Spectral Insights from the Unconstrained Feature Model for Neural Multi-Output Regression

George Andriopoulos* NYU Abu Dhabi

Bimarsha Adhikari* NYU Abu Dhabi

Soyuj Jung Basnet NYU

Juan Guevara MBZUAI

Li Guo NYU Shanghai

Keith W. Ross NYU Abu Dhabi

Abstract

Multi-output regression can be approached either by fitting one joint vector-valued predictor or by training separate univariate predictors for each target coordinate. This holistic-versus-separate choice is classical in the largely non-neural multi-output regression literature, but is less characterized analytically for neural networks, where joint prediction arises naturally through shared features and vector-valued output layers. Because neural networks are highly non-linear and high-dimensional, exact analysis is notoriously difficult. We study two questions using the Unconstrained Feature Model (UFM) as a tractable surrogate for exact analysis: **(1)** how does joint-output regression compare with independent coordinate-wise regression under regularization? **(2)** when can target whitening or normalization improve original-scale MSE? Under UFM, the minimal training MSE depends only on the regularization product and the spectrum of the target covariance. This yields two theoretical results: joint-output regression has no larger training MSE than independent coordinate-wise regression under matched regularization, and whitening or normalizing targets can help or hurt original-scale MSE depending on the average target variance. Experiments on neural networks trained with standard weight decay across robotic imitation-learning and autonomous-driving datasets support these theoretical results, with test-MSE trends following the same qualitative pattern. Our results show that simplified analytical surrogates can provide useful guidance for neural multi-output regression.

1 Introduction

Multi-output regression predicts a vector of continuous target variables from a common input. A basic modeling choice is whether to fit one joint vector-valued predictor or separate univariate predictors for each target coordinate. This setting appears in neural imitation learning, robotic control, autonomous driving, and continuous-control reinforcement learning, where outputs include joint torques, motor commands, steering and acceleration, or continuous action vectors [Levine et al., 2016, Lillicrap et al., 2016, Codevilla et al., 2018]. It also appears beyond robotics in vegetation-condition modeling, remote-sensing biophysical estimation, industrial process monitoring, and multi-response genomics [Kocev et al., 2009, Tuia et al., 2011, Han et al., 2012, Kim and Xing, 2012]. Such outputs can have different physical units, variances, and covariance structure.

This holistic-versus-separate choice is classical in the multi-output regression literature. Survey work distinguishes problem-transformation methods, which reduce the problem to independent single-target regressions, from algorithm-adaptation methods, which directly handle multi-output data and

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

can model target dependencies [Borchani et al., 2015]. Recent work on tree-based ensembles asks this question directly: when should a holistic multivariate approach be preferred over separate univariate predictions [Schmid et al., 2023]? These works establish the modeling choice as important, but study it mainly through non-neural algorithmic and empirical comparisons.

Neural networks make the joint-output version architecturally natural: a shared feature extractor feeds a vector-valued final layer. However, this architectural convenience does not by itself explain when joint fitting should reduce training error under regularization, nor how the geometry and scale of the target vectors affect the result. We use the term multi-output regression rather than multi-task learning because all target coordinates are observed for the same examples and share the same input features; our comparison is between a joint vector-valued predictor and independent coordinate-wise predictors.

Target preprocessing raises a second, closely related question. Multi-output losses aggregate errors across coordinates, so target scale and covariance affect both training and evaluation. Classical multi-output work often standardizes or normalizes outputs when their scales differ [Borchani et al., 2015, Schmid et al., 2023]. However, if predictions are ultimately evaluated in the original target units, it is not obvious whether training on whitened or normalized targets improves original-scale MSE after the inverse transformation.

Because neural networks are highly non-linear and high-dimensional, exact analysis is notoriously difficult. We study the two questions using the Unconstrained Feature Model (UFM) and the closely related Layer-Peeled Model as tractable surrogates for exact analysis[Fang et al., 2021, Mixon et al., 2022]. The UFM deliberately removes the parameterization of the feature extractor and treats training features as free variables, replacing parameter regularization with feature regularization. Our goal is not to claim that UFM gives exact quantitative predictions for finite neural networks. Instead, we ask whether its closed-form solution gives useful qualitative predictions for design choices in neural multi-output regression.

Our contributions are threefold:

- We recast the joint-versus-separate modeling choice from classical multi-output regression in the neural setting and analyze it through UFM.
- We show that target whitening and normalization have a conditional effect on original-scale training MSE, governed by the target covariance spectrum.
- We empirically test these UFM-derived predictions in standard finite neural networks across robotic imitation-learning and autonomous-driving datasets, observing qualitatively consistent training-MSE and test-MSE patterns.

Our theoretical results concern the UFM training MSE. We do not claim a generalization result. The test-MSE results reported in Section 6 are additional empirical observations and are not covered by our theorems.

2 Related work

2.1 Multi-output regression

Multi-output regression, also called multi-target, multivariate, or multi-response regression, studies the simultaneous prediction of several real-valued targets from common inputs [Borchani et al., 2015]. A standard taxonomy separates problem-transformation methods, which train independent single-target models, from algorithm-adaptation methods, which directly learn multi-output predictors and can exploit target dependencies. This taxonomy motivates the comparison studied here: independent coordinate-wise neural regressors correspond to the separate single-target strategy, while a joint vector-valued network corresponds to a holistic multi-output predictor.

Schmid et al. [2023] study this holistic-versus-separate choice for tree-based ensembles, comparing univariate Random Forests and Extra Trees with multivariate variants. Their motivating question is whether a holistic multivariate regression approach should be preferred over separate univariate predictions. Our work studies the analogous modeling choice for neural vector-valued regression, but with a different contribution: rather than an algorithm-specific empirical comparison, we give a

spectral training-MSE analysis through UFM and then test the resulting qualitative predictions in finite networks.

The joint-versus-separate choice can refer to two different questions. The first is whether a shared encoder learns better input-conditional features than separate encoders. The second is how joint regularized fitting interacts with the geometry of the target vectors. This paper addresses the second question. The UFM is suited to it because it retains the target covariance while removing the input-to-feature mapping.

2.2 Relation to multi-task learning

Multi-task learning is related but broader: tasks may involve different datasets, input features, losses, or prediction spaces [Ruder, 2017, Vandenhende et al., 2021]. In contrast, our setting follows multi-output regression, where all target coordinates are observed for the same examples and share the same input features. Accordingly, our claims are restricted to joint versus coordinate-wise prediction for vector-valued regression, not general multi-task learning. In multi-task learning, sharing a representation can also hurt some tasks, which is known as negative transfer [Standley et al., 2020, Wu et al., 2020]. Our setting excludes several of its sources, since all target coordinates share the same inputs, the same loss, and complete target vectors. Interference between output coordinates remains possible in principle.

2.3 Target scaling, whitening, and normalization

Target scaling is a recurring issue in multi-output regression because aggregate losses combine errors from different coordinates. Borchani et al. [2015] note that when target variables have distinct ranges, normalization may be useful before averaging errors, and for MSE/RMSE scaling by target standard deviation is natural. Schmid et al. [2023] also standardize outputs in their real-data analysis and note that some multi-output extensions require outputs to be measured on the same scale. This distinction matters because target scaling may improve the transformed training objective while worsening MSE after mapping predictions back to the original units. Whitening and decorrelation methods are widely used for inputs and hidden representations [LeCun et al., 1998, Ioffe and Szegedy, 2015, Huang et al., 2018, 2019, Kessy et al., 2018]. Our focus is different: we ask when target whitening or normalization improves original-scale MSE after the inverse transformation.

2.4 Unconstrained feature and layer-peeled models

The Unconstrained Feature Model (UFM) [Mixon et al., 2022] and the Layer-Peeled Model [Fang et al., 2021] replace the parameterized feature extractor by free feature variables, making otherwise difficult neural objectives analytically tractable. UFM has been used to study neural collapse [Papyan et al., 2020, Zhu et al., 2021, Weinan and Wojtowytsch, 2022], class imbalance [Hong and Ling, 2024, Thrampoulidis et al., 2022, Yang et al., 2022], normalized features [Yaras et al., 2022], MSE and other losses [Han et al., 2021, Zhou et al., 2022, Guo et al., 2024], multi-label classification [Li et al., 2023], many-class regimes [Jiang et al., 2023], and multivariate regression [Andriopoulos et al., 2024]. Andriopoulos et al. [2024] derive UFM structure for neural multivariate regression. We use these closed-form tools to analyze two modeling choices in neural multi-output regression: joint versus coordinate-wise fitting and target preprocessing. We use UFM as a tractable surrogate for deriving qualitative theoretical predictions in neural multi-output regression; we do not treat it as an exact quantitative model of finite-network training or generalization.

3 UFM setup and closed-form training MSE

We consider the multi-output 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

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

where $\lambda_{\mathbf{H}}$ and $\lambda_{\mathbf{W}}$ are strictly positive regularization parameters.[1] 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 equivalent, feature regularization provides a tractable surrogate that penalizes representation magnitude directly. This surrogate is the step that enables the UFM reduction below.

Our analysis uses the UFM-loss function [Fang et al., 2021, Mixon et al., 2022]

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

$\mathbf{H} := [\mathbf{h}_1 \cdots \mathbf{h}_M] \in \mathbb{R}^{d\times M}$, $\mathbf{Y} := [\mathbf{y}_1 \cdots \mathbf{y}_M] \in \mathbb{R}^{n\times M}$. The UFM-loss function depends on $\mathbf{H}$, $\mathbf{W}$ and $\mathbf{b}$ and not 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 two minimal values are equal if the feature extractor is expressive enough to attain any feature vector configuration on the training set. We use this model to obtain closed-form training-MSE predictions, then ask empirically whether the same behavior appears under standard parameter weight decay.

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:

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

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 $[\mathbf{C}]_j := [\mathbf{c}_1\ \mathbf{c}_2 \cdots \mathbf{c}_j\ \mathbf{0} \cdots \mathbf{0}]$. In the UFM analysis, $c = \lambda_{\mathbf{H}}\lambda_{\mathbf{W}}$ controls the amount of shrinkage. In the finite-network experiments, we use standard parameter weight decay; the comparison to UFM is therefore qualitative rather than an identification of the two objectives.

Our main analytical results will be based on the closed-form solutions given below. Theorem 3.1 builds on Lemma B.1 of [Zhou et al., 2022] (see Appendix C.1).

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

$$\text{MSE}(\mathbf{H}^*, \mathbf{W}^*, \mathbf{b}^*) = j^*c + \sum_{i=j^*+1}^{n} \lambda_i. \tag{5}$$

If $n = 1$, we have that

$$\text{MSE}(\mathbf{H}^*, \mathbf{w}^*, b^*) = \begin{cases} c, & \text{if } c \leq \sigma^2, \\ \sigma^2, & \text{if } c > \sigma^2. \end{cases} \tag{6}$$

[1] If exactly one of $\lambda_{\mathbf{H}}$ and $\lambda_{\mathbf{W}}$ is zero, a minimizer need not exist. For $\lambda_{\mathbf{H}} = 0$ and $\lambda_{\mathbf{W}} > 0$, take any exact fit $(\mathbf{W}_0, \mathbf{H}_0, \mathbf{b})$, which exists when $d \geq n$, with $\mathbf{W}_0\mathbf{H}_0 + \mathbf{b}\mathbf{1}_M^T = \mathbf{Y}$ and consider $(\varepsilon\mathbf{W}_0, \mathbf{H}_0/\varepsilon, \mathbf{b})$. The fit term is unchanged, and the only active penalty, $\frac{\lambda_{\mathbf{W}}}{2}||\varepsilon\mathbf{W}_0||_F^2$, tends to zero as $\varepsilon \to 0$. The infimum is therefore zero, but it is not attained, since $\mathbf{W} = \mathbf{0}$ cannot fit non-constant targets. The case $\lambda_{\mathbf{W}} = 0$ and $\lambda_{\mathbf{H}} > 0$ is symmetric.

4 Joint-output versus coordinate-wise neural regression

We now compare a joint vector-valued neural regressor with n independent coordinate-wise regressors. The joint model uses shared features and an n-dimensional final layer. The coordinate-wise baseline trains n separate univariate models, one per target coordinate. This comparison is the neural analogue of the holistic-versus-separate distinction in classical multi-output regression.

To preserve the notation of the UFM derivation, we write MSE(multi, c) for the joint-output UFM and MSE(n-single, $\tilde{c}$) for the sum of n coordinate-wise UFMs; here "single" refers only to a single output coordinate.

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-coordinate 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 coordinate-wise model, and let $\text{MSE}^{(i)}(\mathbf{H}^{(i)}, \mathbf{w}^{(i)}, b^{(i)})$ denote its corresponding MSE. The total MSE across the n coordinate-wise models is then given by

$$\text{MSE(n-single, } \tilde{c}) := \sum_{i=1}^{n} \text{MSE}^{(i)}(\mathbf{H}^{(i)}, \mathbf{w}^{(i)}, b^{(i)}). \tag{7}$$

Let σ_i^2 denote the variance of the targets for the i-th single-coordinate 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 coordinate-wise UFM problems is given by*

$$\text{MSE}(\textit{n-single}, \tilde{c}) = k^*\tilde{c} + \sum_{i=k^*+1}^{n} \sigma_i^2. \tag{8}$$

We now present the main result of this section.

Theorem 4.2. *Let MSE(multi, c) be a shorthand of (5).*

(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 gives a training-MSE comparison under UFM: with matched regularization, the joint-output predictor has no larger training MSE than the collection of coordinate-wise predictors. The mechanism is spectral. The joint-output UFM shrinks along eigen-directions of the target covariance, whereas coordinate-wise UFMs shrink marginal variances independently. Thus the result is not simply a statement that highly correlated outputs help; it follows from how the covariance spectrum compares with the diagonal variances under regularization.

The theorem does not follow from a feasibility argument alone. Placing the n coordinate-wise solutions in disjoint feature blocks, with the matching blocks of $\mathbf{W}$ and zeros elsewhere, gives a feasible point of the joint problem when the joint feature dimension is at least the sum of the coordinate-wise ones. Its objective equals the sum of the n coordinate-wise objectives, since the fit term, $||\mathbf{H}||_F^2$, and $||\mathbf{W}||_F^2$ all decompose over blocks. With $\tilde{\lambda}_{\mathbf{H}} = \lambda_{\mathbf{H}}$ and $\tilde{\lambda}_{\mathbf{W}} = \lambda_{\mathbf{W}}$, the minimal joint objective is therefore no larger than the sum of the minimal coordinate-wise objectives. Theorem 4.2, however, compares the MSE terms at the respective minimizers, not the objectives. A smaller objective is compatible with a larger fit error offset by smaller regularization terms, so the inequality between objectives does not order the MSE terms. The comparison instead rests on the relation between the eigenvalues of $\mathbf{\Sigma}$ and its diagonal entries. By Theorem 3.1 and Corollary 4.1, the

two quantities are $\sum_{i=1}^{n} \min\{c, \lambda_i\}$ and $\sum_{i=1}^{n} \min\{\tilde{c}, \sigma_i^2\}$, and the proof in Appendix C.2 obtains the inequality from the Schur-Horn relations. The theorem also identifies when the advantage is strict and when the two designs coincide, which the feasibility argument does not.

5 Target preprocessing in neural multi-output regression

Target preprocessing changes the geometry of multi-output regression because the loss aggregates errors across output coordinates. We analyze ZCA whitening and coordinate-wise normalization. The question is not whether transformed targets have unit covariance or unit marginal variances; it is whether predictions trained in the transformed coordinates have lower MSE after being mapped back to the original target units.

For whitening, we transform the targets using

$$\mathbf{Y}^{ZCA} = \mathbf{\Sigma}^{-1/2}(\mathbf{Y} - \bar{\mathbf{Y}}). \tag{9}$$

The UFM residual structure gives one motivation for this choice: by prior UFM analysis [Andriopoulos et al., 2024, Corollary 4.2(v)], the residual error is proportional to the ZCA-whitened targets,

$$\mathbf{E} := \mathbf{WH} + \mathbf{b}\mathbf{1}_M^T - \mathbf{Y} = -\sqrt{c}\mathbf{Y}^{ZCA}. \tag{10}$$

For the UFM analysis, we solve the UFM objective on $\mathbf{Y}^{ZCA}$, obtain predictions $\tilde{\mathbf{Y}} := \tilde{\mathbf{W}}\tilde{\mathbf{H}}$ in whitened coordinates, de-whiten them as $\hat{\mathbf{Y}} := [\mathbf{\Sigma}^{1/2}]\tilde{\mathbf{Y}} + \bar{\mathbf{Y}}$, and evaluate original-scale MSE:

$$\text{MSE(de-whiten)} := M^{-1}||\hat{\mathbf{Y}} - \mathbf{Y}||_F^2. \tag{11}$$

The next theorem gives the resulting closed form.

Theorem 5.1.

$$\text{MSE(de-whiten)} = \min\{c, 1\} \sum_{i=1}^{n} \lambda_i. \tag{12}$$

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*

$$\sum_{i=1}^{j^*} \lambda_i - j^* < c^{-1}(1-c) \sum_{i=j^*+1}^{n} \lambda_i, \tag{13}$$

we have that

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

When the converse inequality of (13) holds, then

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

When the inequality of (13) 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 (13) reduces to* $\bar{\lambda} := \sum_{i=1}^{n} \lambda_i/n < 1$.

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

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

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

Normalization sets variances to 1 but leaves correlations intact:

$$\mathbf{Y}^{nrm} = \mathbf{V}^{-1/2}(\mathbf{Y} - \bar{\mathbf{Y}}), \tag{14}$$

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. 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.

$$\text{MSE(de-normalize)} = \begin{cases} c\sum_{i=1}^{n}\lambda_i, & \text{if } 0 < c < \tilde{\lambda}_{\min}, \\ \sum_{i=1}^{n}\lambda_i, & \text{if } c > \tilde{\lambda}_{\max}. \end{cases} \tag{15}$$

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,*

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

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,*

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

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

Together, Theorems 5.2 and 5.4 show that target preprocessing is conditional. For whitening, the special case $c < \lambda_{\min}$ reduces the comparison to the average target variance $\bar{\lambda}$. For normalization, the analogous comparison holds when $0 < c < \min\{\lambda_{\min}, \tilde{\lambda}_{\min}\}$. In these regimes, preprocessing improves original-scale training MSE when $\bar{\lambda} < 1$ and worsens it when $\bar{\lambda} > 1$. The result therefore does not recommend universal preprocessing; it gives a spectral criterion for when preprocessing should help or hurt under UFM. The threshold is relative to the target units used for evaluation; rescaling the raw targets changes $\bar{\lambda}$, which is precisely why target preprocessing is not a cosmetic choice.

6 Experiments

The theory concerns UFM training MSE, while the experiments use finite neural networks trained with standard parameter weight decay. We therefore evaluate whether the qualitative predictions, rather than the exact loss values, appear in practice. For each dataset and weight-decay value, we report training MSE as the empirical counterpart of the UFM training-MSE results, and test MSE as supporting evidence about whether the same pattern appears beyond the training set.

The UFM product c and the weight-decay parameter λ_{WD} are related but not identical, and our conclusions do not rely on an exact mapping between them. The relation has four parts. First, weight decay is applied to all parameters, including the final layer, so the term $\frac{\lambda_{\mathbf{W}}}{2}||\mathbf{W}||_F^2$ appears identically in (1) and in the UFM loss (3). Second, the feature penalty in (2) is a surrogate for weight decay on θ: features are produced by weight-bounded layers, so penalizing $||\theta||_2^2$ also controls the magnitude of $\mathbf{h}_\theta(\mathbf{x})$, which is the quantity that $\lambda_{\mathbf{H}}$ penalizes directly. Third, every prediction we test is an ordering that holds uniformly over a range of c, not a value at a particular c. It is therefore sufficient that the effective shrinkage increases with λ_{WD}, and the value of c need not be identified. This is why we sweep λ_{WD} over four orders of magnitude. Fourth, only the monotonicity of this correspondence is required, not its form. Weight decay is applied identically to the two designs being compared, and to the raw and preprocessed runs, so any common monotone reparametrization of the horizontal axis leaves the tested orderings unchanged.

6.1 Datasets and spectral regimes

We test the UFM-derived predictions on three MuJoCo imitation-learning datasets and one autonomous-driving dataset. Reacher, Swimmer, and Hopper use raw robotic states as inputs and expert actions as targets, derived from MuJoCo environments [Brockman et al., 2016, Todorov et al., 2012, Towers et al., 2024, Gallouédec et al., 2024, Fu et al., 2020]. CARLA 2D uses camera images from the CARLA simulator [Codevilla et al., 2018] and predicts speed and steering angle.

For the MuJoCo datasets we use four-layer MLPs with three hidden layers of width 256. For CARLA 2D we use ResNet18 [He et al., 2016]. Full hyperparameters are given in Appendix A.

Table 1 is central to the target-preprocessing prediction. Reacher, Swimmer, and Hopper have $\bar{\lambda} < 1$, so Theorems 5.2 and 5.4 predict that whitening or normalization should reduce original-scale MSE in the relevant small-regularization regimes. CARLA 2D has $\bar{\lambda} \gg 1$, so the same theory predicts the opposite sign. Appendix Table 4 shows that pairwise target correlations range from near zero to modest negative values, so the empirical joint-output advantage is not restricted to settings with visibly high linear target correlation.

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

All experimental results are averaged over random seeds, with variance across seeds represented by error bars. Although practical weight decay is often small, we sweep values from 10^{-5} to 10^{-1} to test whether the same qualitative behavior appears across a broad range of regularization strengths in finite networks.

6.2 Joint-output versus coordinate-wise prediction

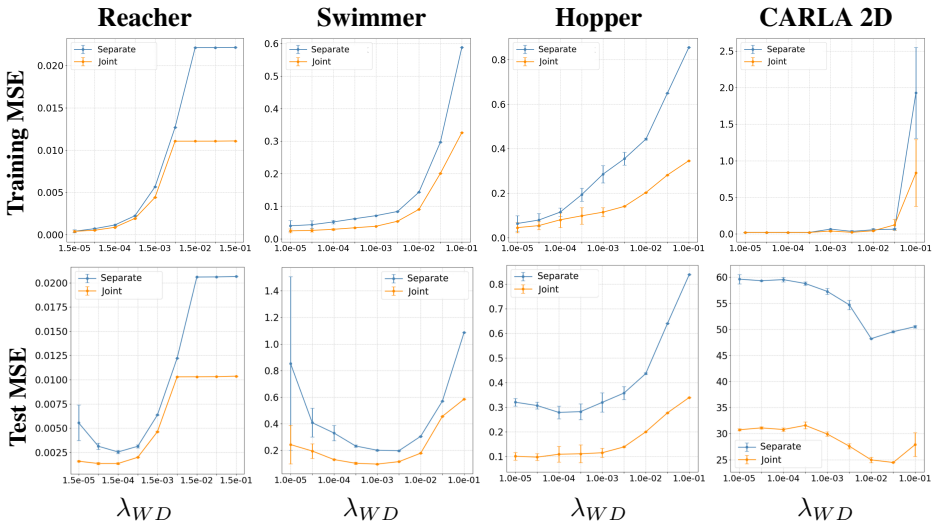


Figure 1: Training and test MSE for *Joint*, a single vector-valued model with shared features and an n-dimensional output layer, and *Separate*, n independently trained coordinate-wise models. The UFM result predicts that the joint-output training MSE is no larger under matched regularization; test MSE is reported only as empirical supporting evidence.

Figure 1 compares the joint-output model with independent coordinate-wise models. Across all four datasets and all displayed weight-decay values, the joint-output model achieves lower training MSE, consistent with Theorem 4.2. The test set shows the same qualitative pattern, and Appendix B shows that the training-MSE pattern is stable across a broader range of MLP architectures.

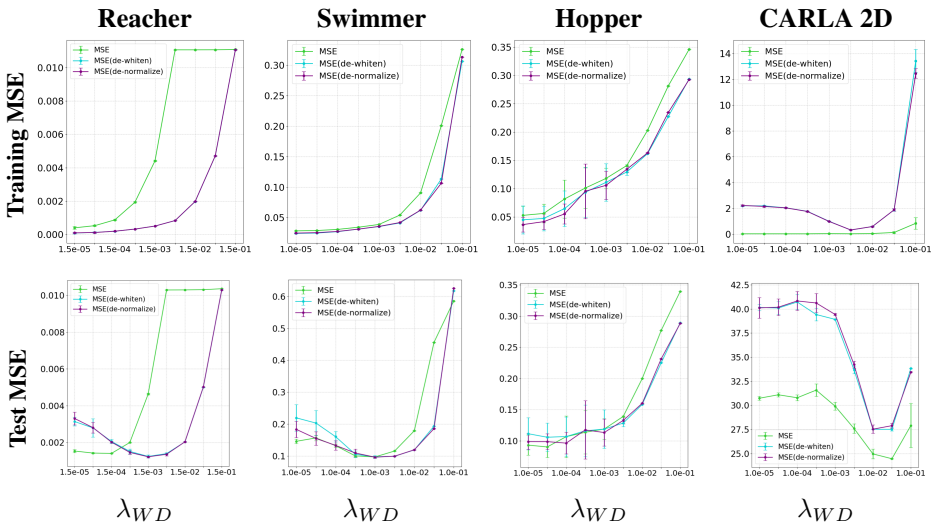


Figure 2: Original-scale training and test MSE after training with raw, whitened, or normalized targets. In the displayed architecture, whitening and normalization reduce MSE for the MuJoCo datasets with $\bar{\lambda} < 1$, and increase MSE for CARLA 2D where $\bar{\lambda} \gg 1$, matching the predicted sign pattern.

6.3 Target whitening and normalization

For target preprocessing, we train with raw targets, whitened targets, or normalized targets, applying the inverse transformation before computing MSE in the original target units. Figure 2 shows the sign pattern predicted by the UFM analysis: preprocessing reduces original-scale training MSE in Reacher, Swimmer, and Hopper, but increases it in CARLA 2D. The same qualitative trend appears for test MSE, which is supporting evidence rather than a theoretical claim. Appendix B reports architecture sweeps showing the training-MSE pattern is stable.

7 Discussion

Both analyses show that design choices in neural multi-output regression can be organized around the spectrum of the target covariance. For joint-output prediction, the UFM objective acts along covariance eigen-directions rather than along marginal output coordinates. This is why the comparison between joint and coordinate-wise fitting is not determined only by whether two outputs are pairwise correlated. The relevant object is the full covariance geometry and how it interacts with regularization. Coordinate-wise fitting treats the output coordinates independently, so its behavior depends on the diagonal variances. Joint fitting, in contrast, exposes the regularizer to the eigenstructure of the full target covariance. The resulting comparison is therefore about how output geometry is used by the predictor, not merely about model size or capacity.

This perspective also clarifies why target preprocessing has a nontrivial effect on original-scale MSE. Whitening and standardization are often treated as default preprocessing choices because they make the transformed targets numerically balanced during optimization. However, the evaluation metric is usually computed after mapping predictions back to the original target units. In that metric, preprocessing changes how error is distributed across output directions. The UFM analysis makes this dependence explicit. In the small-regularization regimes of Theorems 5.2 and 5.4, the average covariance scale $\bar{\lambda} = \mathrm{Tr}(\Sigma)/n$ gives a simple first diagnostic: preprocessing is predicted to help when the original target covariance scale is small and to hurt when it is large. Outside these simplified regimes, the full spectral conditions provide the more precise criterion.

The UFM solutions show how this redistribution happens, in each eigen-direction of $\mathbf{\Sigma}$. With raw targets, Theorem 3.1 gives a training MSE of $\sum_{i=1}^{n} \min\{c, \lambda_i\}$, where the i-th term is the error in the i-th eigen-direction. Directions with $\lambda_i \geq c$ are fitted with a fixed absolute error c. Directions with $\lambda_i < c$ are not fitted at all: the model predicts their mean and incurs the full variance λ_i.

With whitened targets and $c < 1$, the proof of Theorem 5.1 shows that the de-whitened residual is exactly $-\sqrt{c}(\mathbf{Y} - \bar{\mathbf{Y}})$, so the error in eigen-direction i is $c\lambda_i$. Every direction is then fitted, and the error is proportional to the variance of the direction rather than uniform across directions. Whitening therefore moves error from low-variance directions to high-variance directions. This explains the two cases in our experiments. On CARLA 2D, $\lambda_{\max} = 209.097$ (Table 1), so whitening raises the error in the dominant eigen-direction from c to $209.097\,c$, while the small direction, with $\lambda_{\min} = 0.024$, improves only slightly. The original-scale MSE therefore becomes worse, consistent with $\bar{\lambda} = 104.561 > 1$. On Reacher all eigenvalues lie between 0.010 and 0.012. Once c exceeds $\lambda_{\max} = 0.012$, training on raw targets predicts the mean in every direction and incurs the full $\sum_i \lambda_i$, while whitening continues to fit every direction for any $c < 1$ and incurs only $c\sum_i \lambda_i$.

The MuJoCo and CARLA 2D experiments are designed to test both signs of this prediction. In MuJoCo, the target covariance scale lies in the regime where whitening or normalization is predicted to improve original-scale MSE. In CARLA 2D, the covariance scale is much larger, and the theory predicts that preprocessing can worsen original-scale MSE. The experiments support this qualitative distinction in finite neural networks trained with standard weight decay. This is important because it shows that the UFM calculations are not merely algebraic properties of the surrogate model: they identify regimes in which familiar empirical choices have different consequences.

A useful consequence of the theory is that it suggests what should be reported in multi-output regression experiments. Pairwise target correlations are helpful, but they do not fully determine the behavior of regularized multi-output prediction. Two datasets can have similar marginal correlations but different covariance spectra, and those spectra can lead to different predictions under UFM. Likewise, transformed-coordinate losses can obscure whether preprocessing improves the metric that is actually used for evaluation. Reporting the covariance spectrum, the marginal variances, and the original-scale MSE makes the comparison between joint fitting, coordinate-wise fitting, and target preprocessing more interpretable.

8 Limitations

Our theory concerns training MSE under UFM, not test MSE. The closed-form results characterize the surrogate optimization problem, rather than providing a generalization theorem for finite neural networks. The experiments therefore test whether the qualitative predictions of the UFM training-MSE analysis also appear in held-out test MSE; across the settings studied here, the same directional effects are observed.

A second limitation is that UFM regularization is not in one-to-one correspondence with standard neural-network weight decay. In UFM, features are optimized directly and regularized explicitly, yielding the product $c = \lambda_H \lambda_W$; in finite networks, features are produced by parameterized layers and the experiments use parameter weight decay. Thus, we do not estimate c or predict exact finite-network losses. The intended comparison is qualitative: UFM identifies spectral regimes in which joint fitting or preprocessing should help or hurt, and the experiments test whether those regimes are reflected in finite-network behavior.

9 Conclusion

We studied joint-output versus coordinate-wise fitting and target whitening or normalization in neural multi-output regression. Under UFM, both questions reduce to spectral properties of the target covariance and regularization product: joint-output regression has no larger UFM training MSE under matched regularization, while preprocessing can improve or worsen original-scale MSE depending on covariance scale.

Practically, these choices can be diagnosed from the target covariance. Joint versus coordinate-wise fitting depends on covariance eigenvalues and marginal variances. For whitening and normalization, $\bar{\lambda} = \mathrm{Tr}(\Sigma)/n$ gives a small-regularization rule, while the full spectral conditions give the general comparison. Finite-network experiments support these predictions, suggesting UFM can guide neural multi-output regression design.

Acknowledgments and Disclosure of Funding

This work was supported by Tamkeen under Research Institute Award CG010 to the NYU Abu Dhabi Center for Artificial Intelligence and Robotics, and by the NYU Abu Dhabi Center of Interdisciplinary Data Science and Artificial Intelligence. Some of the experiments were run on the High Performance Computing resources of New York University Abu Dhabi.

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.

Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. A survey on multi-output regression. *WIREs Data Mining and Knowledge Discovery*, 5(5):216–233, 2015. doi: 10.1002/widm.1157.

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)*, pages 4693–4700. IEEE, 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.

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.

Zhongyang Han, Ying Liu, Jun Zhao, and Wei Wang. Real time prediction for converter gas tank levels based on multi-output least square support vector regressor. *Control Engineering Practice*, 20(12):1400–1409, 2012. doi: 10.1016/j.conengprac.2012.08.006.

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

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

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 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*, pages 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*, pages 448–456. PMLR, 2015.

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.

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

Seyoung Kim and Eric P. Xing. Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eqtl mapping. *The Annals of Applied Statistics*, 6(3):1095–1117, 2012. doi: 10.1214/12-AOAS549.

Dragi Kocev, Sašo Džeroski, Matt D. White, Graeme R. Newell, and Peter Griffioen. Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecological Modelling*, 220(8):1159–1168, 2009. doi: 10.1016/j.ecolmodel.2009.01.037.

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.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

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.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 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.

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.

Lena Schmid, Alexander Gerharz, Andreas Groll, and Markus Pauly. Tree-based ensembles for multi-output regression: Comparing multivariate approaches with separate univariate ones. *Computational Statistics & Data Analysis*, 179:107628, 2023. doi: 10.1016/j.csda.2022.107628.

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

Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9120–9132. PMLR, 2020.

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

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.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. 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.

Devis Tuia, Jochem Verrelst, Luis Alonso, Fernando Pérez-Cruz, and Gustavo Camps-Valls. Multioutput support vector regression for remote sensing biophysical parameter estimation. *IEEE Geoscience and Remote Sensing Letters*, 8(4):804–808, 2011. doi: 10.1109/LGRS.2011.2109934.

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.

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*, pages 270–290. PMLR, 2022.

Sen Wu, Hongyang R. Zhang, and Christopher Ré. Understanding and improving information transfer in multi-task learning. In *International Conference on Learning Representations*, 2020.

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

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.

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

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

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 6. The epoch counts in Tables 2 and 3 were chosen so that training losses had stabilized in our runs.

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 layers ($\mathbf{W}$)	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

We utilize expert data from previous work [Gallouédec et al., 2024, Fu et al., 2020] for the Reacher, Swimmer, and Hopper environments. The robots are controlled by applying torques to their joints, which serve as action targets. To construct the datasets, expert policies were deployed in the environments to generate offline state-action pairs. We use 20 expert demonstrations for Reacher, 1 for Swimmer, and 10 for Hopper, translating to 1,000, 1,000, and 10,000 training samples respectively. For each environment, we construct a held-out test set containing 20% of the training-data size.

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 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.

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], we use the full available training split for optimization and report MSE on a held-out evaluation split. We replace the final classification layer with a fully connected layer that outputs continuous target values. 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 Target correlations

Table 4 reports dataset sizes, input types, target dimensions, and Pearson correlations between target components. These correlations are generally small to modest, so the joint-output advantage observed in Figure 1 is not restricted to highly correlated targets.

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	1K	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 Architecture sweeps for training MSE

We repeated the MuJoCo experiments across multiple architectures, covering a range of widths and depths. Figures 3-5 compare joint-output and coordinate-wise training across architectures. Figures 6-8 compare raw targets, whitening, and normalization across the same architecture sweep. The qualitative patterns remain stable.

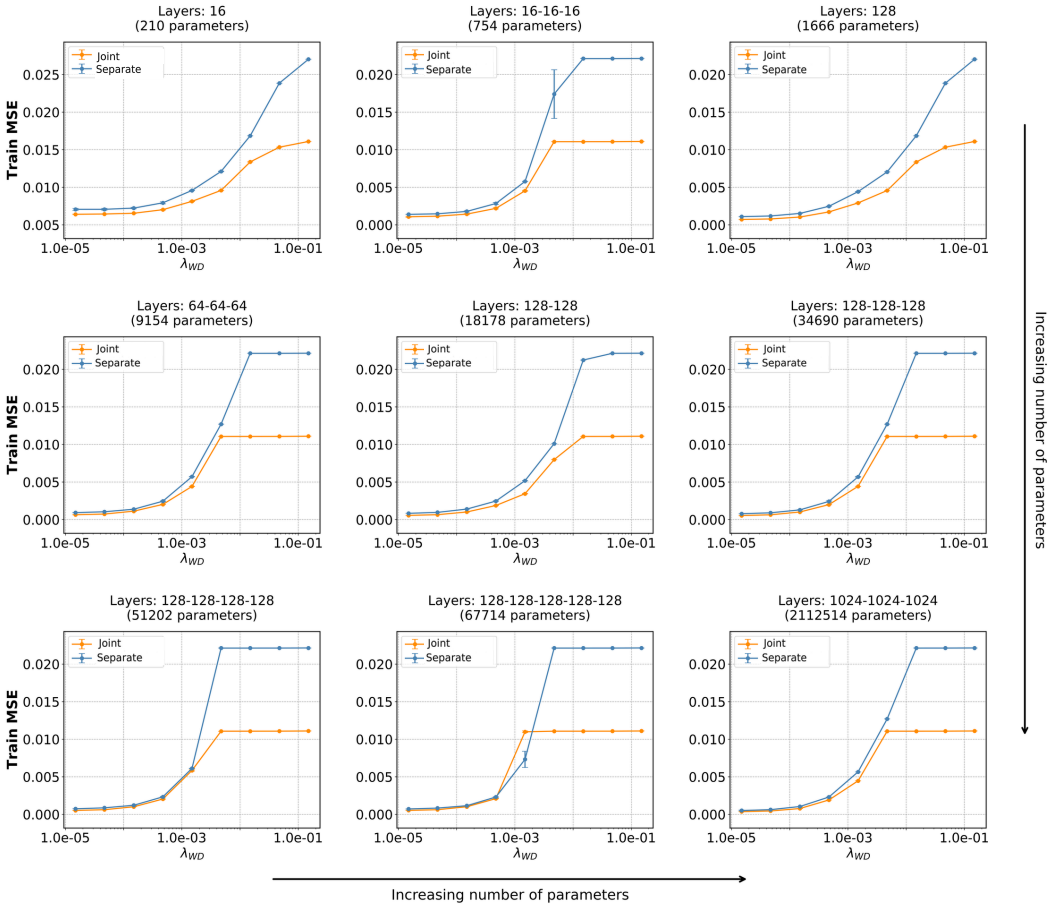


Figure 3: **Effect of network architecture on Joint vs. Separate training: Reacher.** Joint denotes one vector-valued model; Separate denotes independent coordinate-wise models. Training MSE is shown across different architectures and weight decay values.

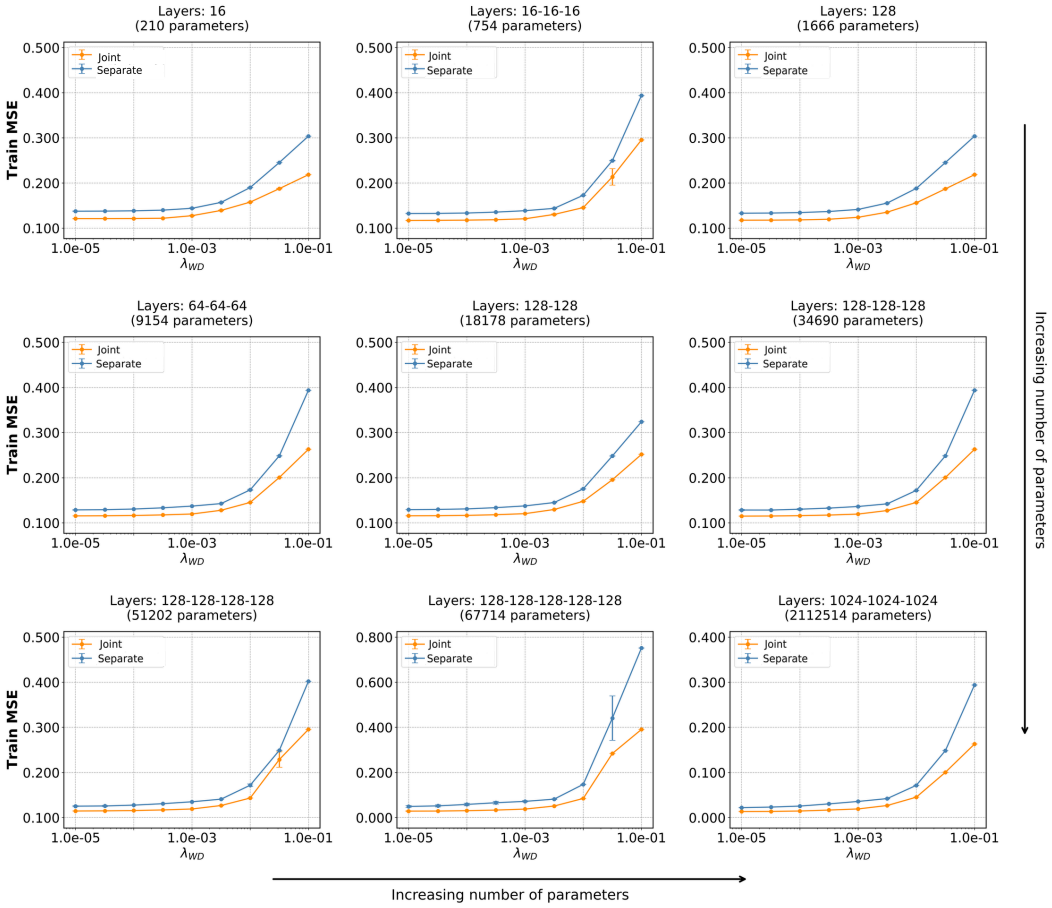


Figure 4: **Effect of network architecture on Joint vs. Separate training: Swimmer.** Joint denotes one vector-valued model; Separate denotes independent coordinate-wise models. Training MSE is shown across different architectures and weight decay values.

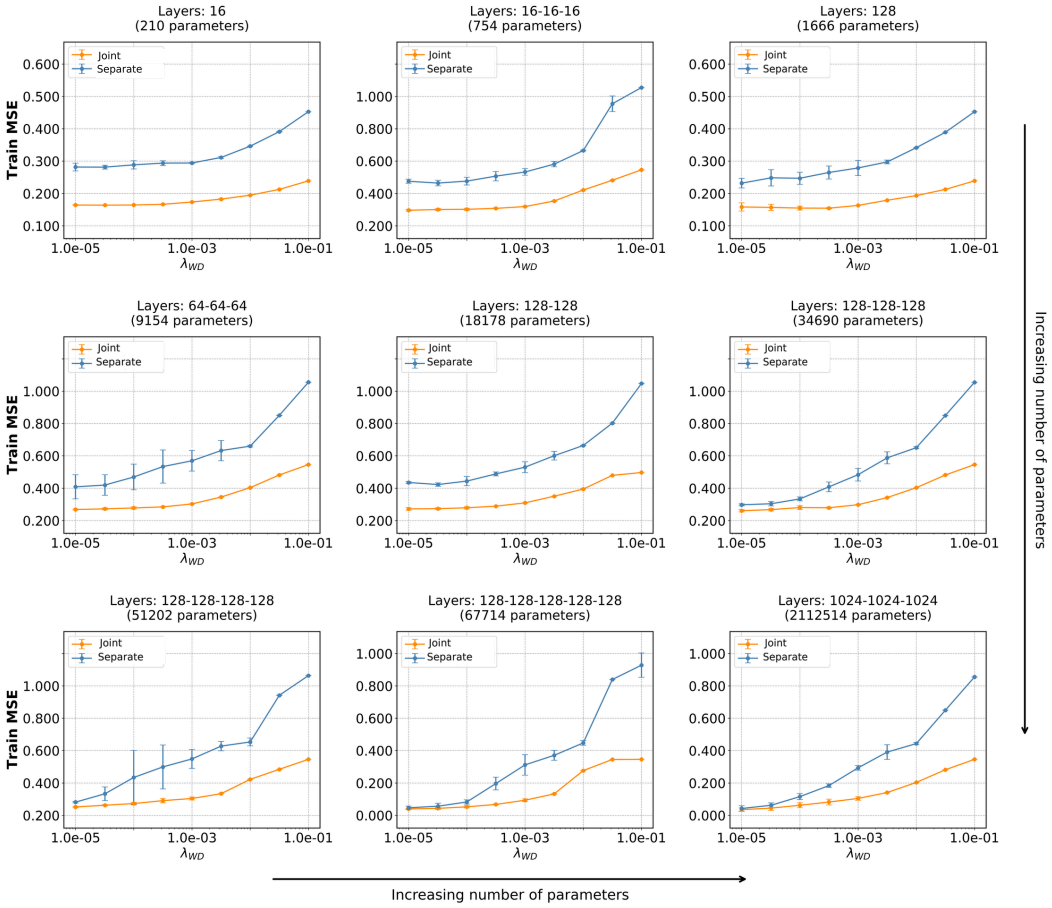


Figure 5: **Effect of network architecture on Joint vs. Separate training: Hopper.** Joint denotes one vector-valued model; Separate denotes independent coordinate-wise models. Training MSE is shown across different architectures and weight decay values.

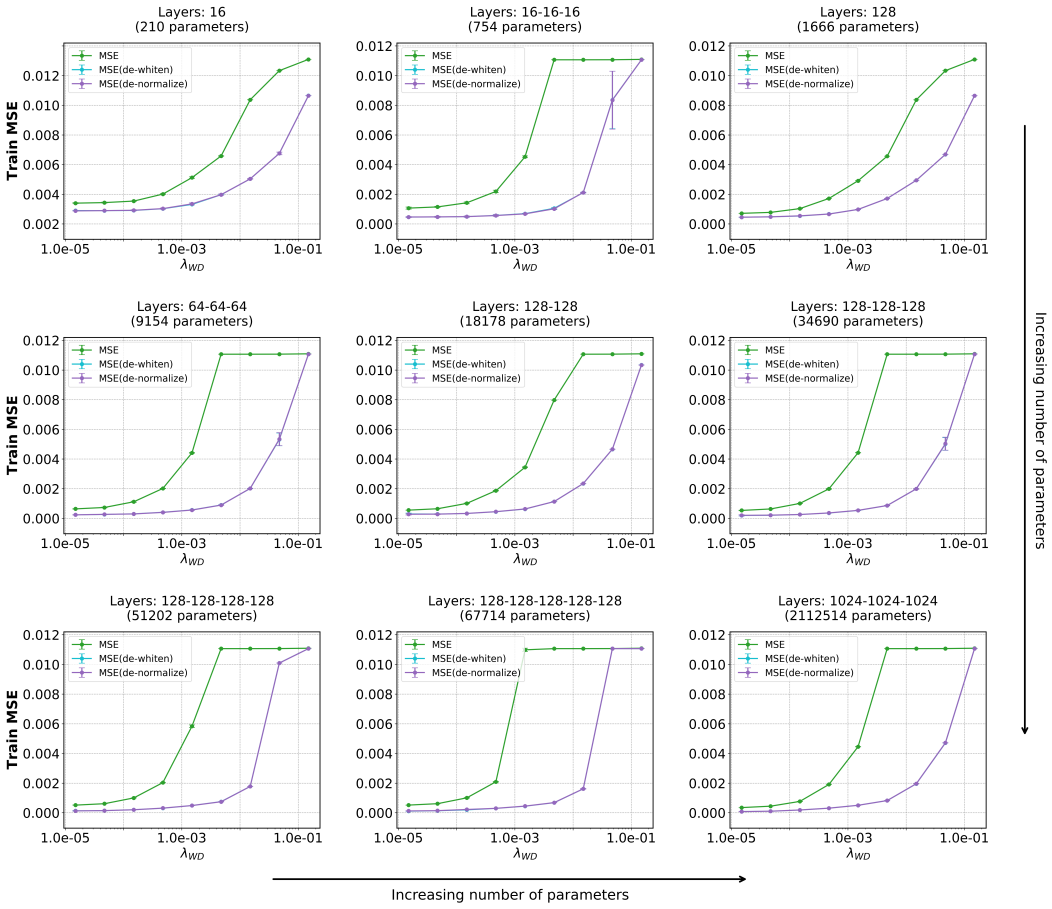


Figure 6: **Whitening vs. normalization vs. raw targets: Reacher.** Training-MSE comparisons across architectures and weight decay values.

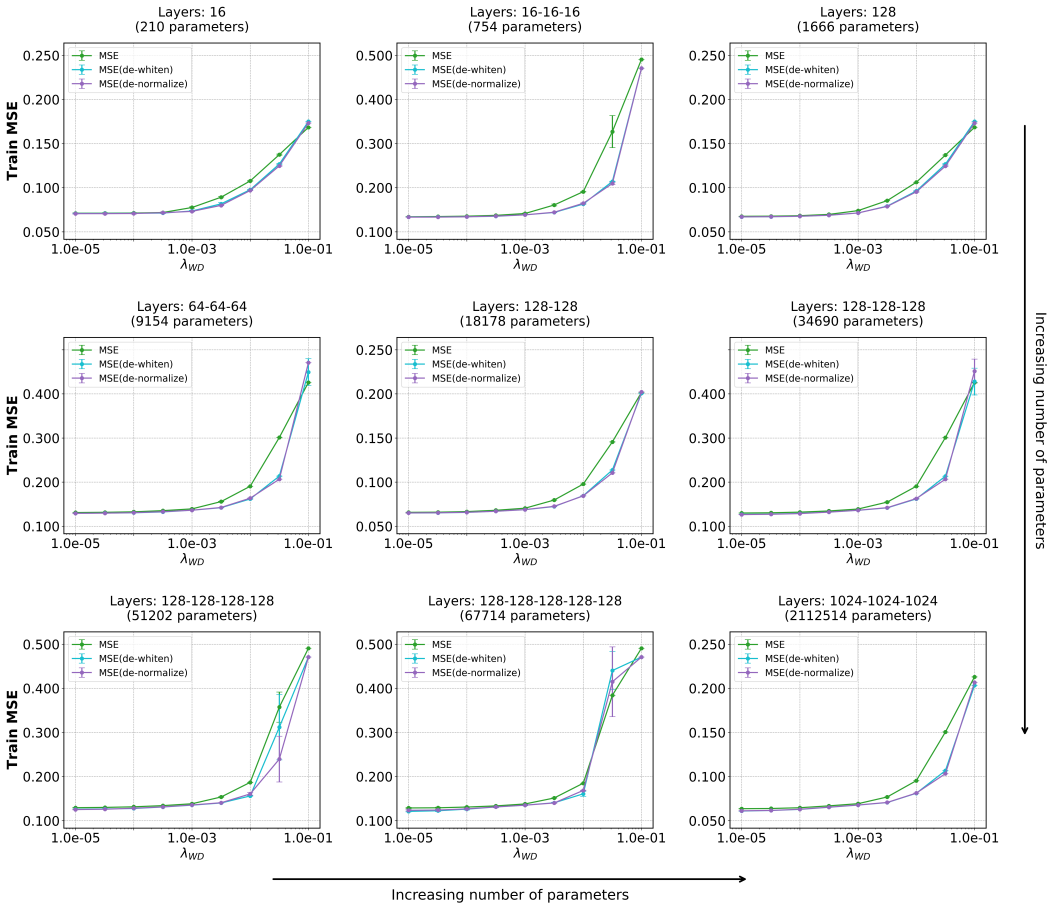


Figure 7: **Whitening vs. normalization vs. raw targets: Swimmer.** Training-MSE comparisons across architectures and weight decay values.

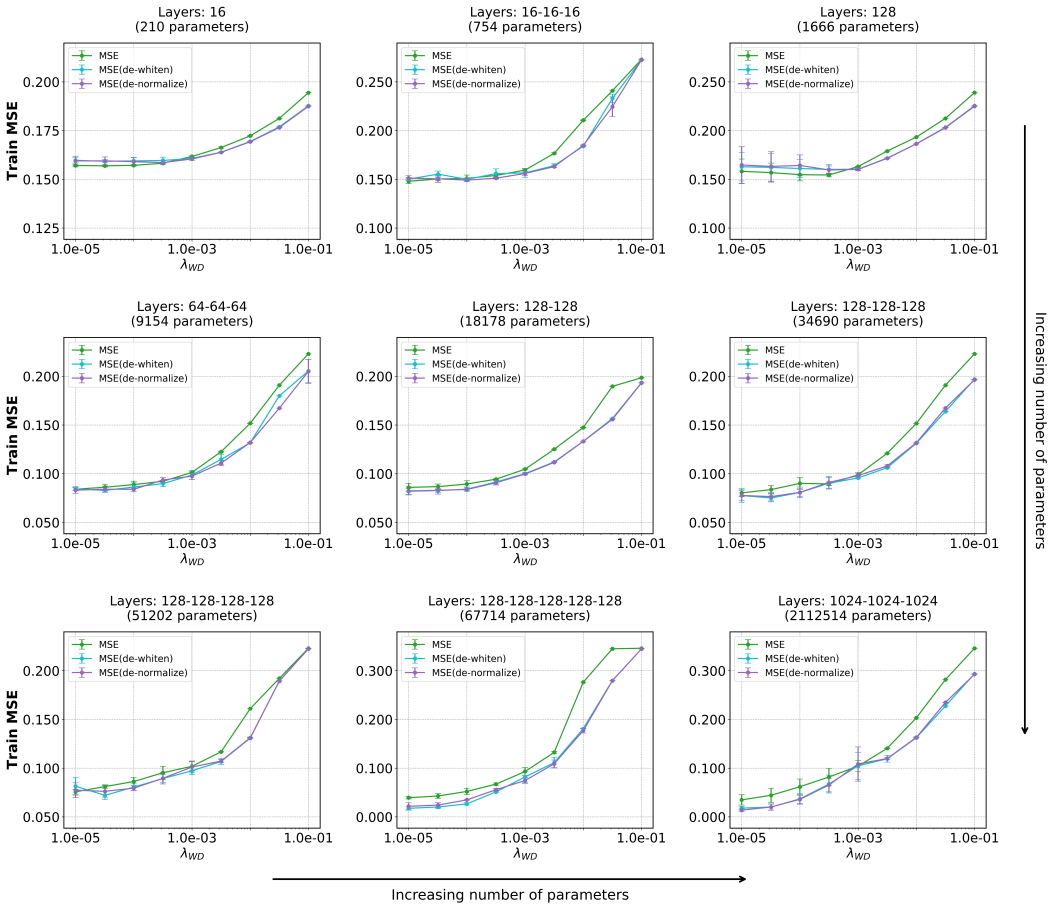


Figure 8: **Whitening vs. normalization vs. raw targets: Hopper.** Training-MSE comparisons across architectures and weight decay values.

B.2 Explicit feature regularization

The experiments in Section 6 use weight decay on all parameters, as in the parameter-regularized loss (1). Penalizing the features directly is closer to the analyzed setting. We therefore also trained the MuJoCo MLPs with the feature-regularized loss (2), which penalizes $\frac{\lambda_{\mathbf{H}}}{2M}\sum_{i=1}^{M}||\mathbf{h}_\theta(\mathbf{x}_i)||_2^2$ together with $\frac{\lambda_{\mathbf{W}}}{2}||\mathbf{W}||_F^2$ on the final layer, in place of weight decay on all parameters. We set $\lambda_{\mathbf{H}} = \lambda_{\mathbf{W}} = \lambda$ and keep the architecture, optimizer, seeds, and regularization grid of the weight-decay runs. Table 5 reports training MSE averaged over three seeds. The λ column gives the grid for Swimmer and Hopper; for Reacher, as in the weight-decay runs, the grid is 1.5 times these values.

The joint model has lower training MSE than the coordinate-wise models in all rows of Table 5 except three: the row 10^{-2} for Reacher, and the row 10^{-1} for Swimmer and Hopper. The qualitative picture is therefore close to that under weight decay.

Table 5: Training MSE under the feature-regularized loss (2) with $\lambda_{\mathbf{H}} = \lambda_{\mathbf{W}} = \lambda$, averaged over three seeds. Joint denotes one vector-valued model; Separate denotes independent coordinate-wise models.

	Reacher		Swimmer		Hopper	
λ	Joint	Separate	Joint	Separate	Joint	Separate
1.0e-05	0.0004	0.0006	0.0275	0.0507	0.0487	0.0641
3.2e-05	0.0004	0.0006	0.0282	0.0498	0.0560	0.0597
1.0e-04	0.0005	0.0007	0.0285	0.0497	0.0527	0.0608
3.2e-04	0.0006	0.0008	0.0296	0.0521	0.0453	0.0912
1.0e-03	0.0007	0.0010	0.0308	0.0540	0.0605	0.1713
3.2e-03	0.0010	0.0012	0.0309	0.0562	0.0585	0.1202
1.0e-02	0.0029	0.0022	0.0325	0.0581	0.0610	0.1296
3.2e-02	0.0111	0.0120	0.0409	0.0633	0.0750	0.1581
1.0e-01	0.0111	0.0221	0.1152	0.1068	0.2458	0.1840

C Proofs

C.1 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., 2022], the associated MSE is

$$\text{MSE}(\mathbf{H}^*, \mathbf{W}^*, \mathbf{b}^*) = \frac{1}{M}\sum_{i=1}^{n}([\eta_i - \sqrt{Mc}]_+ - \eta_i)^2. \tag{16}$$

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

$$\sqrt{M}[\mathbf{\Sigma}^{1/2} - \sqrt{c}\mathbf{I}_n] = U\left[\tilde{\mathbf{\Sigma}} - \sqrt{Mc}\mathbf{I}_n\right]U^{-1}, \tag{17}$$

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 (17) 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 (17) has eigenvalues $\eta_i - \sqrt{Mc}$, $i = 1, ..., n$. Since the eigenvalues in these two sets are both arranged in descending order, we have

$$\sqrt{\lambda_i} = \frac{\eta_i}{\sqrt{M}}, \qquad \text{for all } i = 1, ..., n. \tag{18}$$

Correspondingly, by (16) and (18) 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. □

Remark C.1. The assumption $d \geq n$ is used in (16). Lemma B.1 of Zhou et al. [2022] holds for any d, and its global minimizers are supported on the top $\min(d, n)$ singular directions of $\tilde{\mathbf{Y}}$, with singular values $[\eta_i - \sqrt{Mc}]_+$. More precisely, (5) holds whenever $d \geq j^*$, since the directions with $i > j^*$ are not fitted. Because $j^* = n$ whenever $c \leq \lambda_{\min}$, the assumption $d \geq n$ is needed in general. If $d < j^*$, only the top d directions can be fitted, and the training MSE at a minimizer becomes $dc + \sum_{i=d+1}^{n}\lambda_i$. All experiments in this paper satisfy $d \geq n$, with $d = 256$ for the MLPs and $d = 512$ for ResNet18, against $n \leq 3$.

C.2 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.,

$$\sum_{i=1}^{k} \lambda_i \geq \sum_{i=1}^{k} \sigma_i^2, \qquad \text{for all } k = 1, \ldots, n-1, \tag{19}$$

$$\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \sigma_i^2. \tag{20}$$

From (19) and (20) we have

$$\lambda_1 \geq \sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_n^2 \geq \lambda_n. \tag{21}$$

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 (21), we have that $\tilde{c} < \sigma_n^2$. Thus, by Theorem 3.1 and Corollary 4.1, the two MSEs are given by

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

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

Case II: Suppose $\tilde{c} \geq c > \lambda_1$. By (21), we have that $\tilde{c} > \sigma_1^2$. Thus, by Theorem 3.1 and Corollary 4.1, the two MSEs are given by

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

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

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

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

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

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 (19) 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 consider two subcases.

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

$$\begin{aligned}
&\text{MSE(multi}, c) - \text{MSE(n-single}, \tilde{c}) \\
&= \sum_{i=j^*+1}^{n} \lambda_i - \sum_{i=k^*+1}^{n} \sigma_i^2 - (k^* - j^*)c + k^*(c - \tilde{c}) \\
&= \sum_{i=j^*+1}^{k^*} \lambda_i + \sum_{i=k^*+1}^{n} \lambda_i - \sum_{i=k^*+1}^{n} \sigma_i^2 - (k^* - j^*)c + k^*(c - \tilde{c}) \\
&= \sum_{i=k^*+1}^{n} (\lambda_i - \sigma_i^2) + \sum_{i=j^*+1}^{k^*} (\lambda_i - c) + k^*(c - \tilde{c}) < 0. \qquad (22)
\end{aligned}$$

The first sum in (22) is non-positive by (C.2). The second sum is strictly negative since $\lambda_i < c$, for all $i > j^*$. Therefore, MSE(multi, c) < 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

$$\begin{aligned}
&\text{MSE(multi}, c) - \text{MSE(n-single}, \tilde{c}) \\
&= \sum_{i=k^*+1}^{j^*} (c - \sigma_i^2) + \sum_{i=j^*+1}^{n} (\lambda_i - \sigma_i^2) + k^*(c - \tilde{c}) \\
&= \sum_{i=k^*+1}^{j^*} (c - \sigma_i^2) + \sum_{i=1}^{j^*} (\sigma_i^2 - \lambda_i) + k^*(c - \tilde{c}) \\
&= \sum_{i=k^*+1}^{j^*} (c - \sigma_i^2) + \sum_{i=1}^{k^*} (\sigma_i^2 - \lambda_i) + \sum_{i=k^*+1}^{j^*} (\sigma_i^2 - \lambda_i) + k^*(c - \tilde{c}) \\
&\leq \sum_{i=k^*+1}^{j^*} (\lambda_i - \sigma_i^2) + \sum_{i=1}^{k^*} (\sigma_i^2 - \lambda_i) + \sum_{i=k^*+1}^{j^*} (\sigma_i^2 - \lambda_i) + k^*(c - \tilde{c}) \qquad (23) \\
&= \sum_{i=1}^{k^*} (\sigma_i^2 - \lambda_i) + k^*(c - \tilde{c}) \leq 0, \qquad (24)
\end{aligned}$$

where the inequality in (23) holds due to the fact that $c \leq \lambda_i$, for all $i \leq j^*$, and the inequality (24) is due to (19). Therefore, MSE(multi, c) $\leq$ MSE(n-single, $\tilde{c}$). □

C.3 Proofs of Theorems 5.1 and 5.2

The whitened UFM-loss used in the proof is

$$\frac{1}{2M} ||\mathbf{W}\mathbf{H} - \mathbf{Y}^{ZCA}||_F^2 + \frac{\lambda_{\mathbf{H}}}{2M} ||\mathbf{H}||_F^2 + \frac{\lambda_{\mathbf{W}}}{2} ||\mathbf{W}||_F^2. \qquad (25)$$

Proof of Theorem 5.1. It is easily seen that

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

Thus, the covariance matrix for the whitened targets is the $n \times n$ identity matrix.
Case $c < 1$:

By prior UFM analysis [Andriopoulos et al., 2024, Theorem 4.1], for a semi-orthogonal matrix $\mathbf{R}$, any global minimum $(\tilde{\mathbf{H}}, \tilde{\mathbf{W}})$ for (25) takes the form:

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

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 prior UFM analysis [Andriopoulos et al., 2024, Theorem 4.1], 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 (13) 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 $i \leq j^*$. □

C.4 Proof of Theorem 5.3

The normalized UFM-loss used in the proof is

$$\frac{1}{2M}||\mathbf{W}\mathbf{H}-\mathbf{Y}^{nrm}||_F^2+\frac{\lambda_{\mathbf{H}}}{2M}||\mathbf{H}||_F^2+\frac{\lambda_{\mathbf{W}}}{2}||\mathbf{W}||_F^2. \tag{26}$$

Proof of Theorem 5.3. 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 prior UFM analysis [Andriopoulos et al., 2024, Theorem 4.1], for a semi-orthogonal matrix $\mathbf{R}$, any global minimum $(\bar{\mathbf{H}},\bar{\mathbf{W}})$ for (26) 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 prior UFM analysis [Andriopoulos et al., 2024, Theorem 4.1], 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. □

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and introduction state that the theory concerns UFM training MSE and that finite-network experiments test qualitative orderings. The limitations section further narrows the claims and excludes generalization-theorem or deployment claims.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes].

   Justification: The limitations section discusses the UFM simplification, feature versus parameter regularization, the lack of quantitative finite-network or generalization guarantees, and the conditional nature of the conclusions.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes].

Justification: The UFM setup states the data, rank, covariance, and regularization assumptions, and the appendix provides proofs for the main theorems.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes].

Justification: The paper describes datasets, models, preprocessing comparisons, weight-decay sweeps, optimizers, batch sizes, epochs, seeds, and compute details in the experiments and appendix.

Guidelines:

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

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No].

   Justification: The paper cites the public data sources and gives experimental details, but we do not release code or reproduction scripts.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: The main experiments section and appendix specify dataset sizes, architectures, optimizers, learning rates, epochs, batch sizes, random seeds, and validation/test reporting.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes].

   Justification: The figures report variability across random seeds using error bars; captions and experimental text describe the aggregation over seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes].

   Justification: The appendix tables list GPU type, number of compute workers, requested memory, and approximate runtime for the MuJoCo and CARLA experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes].

   Justification: The work is theoretical and empirical analysis on cited benchmark-style datasets, with no human-subject intervention, deceptive data collection, or high-risk release.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes].

    Justification: The limitations section discusses the foundational nature of the work, possible benefits for neural regression design, and the need for downstream safety validation in robotics or autonomous-driving settings.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A].

    Justification: The paper does not release a high-risk model, scraped dataset, generative model, or dual-use system.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes].

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

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

    Answer: [N/A].

    Justification: The paper does not introduce a new dataset, benchmark, model release, or software asset.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or new research with human subjects.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A].

    Justification: The work does not conduct new human-subjects research and therefore does not require IRB approval for the reported experiments.

    Guidelines:

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

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core methodology does not use LLMs as a scientific component; any writing or formatting assistance does not affect the method, results, or originality.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.