFM assumption, can we mathematically show that one approach has superior performance over the other?
2. **Whitening and normalizing regression targets:** Whitening transforms data to have a covariance equal to the identity matrix, while normalization adjusts variances to be equal to one while preserving correlations. These techniques have been extensively applied to inputs and intermediate features but rarely to regression targets. Under the UFM assumption, can we mathematically show that whitening or normalizing the targets improves performance?

To respond to these questions, we combine theoretical insights from the UFM with extensive empirical analyses. Our empirical evaluations are conducted on four datasets: three robotic locomotion datasets and one autonomous driving dataset. Specifically:

- For the multi-task model versus multiple single-task models problem, we use UFM theory to show that multi-task models achieve strictly smaller training MSE than the single-task models when using stronger regularization for the latter. Even with equivalent regularization, multi-task models are theoretically guaranteed to have training MSE less than or equal to that of single-task models. These qualitative results are then confirmed empirically: multi-task models consistently outperform single-task models across all regularization settings in terms of training MSE.
- UFM theory and empirical evidence demonstrate that whitening and normalizing the targets have an important effect on performance. When applied, these techniques adjust the training MSE in a manner dependent on the inherent variance of the target data. Notably, they improve training performance when the average variance per target dimension is less than one, but harm training performance when the converse holds. This highlights the importance of examining the variance structure of target data when considering the pre-processing strategies of whitening and normalization.

Beyond these specific results, our work demonstrates the broader potential of the UFM as a theoretical and practical tool. By enabling closed-form solutions, the UFM provides an opportunity to gain important qualitative insights into DNNs. Our findings highlight how the UFM can guide design decisions such as choosing between multi-task and single-task models, and on choosing data pre-processing strategies such as whitening and normalization.

2 Related work

The development of deep learning has primarily been driven by heuristics and empirical experience, with limited progress in establishing a solid theoretical framework. The primary challenge lies in the highly non-convex nature of the loss landscape, which complicates optimization and theoretical analysis. Among existing approaches, Neural Tangent Kernel (NTK) theory (Arora et al., 2019; Du et al., 2018; 2019; Jacot et al., 2018; Zou et al., 2020) provides a valuable tool for understanding optimization and convergence behavior in neural networks within the infinite-width regime. However, NTK focuses on the early stages of training and neglects the rich nonlinear feature learning that characterizes practical neural networks.

In contrast, the Unconstrained Feature Model (UFM) (Mixon et al., 2022) and the Layer-Peeled Model (Fang et al., 2021) offer an alternative perspective that emphasizes the nonlinearity of feature representations. Unlike NTK, which focuses on parameter dynamics, UFM assumes universal feature

representations and treats the last-layer features as free parameters. This distinctive formulation allows UFM to capture phenomena that NTK cannot explain, such as neural collapse (Papyan et al., 2020; Weinan & Wojtowytsch, 2022; Zhu et al., 2021), where class features converge to their means, and class means form a simplex equiangular tight frame (ETF).

Research with the UFM includes training on imbalanced data (Hong & Ling, 2024; Thrampoulidis et al., 2022; Yan et al., 2024; Yang et al., 2022), using normalized features (Yaras et al., 2022), and using various loss functions such as mean-squared error (MSE) (Han et al., 2021; Zhou et al., 2022a), label smoothing, and focal losses (Guo et al., 2024; Zhou et al., 2022b). Additionally, UFM has been applied to analyze multi-label classification (Li et al., 2023), scenarios involving a large number of classes (Jiang et al., 2023), and multivariate regression tasks (Andriopoulos et al., 2024). By examining the landscape of the loss function, UFM provides valuable insights into the optimization and convergence behavior of neural networks (Yaras et al., 2022; Zhou et al., 2022a; Zhu et al., 2021). Recent research has extended the classical UFM framework to investigate the propagation of neural collapse beyond the last layer to earlier layers of DNNs, i.e., UFM with two layers connected by nonlinearity in (Tirer & Bruna, 2022), UFM with multiple linear layers in (Dang et al., 2023), deep UFM (DUFM) with nonlinear activations in the context of classification (Súkeník et al., 2024b;a). Such extensions aim to generalize UFM principles to more complex architectures, thereby bridging the gap between theoretical analysis and real-world DNN implementations.

There is a growing body of literature that explores trade-offs between multi-task learning (MTL) and multiple single-task learning (STL) for computer vision to train models that can perform multiple vision tasks (e.g., object detection, segmentation, depth estimation) simultaneously (Kendall et al., 2018; Misra et al., 2016; Vandenhende et al., 2021), and also for reinforcement learning to train a single agent to perform multiple tasks, leveraging shared representations and knowledge transfer across tasks (Rusu et al., 2015; Teh et al., 2017; Yu et al., 2020). Several works explicitly compare MLT and SLT, analyzing their strengths and weaknesses (Fifty et al., 2021; Ruder, 2017).

In parallel, normalization and whitening techniques have been widely studied in machine learning. Beyond their conventional application to raw input data (LeCun et al., 1998), these techniques have also been applied in intermediate layers to not only normalize (Ioffe & Szegedy, 2015), but to also decorrelate features in order to stabilize training, reduce complexity in latent space representation, and improve convergence (Huang et al., 2018; Siarohin et al., 2018; Vincent et al., 2010). Huang et al. (2019) also proposed iterative methods for approximating full whitening in deep layers with the aim of improving optimization efficiency. Furthermore, whitening features helps to reduce domain shifts to align distributions across source and target domains, aiding in transfer learning Sun & Saenko (2016).

Although, there has been significant work on the UFM, to the best of our knowledge, this is the first paper that uses the UFM model explicitly in order to mathematically address the theoretical behavior of training MSE loss in multi-task regression vs single-tasks regressions, and when using target whitening and normalization.

3 Approximations motivated by the UFM

We consider the multivariate regression problem with M training examples $\{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, M\}$ with input $\mathbf{x}_i \in \mathbb{R}^D$ and target $\mathbf{y}_i \in \mathbb{R}^n$. For univariate regression, $n = 1$. For the regression task, the 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 nonlinear 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. The parameters θ, $\mathbf{W}$, and $\mathbf{b}$ are all trainable.

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

$$\mathcal{L}_{\text{params}}(\theta, \mathbf{W}, \mathbf{b}) := \frac{1}{2M} \sum_{i=1}^{M} ||f_{\theta,\mathbf{W},\mathbf{b}}(\mathbf{x}_i) - \mathbf{y}_i||_2^2 + \frac{\lambda_\theta}{2}||\theta||_2^2 + \frac{\lambda_\mathbf{W}}{2}||\mathbf{W}||_F^2, \tag{1}$$

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 $\lambda_\theta = \lambda_\mathbf{W} = \lambda_{WD}$.

In this paper, we consider a modified version of the standard problem, namely, to train the DNN to minimize

$$\mathcal{L}_{\text{features}}(\theta, \mathbf{W}, \mathbf{b}) := \frac{1}{2M}\sum_{i=1}^{M}||f_{\theta,\mathbf{W},\mathbf{b}}(\mathbf{x}_i) - \mathbf{y}_i||_2^2 + \frac{\lambda_{\mathbf{H}}}{2M}\sum_{i=1}^{M}||\mathbf{h}_\theta(\mathbf{x}_i)||_2^2 + \frac{\lambda_{\mathbf{W}}}{2}||\mathbf{W}||_F^2, \quad (2)$$

where $\lambda_{\mathbf{H}}$ and $\lambda_{\mathbf{W}}$ are non-negative regularization parameters. We refer to the standard loss function (1) as the *parameter-regularized loss function* and to the modified loss function (2) as the *feature-regularized loss function*. Although the two problems are not the same, by regularizing the features, we are implicitly constraining the internal parameters θ.

Our goal is to derive mathematically motivated approximations for the training error related with (2) and then use these approximations to gain qualitative insights into fundamental issues in neural multivariate regression. To this end, we consider the *Unconstrained Feature Model (UFM)-loss function* (Fang et al., 2021; Mixon et al., 2022) which is defined as follows:

$$\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b}) := \frac{1}{2M}||\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y}||_F^2 + \frac{\lambda_{\mathbf{H}}}{2M}||\mathbf{H}||_F^2 + \frac{\lambda_{\mathbf{W}}}{2}||\mathbf{W}||_F^2, \quad (3)$$

$\mathbf{H} := [\mathbf{h}_1 \cdots \mathbf{h}_M] \in \mathbb{R}^{d\times M}$, $\mathbf{Y} := [\mathbf{y}_1 \cdots \mathbf{y}_M] \in \mathbb{R}^{n\times M}$. Note that the UFM-loss function solely depends on $\mathbf{H}$, $\mathbf{W}$ and $\mathbf{b}$ and does not depend on the inputs or the parameters θ. It is easily seen that $\min_{\mathbf{H},\mathbf{W},\mathbf{b}} \mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b}) \leq \min_{\theta,\mathbf{W},\mathbf{b}} \mathcal{L}_{\text{features}}(\theta, \mathbf{W}, \mathbf{b})$. The minimal values of the two loss functions are equal if the feature extractor is so expressive that any feature vector configuration is attainable for the inputs in the training set. This simple observation is very powerful since minimizing the UFM-loss function is mathematically tractable and can be solved in closed form without requiring any training.

3.1 Closed-form expressions

Let $\mathbf{\Sigma}$ denote the $n \times n$ sample 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 $\bar{\mathbf{Y}} = [\bar{\mathbf{y}} \cdots \bar{\mathbf{y}}]$, 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 $\mathbf{\Sigma}$ has a positive definite square root, which we denote by $\mathbf{\Sigma}^{1/2}$. We define the *training MSE* for the UFM as:

$$\text{MSE}(\mathbf{H}, \mathbf{W}, \mathbf{b}) := \frac{1}{M}||\mathbf{W}\mathbf{H} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y}||_F^2. \quad (4)$$

Reorder the eigenvalues of $\mathbf{\Sigma}$ so that $\lambda_{\max} := \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n := \lambda_{\min} > 0$. Let $c := \lambda_{\mathbf{H}}\lambda_{\mathbf{W}}$ and let $j^* := \max\{j : \lambda_j \geq c\}$ with the convention $j^* = 0$ when the set in question is the empty set. When $n = 1$, let σ^2 denote the variance of the 1-d targets over the M samples. For any $p \times q$ matrix $\mathbf{C}$ with columns $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_q$, we denote $[\mathbf{C}]_j := [\mathbf{c}_1\ \mathbf{c}_2 \cdots \mathbf{c}_j\ \mathbf{0} \cdots \mathbf{0}]$. Our main analytical results will be based on the closed-form solutions given below.

Theorem 3.1. *Suppose* $(\mathbf{H}^*, \mathbf{W}^*, \mathbf{b}^*)$ *minimizes the UFM-loss* $\mathcal{L}(\mathbf{H}, \mathbf{W}, \mathbf{b})$ *given by (3). Then,*

$$\mathcal{L}(\mathbf{H}^*, \mathbf{W}^*, \mathbf{b}^*) = \text{MSE}(\mathbf{H}^*, \mathbf{W}^*, \mathbf{b}^*) + \sqrt{c}\sum_{i=1}^{n}\eta_i, \quad (5)$$

where η_i *is the* i*-th diagonal entry of* $[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n]_{j^*}$*, and*

$$\text{MSE}(\mathbf{H}^*, \mathbf{W}^*, \mathbf{b}^*) = j^*c + \sum_{i=j^*+1}^{n}\lambda_i. \quad (6)$$

If $n = 1$*, we have that*

$$\text{MSE}(\mathbf{H}^*, \mathbf{w}^*, b^*) = \begin{cases} c, & \text{if } c \leq \sigma^2, \\ \sigma^2, & \text{if } c > \sigma^2. \end{cases} \quad (7)$$

4 Multi-task vs multiple single-task models

In this section, we leverage the closed-form solutions provided in Theorem 3.1 to study a fundamental problem in multivariate regression: which is better, a single multi-task model with n tasks (target

types) or n dedicated single-task models? Multi-task regression provides several advantages in terms of computation, memory efficiency, training time, and hyper-parameter tuning. Multi-task regression trains a single DNN to predict multiple outputs simultaneously, enabling weight sharing and the reuse of learned features across tasks. With n dedicated single-task models, we need roughly n times the number of parameters, and we must separately train and perform hyper-parameter tuning for each of the n models. Also, dedicated single-task models are less efficient during inference since the single-task approach requires running the input through each of the n models.

But how does multi-task regression compare to n single-task regressions in terms of the MSE training performance? We will first address this question mathematically through the lens of the UFM framework. We will then supplement the mathematical results with empirical findings.

4.1 Insights from the UFM framework

Consider n univariate regression problems with training sets $\{(\mathbf{x}_j, \mathbf{y}_j^{(i)}), j = 1, ..., M\}$, where $\mathbf{y}_j^{(i)}$ corresponds to the target value in the i-th dimension of the j-th training example. For each one of these univariate problems, consider the corresponding single-task UFM, each with regularization parameters $\tilde{\lambda}_{\mathbf{W}}$ and $\tilde{\lambda}_{\mathbf{H}}$. Define $\tilde{c} := \tilde{\lambda}_{\mathbf{H}}\tilde{\lambda}_{\mathbf{W}}$. Let $(\mathbf{H}^{(i)}, \mathbf{w}^{(i)}, b^{(i)})$ denote an optimal solution for the i-th such single-task model, and let $\text{MSE}^{(i)}(\mathbf{H}^{(i)}, \mathbf{w}^{(i)}, b^{(i)})$ denote its corresponding MSE. The total MSE across the n single-task models is then given by

$$\text{MSE(n-single}, \tilde{c}) := \sum_{i=1}^{n} \text{MSE}^{(i)}(\mathbf{H}^{(i)}, \mathbf{w}^{(i)}, b^{(i)}). \tag{8}$$

Let σ_i^2 denote the variance of the targets for the i-th single task regression problem: $\sigma_i^2 := M^{-1}\sum_{j=1}^{M}(\mathbf{y}_j^{(i)} - \bar{\mathbf{y}}^{(i)})^2$. Re-order the indices so that $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_n^2$. Define $k^* := \max\{j : \sigma_j^2 \geq \tilde{c}\}$. The corollary below follows directly from case $n = 1$ of Theorem 3.1.

Corollary 4.1. *The total MSE across the n single-task problems is given by*

$$\text{MSE}(\textit{n-single}, \tilde{c}) = k^*\tilde{c} + \sum_{i=k^*+1}^{n} \sigma_i^2. \tag{9}$$

We now present the main result of this section.

Theorem 4.2. *Let MSE$(\textit{multi}, c)$ be a shorthand of (6).*

(i) *Suppose $\tilde{c} = c$. Then,*

$$\text{MSE}(\textit{multi}, c) \leq \text{MSE}(\textit{n-single}, c).$$

Furthermore, for $\lambda_{\min} < c < \lambda_{\max}$ and $j^* < k^*$, the inequality is strict, while for $0 < c < \lambda_{\min}$ or $c > \lambda_{\max}$ the inequality holds with equality.

(ii) *Suppose $c < \tilde{c}$. Then,*

$$\text{MSE}(\textit{multi}, c) \leq \text{MSE}(\textit{n-single}, \tilde{c}),$$

Furthermore, for $c < \min\{\tilde{c}, \lambda_{\max}\}$ the inequality is strict, and if $\lambda_{\max} < c < \tilde{c}$ the inequality holds with equality.

Theorem 4.2 states that if the multi-task regularization constant c is not greater than the single-task regularization constant $\tilde{c}$, then under the UFM approximation, the MSE training error for the multi-task model is always less than that of the n single-task models. Furthermore, the theorem provides refined information for when the inequality is strict or actually an equality. Thus, not only is the multi-task approach more efficient in terms of memory, training and inference computation, it also has lower training MSE in situations of practical interest under the UFM assumption. We briefly note that the case of $c > \tilde{c}$ remains an open problem for future research.

4.2 Experimental results: multi-task vs multiple single-task models

Datasets. Our empirical experiments utilize the Swimmer, Reacher, and Hopper datasets, derived from MuJoCo (Brockman et al., 2016; Todorov et al., 2012; Towers et al., 2024), a physics engine

designed for simulating continuous multi-joint robotic control. These datasets have been widely used as benchmarks in deep reinforcement learning research. Each dataset consists of raw robotic states as inputs and corresponding robotic actions as targets. To adapt them for imitation learning, we reduce their size by selecting a subset of episodes. In addition, we use the CARLA dataset, sourced from the CARLA Simulator—an open-source platform for autonomous driving research. Specifically, we utilize an expert-driven offline dataset (Codevilla et al., 2018), where input images from vehicle-mounted cameras are recorded alongside corresponding expert driving actions as the vehicle navigates a simulated environment. For convenience, we consider a simplified 2D version, which includes only speed and steering angle. Table 1 summarizes the datasets, including target dimensions and spectral properties.

Table 1: **Spectral Properties of Datasets**. $\lambda_{\min}$, $\lambda_{\max}$, and $\bar{\lambda}$ denote the minimum, maximum, and average eigenvalues of the target variable's covariance matrix. $\tilde{\lambda}_{\min}$ and $\tilde{\lambda}_{\max}$ represent the minimum and maximum eigenvalues of the target variable's correlation matrix.

Dataset	n	$\lambda_{\min}$	$\tilde{\lambda}_{\min}$	$\lambda_{\max}$	$\tilde{\lambda}_{\max}$	$\bar{\lambda}$
Reacher	2	0.010	0.991	0.012	1.009	0.011
Swimmer	2	0.276	0.756	0.466	1.244	0.371
Hopper	3	0.215	0.782	0.442	1.258	0.345
CARLA 2D	2	0.024	0.996	209.097	1.004	104.561

Experimental 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., 2024). For CARLA 2D, we employed ResNet18 (He et al., 2016) as the backbone model. To minimize the impact of extraneous factors, we applied standard pre-processing techniques without data augmentation.

All experimental results were averaged over random seeds, with variance across seeds represented by error bars. While weight decay is generally set to small values in practice, we tested a range of values from $1\mathrm{e}{-5}$ to $1\mathrm{e}{-1}$ to thoroughly explore its effect on model training MSE. For each weight decay value, the model was trained for a fixed number of epochs, and training MSE was recorded. The number of training epochs was adjusted based on dataset size, with smaller datasets requiring more epochs to reach convergence. The full experimental setup is provided in Appendix A.

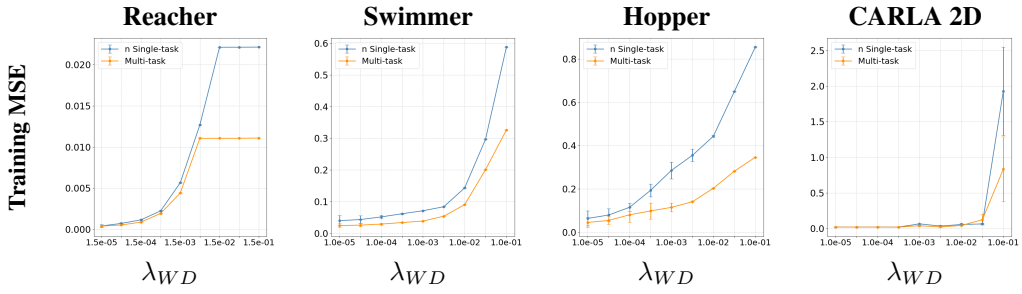

Figure 1: Comparison of the training error of a single multi-task model with that of multiple single task models for different weight decay values after training with the standard parameter-regularized loss function.

Empirical results. Figure 1 shows, regardless of the value of weight decay, that the multi-task model has lower training MSE than the multiple single-task models. These empirical results are consistent with our results in Theorem 4.2, which established that under the UFM assumption, multi-task models achieve smaller training MSE when using equivalent or stronger regularization for single-task models. These empirical results are invariable to changes in the choice of architecture, cf. Figure 1 with Figures 3-5 in Appendix B. Moreover, the trends for testing MSE align closely with those shown for training MSE in Figure 1, see Figure 9 in Appendix B. We mitigate a theoretical explanation of this alignment in Appendix H.

We have also performed experiments directly comparing the training MSE predicted by the UFM and the empirical MSE obtained by training the standard parameter-regularized model. We found that the UFM tends to underestimate the empirical MSE values. A challenging direction for future research is to refine the UFM model so that it provides more accurate estimates while remaining mathematically tractable.

5 TARGET WHITENING AND NORMALIZATION

Whitening and decorrelation. In statistical analysis, whitening (or sphering) refers to a common pre-processing step to transform random variables to orthogonality. A whitening transformation is a linear transformation that converts a random vector with a known covariance matrix into a new random vector of the same dimension and with a covariance matrix given by the identity matrix. Orthogonality among random vectors greatly simplifies multivariate data analysis both from a computational as well as from a statistical standpoint. Whitening is employed mostly in pre-processing but is also part of modeling (Hao et al., 2015; Zuber & Strimmer, 2009).

In this paper we consider whitening the targets using a natural and common form of whitening called zero-phase component analysis (ZCA). In ZCA, in order to whiten a target vector $\mathbf{y}_i$, we simply subtract from $\mathbf{y}_i$ the mean $\bar{\mathbf{y}}$ and then "divide" by the square root of the sample covariance matrix. More specifically,

$$\mathbf{Y}^{ZCA} = \mathbf{\Sigma}^{-1/2}(\mathbf{Y} - \bar{\mathbf{Y}}). \tag{10}$$

Connection with UFM. Importantly, after examining closely the global minima for the UFM-loss function, a significant implication arises. The residual errors will be zero mean and uncorrelated across the n target dimensions, with each variance equal to c. More specifically, by (Andriopoulos et al., 2024)[Corollary 4.2(v)], the residual error is proportional to the ZCA-whitened targets:

$$\mathbf{E} := \mathbf{WH} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y} = -\sqrt{c}\mathbf{Y}^{ZCA}. \tag{11}$$

Training MSE with whitened targets. The emergence of ZCA whitening in the UFM model, as well as the fact that ZCA whitening guarantees no loss of information between the unprocessed and the transformed targets (see the discussion in Appendix E), motivates us to investigate how whitening of the targets will affect the training MSE, from both the UFM and empirical perspectives. For the UFM analysis, we consider the following whitening process:

- First, we whiten the targets using $\mathbf{Y}^{ZCA} = \mathbf{\Sigma}^{-1/2}(\mathbf{Y} - \bar{\mathbf{Y}})$.
- Next, we obtain the optimal $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{H}}$ for the UFM-loss using $\mathbf{Y}^{ZCA}$.
- We then obtain the associated predictions for the whitened training data $\tilde{\mathbf{Y}} := \tilde{\mathbf{W}}\tilde{\mathbf{H}}$.
- We then de-whiten these predictions: $\hat{\mathbf{Y}} := [\mathbf{\Sigma}^{1/2}]\tilde{\mathbf{Y}} + \bar{\mathbf{Y}}$.
- Finally, we calculate the MSE for the whitening approach as

$$\text{MSE(de-whiten)} := M^{-1}||\hat{\mathbf{Y}} - \mathbf{Y}||_F^2. \tag{12}$$

Following the procedure described in the bullet points above, our next theorem provides a closed-form expression for MSE(de-whiten).

Theorem 5.1.

$$\text{MSE(de-whiten)} = \min\{c, 1\} \sum_{i=1}^{n} \lambda_i. \tag{13}$$

We now examine how MSE(de-whiten) compares to the training MSE with the unprocessed targets.

Theorem 5.2. (i) *Suppose* $0 < c \leq 1$. *If*

$$\sum_{i=1}^{j^*} \lambda_i - j^* < c^{-1}(1-c) \sum_{i=j^*+1}^{n} \lambda_i, \tag{14}$$

we have that

$$\textit{MSE(de-whiten)} < \textit{MSE(no-whitening)}.$$

When the converse inequality of (14) holds, then

$$\textit{MSE}(\textit{de-whiten}) > \textit{MSE}(\textit{no-whitening}).$$

When the inequality of (14) is replaced with equality, then the two MSEs are equal.

In the special case when $c < \lambda_{\min}$, i.e., $j^* = n$, the condition in (14) reduces to $\bar{\lambda} := \sum_{i=1}^{n} \lambda_i/n < 1$.

(ii) *Suppose $c > 1$. Then,*

$$\textit{MSE}(\textit{de-whiten}) \geq \textit{MSE}(\textit{no-whitening}),$$

which is an equality if and only if $\lambda_i = c$, for all $i \leq j^*$.

Theorem 5.2 offers valuable qualitative insights into the target whitening approach. In many practical situations, we have $\lambda_{\min} < 1$ and $\bar{\lambda} < 1$, as is the case with the Reacher, Swimmer, and Hopper training datasets (see Table 1). Furthermore, test error is typically minimized with relatively small values of weight decay, corresponding to $c < 1$. In such cases, Theorem 5.2 tells us that MSE(de-whiten) $<$ MSE(no-whitening), meaning that, under the UFM approximation, training with whitened targets strictly reduces the training MSE. Conversely, if $\bar{\lambda} > 1$, as is the case for our CARLA 2D dataset (see Table 1), Theorem 5.2 tells us that MSE(de-whiten) $>$ MSE(no-whitening). In this scenario, training with whitened targets increases the training MSE. This duality highlights the critical role of $\bar{\lambda}$ in determining whether whitening improves or worsens training performance.

Normalization. Another transformation closely related to whitening is normalization, which sets variances to 1 but leaves correlations intact:

$$\mathbf{Y}^{nrm} = \mathbf{V}^{-1/2}(\mathbf{Y} - \bar{\mathbf{Y}}), \tag{15}$$

where $\mathbf{V}$ is the diagonal matrix $\mathbf{V} := \text{diag}(\sigma_1^2, ..., \sigma_n^2)$, and σ_j^2 denotes the variance of the jth target component.

As we did previously for whitening, under the lens of the UFM theory, we can examine the effect of normalizing the targets with respect to the training MSE. To this end, we follow the same procedure as for whitening, but instead of using $\mathbf{Y}^{ZCA}$ we use $\mathbf{Y}^{nrm}$. Denote MSE(de-normalize) for the resulting de-normalized training MSE. Following this procedure, our next theorem provides a closed-form expression for MSE(de-normalization).

Let $\tilde{\lambda}_{\min}$, $\tilde{\lambda}_{\max}$ be the min and max eigenvalues of the sample correlation matrix of the target data given by $\mathbf{P} = \mathbf{V}^{-1/2}\mathbf{\Sigma}\mathbf{V}^{-1/2}$. Note that $0 \leq \tilde{\lambda}_{\min} \leq 1$, and $\tilde{\lambda}_{\max} \geq 1$.

Theorem 5.3.

$$\text{MSE(de-normalize)} = \begin{cases} c\sum_{i=1}^{n} \lambda_i, & \textit{if } 0 < c < \tilde{\lambda}_{\min}, \\ \sum_{i=1}^{n} \lambda_i, & \textit{if } c > \tilde{\lambda}_{\max}. \end{cases} \tag{16}$$

Let us now examine how MSE(de-normalize) compares to MSE(de-whiten) and to the training MSE with the unprocessed targets. The next theorem follows directly from Theorems 5.1-5.3.

Theorem 5.4. (i) *Suppose $0 < c < \min\{\lambda_{\min}, \tilde{\lambda}_{\min}\}$. Then,*

$$\textit{MSE}(\textit{de-normalize}) = \textit{MSE}(\textit{de-whiten}).$$

Furthermore, if $\bar{\lambda} < 1$, then MSE(de-normalize) $<$ MSE, where MSE is the training MSE using unprocessed targets. If $\bar{\lambda} > 1$, then MSE(de-normalize) $>$ MSE. If $\bar{\lambda} = 1$, then MSE(de-normalize) $=$ MSE.

(ii) Suppose $c > \tilde{\lambda}_{\max}$. Then,

$$\textit{MSE}(\textit{de-normalize}) = \textit{MSE}(\textit{de-whiten}).$$

Furthermore, MSE(de-normalize) $\geq$ MSE.

Theorem 5.4 provides important qualitative insights into the target normalization approach. In practical scenarios, such as with all of our 4 training datasets, we observe that $\lambda_{\min} < 1$ ($\tilde{\lambda}_{\min} \leq 1$

for any arbitrary correlation matrix), as indicated in Table 1. Additionally, test error is typically minimized with small weight decay values, corresponding to $c < 1$. Under these conditions, MSE(de-normalize) = MSE(de-whiten), implying that under the UFM approximation, one can use either target whitening or normalization. As in Theorem 5.2, Theorem 5.4 also highlights the crucial role of $\bar{\lambda}$ in determining whether normalization improves or worsens training performance. We also note that Theorem 5.4 does not provide a characterization for the case $\min\{\lambda_{\min}, \tilde{\lambda}_{\min}\} < c < \tilde{\lambda}_{\max}$. We leave this as an open research problem.

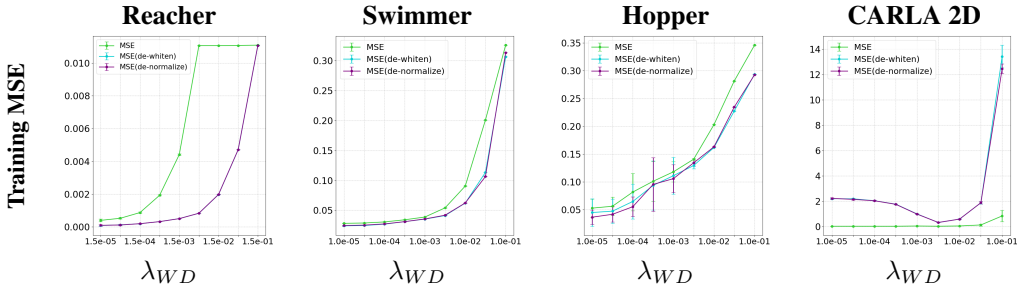

Figure 2: Comparison of the effect that target whitening and normalization have on training error for different weight decay values after training with the standard parameter-regularized loss function. The green curve (in short MSE) records the training error for different weight decay values after training with the original unprocessed targets.

5.1 Experimental results: target whitening and normalization

To validate the theoretical results, we conduct empirical experiments on three MuJoCo datasets and CARLA 2D. We adopt the same network structures as in Section 4.2. For different weight decay values λ_{WD}, we evaluate three settings: (1) using the raw target y as a baseline, (2) using the whitened targets, and (3) using the normalized targets. When whitening or normalizing, we train the model with the transformed y and then apply the inverse transformation (de-whitening or de-normalizing) to compute the training MSE.

Figure 2 presents the experimental results, which turn out to closely align with the theory. First, as the theory suggests, there is minimal difference in training MSE between whitening and normalizing. Second, the impact of whitening/normalizing on training MSE depends on the average variance over target dimensions, $\bar{\lambda}$. Whitening/normalizing reduces the training MSE when $\bar{\lambda} < 1$, as is the case in the three MuJoCo datasets. In contrast, when $\bar{\lambda} > 1$, as is the case in the CARLA 2D dataset, whitening/normalizing leads to an increase in training MSE. Once again, these empirical results are invariable to changes in the choice of architecture, cf. Figure 2 with Figures 6-8 in Appendix B. Moreover, the trends for testing MSE align closely with those shown for training MSE in Figure 2, see Figure 10 in Appendix B. These results not only validate the UFM theory but also provide practical insights for selecting appropriate data pre-processing strategies in machine learning pipelines.

6 Conclusion

This paper demonstrates the power of the Unconstrained Feature Model (UFM) as a theoretical tool for understanding neural multivariate regression tasks. By deriving and analyzing closed-form expressions for training mean-squared error (MSE), we explored two critical problems: multi-task versus single-task regression models and the benefits of target whitening and normalization. We found that for both of these problems, the UFM was applicable and provided novel qualitative insights. Our experiments then confirmed the correctness of the UFM predictions. Beyond these specific insights, the broader significance of this work lies in its demonstration of how the UFM can bridge theory and practice. The UFM's ability to yield qualitative insights into key design choices, such as model architecture and data pre-processing strategies, establishes it as a valuable tool for advancing the design and optimization of DNNs. Additionally, developing tractable models within the UFM framework that extend to understanding and predicting generalization performance would be a valuable direction for further research.

REFERENCES

George Andriopoulos, Zixuan Dong, Li Guo, Zifan Zhao, and Keith W. Ross. The prevalence of neural collapse in neural multivariate regression. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332. PMLR, 2019.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 4693–4700. IEEE, 2018.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Hien Dang, Tho Tran, Stanley Osher, Hung Tran-The, Nhat Ho, and Tan Nguyen. Neural collapse in deep linear networks: from balanced to imbalanced data. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 6873–6947, 2023.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pp. 1675–1685. PMLR, 2019.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

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.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34: 27503–27516, 2021.

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.

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.

Li Guo, Keith Ross, Zifan Zhao, George Andriopoulos, 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.

Ning Hao, Bin Dong, and Jianqing Fan. Sparsifying the Fisher linear discriminant by rotation. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 77(4):827–851, 2015.

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*, pp. 770–778, 2016.

Wanli Hong and Shuyang Ling. Neural collapse for unconstrained feature model under cross-entropy loss with imbalanced data. *Journal of Machine Learning Research*, 25(192):1–48, 2024.

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

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 791–800, 2018.

Lei Huang, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Iterative normalization: Beyond standardization towards efficient whitening. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4874–4883, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Jiachen Jiang, Jinxin Zhou, Peng Wang, Qing Qu, Dustin Mixon, Chong You, and Zhihui Zhu. Generalized neural collapse for a large number of classes. *arXiv preprint arXiv:2310.05351*, 2023.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Pengyu Li, Xiao Li, Yutong Wang, and Qing Qu. Neural collapse in multi-label learning with pick-all-label loss. *arXiv preprint arXiv:2310.15903*, 2023.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3994–4003, 2016.

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

Mehryar Mohri. *Foundations of machine learning*. MIT press, 2nd Edition, 2018.

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.

S Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening and coloring batch transform for gans. *arXiv preprint arXiv:1806.00420*, 2018.

Peter Súkeník, Marco Mondelli, and Christoph Lampert. Neural collapse versus low-rank bias: Is deep neural collapse really optimal? *arXiv preprint arXiv:2405.14468*, 2024a.

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, 2024b.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

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*, pp. 843–852, 2017.

Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. CORL: Research-oriented deep offline reinforcement learning library. *Advances in Neural Information Processing Systems*, 36, 2024.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

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*, pp. 21478–21505. PMLR, 2022.

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*, pp. 5026–5033. IEEE, 2012.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3614–3633, 2021.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

E Weinan and Stephan Wojtowytsch. On the emergence of simplex symmetry in the final and penultimate layers of neural network classifiers. In *Mathematical and Scientific Machine Learning*, pp. 270–290. PMLR, 2022.

Hongren Yan, Yuhua Qian, Furong Peng, Jiachen Luo, Feijiang Li, et al. Neural collapse to multiple centers for imbalanced data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

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.

Can Yaras, Peng Wang, Zhihui Zhu, Laura Balzano, and Qing Qu. Neural collapse with normalized features: A geometric analysis over the Riemannian manifold. *Advances in neural information processing systems*, 35:11547–11560, 2022.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

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*, pp. 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.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep ReLU networks. *Machine learning*, 109:467–492, 2020.

Verena Zuber and Korbinian Strimmer. Gene ranking and biomarker discovery under correlation. *Bioinformatics*, 25(20):2700–2707, 2009.

A EXPERIMENTAL DETAILS

A.1 MUJOCO

The datasets Reacher and Swimmer are sourced from an open-source repository (Gallouédec et al., 2024) and consist of expert data generated by a policy trained using Proximal Policy Optimization (PPO) (Schulman et al., 2017). For Hopper, the dataset is part of the D4RL benchmark (Fu et al., 2020), which is widely recognized in offline reinforcement learning research. Table 2 provides a summary of all model hyper-parameters and experimental configurations used in Sections 4.2 and 5.1. In all experiments, the models are trained until their weights converge. Additional details regarding the MuJoCo datasets and dataset-specific hyper-parameter settings are provided below.

Table 2: Hyper-parameter settings for experiments with weight decay on MuJoCo datasets.

	Hyper-parameter	**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	2e5, Reacher 2e5, Swimmer 4e4, Hopper
	Batch size	256
	Optimizer	SGD
	Learning rate	1e-2
	Seeds	0, 1, 2
	Compute resources	NVIDIA A100 8358 80GB
	Number of compute workers	4
	Requested compute memory	16 GB
	Approximate average execution time	5 hours

MuJoCo environment descriptions. We utilize expert data from previous work (Gallouédec et al., 2024; Fu et al., 2020) for the Reacher, Swimmer, and Hopper environments. The Reacher environment features a two-jointed robotic arm, where the objective is to control the arm's tip to reach a randomly placed target in a 2D plane. The Swimmer environment consists of a three-segment robot connected by two rotors, designed to propel itself forward as quickly as possible. Similarly, the Hopper environment is a single-legged robot with four connected body parts, aiming to hop forward efficiently. In all three cases, the robots are controlled by applying torques to their joints, which serve as the action inputs. To construct the datasets, an online reinforcement learning algorithm was used to train expert policies (Gallouédec et al., 2024; Fu et al., 2020). These expert policies were then deployed in the environments to generate offline datasets, consisting of state-action pairs where the states $\mathbf{x}_i$ encompass the robot's positions, angles, velocities, and angular velocities, while the targets $\mathbf{y}_i$ correspond to the torques applied to the joints.

Low data regime. Training neural networks with expert state-action data using regularized regression is commonly known as *imitation learning*. Following standard practices for MuJoCo environments (Tarasov et al., 2024), we employ relatively small multi-layer perceptron (MLP) architectures. Since the goal in imitation learning is to achieve strong performance with minimal expert data, we train models using only a fraction of the available datasets. Specifically, we use 20 expert demonstrations (episodes) for Reacher, 1 for Swimmer, and 10 for Hopper, translating to datasets of 1,000, 1,000, and 10,000 samples respectively. Additionally, for each environment, we construct a validation set that contains 20% of the size of the training data.

A.2 CARLA

The CARLA dataset is created by capturing the vehicle's surroundings through automotive cameras while a human driver controls the vehicle in a simulated urban environment (Codevilla et al., 2018).

The recorded images represent the vehicle's states $\mathbf{x}_i$, and the expert driver's control inputs, including speed and steering angles, are treated as the actions $\mathbf{y}_i \in [0, 85] \times [-1, 1]$ in the dataset. A model trained on this data is expected to navigate the vehicle safely within the virtual environment.

For feature extraction from images, we use ResNet-18 (He et al., 2016) as the backbone model. Since a large number of images is required to train a robust feature extractor from visual inputs (He et al., 2016; Sun et al., 2017), the entire dataset is utilized for training. To adapt the ResNet architecture, which was initially designed for classification tasks, to a regression setting, we replace the final classification layer with a fully connected layer that outputs continuous values corresponding to the targets. The experimental setup for CARLA is detailed in Table 3.

Table 3: Hyper-parameters of ResNet for CARLA dataset.

	Hyper-parameter	**Value**
Architecture	Backbone of hidden layers	ResNet18
	Last layer hidden dim	512
Training	Epochs	100
	Batch size	512
	Optimizer	SGD
	Momentum	0.9
	Learning rate	0.001
	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 Inter-task correlations

Our chosen datasets exhibit a range of task correlations. Table 4 below includes the Pearson correlation coefficient between the i-th and j-th target components for $i \neq j$. When the target dimension is 2, there is one correlation value between the two target components; when the target dimension is 3, there are three correlation values between the three target components. From Table 4, we observe that the target components in CARLA 2D and Reacher are nearly uncorrelated, whereas those in Hopper and Swimmer exhibit stronger correlations. This demonstrates that multi-task learning's advantages in our experiments are not solely attributable to high-correlation scenarios. While strongly correlated tasks do provide a "best-case" benchmark, our results also highlight settings where task relationships are weak - underscoring multi-task learning's ability to leverage even limited shared structure.

Table 4: Overview of datasets employed in our analysis.

Dataset	**Data Size**	**Input Type**	**Target Dimension n**	**Target Correlation**
Swimmer	1K	raw state	2	-0.244
Reacher	10K	raw state	2	-0.00933
Hopper	10K	raw state	3	[-0.215, -0.090, 0.059]
CARLA 2D	600K	RGB image	2	-0.0055

B ADDITIONAL EXPERIMENTAL RESULTS

B.1 SUMMARY OF TRAIN MSE EMPIRICAL RESULTS ACROSS A BROAD RANGE OF ARCHITECTURES

The UFM formulation depends on the idea that the nonlinear feature extractor is flexible enough to be capable of approximating any function. To explore what happens when this part of the network has dramatically reduced or increased capacity, we cover in depth how the training of the networks impacts the experimental results of Section 4.2 and Section 5.1.

To provide further analysis of how the size of neural networks influences the empirical results of Figure 1 and Figure 2, we have repeated our experiments across multiple architectures, covering a broad range of widths and depths. For this ablation, we selected the MuJoCo environments of Reacher, Swimmer, and Hopper. The results summarized in Figures 3-5 are organized from left to right and top to bottom based on the number of parameters of the neural networks. We find that, regardless of the architecture, our experimental results align with the predictions of Theorem 4.2 regarding multi-task and single-task models. Similarly, in Figures 6-8, we summarize experiments across a broad range of architectures to test the claims of Theorem 5.4 regarding training with whitened and normalized targets as opposed to training with raw targets. The results remain consistent.

Thus, our choice of datasets with input data ranging from raw robotic states (vectors) to images and training networks ranging from shallow MLPs to the deep and wide ResNet architecture, strikes a balance, and demonstrates the soundness of our empirical findings for low and high capacity networks.

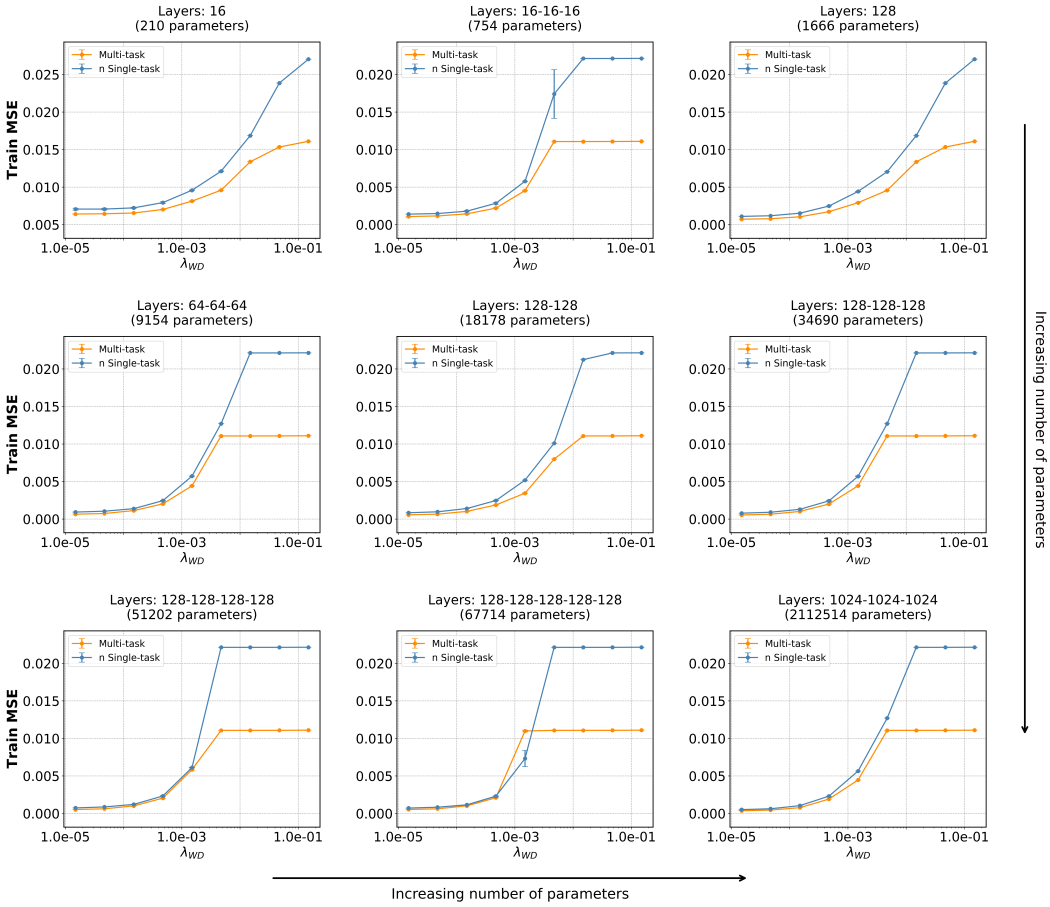

Figure 3: **Effect of network architecture on multi- vs. single-task training: Reacher.** Comparison of the training error of a single multi-task model with that of multiple single task models for different weight decay values after training with the standard parameter-regularized loss function across different architectures. The architectures are denoted by by their layer sizes (input and output layers omitted for simplicity). The number of parameters increases from left to right and from top to bottom.

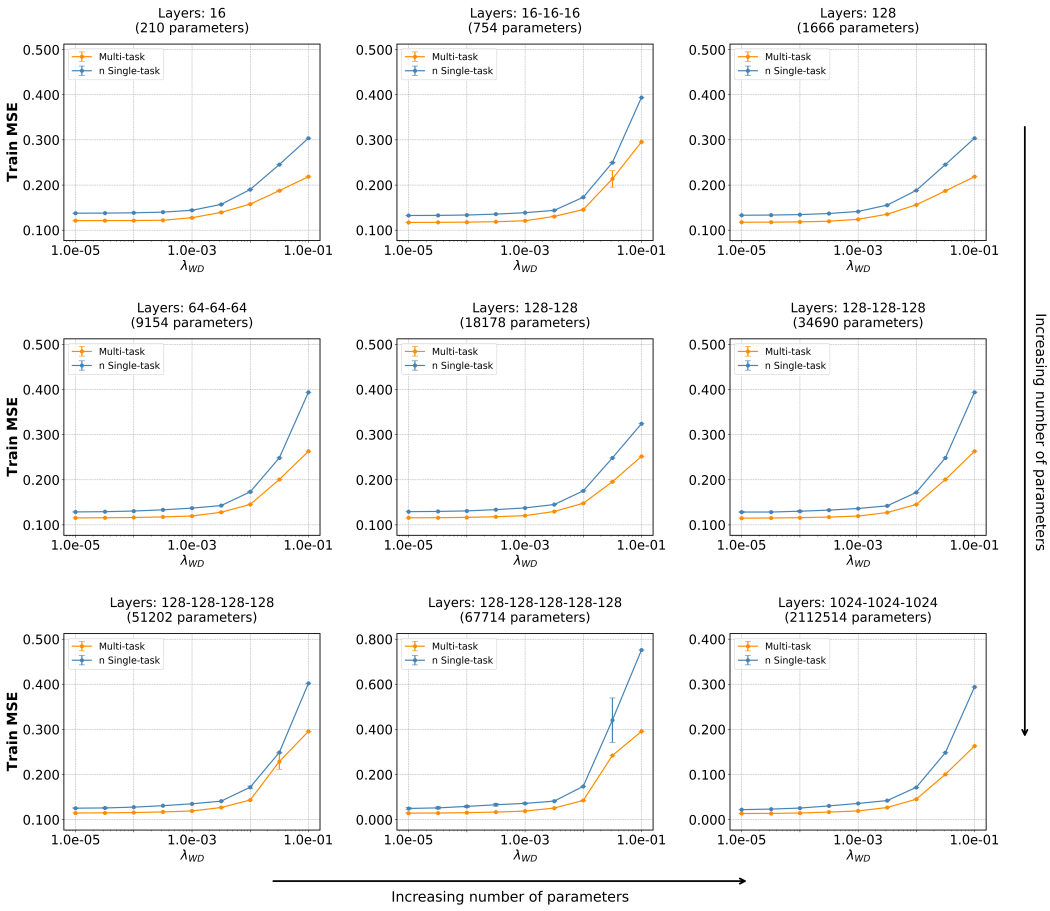

Figure 4: **Effect of network architecture on multi- vs. single-task training: Swimmer.** Comparison of the training error of a single multi-task model with that of multiple single task models for different weight decay values after training with the standard parameter-regularized loss function across different architectures. The architectures are denoted by by their layer sizes (input and output layers omitted for simplicity). The number of parameters increases from left to right and from top to bottom.

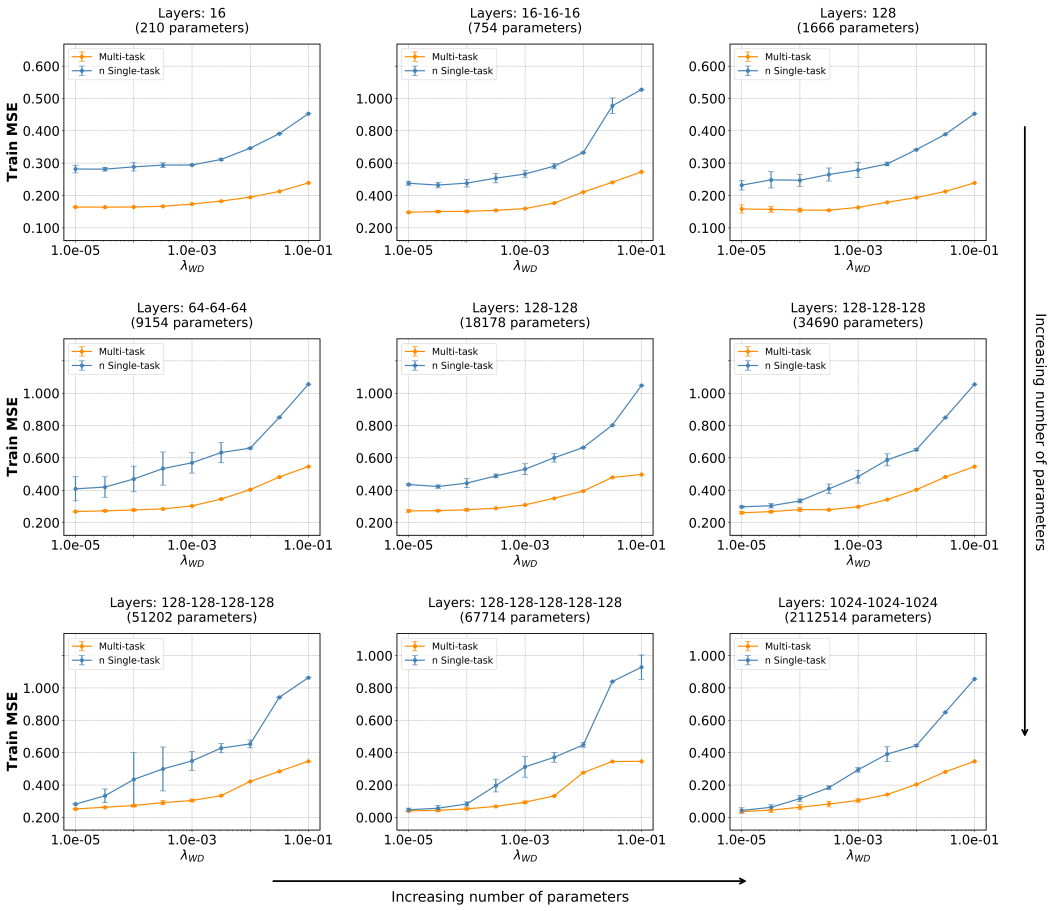

Figure 5: **Effect of network architecture on multi- vs. single-task training: Hopper.** Comparison of the training error of a single multi-task model with that of multiple single task models for different weight decay values after training with the standard parameter-regularized loss function across different architectures. The architectures are denoted by by their layer sizes (input and output layers omitted for simplicity). The number of parameters increases from left to right and from top to bottom.

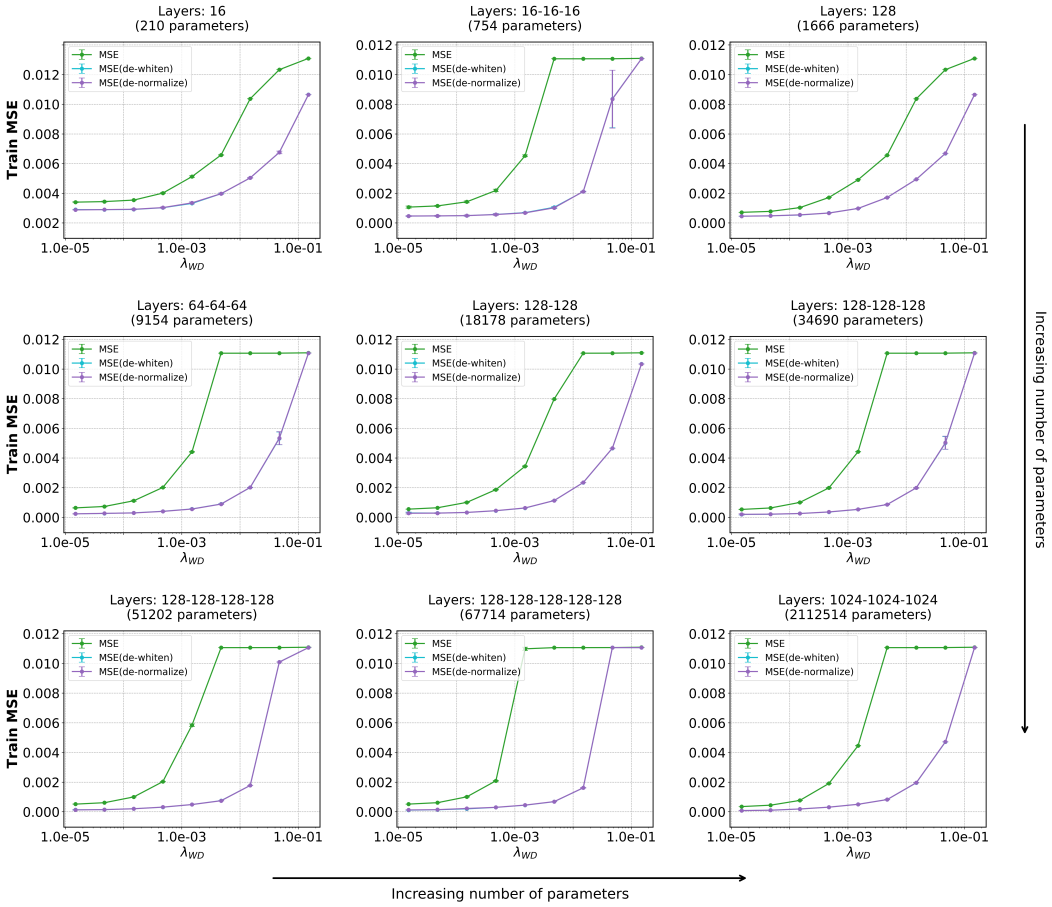

Figure 6: **Whitening vs. normalization vs. raw targets (Swimmer).** Training-MSE comparisons for the Reacher environment. Comparison of the effect that target whitening and normalization have on training error for different weight decay values after training with the standard parameter-regularized loss function across different architectures. The green curve (in short MSE) records the training error for different weight decay values after training with the original unprocessed targets. The architectures are denoted by their layer sizes (input and output layers omitted for simplicity). The number of parameters increases from left to right and from top to bottom.

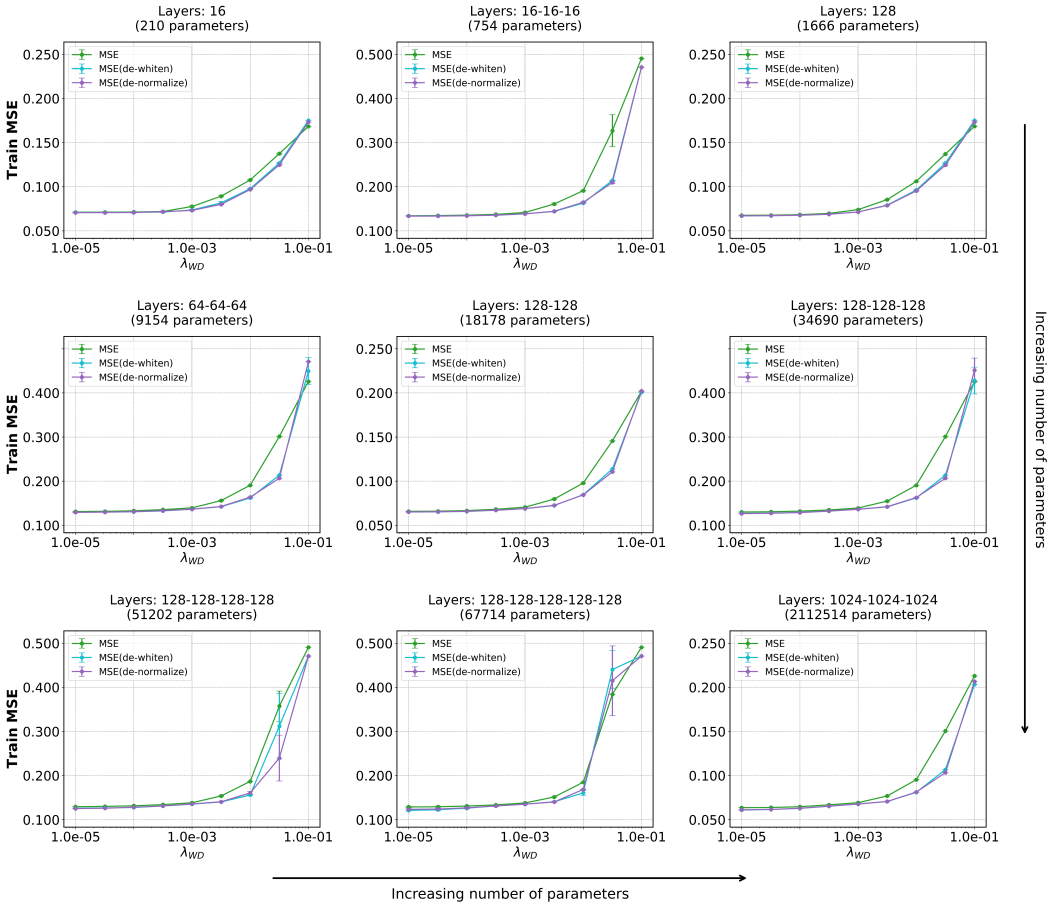

Figure 7: **Whitening vs. normalization vs. raw targets (Reacher).** Training-MSE comparisons for the Reacher environment. Comparison of the effect that target whitening and normalization have on training error for different weight decay values after training with the standard parameter-regularized loss function across different architectures. The green curve (in short MSE) records the training error for different weight decay values after training with the original unprocessed targets. The architectures are denoted by their layer sizes (input and output layers omitted for simplicity). The number of parameters increases from left to right and from top to bottom.

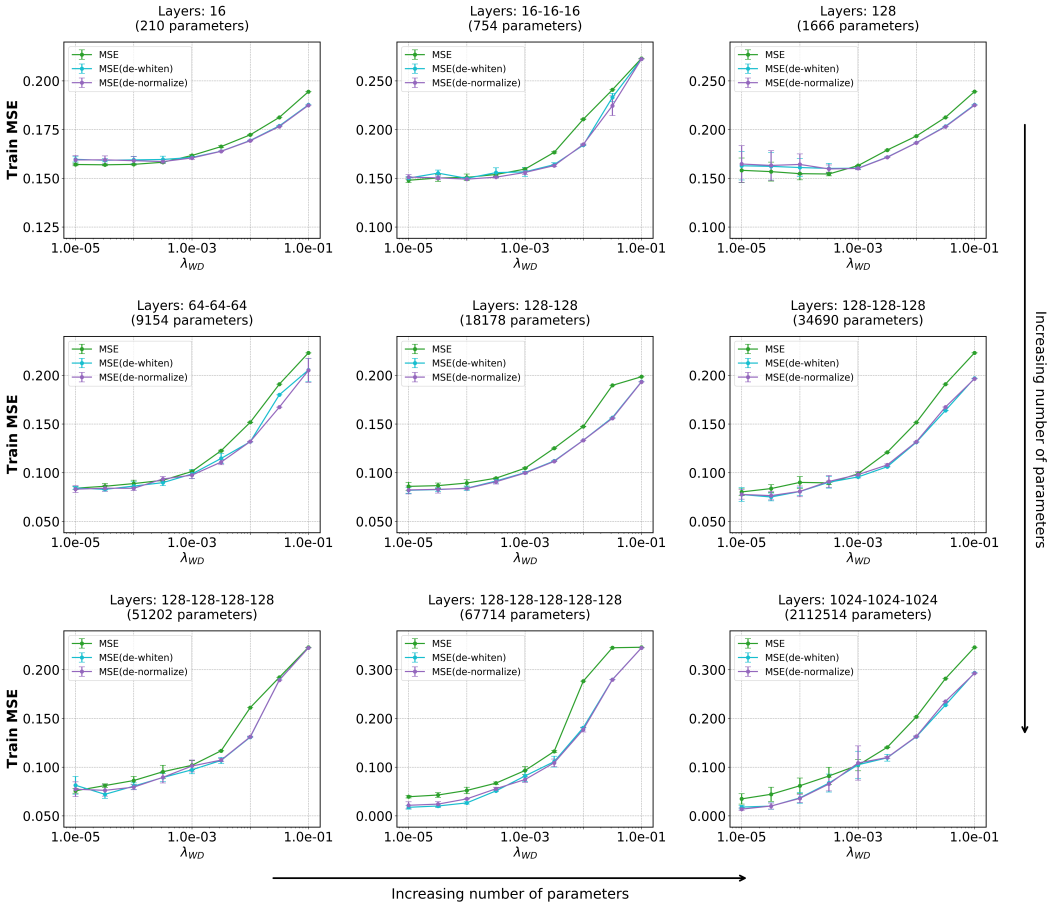

Figure 8: **Whitening vs. normalization vs. raw targets (Hopper).** Training-MSE comparisons for the Reacher environment. Comparison of the effect that target whitening and normalization have on training error for different weight decay values after training with the standard parameter-regularized loss function across different architectures. The green curve (in short MSE) records the training error for different weight decay values after training with the original unprocessed targets. The architectures are denoted by their layer sizes (input and output layers omitted for simplicity). The number of parameters increases from left to right and from top to bottom.

B.2 SUMMARY OF TESTING MSE EMPIRICAL RESULTS

We have conducted extensive experiments to evaluate testing MSE, complementing our theoretical and empirical analysis on training MSE. Regarding the inclusion of test results, these serve two purposes. One, to illustrate that even with zero training MSE, e.g., in the unregularized cases, modest weight decay can still improve generalization, and another, to provide preliminary evidence that our theoretical findings on training MSE do not lead to impractical model behavior. The results are summarized in the figures below.

For the vast majority of cases, the trends for testing MSE align closely with those shown for training MSE in Figures 1 and 2:

- Figure 9: Multi-task learning consistently achieves lower testing MSE than single-task learning across all datasets and weight decay values, supporting the intuition that, by taking on account task dependencies and learning shared patterns through weights and representation sharing in a single model, multi-task regression improves generalization. For an attempt to explain this theoretically using generalization bounds, we refer the reader to the discussion that follows the derivation of (35) and (36) in Appendix H.
- Figure 10: The difference between whitening and normalization is minor, in line with Figure 2 of the main body. The effect depends on the average eigenvalue of the covariance matrix. When the average eigenvalue is < 1 (e.g., the three MuJoCo datasets), whitening/normalization generally reduces testing MSE, except for very small weight decay values. When the average eigenvalue is > 1 (e.g., CARLA 2D), whitening/normalization increases testing MSE, reflecting the same trend observed for training MSE.

In summary, the testing MSE results reinforce the robustness of our findings: modest regularization improves generalization, multi-task learning offers consistent benefits, and whitening/normalization effects align with the spectral properties of the target data covariance.

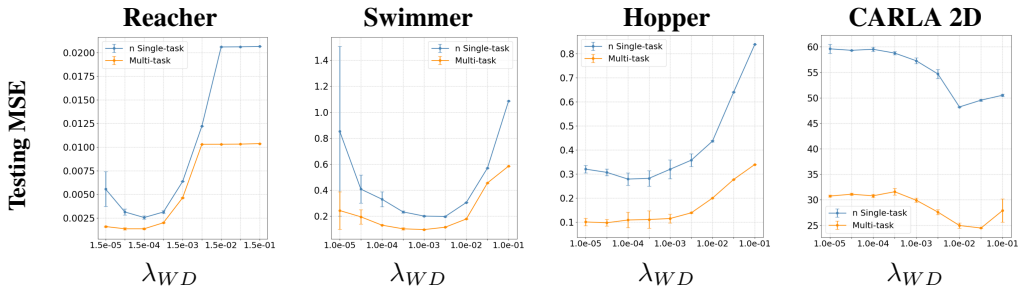

Figure 9: Comparison of the test error of a single multi-task model with that of multiple single task models for different weight decay values after training with the standard parameter-regularized loss function. Values shown as mean±std across random seeds.

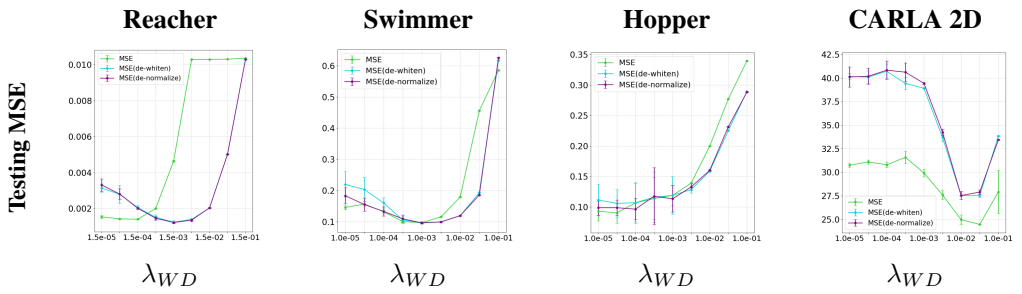

Figure 10: Comparison of the test error of a single multi-task model with that of multiple single task models for different weight decay values after training with the standard parameter-regularized loss function. Values shown as mean±std across random seeds.

C PROOF OF THEOREM 3.1

Proof of Theorem 3.1. Let $\tilde{\mathbf{Y}} = \mathbf{Y} - \bar{\mathbf{Y}} = U\tilde{\mathbf{\Sigma}}V^T$ denote the compact SVD of $\tilde{\mathbf{Y}}$, where $U \in \mathbb{R}^{n\times n}$ is orthogonal, $V \in \mathbb{R}^{M\times n}$ is semi-orthogonal, and $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{n\times n}$ is diagonal containing the singular values $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_n > 0$. Let $(\mathbf{H}^*, \mathbf{W}^*, \mathbf{b}^*)$ be a global minimum of (3). By Lemma B.1(Zhou et al., 2022a), the associated MSE is

$$\text{MSE}(\mathbf{H}^*, \mathbf{W}^*, \mathbf{b}^*) = \frac{1}{M}\sum_{i=1}^{n}([\eta_i - \sqrt{Mc}]_+ - \eta_i)^2. \tag{17}$$

Furthermore, using the SVD of $\tilde{\mathbf{Y}} = U\tilde{\mathbf{\Sigma}}V^T$,

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

from which we have $\mathbf{\Sigma}^{1/2} = U\frac{\tilde{\mathbf{\Sigma}}}{\sqrt{M}}U^T$. This further yields

$$\sqrt{M}[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n] = U\tilde{\mathbf{\Sigma}}U^T - U\sqrt{Mc}\mathbf{I}_nU^T$$

Since $U^T = U^{-1}$,

$$\sqrt{M}[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n] = U\left[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n\right]U^{-1}, \tag{18}$$

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 (18) has eigenvalues given by $\sqrt{M\lambda_i} - \sqrt{Mc}$, $i = 1, ..., n$, where λ_i is the i-th eigenvalue of $\mathbf{\Sigma}$, whereas the $n \times n$ matrix on the right-hand side of (18) has eigenvalues $\eta_i - \sqrt{Mc}$, $i = 1, ..., n$. Since the eigenvalues in these two sets are both arranged in descending order, we have

$$\sqrt{\lambda_i} = \frac{\eta_i}{\sqrt{M}}, \qquad \text{for all } i = 1, ..., n. \tag{19}$$

Correspondingly, by (17) and (19) we obtain

$$\begin{aligned}
\text{MSE}(\mathbf{H}^*, \mathbf{W}^*, \mathbf{b}^*) &= \frac{1}{M}\sum_{i=1}^{n}([\eta_i - \sqrt{Mc}]_+ - \eta_i)^2 \\
&= \frac{1}{M}\sum_{i=1}^{n}([\sqrt{M\lambda_i} - \sqrt{Mc}]_+ - \sqrt{M\lambda_i})^2 \\
&= j^*c + \sum_{i=j^*+1}^{n}\lambda_i
\end{aligned}$$

as desired. When $n = 1$, $\mathbf{\Sigma}$ is simply the scalar σ^2, which together with the equation above completes the proof. □

Next, we give a simple upper bound of the form (20) as a corollary. The bound is explicitly given as the minimum of the MSEs when $c < \lambda_{\min}$ ($j^* = n$) and $c > \lambda_{\max}$ ($j^* = 0$) respectively.

Corollary C.1.

$$\text{MSE}(\mathbf{H}, \mathbf{W}, \mathbf{b}) \leq \min\left\{nc, \sum_{i=1}^{n}\lambda_i\right\}. \tag{20}$$

Proof. Note that since $c \leq \lambda_i$, for all $i \leq j^*$, and $c > \lambda_i$, for all $i > j^*$, we have that

$$\text{MSE}(\mathbf{H}, \mathbf{W}, \mathbf{b}) = j^*c + \sum_{i=j^*+1}^{n}\lambda_i < j^*c + \sum_{i=j^*+1}^{n}c = nc,$$

$$\text{MSE}(\mathbf{H}^*, \mathbf{W}^*, \mathbf{b}^*) = j^*c + \sum_{i=j^*+1}^{n}\lambda_i = \sum_{i=1}^{j^*}c + \sum_{i=j^*+1}^{n}\lambda_i \leq \sum_{i=1}^{j^*}\lambda_i + \sum_{i=j^*+1}^{n}\lambda_i = \sum_{i=1}^{n}\lambda_i.$$

The desired result readily follows. □

D PROOF OF THEOREM 4.2

Proof of Theorem 4.2. We begin by using the Schur-Horn theorem to establish some relations between the eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\mathbf{\Sigma}$ and the diagonal elements (variances) $\sigma_1^2, \ldots, \sigma_n^2$ of $\mathbf{\Sigma}$. Recall that both $\lambda_1, \ldots, \lambda_n$ and $\sigma_1^2, \ldots, \sigma_n^2$ are arranged in descending order. By the Schur-Horn theorem, the vector containing the diagonal elements of $\mathbf{\Sigma}$ is majorized by the vector that contains the ordered eigenvalues of $\mathbf{\Sigma}$, i.e.,

$$\sum_{i=1}^{k} \lambda_i \geq \sum_{i=1}^{k} \sigma_i^2, \qquad \text{for all } k = 1, \ldots, n-1, \tag{21}$$

$$\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \sigma_i^2. \tag{22}$$

From (21) and (22) we have

$$\lambda_1 \geq \sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_n^2 \geq \lambda_n. \tag{23}$$

and also the following inequalities for the tail partial sums:

$$\sum_{i=k}^{n} \lambda_i \leq \sum_{i=k}^{n} \sigma_i^2, \qquad \text{for all } k = 1, \ldots, n.$$

Fix $0 < c < \tilde{c}$. We will consider 6 cases:

Case I: Suppose $c < \tilde{c} < \lambda_n$. By (23), we have that $\tilde{c} < \sigma_n^2$. Thus, by Theorem 3.1 and Corollary 4.1, the two MSEs are given by

$$nc = \text{MSE}(\text{multi}, c) < \text{MSE}(\text{n-single}, \tilde{c}) = n\tilde{c}.$$

Observe that when $c = \tilde{c}$, the two MSEs are equal to nc.

Case II: Suppose $\tilde{c} \geq c > \lambda_1$. By (23), we have that $\tilde{c} > \sigma_1^2$. Thus, by Theorem 3.1 and Corollary 4.1, the two MSEs are given by

$$\text{MSE}(\text{multi}, c) = \sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \sigma_i^2 = \text{MSE}(\text{n-single}, \tilde{c}).$$

Case III: Suppose $c < \lambda_n$, $\tilde{c} > \lambda_1$. From Theorem 3.1 and Corollary 4.1, the difference of the two MSEs is given by

$$\text{MSE}(\text{multi}, c) - \text{MSE}(\text{n-single}, \tilde{c}) = nc - \sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} (c - \lambda_i) < 0,$$

since $c < \lambda_n \leq \lambda_i$, for all $i = 1, ..., n$.

Case IV: Suppose $\lambda_n < c < \lambda_1 < \tilde{c}$. From Theorem 3.1 and Corollary 4.1, the difference of the two MSEs is given by

$$\text{MSE}(\text{multi}, c) - \text{MSE}(\text{n-single}, \tilde{c}) = j^* c - \sum_{i=1}^{j^*} \lambda_i = \sum_{i=1}^{j^*} (c - \lambda_i) < 0,$$

since $c < \lambda_1$ and $c \leq \lambda_i$, for all $i \leq j^*$.

Case V: Suppose $c < \lambda_n < \tilde{c} < \lambda_1$. From Theorem 3.1 and Corollary 4.1, the difference of the two MSEs is given by

$$\begin{aligned}\text{MSE(multi}, c) - \text{MSE(n-single}, \tilde{c}) &= nc - k^*\tilde{c} - \sum_{i=k^*+1}^{n} \sigma_i^2 \\ &= \sum_{i=k^*+1}^{n} (c - \sigma_i^2) + k^*(c - \tilde{c}) \\ &< \sum_{i=k^*+1}^{n} (\lambda_i - \sigma_i^2) + k^*(c - \tilde{c}) \\ &= \sum_{i=1}^{k^*} (\sigma_i^2 - \lambda_i) + k^*(c - \tilde{c}) < 0,\end{aligned}$$

where the first inequality holds since $c < \lambda_n \leq \lambda_i$, for all $i = 1, ..., n$, and the last inequality is due to (21) and since $c < \tilde{c}$.

Case VI: Suppose $\lambda_n < c \leq \tilde{c} < \lambda_1$. Recall that $j^* := \max\{j : \lambda_j \geq c\}$ and $k^* := \max\{j : \sigma_j^2 \geq \tilde{c}\}$. We will consider three subcases.

VIa): Suppose $j^* < k^*$. From Theorem 3.1 and Corollary 4.1, the difference of the two MSEs is given by

$$\begin{aligned}&\text{MSE(multi}, c) - \text{MSE(n-single}, \tilde{c}) \\ &= \sum_{i=j^*+1}^{n} \lambda_i - \sum_{i=k^*+1}^{n} \sigma_i^2 - (k^* - j^*)c + k^*(c - \tilde{c}) \\ &= \sum_{i=j^*+1}^{k^*} \lambda_i + \sum_{i=k^*+1}^{n} \lambda_i - \sum_{i=k^*+1}^{n} \sigma_i^2 - (k^* - j^*)c + k^*(c - \tilde{c}) \\ &= \sum_{i=k^*+1}^{n} (\lambda_i - \sigma_i^2) + \sum_{i=j^*+1}^{k^*} (\lambda_i - c) + k^*(c - \tilde{c}) < 0. \qquad (24)\end{aligned}$$

The first sum in (24) is non-positive by (D). The second sum is strictly negative since $\lambda_i < c$, for all $i > j^*$. Therefore, $\text{MSE(multi}, c) < \text{MSE(n-single}, \tilde{c})$.

VIb): Suppose $k^* \leq j^*$. From Theorem 3.1 and Corollary 4.1, the difference of the two MSEs is given by

$$\begin{aligned}&\text{MSE(multi}, c) - \text{MSE(n-single}, \tilde{c}) \\ &= \sum_{i=k^*+1}^{j^*} (c - \sigma_i^2) + \sum_{i=j^*+1}^{n} (\lambda_i - \sigma_i^2) + k^*(c - \tilde{c}) \\ &= \sum_{i=k^*+1}^{j^*} (c - \sigma_i^2) + \sum_{i=1}^{j^*} (\sigma_i^2 - \lambda_i) + k^*(c - \tilde{c}) \\ &= \sum_{i=k^*+1}^{j^*} (c - \sigma_i^2) + \sum_{i=1}^{k^*} (\sigma_i^2 - \lambda_i) + \sum_{i=k^*+1}^{j^*} (\sigma_i^2 - \lambda_i) + k^*(c - \tilde{c}) \\ &\leq \sum_{i=k^*+1}^{j^*} (\lambda_i - \sigma_i^2) + \sum_{i=1}^{k^*} (\sigma_i^2 - \lambda_i) + \sum_{i=k^*+1}^{j^*} (\sigma_i^2 - \lambda_i) + k^*(c - \tilde{c}) \qquad (25) \\ &= \sum_{i=1}^{k^*} (\sigma_i^2 - \lambda_i) + k^*(c - \tilde{c}) \leq 0, \qquad (26)\end{aligned}$$

where the inequality in (25) holds due to the fact that $c \leq \lambda_i$, for all $i \leq j^*$, and the inequality (26) is due to (21). Therefore, $\text{MSE(multi}, c) \leq \text{MSE(n-single}, \tilde{c})$. □

Remark D.1. *Applicability without L2-regularization*: Our analysis relies on L2-regularization to yield non-trivial closed-form results, that is the UFM-approximation of training MSE in Theorem 3.1 holds when the UFM-regularization constant $c > 0$. If $c = 0$, it is easy to see that, for any $n \times d$ matrix $\mathbf{W}$ with full-rank n, considering the set of $\mathbf{H}$ that satisfy $||\mathbf{WH} - \mathbf{Y}||_F^2 = 0$, gives:

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

where $\mathbf{W}^+$ is the pseudoinverse of $\mathbf{W}$ and $\mathbf{Z}$ is any $d \times M$ matrix, and this is the well-known solution of the standard least squares problem. Thus, comparing multi-task ($c = 0$) with single-tasks ($\tilde{c} = 0$) is trivial as both training MSEs are identical and equal to zero. Such a UFM-inspired approximation agrees with classical notions of overfitting.

E Proofs of Theorems 5.1 and 5.2

Before delving into the proof of Theorems 5.1 and 5.2, we provide a brief introduction to the motivation and key concepts relevant to whitening.

In statistical analysis, whitening (or sphering) refers to a common pre-processing step to transform random variables to orthogonality. A whitening transformation (or sphering transformation) is a linear transformation that converts a random vector with a known covariance matrix into a new random vector of the same dimension and with covariance matrix given by the identity matrix. Orthogonality among random vectors greatly simplifies multivariate data analysis both from a computational as well as from a statistical standpoint. Whitening is employed mostly in pre-processing but is also part of modeling, see for instance (Hao et al., 2015; Zuber & Strimmer, 2009).

Due to rotational freedom there are infinitely many whitening transformations. All produce orthogonal but different sphered random variables. To understand differences between whitening transformations, and to select an optimal whitening procedure for a particular situation, the work of (Kessy et al., 2018), provided an overview of the underlying theory and discussed several natural whitening procedures. For example, they identified PCA whitening as the unique procedure that maximizes the compression of all components of the unprocessed vector in each component of the sphered vector. Of a particular interest for our work is *ZCA whitening*, ZCA standing for zero-phase component analysis. Rather than dimensionality reduction and data compression, ZCA whitening is useful for retaining maximal similarity between the unprocessed and the transformed variables.

Definition E.1. ZCA whitening employs the sphering matrix $\mathbf{W}^{ZCA} = \mathbf{\Sigma}^{-1/2}$, i.e.,

$$\mathbf{Y}^{ZCA} := \mathbf{\Sigma}^{-1/2}(\mathbf{Y} - \bar{\mathbf{Y}}).$$

We make the following observations:

1. Clearly, writing $\mathbf{W}^{ZCA} = \mathbf{Q}_1\mathbf{\Sigma}^{-1/2}$, where $\mathbf{Q}_1$ is an orthogonal matrix, $\mathbf{W}^{ZCA}$ satisfies
$$\mathbf{W}^{ZCA}\mathbf{\Sigma}(\mathbf{W}^{ZCA})^T = \mathbf{I}_n,$$
thus leading to new (of the infinitely many) whitening transformations. In fact, with $\mathbf{Q}_1 = \mathbf{I}_n$, ZCA whitening is the unique sphering method with a symmetric whitening matrix.
2. Breaking the rotational invariance by investigating the cross-covariance between unprocessed (centered) and sphered targets is key to identifying the optimal whitening transformations. The sample *cross-covariance* between $\mathbf{Y}^{ZCA}$ and $\mathbf{Y}$ is given by
$$\mathbf{\Phi} := \frac{\mathbf{Y}^{ZCA}(\mathbf{Y} - \bar{\mathbf{Y}})^T}{M} = \mathbf{W}^{ZCA}\frac{(\mathbf{Y} - \bar{\mathbf{Y}})(\mathbf{Y} - \bar{\mathbf{Y}})^T}{M} = \mathbf{W}^{ZCA}\mathbf{\Sigma} = \mathbf{Q}_1\mathbf{\Sigma}^{1/2}.$$
Note that $\mathbf{\Phi}$ is in general not symmetric, unless $\mathbf{Q}_1 = \mathbf{I}_n$. Using targets and their (ZCA)-whitened counterparts, the least squares objective is minimized when the trace of the cross-covariance is maximized, e.g., (Kessy et al., 2018)[eq. (13) and (14)],
$$\frac{1}{M}||\mathbf{Y}^{ZCA} - (\mathbf{Y} - \bar{\mathbf{Y}})||_F^2 = n - 2\text{tr}(\mathbf{\Phi}) + \sum_{i=1}^{n}\sigma_i^2 = n - 2\text{tr}(\mathbf{Q}_1\mathbf{\Sigma}^{1/2}) + \sum_{i=1}^{n}\sigma_i^2.$$

It can be shown that the minimum of the latter is attained at $\mathbf{Q}_1 = \mathbf{I}_n$. Therefore, not only ZCA whitening is the unique sphering method with a symmetric whitening matrix. It is also the optimal whitening apporach identified by evaluating the objective function of the total squared distance between the unprocessed and the whitened targets, computed from the cross-covariance $\mathbf{\Phi}$.

To summarize, ZCA whitening is the unique procedure used with the aim of making the transformed targets as similar as possible to the unprocessed targets, which is appealing since in many applications it is desirable to remove dependencies with minimal additional adjustments.

We reformulate Theorem 5.1 to also include information about the structure of the global minima, and consequently the target predictions regarding those.

Theorem E.2. *Let $c := \lambda_{\mathbf{H}}\lambda_{\mathbf{W}}$. Any global minimum $(\tilde{\mathbf{H}}, \tilde{\mathbf{W}})$ of the regularized UFM-loss*

$$\frac{1}{2M}||\mathbf{W}\mathbf{H} - \mathbf{Y}^{ZCA}||_F^2 + \frac{\lambda_{\mathbf{H}}}{2M}||\mathbf{H}||_F^2 + \frac{\lambda_{\mathbf{W}}}{2}||\mathbf{W}||_F^2, \tag{27}$$

where $\lambda_{\mathbf{H}}$ and $\lambda_{\mathbf{W}}$ are non-negative regularization parameters, takes the following form.

If $0 < c < 1$, then for any semi-orthogonal matrix $\mathbf{R}$,

$$\tilde{\mathbf{W}} = \left(\frac{\lambda_{\mathbf{H}}}{\lambda_{\mathbf{W}}}\right)^{1/4} \tilde{\mathbf{A}}^{1/2}\mathbf{R}, \qquad \tilde{\mathbf{H}} = \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}\tilde{\mathbf{W}}^T\mathbf{Y}^{ZCA}, \tag{28}$$

$$\tilde{\mathbf{W}}\tilde{\mathbf{H}} = (1 - \sqrt{c})\mathbf{Y}^{ZCA},$$

where $\tilde{\mathbf{A}} = (1 - \sqrt{c})\mathbf{I}_n$.

If $c > 1$, then $(\tilde{\mathbf{H}}, \tilde{\mathbf{W}}) = (\mathbf{0}, \mathbf{0})$.

Furthermore,

$$\text{MSE(de-whiten)} = \begin{cases} c\sum_{i=1}^{n} \lambda_i, & \text{if } c < 1, \\ \sum_{i=1}^{n} \lambda_i, & \text{if } c \geq 1, \end{cases}$$

where λ_i is the i-th eigenvalue (in descending order) of the original sample covariance matrix $\mathbf{\Sigma}$.

Before giving the proof, let us first discuss the nature of $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{H}}$ when whitening is applied. In light of (28), properties that hold are:

1. The rows of $\tilde{\mathbf{W}}$ are orthogonal (due to $\tilde{\mathbf{A}} = (1 - \sqrt{c})\mathbf{I}_n$ being diagonal).
2. The rows of $\tilde{\mathbf{W}}$ are equinorm. More specifically,
$$||\tilde{\mathbf{w}}_j||_2^2 = \lambda_{\mathbf{H}}\left(\frac{1}{\sqrt{c}} - 1\right), \qquad j = 1, ..., n.$$
3. The angles between the columns of $\tilde{\mathbf{H}}$ are equal to angles between the whitened $\tilde{\mathbf{y}}_i$'s, i.e.,
$$\tilde{\mathbf{H}}^T\tilde{\mathbf{H}} = \lambda_{\mathbf{W}}\left(\frac{1}{\sqrt{c}} - 1\right)(\mathbf{Y}^{ZCA})^T\mathbf{Y}^{ZCA}.$$

Proof of Theorem E.2. It is easily seen that

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

Thus, the covarance matrix for the whitened targets is the $n \times n$ identity matrix.
Case $c < 1$:
By (Andriopoulos et al., 2024)[Theorem 4.1], we have that any global minimum $(\tilde{\mathbf{H}}, \tilde{\mathbf{W}})$ for (27) takes the form:

$$\tilde{\mathbf{W}} = \left(\frac{\lambda_{\mathbf{H}}}{\lambda_{\mathbf{W}}}\right)^{1/4} \tilde{\mathbf{A}}^{1/2}\mathbf{R}, \qquad \tilde{\mathbf{H}} = \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}\tilde{\mathbf{W}}^T\mathbf{Y}^{ZCA},$$

where $\tilde{\mathbf{A}} = (1-\sqrt{c})\mathbf{I}_n$. Note that, under whitening $j^* = \max\{j : c \leq 1\} = n$. The optimal predictions after whitening (but before de-whitening) are

$$\tilde{\mathbf{W}}\tilde{\mathbf{H}} = \tilde{\mathbf{A}}^{1/2}\mathbf{R}\mathbf{R}^T\tilde{\mathbf{A}}^{1/2}\mathbf{Y}^{ZCA} = (1-\sqrt{c})\mathbf{Y}^{ZCA}.$$

Our final de-whitened predictions satisfy

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

Therefore,

$$\hat{\mathbf{Y}} - \mathbf{Y} = -\sqrt{c}(\mathbf{Y}-\bar{\mathbf{Y}}),$$

and

$$\frac{(\hat{\mathbf{Y}}-\mathbf{Y})(\hat{\mathbf{Y}}-\mathbf{Y})^T}{M} = c\frac{(\mathbf{Y}-\bar{\mathbf{Y}})(\mathbf{Y}-\bar{\mathbf{Y}})^T}{M} = c\mathbf{\Sigma}.$$

The result for MSE(de-whiten) readily follows by taking traces in both sides.
Case $c \geq 1$:
By (Andriopoulos et al., 2024)[Theorem 4.1], we have that the only global minimum is $(\tilde{\mathbf{H}}, \tilde{\mathbf{W}}) = (\mathbf{0}, \mathbf{0})$. Therefore,

$$\hat{\mathbf{Y}} = \bar{\mathbf{Y}},$$

and

$$\frac{(\hat{\mathbf{Y}}-\mathbf{Y})(\hat{\mathbf{Y}}-\mathbf{Y})^T}{M} = \frac{(\mathbf{Y}-\bar{\mathbf{Y}})(\mathbf{Y}-\bar{\mathbf{Y}})^T}{M} = \mathbf{\Sigma}.$$

The result for MSE(de-whiten) readily follows by taking traces in both sides. □

Proof of Theorem 5.2. Suppose $0 < c \leq 1$. By Theorem 3.1 and Theorem 5.1,

$$\begin{aligned}
\text{MSE(de-whiten)} - \text{MSE(multi)} &= c\sum_{i=1}^{n}\lambda_i - j^*c - \sum_{i=j^*+1}^{n}\lambda_i \\
&= c\sum_{i=1}^{j^*}\lambda_i + c\sum_{i=j^*+1}^{n}\lambda_i - j^*c - \sum_{i=j^*+1}^{n}\lambda_i \\
&= c\left[\sum_{i=1}^{j^*}\lambda_i - j^*\right] + (c-1)\sum_{i=j^*+1}^{n}\lambda_i < 0
\end{aligned}$$

if and only if

$$\sum_{i=1}^{j^*}\lambda_i - j^* < c^{-1}(1-c)\sum_{i=j^*+1}^{n}\lambda_i,$$

which is condition (14) as postulated in the assumptions of Theorem 5.2(i).

Suppose $c > 1$. By Theorem 3.1 and Theorem 5.1,

$$\begin{aligned}
\text{MSE(de-whiten)} - \text{MSE(multi)} &= \sum_{i=1}^{n}\lambda_i - j^*c - \sum_{i=j^*+1}^{n}\lambda_i \\
&= \sum_{i=1}^{j^*}\lambda_i + \sum_{i=j^*+1}^{n}\lambda_i - j^*c - \sum_{i=j^*+1}^{n}\lambda_i \\
&= \sum_{i=1}^{j^*}(\lambda_i - c) \geq 0.
\end{aligned}$$

since by definition $\lambda_i \geq c$, for all $i \leq j^*$. □

F PROOF OF THEOREM 5.3

We reformulate Theorem 5.3 to also include information about the structure of the global minima, and consequently the target predictions regarding those.

Theorem F.1. *Let $c := \lambda_{\mathbf{H}}\lambda_{\mathbf{W}}$. Any global minimum $(\bar{\mathbf{H}}, \bar{\mathbf{W}})$ of the regularized UFM-loss*

$$\frac{1}{2M}||\mathbf{W}\mathbf{H} - \mathbf{Y}^{nrm}||_F^2 + \frac{\lambda_{\mathbf{H}}}{2M}||\mathbf{H}||_F^2 + \frac{\lambda_{\mathbf{W}}}{2}||\mathbf{W}||_F^2, \tag{29}$$

where $\lambda_{\mathbf{H}}$ and $\lambda_{\mathbf{W}}$ are non-negative regularization parameters, takes the following form.

If $0 < c < \tilde{\lambda}_{\min}$, then for any semi-orthogonal matrix $\mathbf{R}$,

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

$$\bar{\mathbf{W}}\bar{\mathbf{H}} = [\mathbf{I}_n - \sqrt{c}\mathbf{P}^{-1/2}]\mathbf{Y}^{nrm},$$

where $\bar{\mathbf{A}} = \mathbf{P}^{1/2} - \sqrt{c}\mathbf{I}_n$.

If $c > \tilde{\lambda}_{\max}$, then $(\bar{\mathbf{H}}, \bar{\mathbf{W}}) = (\mathbf{0}, \mathbf{0})$.

Furthermore,

$$\text{MSE(de-normalize)} = \begin{cases} c\sum_{i=1}^{n} \lambda_i, & \text{if } 0 < c < \tilde{\lambda}_{\min}, \\ \sum_{i=1}^{n} \lambda_i, & \text{if } c > \tilde{\lambda}_{\max}, \end{cases}$$

where $\tilde{\lambda}_{\min}$ and $\tilde{\lambda}_{\max}$ are the min and max eigenvalues of the original sample correlation matrix $\mathbf{P}$.

Proof of Theorem F.1. Using the decomposition of $\mathbf{\Sigma} = \mathbf{V}^{1/2}\mathbf{P}\mathbf{V}^{1/2}$, it is readily deduced that

$$M^{-1}\mathbf{Y}^{nrm}(\mathbf{Y}^{nrm})^T = \mathbf{V}^{-\frac{1}{2}}\mathbf{\Sigma}\mathbf{V}^{-\frac{1}{2}} = \mathbf{P}.$$

Case $0 < c < \tilde{\lambda}_{\min}$:
By (Andriopoulos et al., 2024)[Theorem 4.1], we have that any global minimum $(\bar{\mathbf{H}}, \bar{\mathbf{W}})$ for (29) takes the form:

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

where $\bar{\mathbf{A}} = \mathbf{P}^{1/2} - \sqrt{c}\mathbf{I}_n$. The optimal predictions after normalization (but before de-normalizing) are

$$\begin{aligned}\bar{\mathbf{W}}\bar{\mathbf{H}} = \bar{\mathbf{A}}^{1/2}\mathbf{R}\mathbf{R}^T\bar{\mathbf{A}}^{1/2}\mathbf{P}^{-1/2}\mathbf{Y}^{nrm} = \bar{\mathbf{A}}\mathbf{P}^{-1/2}\mathbf{Y}^{nrm} &= [\mathbf{P}^{1/2} - \sqrt{c}\mathbf{I}_n]\mathbf{P}^{-1/2}\mathbf{Y}^{nrm} \\ &= [\mathbf{I}_n - \sqrt{c}\mathbf{P}^{-1/2}]\mathbf{Y}^{nrm}\end{aligned}$$

Our final de-normalized predictions satisfy

$$\begin{aligned}\check{\mathbf{Y}} = [\mathbf{V}^{1/2}]\bar{\mathbf{W}}\bar{\mathbf{H}} + \bar{\mathbf{Y}} &= \mathbf{V}^{1/2}[\mathbf{V}^{-1/2} - \sqrt{c}\mathbf{P}^{-1/2}\mathbf{V}^{-1/2}](\mathbf{Y} - \bar{\mathbf{Y}}) + \bar{\mathbf{Y}} \\ &= \mathbf{Y} - \sqrt{c}\mathbf{V}^{1/2}\mathbf{P}^{-1/2}\mathbf{V}^{-1/2}(\mathbf{Y} - \bar{\mathbf{Y}})\end{aligned}$$

Therefore,

$$\check{\mathbf{Y}} - \mathbf{Y} = -\sqrt{c}\mathbf{V}^{1/2}\mathbf{P}^{-1/2}\mathbf{V}^{-1/2}(\mathbf{Y} - \bar{\mathbf{Y}}),$$

and

$$\begin{aligned}\frac{(\check{\mathbf{Y}} - \mathbf{Y})(\check{\mathbf{Y}} - \mathbf{Y})^T}{M} &= c\mathbf{V}^{1/2}\mathbf{P}^{-1/2}\left[\mathbf{V}^{-1/2}\mathbf{\Sigma}\mathbf{V}^{-1/2}\right]\mathbf{P}^{-1/2}\mathbf{V}^{1/2} \\ &= c\mathbf{V}^{1/2}\mathbf{P}^{-1/2}\mathbf{P}\mathbf{P}^{-1/2}\mathbf{V}^{1/2} \\ &= c\mathbf{V}\end{aligned}$$

The result for MSE(de-normalize) readily follows by taking traces in both sides.

Case $c > \tilde{\lambda}_{\max}$:

By (Andriopoulos et al., 2024)[Theorem 4.1], we have that the only global minimum is $(\bar{\mathbf{H}}, \bar{\mathbf{W}}) = (\mathbf{0}, \mathbf{0})$. Therefore,

$$\check{\mathbf{Y}} = \bar{\mathbf{Y}},$$

and

$$\frac{(\check{\mathbf{Y}} - \mathbf{Y})(\check{\mathbf{Y}} - \mathbf{Y})^T}{M} = \frac{(\mathbf{Y} - \bar{\mathbf{Y}})(\mathbf{Y} - \bar{\mathbf{Y}})^T}{M} = \mathbf{\Sigma}.$$

The result for MSE(de-normalize) readily follows by taking traces in both sides. □

G FEATURE RELATIONSHIPS ACROSS METHODS

Since the UFM was first proposed to understand feature learning, and especially the neural collapse phenomenon in classification (Papyan et al., 2020) and neural multivariate regression (Andriopoulos et al., 2024), it can also provide insights on the feature $\mathbf{H}$ learned by different methods. For example, how do the features learned by training with original targets or by whitening, and normalization, compare with each other?

Regarding this insightful question, our analysis in the Appendix (Theorem E.2 and the remarks before its proof, and Theorem F.1) explicitly discuss the nature of the optimal $\mathbf{H}_*$ when whitening and normalization are applied respectively. Let us collect the globally optimal learned features and compare them across methods here as well.

Training with original targets $\mathbf{Y}$:

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

where $\mathbf{R} \in \mathbb{R}^{n \times d}$ is semi-orthogonal and $\mathbf{Y}^{ZCA} = \mathbf{\Sigma}^{-1/2}(\mathbf{Y} - \bar{\mathbf{Y}})$.

The key observation is that the optimal learned features are formed by two procedures. The term $\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n$ adjusts the covariance of the original target data $\mathbf{Y}$. By subtracting $\sqrt{c}\mathbf{I}_n$ from $\mathbf{\Sigma}^{1/2}$, small eigenvalues are regularized or "shrunk", effectively denoising the original target data. Taking the root of the result further scales the eigenvalues nonlinearly, emphasizing stronger signal directions. The first procedure consists of the whitened target data undergoing adaptive scaling, i.e., the multiplication $[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n]^{1/2}\mathbf{Y}^{ZCA}$ re-weights the whitened target data using the thresholded eigenvalues from $\mathbf{\Sigma}$. Directions aligned with strong original covariance are amplified, while weak/noisy directions are zeroed out. The second procedure consists of a rotation of the first procedure's outcome.The semi-orthogonal matrix $\mathbf{R}^T$ acts as a rotation, redistributing the features into a coordinate system that preserves distances but may optimize for properties like orthogonality or sparsity. To summarize, the two procedures ensure a denoised, lower dimensional feature representation that retains only statistically significant components from the covariance of the original target data.

For single-task i:

$$\mathbf{H}_*^{(i)} = \left(\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}\right)^{1/4} \mathbf{R}^T [\sigma_i - \sqrt{c}]^{1/2} \sigma_i^{-1} \mathbf{y}^{(i)},$$

where $\mathbf{y}^{(i)}$ is the i-th row of $\mathbf{Y}$. There is no a priori relationship between $\mathbf{H}_*$ and $\mathbf{H}_*^{(i)}$. However, we note that in the special case when the targets are uncorrelated, it is easy to see that $\mathbf{H}_*^{(i)}$ is the i-th row of $\mathbf{H}_*$.

Training with whitened targets $\mathbf{Y}^{ZCA}$:

$$\mathbf{H}_* = \left(\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}\right)^{1/4} \mathbf{R}^T [\mathbf{I}_n^{1/2} - \sqrt{c}\mathbf{I}_n]^{1/2} \mathbf{Y}^{ZCA}.$$

In this case, the form of the optimal learned features is retained with the difference that the $n \times n$ identity matrix $\mathbf{I}_n$ takes the place of $\mathbf{\Sigma}$. The whitened target data is simply scaled down by $\sqrt{1 - \sqrt{c}}$, if $c < 1$, akin to mild regularization. No denoising (eigenvalue thresholding) takes place, and all the directions are kept.

Training with normalized targets $\mathbf{Y}^{nrm} = \mathbf{V}^{-1/2}(\mathbf{Y} - \bar{\mathbf{Y}})$:

Recall that we have used $\mathbf{P}$ to denote the correlation matrix of the targets. Then,

$$\mathbf{H}_* = \left(\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}\right)^{1/4} \mathbf{R}^T[\mathbf{P}^{1/2} - \sqrt{c}\mathbf{I}_n]^{1/2}\mathbf{Y}^{ZCA-cor},$$

where $\mathbf{Y}^{ZCA-cor} := \mathbf{P}^{-1/2}\mathbf{Y}^{nrm}$ is referred to in the literature as the ZCA-cor whitening (Kessy et al., 2018), and it is the unique whitening procedure that makes the transformed normalized targets as similar as possible to the original normalized targets, the similarity being in terms of the cross-correlation between the former and the latter.

The optimal leaned features are obtained in accordance to the analysis that we have outlined when training with original targets. Here, the role of $\mathbf{\Sigma}$ is played by $\mathbf{P}$ and in the place of $\mathbf{Y}^{ZCA}$, we have $\mathbf{Y}^{ZCA-cor}$.

H GENERALIZATION BOUNDS

Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ the target space, which regarding the learning problem of neural multivariate regression, is a subset of $\mathbb{R}^n$. Here, we adopt the stochastic scenario and will denote by $\mathcal{D}$ a distribution over $\mathcal{X} \times \mathcal{Y}$. In the supervised learning scenario, the learner receives training examples $S := \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, ..., M\} \in (\mathcal{X} \times \mathcal{Y})^M$ drawn in a i.i.d. manner according to $\mathcal{D}$. The deterministic scenario where input points admit a unique target value determined by a target function $f : \mathcal{X} \to \mathcal{Y}$ is a straightforward special case.

We denote by $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ the loss function used to measure the magnitude of the difference between the vector-valued target predicted and the "true" or "correct" one. The most common loss function used in neural multivariate regression is the L_p-loss defined by $L(\mathbf{y}, \mathbf{y}') = ||\mathbf{y} - \mathbf{y}'||_p$, for some $p \geq 1$, and every $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$.

Given a hypothesis set $\mathcal{H}$, that is a set which contains all the functions mapping the input space $\mathcal{X}$ to the target space $\mathcal{Y}$, neural multivariate regression tasks consist of using a set of training examples S to find a hypothesis $h \in \mathcal{H}$ with small generalization error $R(h)$ with respect to the "true" or "correct" *target function* mapping inputs to targets in S.

Definition H.1 (Generalization error)**.** Given a hypothesis $h \in \mathcal{H}$, a *target function*, and an underlying distribution $\mathcal{D}$, the generalization error is defined by

$$R(h) := \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\left[L(h(\mathbf{x}), \mathbf{y})\right],$$

where $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is the loss function used to measure the magnitude of error.

The generalization error is not directly accessible to the learner since both the distribution $\mathcal{D}$ and the *target function* are unknown. However, the learner can measure the empirical error of a hypothesis on a set of training examples S.

Definition H.2 (Training error)**.** Given a hypothesis $h \in \mathcal{H}$, a *target function*, and a training set S, the training error is

$$\hat{R}_S(h) := \frac{1}{M}\sum_{i=1}^{M} L(h(\mathbf{x}_i), \mathbf{y}_i).$$

Thus, the training error of h is its average error over the training set S, while the generalization error is its expected error based on the distribution $\mathcal{D}$. If L is the squared loss, the training error represents the training MSE of h on S.

The theoretical results presented below are based on the assumption that the neural multivariate regression problem is bounded, that is when the loss function is bounded above by some $K > 0$, i.e., $L(\mathbf{y}, \mathbf{y}') \leq K$, for all $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$, or, more strictly, when $L(h(\mathbf{x}), \mathbf{y}) \leq K$, for all $h \in \mathcal{H}$, and $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$.

H.1 GENERALIZATION BOUNDS FOR KERNEL-BASED HYPOTHESES

Generalization bounds based on Rademacher complexity (RC), a term that captures the richness of a family of functions by measuring the degree to which a hypothesis set can fit random noise,

were presented in (Mohri, 2018), e.g., Theorem 11.3 therein. These generalization bounds suggest a trade-off between reducing the training MSE and controlling the RC of the hypothesis set. A richer or more complex hypothesis set achieves a small training MSE but has high RC, while a poorer or more simple hypothesis set has small RC but achieves high training MSE. The aim is to control this trade-off. An important benefit of the learning bounds in Mohri (2018)[Theorem 11.3] is that they are data dependent. This can lead to more accurate learning guarantees. For kernel-based hypotheses upper bounds on the RC can be used directly to derive generalization bounds depending on the trace of the kernel matrix or the maximum diagonal entry.

Definition H.3 (Kernel)**.** A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be a positive definite symmetric (PDS) kernel if for any $\{\mathbf{x}_i : i = 1, ..., M\} \in \mathcal{X}$, the matrix $K := [K(\mathbf{x}_i, \mathbf{x}_j)]_{i,j} \in \mathbb{R}^{M \times M}$ is symmetric positive semi-definite (SPSD).

For a PDS kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a Hilbert space $\mathbb{H}$ and a mapping $\Phi : \mathcal{X} \to \mathbb{H}$ such that:

$$K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle , \qquad \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}.$$

For a proof of this result, we refer the reader to Mohri (2018)[Theorem 6.8].

A generalization bound for (univariate) linear regression with bounded linear hypotheses in a feature space defined by a PDS kernel was presented in Mohri (2018)[Theorem 11.11]. For simplicity, we give the generalization bound for the squared loss.

Theorem H.4. *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a PDS kernel, $\Phi : \mathcal{X} \to \mathbb{H}$ be a feature mapping associated with K, and*

$$\mathcal{H} := \{\mathbf{x} \mapsto \mathbf{w} \cdot \Phi(\mathbf{x}) : ||\mathbf{w}|| \leq \Lambda_{\mathbf{w}}\}$$

be the family of bounded linear hypotheses corresponding to the optimization problem

$$\min_{\mathbf{w}} \frac{1}{M} \sum_{i=1}^{M} (\mathbf{w} \cdot \Phi(\mathbf{x}_i) - \mathbf{y}_i)^2, \text{ subject to } ||\mathbf{w}|| \leq \Lambda_{\mathbf{w}},$$

for a training set $S = \{(\mathbf{x}_i, \mathbf{y}_i) : i = 1, ..., M\}$.

- *Assume that there exists $K > 0$ such that $|h(\mathbf{x}) - \mathbf{y}| \leq K$, for all $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$.*
- *Let tr $([K(\mathbf{x}_i, \mathbf{x}_j)]_{i,j}) \leq Mr^2$, for any training set S of size M.*

Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds for all $h \in \mathcal{H}$:

$$\mathbb{E}_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}}[|h(\mathbf{x}) - \mathbf{y}|^2] \leq \text{MSE}_S(h) + 4K \frac{r\Lambda_{\mathbf{w}}}{\sqrt{M}} + 3K^2 \sqrt{\frac{\log \frac{2}{\delta}}{2M}}. \tag{30}$$

The generalization bound of the theorem above suggest minimizing a trade-off between the training MSE, denoted in (30) by $\text{MSE}_S(h)$, and the norm of the weight vector $\mathbf{w}$. The third term adds an error dependent on the confidence level δ and the size of the training set M.

H.2 Application to the UFM via the Layer-Peeled Model

Another formulation of the UFM is the so-called Layer-Peeled Model introduced by Fang et al. (2021) as:

$$\min_{\mathbf{W},\mathbf{H}} ||\mathbf{W}\mathbf{H} - \mathbf{Y}||_F^2, \text{ subject to } \begin{cases} ||\mathbf{W}||_F & \leq \Lambda_{\mathbf{W}}, \\ \dfrac{||\mathbf{H}||_F}{\sqrt{M}} & \leq \Lambda_{\mathbf{H}}, \end{cases} \tag{31}$$

where $\mathbf{W} \in \mathbb{R}^{n \times d}$ is, as in (3), a linear classifier in the last layer, $\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_M] \in \mathbb{R}^{d \times M}$, where $\mathbf{h}_i := \mathbf{h}_\theta(\mathbf{x}_i)$ is the d-dimensional last-layer activation/feature of the i-th training sample, and $\Lambda_{\mathbf{W}}, \Lambda_{\mathbf{H}}$ are positive scalars constraining the matrix-norms of $\mathbf{W}$ and $\mathbf{H}$ respectively.

A constrained minimization problem can be solved by means of the Karush-Kuhn-Tucker (KKT) multiplier method, which minimizes a function subject to inequality constraints. The KKT multiplier method states that, under some regularity conditions (all met here), there exist constants $\nu_{\mathbf{W}}, \nu_{\mathbf{H}} \geq 0$, called the multipliers, such that the solution $(\mathbf{W}(\nu_{\mathbf{W}}), \mathbf{H}(\nu_{\mathbf{H}}))$ of the constrained minimization problem (31) satisfies the so-called KKT conditions.

- The first condition (referred to as the stationarity condition) demands that the gradients of the Langrangian

$$\Delta(\mathbf{W}, \mathbf{H}) := ||\mathbf{W}\mathbf{H} - \mathbf{Y}||_F^2 + \nu_{\mathbf{W}}(||\mathbf{W}||_F^2 - \Lambda_{\mathbf{W}}^2) + \nu_{\mathbf{H}}(||\mathbf{H}||_F^2 - M\Lambda_{\mathbf{H}}^2),$$

associated with the minimization problem (3), i.e., the UFM, is 0 at the solution $(\mathbf{W}(\nu_{\mathbf{W}}), \mathbf{H}(\nu_{\mathbf{H}}))$. More specifically,

$$\left.\frac{\partial \Delta}{\partial \mathbf{W}}\right|_{(\mathbf{W}(\nu_{\mathbf{W}}),\mathbf{H}(\nu_{\mathbf{H}}))} = 2(\mathbf{W}(\nu_{\mathbf{W}})\mathbf{H}(\nu_{\mathbf{H}}) - \mathbf{Y})\mathbf{H}(\nu_{\mathbf{H}})^T + 2\nu_{\mathbf{W}}\mathbf{W}(\nu_{\mathbf{W}}), \tag{32}$$

$$\left.\frac{\partial \Delta}{\partial \mathbf{H}}\right|_{(\mathbf{W}(\nu_{\mathbf{W}}),\mathbf{H}(\nu_{\mathbf{H}}))} = 2\mathbf{W}(\nu_{\mathbf{W}})^T(\mathbf{W}(\nu_{\mathbf{W}})\mathbf{H}(\nu_{\mathbf{H}}) - \mathbf{Y}) + 2\nu_{\mathbf{H}}\mathbf{H}(\nu_{\mathbf{H}}). \tag{33}$$

- The second KKT condition (referred to as the complimentarity condition) requires that

$$\nu_{\mathbf{W}}(||\mathbf{W}(\nu_{\mathbf{W}})||_F^2 - \Lambda_{\mathbf{W}}^2) = 0, \qquad \nu_{\mathbf{H}}(||\mathbf{H}(\nu_{\mathbf{H}})||_F^2 - M\Lambda_{\mathbf{H}}^2) = 0. \tag{34}$$

If

$$\nu_{\mathbf{W}} = \lambda_{\mathbf{W}}, \qquad \nu_{\mathbf{H}} = \lambda_{\mathbf{H}},$$
$$\Lambda_{\mathbf{W}} = ||\mathbf{W}(\lambda_{\mathbf{W}})||_F, \qquad \Lambda_{\mathbf{H}} = \frac{||\mathbf{H}(\lambda_{\mathbf{H}})||_F}{\sqrt{M}},$$

the UFM solution $(\mathbf{W}(\lambda_{\mathbf{W}}), \mathbf{H}(\lambda_{\mathbf{H}}))$ satisfies (32)-(34). Therefore, the theorem that follows is immediately deduced.

Theorem H.5. *If*

$$\Lambda_{\mathbf{W}} = ||\mathbf{W}(\lambda_{\mathbf{W}})||_F, \qquad \Lambda_{\mathbf{H}} = \frac{||\mathbf{H}(\lambda_{\mathbf{H}})||_F}{\sqrt{M}},$$

the minimization problems of the UFM (3) and the Layer-Peeled Model (31) have the same solution.

Remark H.6. The UFM solutions $\mathbf{W}(\lambda_{\mathbf{W}})$ and $\mathbf{H}(\lambda_{\mathbf{H}})$ are always to be found on the boundary of the Layer-Peeled Model constraints, parameterized by $\{(\mathbf{W}, \mathbf{H}) : ||\mathbf{W}||_F \leq \Lambda_{\mathbf{W}}, ||\mathbf{H}||_F \leq M\Lambda_{\mathbf{H}}\}$ for some $\Lambda_{\mathbf{W}}, \Lambda_{\mathbf{H}} > 0$. The size of the spherical constraints of the Layer-Peeled Model shrink as the regularizing constants of the UFM increase, and eventually in the $\lambda_{\mathbf{W}}, \lambda_{\mathbf{H}} \to \infty$-limit, $\Lambda_{\mathbf{W}}, \Lambda_{\mathbf{H}} \to 0$. This follows from the closed-form functions of the minimizers for the UFM with respect to each other, i.e.,

$$\lim_{\lambda_{\mathbf{w}} \to \infty} \mathbf{W}(\lambda_{\mathbf{W}}) = \lim_{\lambda_{\mathbf{W}} \to \infty} \mathbf{Y}\mathbf{H}^T[\mathbf{H}\mathbf{H}^T + M\lambda_{\mathbf{W}}]^{-1} = 0,$$
$$\lim_{\lambda_{\mathbf{H}} \to \infty} \mathbf{H}(\lambda_{\mathbf{H}}) = \lim_{\lambda_{\mathbf{H}} \to \infty} \mathbf{W}^T[\mathbf{W}\mathbf{W}^T + \lambda_{\mathbf{H}}\mathbf{I}_n]^{-1}\mathbf{Y} = 0.$$

Remark H.7. Suppose $c < \lambda_{\min}$. By applying Andriopoulos et al. (2024)[Corollary 4.2 (ii)-(iii)] to the n-dimensional case, the following relation holds:

$$\Lambda_{\mathbf{W}} = ||\mathbf{W}(\lambda_{\mathbf{W}})||_F = \left(\frac{\lambda_{\mathbf{H}}}{\lambda_{\mathbf{W}}}\right)^{1/4} \left[\text{tr}\left(\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n\right)\right]^{1/2},$$
$$\Lambda_{\mathbf{H}} = \frac{||\mathbf{H}(\lambda_{\mathbf{H}})||_F}{\sqrt{M}} = \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}||\mathbf{W}(\lambda_{\mathbf{W}})||_F = \sqrt{\frac{\lambda_{\mathbf{W}}}{\lambda_{\mathbf{H}}}}\Lambda_{\mathbf{W}}.$$

Recall that $\mathbf{h}_i := \mathbf{h}_\theta(x_i) \in \mathbb{R}^d$ for all $i = 1, ..., M$, where $\mathbf{h}_\theta$ is associated with the PDS kernel $K = \mathbf{H}^T\mathbf{H}$, i.e., $[\mathbf{h}_i \cdot \mathbf{h}_j]_{i,j} \in \mathbb{R}^{M \times M}$ is the covariance matrix of the $\mathbf{h}_i$'s and as such it is symmetric and positive semi-definite. Under the constraints that $\mathbf{W}$ and $\mathbf{H}$ are subject to:

$$\text{tr}(K) = \text{tr}(\mathbf{H}^T\mathbf{H}) = ||\mathbf{H}||_F^2 \leq M\Lambda_{\mathbf{H}}^2.$$

Under the UFM, when $c < \lambda_{\min}$, $\text{MSE}_S(h) = c = \lambda_{\mathbf{W}}\lambda_{\mathbf{H}}$ for any labeled sample S, see Theorem 3.1. Combining the points above, the generalization bound of (30) yields

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})\in\mathcal{D}}[|h(\mathbf{x}) - \mathbf{y}|^2] \leq nc + \mathcal{O}\left(\frac{C}{\sqrt{M}}\right) + \mathcal{O}\left(\sqrt{\frac{\log 2\delta^{-1}}{M}}\right),$$

where $C := \Lambda_{\mathbf{W}}\Lambda_{\mathbf{H}}$. In the special case in which we set $\lambda_{\mathbf{W}} = \lambda_{\mathbf{H}}$, we have $c = \lambda_{\mathbf{W}}^2$, and $C = \Lambda_{\mathbf{W}}^2 = \text{tr}\left(\mathbf{\Sigma}^{1/2} - \lambda_{\mathbf{W}}\mathbf{I}_n\right)$, and the generalization bound reads

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})\in\mathcal{D}}[|h(\mathbf{x}) - \mathbf{y}|^2] \leq n\lambda_{\mathbf{W}}^2 + \mathcal{O}\left(\frac{\text{tr}\left(\mathbf{\Sigma}^{1/2} - \lambda_{\mathbf{W}}\mathbf{I}_n\right)}{\sqrt{M}}\right) + R, \tag{35}$$

where, for a fixed confidence level $\delta \in (0,1)$, $\lim_{M\to\infty} R = 0$.

For single task i, the right-hand side (wit the remainder term) of (35) becomes

$$\text{GB}^{(i)} := \lambda_{\mathbf{W}}^2 + \mathcal{O}(M^{-1/2}(\sigma - \lambda_{\mathbf{W}})) + R.$$

Using this, we can directly derive an upper bound for the generalization error of the n-single tasks neural regression problem:

$$\text{GB(n-single)} \leq \sum_{i=1}^{n} \text{GB}^{(i)} = n\lambda_{\mathbf{W}}^2 + \mathcal{O}\left(\frac{\sum_{i=1}^{n}\sigma_i - n\lambda_{\mathbf{W}}}{\sqrt{M}}\right) + \tilde{R}, \tag{36}$$

where, for a fixed confidence level $\delta \in (0,1)$, $\lim_{M\to\infty} \tilde{R} = 0$.

Because $\sum_{i=1}^{n}\sigma_i = \sum_{i=1}^{n}\sqrt{\Sigma_{ii}} \geq \text{tr}(\mathbf{\Sigma}^{1/2})$, the $\mathcal{O}$-term in (36) is $\geq$ than the $\mathcal{O}$-term in (35), so the whole right-hand side of the bound in (36) is $\geq$ than the right-hand side of the bound in (36).

If the two test MSEs concentrate near their respective bounds, then the multi-task test MSE is smaller than that of the n single tasks; thus the gain from multi-tasking is tightest when $\mathbf{\Sigma}$ is non-diagonal and the features across tasks are correlated.