Geometric Analysis of Neural Regression Collapse via Intrinsic Dimension

Zixuan Dong[1,3†*] **Bimarsha Adhikari**[2*] **George Andriopoulos**[2*] **Keith Ross**[2]

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

Code

Abstract

Neural multivariate regression underpins a wide range of domains such as control, robotics, and finance, yet the geometry of its learned representations remains poorly characterized. While neural collapse has been shown to benefit generalization in classification, we find that analogous collapse in regression consistently degrades performance. To explain this contrast, we analyze regression models through the lens of intrinsic dimension. Across control tasks and synthetic datasets, we estimate the intrinsic dimension of last-layer features ($\mathrm{ID_H}$) and compare it with that of the regression targets ($\mathrm{ID_Y}$). Collapsed models exhibit $\mathrm{ID_H} < \mathrm{ID_Y}$, leading to over-compression and poor generalization, whereas non-collapsed models typically maintain $\mathrm{ID_H} \geq \mathrm{ID_Y}$. For the non-collapsed models, performance with respect to $\mathrm{ID_H}$ depends on the data quantity and noise levels. From these observations, we identify two regimes—over-compressed and under-compressed—that characterize when expanding or reducing feature dimensionality is associated with better performance. Our results provide new geometric insights into neural regression collapse and offer a practical proxy for model selection.

1 Introduction

Neural multivariate regression has emerged as a cornerstone of modern machine learning, powering a wide spectrum of applications where the outputs are continuous and vector-valued. In imitation learning for autonomous driving, regression models predict control actions such as speed and steering angle from human driving demonstrations (Ross et al., 2011). In robotics, they enable agents to replicate expert trajectories (Chen et al., 2021b). In finance, regression underlies predictive analytics ranging from risk estimation to stock price forecasting (Gu et al., 2020). Finally, reinforcement learning employs regression to approximate value functions, with targets derived from Monte Carlo or bootstrapped returns (Gallici et al., 2025). The ubiquity of regression across these domains underscores the importance of a principled understanding of the representational geometry learned by neural networks in multivariate regression tasks.

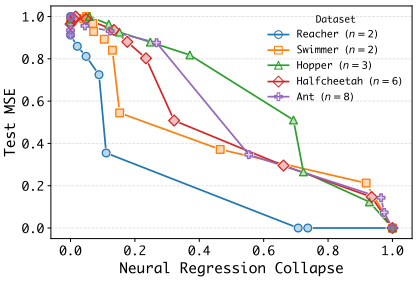


Figure 1: Neural Regression Collapse correlates with high Test MSE. The smaller the NRC value, the closer features lie to the n-dimensional subspace.

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

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

In this work, we empirically investigate the *geometric structure of neural multivariate regression*, with an emphasis on the geometry of last-layer feature vectors. Prior efforts have largely framed this problem through the lens of *neural collapse*. In classification, Neural Collapse (NC) describes the emergence of a highly symmetric configuration: last-layer features converge to the vertices of a Simplex Equiangular Tight Frame (ETF), aligned with the classifier weights (Papyan et al., 2020). In regression, by contrast, Neural Regression Collapse (NRC) manifests as the concentration of last-layer features within a linear subspace spanned by the top n principal components of the last-layer feature matrix, where n is the number of target variates. Since n is typically much smaller than the feature dimension, regression collapse implies a major reduction in representational degrees of freedom (Andriopoulos et al., 2024).

In this paper, we first make a key empirical observation: *In contrast with classification, collapsed regression models consistently exhibit degraded generalization as compared to their non-collapsed counterparts*. Figure 1 illustrates this phenomenon, showing high test MSE for models with highly collapsed features (low NRC metric) for five robotic locomotion tasks. Existing theoretical and empirical treatments of regression collapse, including the work of Andriopoulos et al. (2024), do not account for this degradation. This raises a central open question: **Why does neural collapse hinder generalization in multivariate regression, in contrast to its beneficial role in classification?**

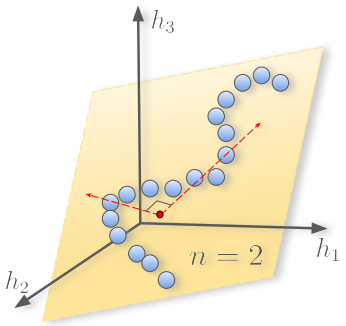


Figure 2: When the target dimension is $n = 2$, the collapsed features (blue points) with intrinsic dimension of 1 can lie close to a subspace (yellow plane) spanned by the first 2 principal components (red arrows) of the features.

We address this question using *intrinsic dimension (ID)*, a more refined tool than neural regression collapse for analyzing regression feature geometry. ID measures the effective dimensionality of the manifold in which data lies, capturing nonlinearities that the PCA approach of NRC cannot (Figure 2). We conduct a systematic study of collapsed and non-collapsed models across diverse regression tasks, including simulated robotic control and synthetic image regression. While ID has been studied in the context of neural classification (Ansuini et al., 2019), to the best of our knowledge, we are the first to systematically analyze neural regression collapse (NRC) from this perspective.

Our findings reveal that the intrinsic dimension of the regression targets ($\mathrm{ID_Y}$) acts as a critical threshold. Collapsed models typically satisfy $\mathrm{ID_H} < \mathrm{ID_Y}$, where $\mathrm{ID_H}$ denotes the intrinsic dimension of the last-layer features, meaning the learned representations are over-compressed and lack the degrees of freedom needed to reconstruct the target manifold. Non-collapsed models, in contrast, typically maintain $\mathrm{ID_H} \geq \mathrm{ID_Y}$. This threshold separates two regimes: an *over-compressed* regime ($\mathrm{ID_H} < \mathrm{ID_Y}$) where models with higher feature dimension generalize better, and an *under-compressed* regime ($\mathrm{ID_H} \geq \mathrm{ID_Y}$) where, in low-data or noisy settings, lower feature dimension is associated with better generalization instead. Together, these results explain why collapse is detrimental in regression and provide a practical signal for model selection. In summary, our contributions are:

- We provide, to the best of our knowledge, the first systematic investigation of neural regression collapse through the lens of intrinsic dimension. We show that intrinsic dimension reveals geometric structure that NRC's linear PCA methodology cannot capture, including nonlinear manifold collapse within the n-dimensional subspace (Section 5).

- We empirically demonstrate that the relationship between $\mathrm{ID_H}$ and $\mathrm{ID_Y}$ identifies over-compressed and under-compressed regimes, explaining both why regression collapse degrades generalization and under what conditions higher or lower feature dimensionality is associated with better performance (Section 6). We complement these findings with a mathematical analysis that provides intuition for how regularization drives over-compression (Section 7).

- Our refined geometric understanding of regression representations yields a practical diagnostic for model selection in applied regression tasks. We demonstrate that the ID gap $|\mathrm{ID_H} - \mathrm{ID_Y}|$ can serve as a practical surrogate for policy evaluation in visual robotic control, motivating principled model selection that reduces costly environment rollouts (Section 6.1).

2 Related Work

NC in classification. Neural collapse (NC) was first observed empirically by Papyan et al. (2020) and has since been analyzed theoretically through the Unconstrained Feature Model (Mixon et al., 2020) and the layer-peeled model (Fang et al., 2021), with subsequent extensions to deep, nonlinear unconstrained-feature models (Tirer & Bruna, 2022; Súkeník et al., 2023). NC has been shown to arise under diverse conditions (Han et al., 2021; Yaras et al., 2022; Zhou et al., 2022a,b; Zhu et al., 2021) and loss functions such as label smoothing (Guo et al., 2024). See also (Hong & Ling, 2023; Thrampoulidis et al., 2022; Yang et al., 2022).

NC beyond single-label classification. Li et al. (2023a) extended NC to multi-label classification, showing that embeddings lie in the span of their label means. Andriopoulos et al. (2024) formalized Neural Regression Collapse (NRC) for multivariate regression. Ma et al. (2025) demonstrated NC in deep ordinal regression under the UFM framework, Wu & Papyan (2024) identified "linguistic collapse" in large language models, and Súkeník et al. (2025) proved NC is globally optimal in modern regularized architectures including ResNets and transformers.

Intrinsic dimension in deep networks. Classical local-neighborhood estimators of intrinsic dimension (Levina & Bickel, 2004; Amsaleg et al., 2015; Facco et al., 2017; Allegra et al., 2020) have been applied in neural settings to distinguish adversarial from natural inputs via local intrinsic dimension (Ma et al., 2018a) and to detect untruthful LLM outputs (Yin et al., 2024). Complementary topological approaches—such as Neural Persistence (Rieck et al., 2018) and graph-based invariants (Corneanu et al., 2019, 2020)—provide an alternative lens on network structure, with recent theoretical grounding via persistent homology (Birdal et al., 2021), which has also been used to regularize regression representations (Zhang et al., 2024). Intrinsic dimension has also been studied as a complexity measure through parameter-space degrees of freedom (Gao & Jojic, 2016; Janson et al., 2015), pruning-based compressibility (Blier & Ollivier, 2018), and direct estimation (Ansuini et al., 2019; Li et al., 2018; Ma et al., 2018b; Pope et al., 2021), with compression-based generalization bounds linking lower-dimensional representations to better generalization (Arora et al., 2018; Barsbey et al., 2021; Hsu et al., 2021; Suzuki et al., 2018, 2019; Zhang et al., 2024). Earlier foundational work on network redundancy includes Denil et al. (2013) and LeCun et al. (1989).

3 Background and Key Metrics

We consider the *neural multivariate regression* problem (Andriopoulos et al., 2024), with M training examples $\{(\mathbf{x}_i, \mathbf{y}_i), i = 1, ..., M\}$, where each input $\mathbf{x}_i$ belongs to $\mathbb{R}^D$ and each target vector $\mathbf{y}_i$ belongs to $\mathbb{R}^n$. Considering a deep regression network of the form:

$$f_{\theta,\mathbf{W},\mathbf{b}}(\mathbf{x}) = \mathbf{W}\mathbf{h}_\theta(\mathbf{x}) + \mathbf{b},$$

where $\mathbf{h}_\theta(\cdot) : \mathbb{R}^D \to \mathbb{R}^d$ is the non-linear multi-layer feature extractor, $\mathbf{W} \in \mathbb{R}^{n\times d}$ represents the final linear layer in the model, and $\mathbf{b} \in \mathbb{R}^n$ is the bias vector. The parameters $\theta, \mathbf{W}, \mathbf{b}$ are all trainable. We typically train the DNN using gradient descent to minimize the regularized L_2 loss:

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

where $||\cdot||_2$ and $||\cdot||_F$ denote the L_2-norm and the Frobenius norm, respectively; and λ_{WD} denotes the full-model weight decay coefficient.

To characterize the geometric properties of last-layer representations of neural networks in regression tasks, we consider two central metrics: the NRC1 metric (Andriopoulos et al., 2024) and the 2-Nearest Neighbor (2-NN) intrinsic dimension estimator (Facco et al., 2017).

Neural Regression Collapse: NRC1 metric. In regression, neural collapse is defined by the extent to which the last-layer feature vectors collapse to a subspace spanned by their top n principal components (PCs). Let $\mathbf{h}_i := \mathbf{h}_\theta(\mathbf{x}_i)$ be the feature vector associated with example $\mathbf{x}_i$, $i = 1, \ldots, M$. Further let $\widetilde{\mathbf{h}}_i$ be the centered normalized feature vector, that is, $\widetilde{\mathbf{h}}_i := (\mathbf{h}_i - \bar{\mathbf{h}}) \cdot ||\mathbf{h}_i - \bar{\mathbf{h}}||^{-1}$ where $\bar{\mathbf{h}} := M^{-1}\sum_{i=1}^{M} \mathbf{h}_i$. For any matrix $\mathbf{C} \in \mathbb{R}^{p\times q}$ and any vector $\mathbf{v} \in \mathbb{R}^p$, let proj$(\mathbf{v}|\mathbf{C})$ denote the projection of $\mathbf{v}$ onto the subspace spanned by the columns of $\mathbf{C}$. Let $\mathbf{H}_{\text{PCA}}$ be the $d \times n$ matrix with

the columns consisting of the first n PCs of the feature matrix $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_M] \in \mathbb{R}^{d \times M}$. The NRC1 metric is defined as

$$\text{NRC1} := \frac{1}{M} \sum_{i=1}^{M} ||\widetilde{\mathbf{h}}_i - \text{proj}(\widetilde{\mathbf{h}}_i | \mathbf{H}_{\text{PCA}})||_2^2.$$

A model is considered collapsed if NRC1 is small, indicating that the features lie almost entirely within an n-dimensional subspace. Non-collapsed models exhibit higher NRC1 values, differing from those of collapsed models by orders of magnitude. Andriopoulos et al. (2024) demonstrated that slightly increased weight decay λ_{WD} quickly leads to model collapse during training. Figure 3 investigates NRC1 during training with varying weight decay parameters λ_{WD}. We see that when λ_{WD} is zero or small, there is no neural regression collapse; but if we increase the weight decay, the NRC1 geometric structure emerges. Although Andriopoulos et al. (2024) attributes collapse primarily to weight decay, feature collapse is not exclusive to this regularization. In Appendix G, we empirically show that dropout regularization (Srivastava et al., 2014) and the low-rank simplicity bias induced by increasing model depth (Huh et al., 2023) can also drive feature collapse, even under zero or mild weight decay.

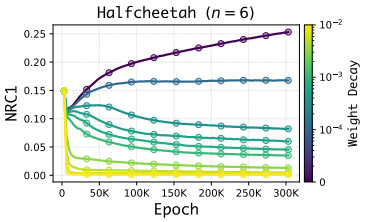


Figure 3: NRC1 decreases with stronger weight decay, leading to model collapse.

Intrinsic Dimension via 2-NN Estimation. To uncover the finer geometry of the learned features, beyond what linear methods like PCA reveal, we turn to intrinsic dimension: the minimal number of degrees of freedom needed to describe the data without significant information loss. Unlike PCA, intrinsic dimension captures the dimensionality of nonlinear manifolds (Figure 2), making it a more informative geometric descriptor.

We estimate intrinsic dimension using the 2-NN estimator introduced by Facco et al. (2017). For a given point, let r_1 and r_2 denote the distances to its first and second nearest neighbors; define the ratio $\mu := r_2/r_1$. Under the assumption of locally uniform data density, the cumulative distribution of the ratio follows a Pareto distribution with parameter d: $F(\mu) = 1 - \mu^{-d}$ for $\mu \geq 1$. The intrinsic dimension d is then estimated by linear regression.

The 2-NN estimator has been widely adopted to analyze data and representation manifolds across deep learning, including adversarial robustness (Ma et al., 2018a), image classification (Ansuini et al., 2019; Pope et al., 2021), representation correlation (Basile et al., 2025), and continuous control via reinforcement learning (Tiwari et al., 2025). Beyond its ability to reliably capture nonlinear manifold structure, it is hyperparameter-free, computationally efficient, and requires only that the data density is approximately uniform within the range of each point's second neighbor, a weaker assumption than the uniformity over larger neighborhoods required by other classical estimators (Levina & Bickel, 2004). In Appendix C, we provide further details and intuitions, and show that our conclusions are robust to two alternative estimators, MLE (Levina & Bickel, 2004) and PHD (Birdal et al., 2021).

4 Datasets

We perform experiments on proprioceptive and visual robot locomotion datasets as well as synthetic image regression datasets, described in this section. Full experimental details are provided in Appendix A. Additionally, Appendix H examines three more challenging datasets with varying data sizes and higher intrinsic dimensions.

Proprioceptive MuJoCo Locomotion. MuJoCo (Brockman et al., 2016; Towers et al., 2023) is a widely used physics simulator for continuous-control reinforcement learning. Following Andriopoulos et al. (2024), we use Reacher, Swimmer, and Hopper datasets, and additionally include the higher-dimensional Halfcheetah and Ant datasets from the D4RL benchmark (Fu et al., 2020). Each dataset contains expert demonstrations with proprioceptive state inputs ($\mathbf{x}_i$) and corresponding action targets ($\mathbf{y}_i$). States encode joint positions, angles, velocities, and angular velocities, while actions represent joint torques. We subsample the expert data to form low- and high-data regimes of 1,000

Table 1: Overview of datasets employed in the main body.

Dataset	Data Size	Input Type	Input Dim (D)	Input ID ($\mathrm{ID_X}$)	Target Dim (n)	Target ID ($\mathrm{ID_Y}$)
Swimmer	20,000	Raw state	8	3.98	2	2.00
Reacher	20,000	Raw state	11	3.65	2	1.99
Hopper	20,000	Raw state	11	4.35	3	2.92
Halfcheetah	20,000	Raw state	17	6.58	6	5.27
Ant	20,000	Raw state	111	7.83	8	7.43
MNIST	50,000	Grayscale image	28×28	12.76	25	8.02
CIFAR-10	50,000	RGB image	$32\times 32\times 3$	27.20	10	9.51
Cheetah-run	80,000	Stacked RGB image	$84 \times 84 \times 9$	8.96	6	6.00

and 20,000 samples, respectively. Trained policy architectures are 3-layer MLPs with varying width, operating directly on the proprioceptive state inputs.

Synthetic Image Regression. We construct two vision-based regression tasks that differ in the amount of task-irrelevant noise present in the targets. In both cases, regression targets are generated by applying a fixed random linear projection to extracted image features. For *MNIST regression*, features are extracted from the penultimate layer of a CNN trained to over 99% accuracy on MNIST, and then projected to 25 dimensions. Because the feature extractor is trained on the same domain, image-feature mismatches are largely suppressed, yielding clean, self-consistent targets. For *CIFAR-10 regression*, features are extracted using a ResNet-18 pretrained on ImageNet and *not* fine-tuned on CIFAR-10, and then projected to 10 dimensions. This domain mismatch causes the projected targets to retain instance-specific, task-irrelevant variation, resulting in noisy targets. For both tasks, regression models are 3-layer MLPs with varying width, operating on the same image tensors used to generate synthetic targets.

Visual Continuous Control. Visual control is particularly challenging because each frame provides only partial information about the system state, effectively forming a Partially Observed Markov Decision Process (POMDP) (Yarats et al., 2022). The model must infer underlying transition dynamics and identify salient visual features directly from high-dimensional pixel inputs. We use the Cheetah-run dataset (Lu et al., 2023), a visual control benchmark derived from the DeepMind Control Suite (Tassa et al., 2018), which also builds on the MuJoCo engine but differs from the proprioceptive tasks above in that inputs are raw RGB image observations. Following standard practice in visual control, we stack three consecutive frames to form $84 \times 84 \times 9$ input tensors, providing temporal context for inferring velocities and dynamics. Targets correspond to continuous control commands ($n = 6$), and we employ up to 80K expert demonstrations from the dataset. During training, the model architecture consists of a 4-layer CNN image encoder followed by a 3-layer MLP policy head.

5 Intrinsic Dimension Refines NRC Geometry

NRC1 measures the degree to which last-layer features collapse to an n-dimensional linear subspace, and as shown in Figure 3, a small amount of weight decay often suffices for such collapse to occur. However, NRC1 does not reveal whether the features further collapse into lower-dimensional nonlinear manifolds, a limitation shared by other spectrum-based metrics such as effective rank (Appendix C.3). To explore this question, we measure the intrinsic dimension of the last-layer features, denoted $\mathrm{ID_H}$, via the 2-NN estimator.

Figure 4 presents scatter plots of $\mathrm{ID_H}$ versus NRC1 for five MuJoCo datasets. The critical value $\mathrm{ID_Y}$, the intrinsic dimension of the targets, denoted by the dashed-red lines in Figure 4, is consistently upper-bounded by n, the dimension of the targets. For highly collapsed models with near-zero NRC1, the intrinsic dimension of last-layer features satisfies $\mathrm{ID_H} \leq \mathrm{ID_Y}$ and continues to shrink even as NRC1 saturates at zero. That is, *collapsed models learn features that lie on increasingly lower-dimensional nonlinear manifolds within an n-dimensional linear subspace*. So, although the NRC1 metric is useful in understanding the linear-subspace structure of the last-layer features in collapsed models, it is inadequate at uncovering this more refined geometric structure.

To further verify this nonlinear structure, Figure 5 visualizes PCA projections of collapsed 1D features in Swimmer ($n = 2$) and Hopper ($n = 3$), revealing visibly curved manifolds that cannot be spanned by a single linear subspace. For the higher-dimensional Halfcheetah dataset, we examine a collapsed

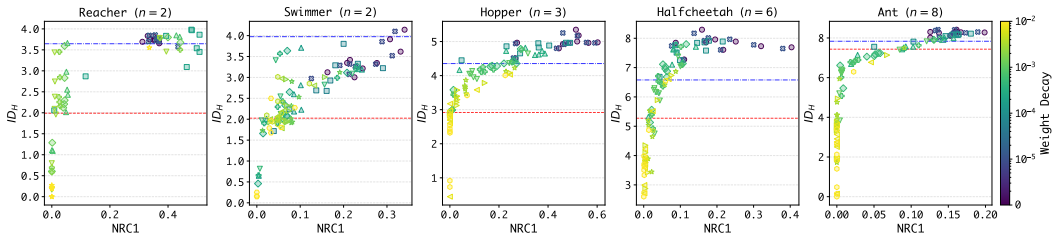
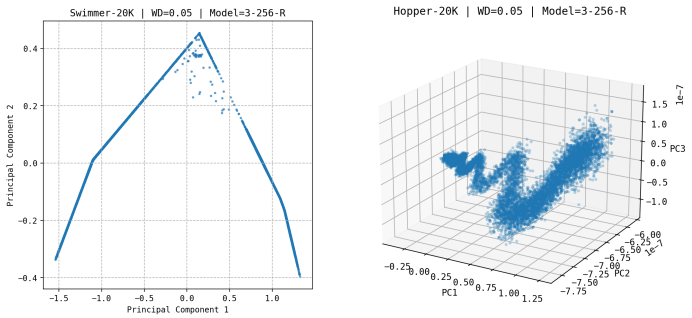


Figure 4: Relationship between NRC1 and intrinsic dimension of the last-layer features $\mathrm{ID_H}$. Dots correspond to models trained with different architectures and weight decay parameters, with colors denoting the degree of weight decay. The horizontal red dashed line is drawn at $\mathrm{ID_Y}$, the intrinsic dimension of the targets. The blue line is drawn at $\mathrm{ID_X}$, the intrinsic dimension of the inputs.



k	Global PCA	Local PCA
6	3.4×10^{-3}	2.8×10^{-5}
5	7.5×10^{-3}	7.9×10^{-5}
4	8.4×10^{-2}	3.5×10^{-4}
3	4.9×10^{-1}	1.5×10^{-3}

Figure 5: Nonlinear structure of collapsed feature manifolds. **Left, Center:** PCA projections of collapsed 1D feature manifolds for Swimmer ($n = 2$) and Hopper ($n = 3$), showing visibly curved structures. **Right:** Global vs. local PCA projection error for collapsed Halfcheetah features ($\mathrm{ID_H} = 3.6$, $n = 6$). Local errors (projection onto the top-k PCs of 64-nearest-neighbor feature matrix) are orders of magnitude smaller, confirming locally flat but globally curved geometry.

feature manifold with $\mathrm{ID_H} = 3.6 < n = 6$. We compare global PCA projection errors (onto the top-k PCs of the full feature matrix) with local PCA projection errors (onto the top-k PCs of each point's 64 nearest neighbors). As shown in the table of Figure 5, the global error increases sharply from $k = 5$ to $k = 4$, while the local error jumps from $k = 4$ to $k = 3$. This indicates that the nearly-4D feature manifold is embedded in a 5D linear subspace and cannot itself be linear. These results echo the analysis of curved feature manifolds in classification by Ansuini et al. (2019).

In contrast, non-collapsed models with higher NRC1 values satisfy $\mathrm{ID_H} > \mathrm{ID_Y}$, and their intrinsic dimension increases monotonically with NRC1, making the two metrics qualitatively interchangeable in this regime. In other words, *non-collapsed models learn features whose manifold dimension exceeds that of the targets*. Intrinsic dimension thus offers two concrete advantages over NRC1: it reveals a soft threshold at $\mathrm{ID_Y}$ separating two geometrically distinct regimes, and it quantifies the degree of collapse ($\mathrm{ID_H}$) across the full range of NRC1 values. For the remainder of this paper, we therefore focus on intrinsic dimension. Appendix D further analyzes the evolution of intrinsic dimension during training.

6 Intrinsic Dimension Predicts NRC Generalization

Having now investigated the relationship between NRC and intrinsic dimension, we now examine what insights intrinsic dimension can provide about NRC generalization. Particularly, we will explore why generalization error increases as $\mathrm{ID_H}$ (and hence as NRC1) decreases, as seen in Figure 1. As we discuss in more detail at the end of this section, this property is in contrast with classification, for which performance typically improves when neural collapse becomes stronger.

Figure 6 shows the relationship between $\mathrm{ID_H}$ and both training and test MSE for five datasets: Halfcheetah-1K, Halfcheetah-20K, CIFAR-10, MNIST, and Cheetah-run-20K datasets. The corresponding figures for other MuJoCo datasets are in Appendix F.

Train MSE decreases when $\mathrm{ID_H}$ increases. This trend is evident in the first row of Figure 6. To explain this, from Figures 3 and 4 we have known stronger regularization reduces $\mathrm{ID_H}$. Theorems 4.1 and 4.3 in Andriopoulos et al. (2024) also tell that stronger regularization reduces the dimension of the linear subspace containing the feature manifold. Thus, by reducing $\mathrm{ID_H}$, the trained features tend to get squashed onto a lower-dimensional and more curved manifold, similar to the "crowding problem" described by Maaten & Hinton (2008). A global linear layer $\mathbf{W}$, performing only an affine transformation, cannot "unbend" such a manifold. Thus, as $\mathrm{ID_H}$ decreases, it becomes more difficult for $\mathbf{WH} + \mathbf{b}$ to accurately match $\mathbf{Y}$ (which lies on its own curved manifold), explaining why train MSE decreases when $\mathrm{ID_H}$ increases.

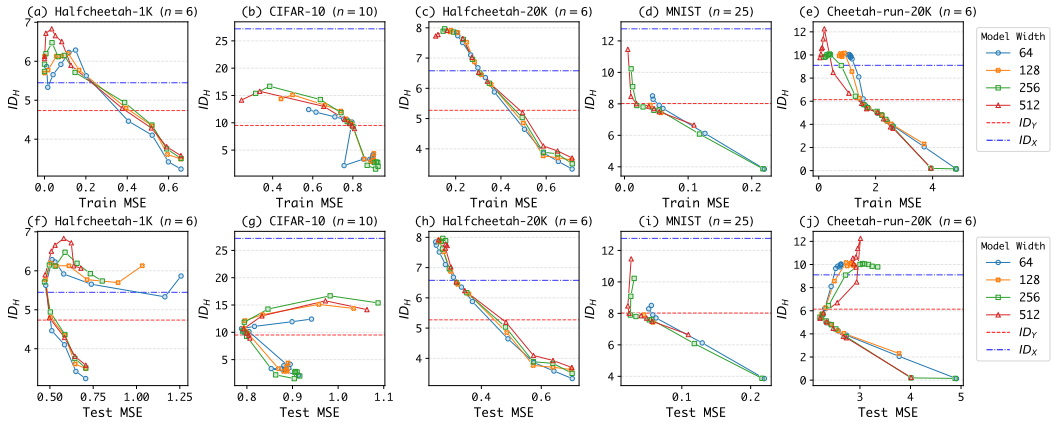


Figure 6: Generalization and Intrinsic Dimension for Halfcheetah, CIFAR-10, MNIST, and Cheetah-run datasets.

Test MSE with respect to $\mathrm{ID_H}$ behaves differently according to its relationship to $\mathrm{ID_Y}$. The second row of Figure 6 shows fundamental differences between collapsed and non-collapsed models:

- $(\mathrm{ID_H} < \mathrm{ID_Y})$: In this regime, the collapsed model's features are confined to a manifold whose intrinsic dimension is lower than that of the targets. This *over-compression* intuitively means the last-layer features lack information essential for reconstructing and generalizing beyond the full target manifold, and Section 7 will present a mathematical formulation of this claim. In this regime, models with higher last-layer intrinsic dimension, obtained for example with weaker regularization or a different architecture, generalize better. We can now answer the question posed in the Introduction: Why does neural collapse hinder generalization in multivariate regression? The explanation simply follows from (1) the monotonic relationship between NRC1 and $\mathrm{ID_H}$; and (2) the reconstruction inability as just described.
- $(\mathrm{ID_H} \geq \mathrm{ID_Y})$: We distinguish these non-collapsed models between two cases:
  - *(i) Low-data or noisy-target tasks.* In this case (Figs. 6 (f), (g), (j)), the test MSE plots exhibit surprising U-shaped dependence on $\mathrm{ID_H}$, with a minimum error near $\mathrm{ID_H} = \mathrm{ID_Y}$. Why does test MSE increase with $\mathrm{ID_H}$ when $\mathrm{ID_H} > \mathrm{ID_Y}$? To explain this, we note that when the amount of training data is small, or when the targets are noisy, f_θ learns a feature manifold whose intrinsic dimension is higher than that of the true feature manifold due to the stronger negative effect of outliers. Then, the extra dimensions in the feature manifold are used to predict training sample-specific noise, leading to overfitting the training set (Ma et al., 2018b; Ansuini et al., 2019). This overfitting is exacerbated when regularization is reduced, or equivalently, when $\mathrm{ID_H}$ is increased, leading to higher test MSE.
  - *(ii) High-data and low-noise tasks.* In this case, test MSE follows the same trend as train MSE, decreasing monotonically with $\mathrm{ID_H}$ (Figs. 6 (h),(i)). To explain, we note that with a large amount of training data and low target noise, f_θ can fit the training data closely while maintaining smoothness to avoid overfitting, and consequently, the training feature manifold $\mathbf{H}_{\text{train}}$ is likely to resemble the test feature manifold $\mathbf{H}_{\text{test}}$.

Table 2 provides the key takeaways concerning generalization and intrinsic dimension. In Appendix B, we review the generalization of neural collapse for classification tasks and contrast it to our study for multivariate regression tasks.

Table 2: Key Takeaways for Generalization.

Regime	ID	Typical behavior
Over-compressed	$\mathrm{ID_H} < \mathrm{ID_Y}$	Underfitting with large train and test MSE
Balanced	$\mathrm{ID_H} \approx \mathrm{ID_Y}$	Sweet spot in low-data and noisy tasks
Under-compressed	$\mathrm{ID_H} > \mathrm{ID_Y}$	Benign overfitting with enough clean data

6.1 Practical Guideline: Intrinsic Dimension as a Surrogate for Policy Evaluation

Our findings regarding NRC geometry have direct implications for evaluating real-world control models, where deployment-time evaluation is expensive, unsafe, and resource-limited (Levine et al., 2020). We demonstrate this on the Cheetah-run visual control task introduced in Section 4, where we train policies via behavior cloning on 20K and 80K expert demonstrations and evaluate them by executing the learned policy in the DeepMind Control Suite simulator (Tassa et al., 2018).

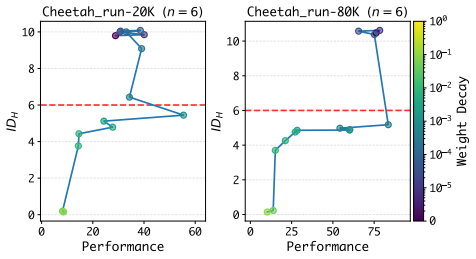


Figure 7: Relationship between $\mathrm{ID_H}$ and normalized model performance ($\in [0, 100]$; the higher, the better). The horizontal red dashed line represents $\mathrm{ID_Y}$.

Figure 7 plots the relationship between $\mathrm{ID_H}$ and *actual* normalized policy performance (ranging from 0 to 100). Consistent with the test MSE analysis, performance exhibits an inverted U-shape, peaking near $\mathrm{ID_H} = \mathrm{ID_Y}$, which mirrors the ID–generalization relationship identified in Section 6. Crucially, validation MSE alone is an unreliable proxy for policy quality: small MSE differences can correspond to large variations in control performance (cf. Fig. 6 (j) and Fig. 7). In contrast, the gap between $\mathrm{ID_H}$ and $\mathrm{ID_Y}$ provides a complementary signal that consistently predicts both low test MSE and high policy performance.

This observation motivates an actionable model selection strategy. Our analysis shows that well-generalizing models either maximize $\mathrm{ID_H}$ in the high-data regime or achieve $\mathrm{ID_H} \approx \mathrm{ID_Y}$ in the low-data and noisy-target regime. Since the regime is rarely known in advance, we retain both types of models: those with the largest $\mathrm{ID_H}$ and those with the smallest ID gap, $|\mathrm{ID_H} - \mathrm{ID_Y}|$. Together, they form a compact candidate pool that contains the potentially best-performing policies regardless of the regime, serving as a principled selection criterion. Compared to exhaustive hyperparameter grid search, this approach offers two advantages. First, it avoids missing optimal configurations in regions where the learned feature geometry changes abruptly across regimes. For instance, in the first row of Figure 10 (Reacher), increasing weight decay by as little as 2×10^{-4} can shift a model into the over-compressed regime. Second, it reduces dependence on costly real-environment rollouts for policy evaluation, which in real robotic systems are inherently risky and resource-constrained.

While these results constitute preliminary evidence on a simulated benchmark, they point toward a promising direction: automated, efficient hyperparameter selection via ID gap minimization, complementing or partially replacing expensive deployment-time evaluation.

7 Mathematical Intuition of Regression Collapse

To provide a principled explanation for the empirical phenomena observed in Section 6, we present a mathematical characterization of regression collapse. We first describe how weight decay induces a dimensional reduction in the feature space (Section 7.1) and subsequently formalize why this necessitates a reconstruction error (Section 7.2).

7.1 Why Weight Decay Causes Collapse

We analyze the effect of regularization through the Unconstrained Feature Model (UFM) framework adopted in prior work (Mixon et al., 2022; Andriopoulos et al., 2024), which treats the last-layer feature matrix as a free optimization variable rather than the output of a parameterized feature extractor. In this section, we slightly abuse some notations introduced in Section 3. Let the feature

matrix be $\mathbf{H} \in \mathbb{R}^{d\times M}$. Given a centered target matrix $\tilde{\mathbf{Y}} \in \mathbb{R}^{n\times M}$ with empirical covariance $\mathbf{\Sigma} = \frac{1}{M}\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top \in \mathbb{R}^{n\times n}$ (assuming $\mathbf{\Sigma}$ is full rank with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n > 0$), the UFM minimizes:

$$\min_{\mathbf{W},\mathbf{H}} \frac{1}{2M}\|\mathbf{W}\mathbf{H} - \tilde{\mathbf{Y}}\|_F^2 \; + \; \frac{\lambda_W}{2}\|\mathbf{W}\|_F^2 \; + \; \frac{\lambda_H}{2}\|\mathbf{H}\|_F^2, \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{n\times d}$ is the final linear layer; and λ_W, λ_H are separate L_2 regularization parameters.[2] Without regularization ($\lambda_W = \lambda_H = 0$), global minimizers satisfy $\mathbf{W}\mathbf{H} = \tilde{\mathbf{Y}}$ and decompose as $\mathbf{H} = \mathbf{W}^\dagger\tilde{\mathbf{Y}} + (\mathbf{I}_d - \mathbf{W}^\dagger\mathbf{W})\mathbf{Z}$ for arbitrary $\mathbf{Z} \in \mathbb{R}^{d\times M}$ (Andriopoulos et al., 2024). The term $\mathbf{W}^\dagger\tilde{\mathbf{Y}}$ lies in the at-most-n-dimensional row space of $\mathbf{W}$, while $(\mathbf{I}_d - \mathbf{W}^\dagger\mathbf{W})\mathbf{Z}$ spans the $(d-n)$-dimensional null space of $\mathbf{W}$, allowing $\mathbf{H}$ to occupy up to d dimensions. With regularization, our following result shows that this null-space freedom is eliminated.

Theorem 7.1 (Limiting Solution Structure; informal, see Appendix I.1)**.** *Let $(\mathbf{W}_\lambda, \mathbf{H}_\lambda)$ be the global minimizer of equation 1 and suppose $\lim_{\lambda_H\to 0,\lambda_W\to 0}(\lambda_H/\lambda_W) = k > 0$. Then, as regularization gradually vanishes, $\mathbf{W}_\lambda\mathbf{H}_\lambda \to \tilde{\mathbf{Y}}$ and the limiting features $\mathbf{H}_0 = \lim \mathbf{H}_\lambda$ satisfy $\mathbf{H}_0 = \mathbf{W}_0^\dagger\tilde{\mathbf{Y}}$, the minimum-norm solution with $\mathbf{Z} = 0$. In particular, $\mathbf{H}_0$ lies entirely within the n-dimensional row space of $\mathbf{W}_0$.*

Thus, even infinitesimally small weight decay biases features toward the minimum-norm solution, collapsing $\mathbf{H}$ from the d-dimensional ambient space onto an n-dimensional subspace. Since typically $d \gg n$, this represents a massive dimensional reduction.

From Neural Regression Collapse to ID Over-compression. Theorem 7.1 explains why features concentrate in an n-dimensional linear subspace, but does not yet explain why $\mathrm{ID_H}$ can fall *below* $\mathrm{ID_Y}$. This transition is governed by the product $c := \lambda_W\lambda_H$. When $c < \lambda_n$ (the smallest eigenvalue of $\mathbf{\Sigma}$), the regularized solution maps $\tilde{\mathbf{Y}}$ to $\mathbf{H}_\lambda$ via an invertible affine transformation, preserving intrinsic dimension, e.g., $\mathrm{ID_H} = \mathrm{ID_Y}$. However, when $c \geq \lambda_n$, the solution reduces $\mathrm{rank}(\mathbf{H}_\lambda) < n$ and breaks the invertibility. As a consequence, affine congruency is not guaranteed to hold, allowing $\mathrm{ID_H}$ to drop below $\mathrm{ID_Y}$, entering the over-compressed regime. The formal characterization is provided in Remark I.5 (Appendix I.1).

This analysis highlights the diagnostic value of intrinsic dimension in practice. The UFM parameter $c = \lambda_W\lambda_H$ is not directly observable, as it arises from a simplified theoretical abstraction rather than an explicit practical training objective. However, intrinsic dimension serves as an empirical proxy: the transition from $\mathrm{ID_H} \approx \mathrm{ID_Y}$ to $\mathrm{ID_H} < \mathrm{ID_Y}$ signals that the effective regularization strength has crossed the critical threshold $c = \lambda_n$, placing the model in the over-compressed regime where affine congruency between features and targets breaks down.

7.2 Why Collapsed Models Fail to Generalize

The over-compression identified above creates an unavoidable reconstruction error, which we formalize by treating the final layer as a smooth map g from the feature manifold to the target space.

Theorem 7.2 (Non-surjectivity of Over-compressed Maps)**.** *Let $\mathcal{M}_H$ be a smooth m-dimensional feature manifold and $\mathcal{N}_Y$ be a smooth n-dimensional target manifold, with $m < n$. A smooth map $g : \mathcal{M}_H \to \mathcal{N}_Y$ cannot be surjective; specifically, the image $g(\mathcal{M}_H)$ has Lebesgue measure zero in $\mathcal{N}_Y$.*

As proven in Appendix I.2 via Sard's Theorem, the condition $m < n$ in the over-compressed regime implies that the set of all possible model predictions is a proper subset of the target manifold. Geometrically, there will always be points on the target manifold that lie outside the model's predictive reach, making perfect reconstruction mathematically impossible. This explains why over-compressed models consistently exhibit high test MSE across all experiments (Figures 6, 13 and 14).

8 Limitations and Future Work

Limitations. Our theoretical results explain why weight decay induces collapse and why over-compressed models cannot fully reconstruct the target manifold, but a complete characterization of

[2] Practical regression objective defined in Section 3 applies the same weight decay regularization to all model parameters.

the relationship between intrinsic dimension and generalization is not yet available. These results are derived under the Unconstrained Feature Model (UFM), which treats the last-layer features as free optimization variables and thus abstracts away the feature extractor, network depth, and architecture. Our empirical evidence linking $\mathrm{ID_H}$ to generalization is correlational: $\mathrm{ID_H}$ varies through the interventions that naturally induce NRC (weight decay, dropout, and depth), which may also affect other aspects of training. Accordingly, the ID gap should be read as a predictive signal for model selection rather than as a quantity whose direct manipulation is guaranteed to improve generalization.

Extending the theory beyond the UFM. A natural path for future work is to follow the progression in the classification literature from the UFM to deep, nonlinear unconstrained-feature models and beyond (Section 2). For regression, this requires replacing the cross-entropy loss with the squared loss and discrete classes with a continuous target manifold. Such an analysis could account for the layer-wise ID reduction we observe in collapsed models (Appendix D), clarifying where along depth over-compression arises and how architecture interacts with the $\mathrm{ID_Y}$ threshold. Deriving generalization bounds that incorporate NRC and ID would further ground the empirical relationships we report.

Characterizing data regimes that favor a small ID gap. Our analysis concerns the regularized regression problem, in which $\mathrm{ID_H}$ varies through the interventions that naturally induce NRC (weight decay, dropout, and depth). A complementary line of work instead imposes intrinsic dimension directly as a training objective for regression, using differentiable persistent-homology estimators (Birdal et al., 2021) to match the intrinsic dimension and topology of the features to those of the targets (Zhang et al., 2024). Our results suggest that whether such matching is beneficial depends on the data regime: a small ID gap $|\mathrm{ID_H} - \mathrm{ID_Y}|$ is favored in low-data or noisy-target settings, whereas performance continues to improve with larger $\mathrm{ID_H}$ in high-data, low-noise settings. Characterizing more precisely which properties of the data, such as sample size, target noise, and target-manifold geometry, favor a small ID gap is an important direction for both regularization-induced and explicitly ID-regularized regression.

Identifying implicit modulators of NRC and ID. Weight decay, dropout, and depth are not designed to control intrinsic dimension, yet each of them shifts $\mathrm{ID_H}$ and, through it, generalization. Other common regularizers and algorithmic design choices, such as normalization layers, data augmentation, the choice of optimizer, learning-rate schedules, batch size, and early stopping, may play a similar implicit role. Systematically studying which of these choices modify NRC and $\mathrm{ID_H}$, in which direction, and with what effect on generalization is both interesting and practically important: it would reveal when standard training recipes unintentionally push models into the over-compressed regime, and would make the ID gap a diagnostic for tuning them. Finally, exploring whether intrinsic dimension provides similar insights for other learning paradigms, such as generative modeling, reinforcement learning, and multi-task learning, could offer a unifying geometric perspective across domains.

9 Conclusion

In this paper, we provided a systematic geometric analysis of neural regression collapse through the lens of intrinsic dimension. We showed that regression collapse corresponds to an over-compressed regime ($\mathrm{ID_H} < \mathrm{ID_Y}$) that discards information essential for reconstructing target manifolds, explaining why collapse, beneficial in classification, is detrimental in regression. For non-collapsed models, generalization behavior depends on whether the task is low-data/noisy or high-data/low-noise. These findings yield a practical guideline: over-compression ($\mathrm{ID_H} < \mathrm{ID_Y}$) signals poor generalization, and models with the largest $\mathrm{ID_H}$ or the smallest ID gap $|\mathrm{ID_H} - \mathrm{ID_Y}|$ form a candidate pool of well-generalizing models regardless of the data regime, providing a principled criterion for model selection in applied multivariate regression tasks such as robotic control.

Acknowledgments

This work was supported in part by NYU Abu Dhabi Center for Artificial Intelligence and Robotics and the Center for Interdisciplinary Data Science and AI, funded by Tamkeen under the Research Institute Award CG010. It was also partially supported by the Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning at NYU Shanghai. Computational resources were provided in part by NYU IT High-Performance Computing resources and services.

References

Michele Allegra, Elena Facco, Francesco Denti, Alessandro Laio, and Antonietta Mira. Data segmentation based on the local intrinsic dimension. *Scientific reports*, 10(1):16449, 2020.

Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E Houle, Ken-ichi Kawarabayashi, and Michael Nett. Estimating local intrinsic dimensionality. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 29–38, 2015.

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.

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International conference on machine learning*, pp. 254–263. PMLR, 2018.

Melih Barsbey, Milad Sefidgaran, Murat A Erdogdu, Gael Richard, and Umut Simsekli. Heavy tails in sgd and compressibility of overparametrized neural networks. *Advances in neural information processing systems*, 34:29364–29378, 2021.

Lorenzo Basile, Santiago Acevedo, Luca Bortolussi, Fabio Anselmi, and Alex Rodriguez. Intrinsic dimension correlation: uncovering nonlinear connections in multimodal representations. In *The Thirteenth International Conference on Learning Representations*, 2025.

Tolga Birdal, Aaron Lou, Leonidas J Guibas, and Umut Simsekli. Intrinsic dimension, persistent homology and generalization in neural networks. *Advances in neural information processing systems*, 34:6776–6789, 2021.

Léonard Blier and Yann Ollivier. The description length of deep learning models. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Johan Samir Obando Ceron, Aaron Courville, and Pablo Samuel Castro. In value-based deep reinforcement learning, a pruned network is a good network. In *Forty-first International Conference on Machine Learning*, 2024.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021a.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021b.

Ciprian A Corneanu, Meysam Madadi, Sergio Escalera, and Aleix M Martinez. What does it mean to learn in deep networks? and, how does one detect adversarial attacks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4757–4766, 2019.

Ciprian A Corneanu, Sergio Escalera, and Aleix M Martinez. Computing the testing error without a testing set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2677–2685, 2020.

Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando De Freitas. Predicting parameters in deep learning. *Advances in neural information processing systems*, 26, 2013.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):12140, 2017.

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.

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

Matteo Gallici, Mattie Fellows, Benjamin Ellis, Bartomeu Pou, Ivan Masmitja, Jakob Nicolaus Foerster, and Mario Martin. Simplifying deep temporal difference learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Quentin Gallouédec, Edward Emanuel Beeching, Clément ROMAC, and Emmanuel Dellandrea. Jack of all trades, master of some, a multi-purpose transformer agent. In *ICML 2024 Workshop: Aligning Reinforcement Learning Experimentalists and Theorists*, 2024.

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

Tianxiang Gao and Vladimir Jojic. Degrees of freedom in deep neural networks. *arXiv preprint arXiv:1603.09260*, 2016.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning*, 2023.

Arna Ghosh, Arnab Kumar Mondal, Kumar Krishna Agrawal, and Blake Richards. Investigating power laws in deep representation learning, 2022.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.

Li Guo, George Andriopoulos, Zifan Zhao, Shuyang Ling, Zixuan Dong, and Keith Ross. 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.

Takuya Hiraoka, Takahisa Imagawa, Taisei Hashimoto, Takashi Onishi, and Yoshimasa Tsuruoka. Dropout q-functions for doubly efficient reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xCVJMsPv3RT.

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

Daniel Hsu, Ziwei Ji, Matus Telgarsky, and Lan Wang. Generalization bounds via distillation. *arXiv preprint arXiv:2104.05641*, 2021.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Lucas Janson, William Fithian, and Trevor J Hastie. Effective degrees of freedom: a flawed metaphor. *Biometrika*, 102(2):479–485, 2015.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations*, 2021.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems*, 17, 2004.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

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*, 2023a.

Qiyang Li, Aviral Kumar, Ilya Kostrikov, and Sergey Levine. Efficient deep reinforcement learning requires regulating overfitting. In *The Eleventh International Conference on Learning Representations*, 2023b.

Xiao Li, Sheng Liu, Jinxin Zhou, Xinyu Lu, Carlos Fernandez-Granda, Zhihui Zhu, and Qing Qu. Understanding and improving transfer learning of deep models via neural collapse. *arXiv preprint arXiv:2212.12206*, 2022.

Zhuang Liu, Xuanlin Li, Bingyi Kang, and Trevor Darrell. Regularization matters in policy optimization - an empirical study on continuous control. In *International Conference on Learning Representations*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Cong Lu, Philip J. Ball, Tim G. J. Rudner, Jack Parker-Holder, Michael A Osborne, and Yee Whye Teh. Challenges and opportunities in offline reinforcement learning from visual observations. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. In *International Conference on Learning Representations*, 2022.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, pp. 23190–23211. PMLR, 2023.

Chuang Ma, Tomoyuki Obuchi, and Toshiyuki Tanaka. Neural collapse in cumulative link models for ordinal regression: An analysis with unconstrained feature model. *arXiv preprint arXiv:2506.05801*, 2025.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018a.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pp. 3355–3364. PMLR, 2018b.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

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

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

Michal Nauman, Michał Bortkiewicz, Piotr Miłoś, Tomasz Trzcinski, Mateusz Ostaszewski, and Marek Cygan. Overestimation, overfitting, and plasticity in actor-critic: the bitter lesson of reinforcement learning. In *Forty-first International Conference on Machine Learning*, 2024.

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.

Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*, 2021.

PyTorch Contributors. Pytorch torch.optim.adamw documentation. `https://docs.pytorch.org/docs/stable/generated/torch.optim.AdamW.html`, 2025.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. *arXiv preprint arXiv:1812.09764*, 2018.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

Peter Súkeník, Marco Mondelli, and Christoph H Lampert. Deep neural collapse is provably optimal for the deep unconstrained features model. In *Advances in Neural Information Processing Systems*, 2023.

Peter Súkeník, Christoph H Lampert, and Marco Mondelli. Neural collapse is globally optimal in deep regularized resnets and transformers. *arXiv preprint arXiv:2505.15239*, 2025.

Taiji Suzuki, Hiroshi Abe, Tomoya Murata, Shingo Horiuchi, Kotaro Ito, Tokuma Wachi, So Hirai, Masatoshi Yukishima, and Tomoaki Nishimura. Spectral pruning: Compressing deep neural networks via spectral analysis and its generalization error. *arXiv preprint arXiv:1808.08558*, 2018.

Taiji Suzuki, Hiroshi Abe, and Tomoaki Nishimura. Compression based bound for non-compressed network: unified generalization error analysis of large compressible deep neural network. *arXiv preprint arXiv:1909.11274*, 2019.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018.

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

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

Saket Tiwari, Omer Gottesman, and George Konidaris. Geometry of neural reinforcement learning in continuous state and action spaces. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

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.

Siwei Wang and Stephanie E Palmer. Towards understanding neural collapse in supervised contrastive learning with the information bottleneck method. *arXiv preprint arXiv:2305.11957*, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, 2020.

Robert Wu and Vardan Papyan. Linguistic collapse: Neural collapse in (large) language models. *Advances in Neural Information Processing Systems*, 37:137432–137473, 2024.

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

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

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=_SJ-_yyes8`.

Fan Yin, Jayanth Srinivasa, and Kai-Wei Chang. Characterizing truthfulness in large language model generations with local intrinsic dimension. *arXiv preprint arXiv:2402.18048*, 2024.

Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K Terry. Minari, 2024. URL `https://doi.org/10.5281/zenodo.13767625`.

Shihao Zhang, Kenji Kawaguchi, and Angela Yao. Deep regression representation learning with topology. In *Forty-first International Conference on Machine Learning*, 2024.

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

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

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

Appendix Contents

A Experiment Details **18**

A.1 MuJoCo experiments 18

A.2 MNIST/CIFAR10 experiments 20

A.3 Visual Control Experiments 20

B Comparison between Regression and Classification **21**

C Intrinsic Dimension Estimation **22**

C.1 The 2-NN Estimator 22

C.2 Robustness to the Choice of Estimator 23

C.3 Intrinsic Dimension versus Spectral Metrics 24

D Evolution of Intrinsic Dimension during Training **25**

E Intrinsic Dimension and Output layer **26**

F Additional Experiments on Generalization **29**

G How does regularization affect NRC and Intrinsic Dimension? **31**

G.1 Weight Decay 31

G.2 Dropout Regularization 31

G.3 Model Depth 32

H Empirical Analysis on More Challenging Tasks **33**

H.1 Visual Control Tasks 34

H.2 Relocate & Humanoid 35

I Mathematical Analysis of Regression Collapse **37**

I.1 Why Weight Decay Leads to Collapse: Analysis via the Unconstrained Feature Model 37

I.2 Why Collapsed Models Fail to Generalize? 39

A Experiment Details

Table 3: Overview of datasets employed in the main body.

Dataset	Data Size	Input Type	Input Dim (D)	Input ID (ID_X)	Target Dim (n)	Target ID (ID_Y)
Swimmer	20,000	Raw state	8	3.98	2	2.00
Reacher	20,000	Raw state	11	3.65	2	1.99
Hopper	20,000	Raw state	11	4.35	3	2.92
Halfcheetah	20,000	Raw state	17	6.58	6	5.27
Ant	20,000	Raw state	111	7.83	8	7.43
MNIST	50,000	Grayscale image	28×28	12.76	25	8.02
CIFAR-10	50,000	RGB image	$32\times 32\times 3$	27.20	10	9.51
Cheetah-run	80,000	Stacked RGB image	$84 \times 84 \times 9$	8.96	6	6.00

A.1 MuJoCo experiments

MuJoCo (Multi-Joint dynamics with Contact) is a physics engine designed for research in robotics, biomechanics, and animation, providing fast and accurate simulations of systems involving complex contact dynamics. It balances physical realism with computational efficiency, enabling reliable modeling of robot–environment interactions (Towers et al., 2024). Environments involved in this work include:

- **Reacher**: A two-jointed robotic arm tasked with moving its tip to a randomly generated target in a 2D plane.
- **Swimmer**: A chain-like robot with three body segments connected by two rotors, aiming to propel itself forward in 2D as quickly as possible.
- **Hopper**: A one-legged, four-part robot that seeks to hop forward at maximum speed in 2D.
- **HalfCheetah**: A planar, bipedal robot with a torso and two legs, each consisting of two joints. It aims to run forward as quickly as possible along a 2D track by coordinating its leg movements.
- **Ant**: A quadrupedal robot with four legs and multiple joints, designed to move in a 3D plane. Its goal is to walk or run forward efficiently, despite the challenge of balancing and coordinating many degrees of freedom. Although Ant's state space has 111 dimensions, 84 of the dimensions related to external contact forces are always zeros in the dataset. Thus, the effective input dimension is 27.

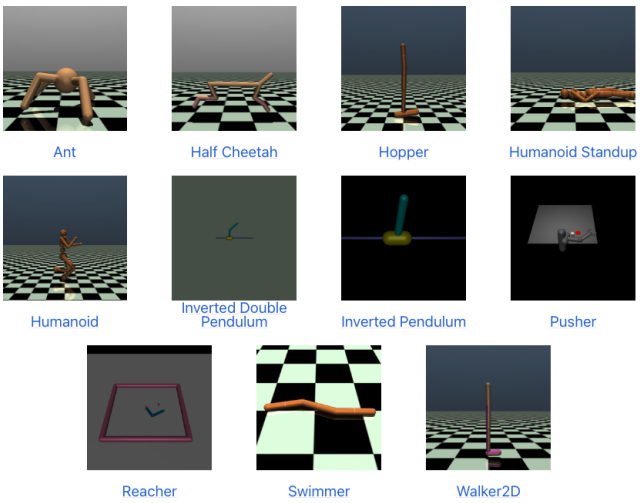


Figure 8: Screenshot of various MuJoCo environments (Towers et al., 2024).

All environments introduce stochasticity by perturbing a fixed initial state with Gaussian noise. Their state spaces combine positions of body and joint with corresponding velocities. Control is achieved

by applying joint torques, which serve as the actions. Expert datasets are generated by first training policies through online reinforcement learning (Fu et al., 2020; Gallouédec et al., 2024) until high performance, then executing these policies to produce trajectories of states $\mathbf{x}_i$ and actions $\mathbf{y}_i$. Here, $\mathbf{x}_i$ encodes robot positions, joint angles, velocities, and angular velocities, while $\mathbf{y}_i$ denotes the applied joint torques.

An episode of expert demonstration has a length of 50 for Reacher, and it has a length of 1,000 for all other environments. Thus, by taking 20,000 data points from each expert dataset, the regression model learns from at least 20 complete trajectories to clone the expert's behavior. For evaluation, we retain a subset of the full validation dataset, keeping the number of data points at 20% of the training data size. For small datasets (1K) used in Figures 6, 13 and 14, the test datasets contain 1,000 unseen samples.

There is no absolute threshold for what constitutes "low-data" versus "high-data," as this depends on problem complexity and data structure. In our experiments, we operationally define this distinction using datasets of 1,000 samples versus 20,000 samples—a 20-fold difference that produces qualitatively different generalization behavior. In the low-data regime, the training set provides sparse coverage of the true data manifold, introducing sampling artifacts such as spurious correlations and outlier effects that do not reflect the true underlying distribution. Models trained on such small datasets, particularly without sufficient regularization, tend to memorize these sample-specific patterns rather than learning generalizable structure. In the high-data regime with 20,000 samples, the training set provides denser coverage, the effect of sampling artifacts diminishes, and the empirical distribution more closely approximates the true underlying distribution.

Table 4: All hyperparameter settings involved for experiments on MuJoCo datasets. Each figure employs a subset of possible hyperparameter combinations.

	Hyperparameter	**Value**
Model Architecture	Number of hidden layers	$\{1, 2, 3, 4, 5\}$
	Hidden layer dimension	$\{64, 128, 256, 512, 1024\}$
	Activation function	ReLU
	Number of linear projection layer ($\mathbf{W}$)	1
Training	Epochs	3×10^5 (20K-datasets) 5×10^6 (1K-datasets)
	Batch size	4096 (20K-datasets) 1000 (1K-datasets)
	Optimizer	SGD
	Learning rate	1×10^{-2}
	Weight decay	$\{0, 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}, 7 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$, Reacher $\{0, 10^{-5}, 1/3/5/7 \times 10^{-4}, 1/3/5/7 \times 10^{-3}, 10^{-2}, 3 \times 10^{-2}\}$, Otherwise
	Seed	0
	Compute resources	NVIDIA A100 40GB
	Number of CPU compute workers	4
	Requested compute memory	16 GB
	Average training time per model	20 hours

Table 4 summarizes all model hyperparameters and experimental settings for MuJoCo datasets. A subset of possible hyperparameter combinations is used for each figure:

- Figure 1 plots the min-max normalized Test MSE as a function of the min-max normalized NRC1 values for the model architecture 3-256 (3 hidden layers and 256 hidden units) and all possible weight decay values.
- Figure 4 establishes the relationship between NRC1 and ID_H. Each subplot includes all weight decays listed in Table 4. And each weight decay is combined with 9 model architectures in {3-64, 3-128, 3-256, 3-512, 3-1024, 1-256, 2-256, 4-256, 5-256}.
- Figs. 6, 13 and 14 empirically reveal how generalization ability is affected by ID_H. We focus on a single model depth of 3, and vary the model width among $\{64, 128, 256, 512\}$. For each model architecture, we evaluate all possible weight decay values listed in Table 4.
- Figure 10 depict how intrinsic dimension evolves for each network layer. The model architecture is fixed at 3-256 and the title of each subplot annotates the weight decay value.
- Figure 11 and 12 follow the same experimental setup as Figs. 6, 13 and 14, but emphasize on the comparison between ID_H and ID_P.

- Figure 15 record NRC1 values along the training process. The model architecture is fixed to 3-256 for all datasets. And we show 10 weight decay values in $\{0, 0.0001, 0.0003, 0.0005, 0.0007, 0.001, 0.003, 0.005, 0.007, 0.01\}$.

A.2 MNIST/CIFAR10 experiments

The regression models for both the MNIST and CIFAR-10 tasks were trained across a spectrum of hyperparameters to thoroughly investigate the effects of architecture and regularization on the learned representations. The specific settings for model architecture, optimizer, and other training parameters are detailed in Table 5.

Table 5: All hyperparameter settings involved for experiments on MNIST and CIFAR-10 datasets.

	Hyperparameter	**Value**
Model Architecture	Number of hidden layers	3
	Hidden layer dimension	$\{32, 64, 128, 256, 512\}$
	Activation function	ReLU
Training	Epochs	200
	Batch size	64
	Optimizer	Adam
	Learning rate	$\{1 \times 10^{-3}, 5 \times 10^{-3}\}$
	Weight decay (MNIST)	$\{0, 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}, 7 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 7 \times 10^{-3}\}$
	Weight decay (CIFAR-10)	$\{0, 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}, 7 \times 10^{-4}, 10^{-3}\}$
	Seed	0
	Compute resources	NVIDIA A100 80GB
	Average training time per model	2 hours

We define noisy-target tasks as tasks where the regression targets contain information that causes models to learn patterns that do not generalize to unseen data. The term "noise" here does not refer to random measurement error or label corruption, but rather to information that varies between semantically similar examples due to instance-specific characteristics rather than reflecting the true underlying function.

The synthetic MNIST regression task represents a low-noise setting. The feature extractor is a CNN trained specifically on MNIST until achieving over 99% classification accuracy, creating a self-consistent target-generation pipeline. During training, the CNN learns to discard instance-specific variations (such as stroke thickness, slight rotations, or pixel-level noise) that are irrelevant for predicting the target label, retaining only the semantically meaningful information. As a result, the mapping from MNIST images to projected features is smooth and well-aligned with the data domain, enabling models to generalize effectively (Figs. 6 (d),(i)).

In contrast, the synthetic CIFAR-10 regression task is a noisy-target setting due to domain mismatch. CIFAR-10 images are processed through a ResNet-18 pretrained on ImageNet to extract features, which are then projected to 10-dimensional targets. The ImageNet encoder captures fine-grained visual details—texture patterns, color distributions, edge structures—optimized for ImageNet's 1,000 classes. When applied to CIFAR-10's 32×32 images, these features encode not only semantic content but also instance-specific characteristics: two images of cars may differ in color, lighting, rendering artifacts, or background elements, all of which significantly influence the extracted features. Models can fit these instance-specific components during training, achieving low training error. However, at test time, new images from the same semantic classes exhibit different instance-specific details. The learned mappings for these non-generalizable components do not transfer, resulting in poor generalization. This is evident in Figs. 6 (b),(g), where models achieve low training MSE but high test MSE, characteristic of overfitting to target noise.

A.3 Visual Control Experiments

Visual D4RL (Lu et al., 2023) provides offline reinforcement learning benchmarks derived from the DeepMind Control Suite (Tassa et al., 2018), where the observations consist of raw RGB pixel inputs rather than proprioceptive states. This introduces additional challenges compared to the state-based MuJoCo tasks described in Appendix A.1, as the model must extract task-relevant features directly from high-dimensional visual observations while discarding irrelevant visual details.

Environment. We use the Cheetah-run environment, where a planar bipedal robot must run forward as fast as possible. Following standard practice in visual continuous control (Yarats et al., 2022), we stack three consecutive $84 \times 84 \times 3$ RGB frames to form $84 \times 84 \times 9$ input tensors, providing temporal context for inferring velocities and dynamics. The target actions are 6-dimensional continuous control commands ($n = 6$). The environment constitutes a Partially Observed Markov Decision Process (POMDP), as individual frames do not fully capture the system state.

Datasets. Expert demonstration datasets are generated by first training policies via online reinforcement learning until high performance, then collecting trajectories of image observations and corresponding actions. We consider two data regimes: 20,000 and 80,000 training samples. Test datasets contain 20,000 samples.

Model architecture. The model consists of two components: a CNN image encoder and an MLP policy network. The encoder contains four convolutional layers, all with 32 output channels and 3×3 kernels (stride 2 for the first layer, stride 1 for the remaining three), each followed by ReLU activation. Input images are normalized to $[-0.5, 0.5]$, and the encoder outputs a flattened representation of dimension $32 \times 35 \times 35 = 39{,}200$. This representation is passed through a trunk consisting of a linear projection to 50 dimensions, followed by layer normalization and a Tanh activation. The policy head is a 3-hidden-layer MLP with hidden dimension 256 and ReLU activations, followed by a final linear projection layer $\mathbf{W}$ that maps to the 6-dimensional action space. During evaluation, actions are squashed through a $\tanh$ nonlinearity to enforce the $[-1, 1]$ action bounds. The last-layer features $\mathbf{h}$ used for intrinsic dimension and NRC1 analysis correspond to the output of the 3-layer MLP policy head (before the final linear projection $\mathbf{W}$).

Training. Models are trained using the Adam optimizer with a fixed learning rate of 3×10^{-4}. The maximum number of training steps is set to $10\times$ the training data size (e.g., 8×10^5 steps for the 80K dataset). Weight decay is applied to the network and is varied across a wide range of values: $\lambda_{\text{WD}} \in \{0,\ 10^{-5},\ 3 \times 10^{-5},\ 5 \times 10^{-5},\ 7 \times 10^{-5},\ 10^{-4},\ 3 \times 10^{-4},\ 5 \times 10^{-4},\ 7 \times 10^{-4},\ 10^{-3},\ 3 \times 10^{-3},\ 5 \times 10^{-3},\ 7 \times 10^{-3},\ 10^{-2},\ 5 \times 10^{-2},\ 10^{-1}\}$.

Evaluation. In addition to train and test MSE, we evaluate the learned policies (of 256 hidden dimensions) by executing each of them in the DeepMind Control Suite simulator for $N_{\text{episodes}} = 100$ episodes. Policy performance is reported as a normalized score $\in [0, 100]$ averaged over N_{episodes} evaluations, obtained by dividing the raw evaluation score by 10 (Tarasov et al., 2023).

B Comparison between Regression and Classification

Previous work on manifold learning for neural classification has demonstrated that the intrinsic dimension of the last hidden layer is negatively correlated with generalization ability. In particular, models achieving lower intrinsic dimension in the penultimate layer were found to exhibit superior test accuracy, with the lowest intrinsic dimension-model attaining the highest top-5 accuracy, see Section 3.2 and Figure 4 in Ansuini et al. (2019). Additionally, Papyan et al. (2020) connect neural collapse to robust decision boundaries, Galanti et al. (2021) demonstrate that collapse patterns improve few-shot and transfer learning, and Li et al. (2022) show the degree of collapse in downstream representations strongly predicts transfer accuracy. Complementing these empirical results, there are also theoretical results showing the benefits of neural collapse for classification (Gao et al., 2023; Wang & Palmer, 2023).

In regression, however, our findings indicate a more nuanced picture. We demonstrated the existence of a "soft" threshold at ID_Y, which delineates distinct generalization regimes. In the under-compressed regime with low-data tasks and high-noise tasks, lower ID_H is associated with better generalization, consistent with the monotonic complexity-performance paradigm observed in classification. However, in the over-compressed regime and in the under-compressed regime with high-data tasks and low-noise tasks, the opposite holds: higher ID_H is associated with better generalization, a phenomenon absent in classification tasks. Thus, in regression, generalization performance depends non-monotonically on the relationship between the learned feature manifold and the intrinsic dimension of the targets.

C Intrinsic Dimension Estimation

The intrinsic dimension (ID) of a dataset is the minimum number of coordinates needed to represent the data faithfully. If data points lie on or near a d-dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^D$ with $d \ll D$, then d is the intrinsic dimension. For example, a circle in 3D space has $d = 1$ and a sphere surface in 10D has $d = 2$.

A critical distinction exists between PCA dimensionality and intrinsic dimension. Consider a 1D spiral embedded in $\mathbb{R}^{10}$ parameterized by t. The spiral winds through space with substantial variance across all 10 axes, requiring multiple PCA components to capture the signal. Yet, the intrinsic dimension is exactly $d = 1$, because specifying a single scalar value, such as the arc length from the origin, is sufficient to uniquely locate any point on the curve. This illustrates that curved or folded manifolds can require many linear directions to approximate while having low intrinsic dimension. Figure 2 demonstrates this for neural regression: collapsed features lie near a 2D linear subspace (yellow plane) yet occupy a nonlinear 1D manifold within it.

Our main analysis uses the 2-NN estimator (Appendix C.1). In Appendix C.2, we verify that our conclusions are robust to this choice by comparing it with two widely used alternatives, MLE (Levina & Bickel, 2004) and PHD (Birdal et al., 2021).

C.1 The 2-NN Estimator

We estimate ID using the 2-NN estimator (Facco et al., 2017), which exploits a fundamental geometric property: in a d-dimensional space, the probability of finding neighbors within a given distance scales with dimension d. The key insight is to consider not absolute distances, but the ratio $\mu = r_2/r_1 \geq 1$ of the second to first nearest-neighbor distances. Remarkably, under the assumption of locally uniform density (density approximately constant within the range of the second neighbor), this ratio has a distribution that depends only on the intrinsic dimension d, with the local density completely canceling out. Specifically, the cumulative distribution function of μ is

$$F(\mu) = 1 - \mu^{-d}, \quad \mu \geq 1$$

This property makes the estimator robust to density variations, since we never need to estimate the density itself. Taking logarithms yields the linear relationship

$$\log(1 - F(\mu)) = -d \log \mu$$

The 2-NN algorithm estimates d by computing μ_i for each point, constructing the empirical CDF F_{emp}, and performing linear regression on the transformed coordinates $\{(\log \mu_i, -\log(1 - F_{\text{emp}}(\mu_i)))\}$. The requirement of local uniformity only within the second-neighbor distance is much weaker than global uniformity, making the estimator practical for real datasets with varying density and curvature.

Algorithm 1: 2-NN Intrinsic Dimension Estimation

Input: Dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^M$.
Output: Estimated intrinsic dimension $\hat{d}$.
for $i \leftarrow 1$ **to** M **do**
  Compute Euclidean distances to the first and second nearest neighbors, $r_1(\mathbf{x}_i)$ and $r_2(\mathbf{x}_i)$;
  Compute ratio $\mu_i \leftarrow r_2(\mathbf{x}_i)/r_1(\mathbf{x}_i)$;
Sort the ratios such that $\mu_{\sigma(1)} \leq \mu_{\sigma(2)} \leq \cdots \leq \mu_{\sigma(M)}$;
for $i \leftarrow 1$ **to** M **do**
  Assign empirical CDF value $F_{\text{emp}}(\mu_{\sigma(i)}) \leftarrow \frac{i}{M}$;
Construct the coordinate set for regression:

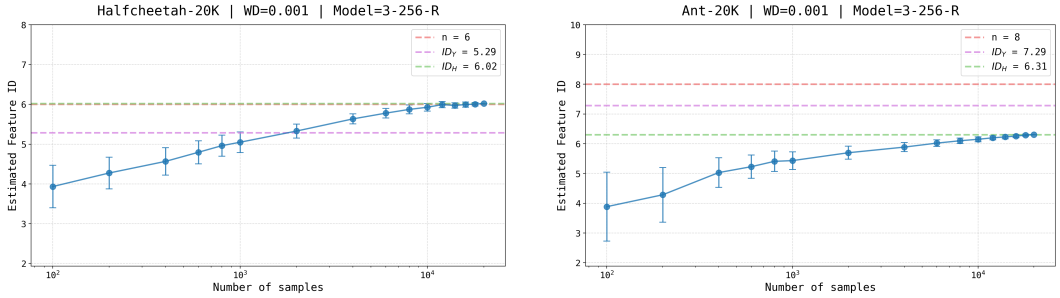
$$\mathcal{S} \leftarrow \left\{ \left(\log \mu_{\sigma(i)},\ -\log\big(1 - F_{\text{emp}}(\mu_{\sigma(i)})\big) \right) \right\}_{i=1}^{M-1}$$

Fit a line through the origin to $\mathcal{S}$ using least squares;
return Slope of the fitted line ($\hat{d}$);

Robustness of the 2-NN estimator. The robustness of the 2-NN estimator is well justified in the literature and has been widely adopted across various domains (Facco et al., 2017; Ansuini et al.,

2019; Basile et al., 2025). It features: (1) the capability to capture intrinsic dimension on nonlinear manifolds; (2) fast computation; and (3) hyperparameter-free estimation. Moreover, as suggested by Facco et al. (2017), we follow the standard block analysis procedure to evaluate the robustness of our ID estimates. The motivation behind block analysis is to distinguish true manifold structure from noise: intrinsic dimensions corresponding to meaningful geometry should remain stable across sample scales, whereas noise-driven dimensions diminish under subsampling. Concretely, we repeatedly estimate ID on progressively smaller random subsets of the data and observe how the estimate changes; a stable plateau indicates genuine intrinsic structure rather than a sampling artifact.

We demonstrate block analyses for two representative cases in Figure 9: (a) non-collapsed Halfcheetah last-layer features and (b) collapsed Ant last-layer features. Block sizes range from 100 to 20,000, and for each block size, we subsample 100 times from the training dataset and report the mean and standard deviation of the estimated ID. In both cases, we observe that the ID estimate converges to a stable plateau once the block size exceeds approximately 10,000 samples, with diminishing variance as the block size increases. This confirms that the reported ID values reflect true manifold structure rather than finite-sample noise, and that our default subsample size of 18,000 lies well within the stable regime.



(a) Non-collapsed Halfcheetah features.

(b) Collapsed Ant features.

Figure 9: Block analysis of the 2-NN intrinsic dimension estimator. Each point shows the mean estimated ID over 100 random subsamples of the given block size. Both cases exhibit a stable plateau for block sizes above approximately 10,000, confirming the reliability of the ID estimates at our default subsample size of 18,000.

C.2 Robustness to the Choice of Estimator

To verify that our conclusions do not hinge on the 2-NN estimator, we repeat the ID estimation with two widely used alternatives. The maximum likelihood estimator (MLE) of Levina & Bickel (2004) is, like 2-NN, a local estimator. Around each sample, it models the number of neighbors falling within a growing radius as a Poisson process whose rate scales with the radius according to the intrinsic dimension, assuming that the density is approximately constant within the neighborhood. Maximizing the resulting likelihood yields a closed-form local dimension estimate computed from the distances to the sample's k nearest neighbors, and the global estimate averages these local estimates over all samples. Compared with 2-NN, which uses only the first two neighbors, MLE pools information from a larger neighborhood, trading some locality for lower variance at the cost of introducing the neighborhood size k as a hyperparameter.

The persistent homology dimension (PHD) of Birdal et al. (2021) instead takes a global, topological view of the point cloud. It computes the total lifetime of the 0-dimensional persistent homology, which coincides with the total edge length of the minimum spanning tree over the samples. For data lying on a d-dimensional manifold, this quantity grows with the number of samples according to a power law whose exponent is determined by d, so PHD estimates d by fitting this growth rate across subsamples of increasing size. Because it depends on the connectivity of the entire point cloud rather than on individual neighborhoods, PHD complements the two local estimators. It is also differentiable with respect to the sample locations, which allows it to be used as a training objective. Together, the three estimators span two families: local neighborhood statistics (2-NN, MLE) and global topological geometry (PHD).

Table 6 reports $\mathrm{ID_H}$ under all three estimators along the weight-decay sweeps for Halfcheetah and Cheetah-run (20K training samples each), together with the estimator-independent quantities (test MSE, NRC1, and, for Cheetah-run, policy return) and each estimator's estimate of $\mathrm{ID_Y}$. Two observations show that our conclusions are not sensitive to the choice of estimator.

First, the three estimators agree closely in absolute value. Relative to 2-NN, the mean absolute deviation in $\mathrm{ID_H}$ along a sweep is 0.25–0.50 and the largest single deviation is 1.13, small compared with the range of roughly 4.5–6.3 over which $\mathrm{ID_H}$ varies along each sweep. The estimate of $\mathrm{ID_Y}$ shifts by at most 0.29. Since test MSE, return, and NRC1 do not depend on the estimator, estimators that agree this closely trace out the same relationships between $\mathrm{ID_H}$, NRC1, and performance that underlie Sections 5 and 6.

Second, and more fundamentally, our conclusions rely on the ordering of models along $\mathrm{ID_H}$ and on where $\mathrm{ID_H}$ crosses $\mathrm{ID_Y}$, rather than on absolute ID values. The ordering is preserved across estimators, apart from swaps among near-tied non-collapsed models at the smallest weight decays. The regime boundary is also preserved: on Halfcheetah, all three estimators place the transition from $\mathrm{ID_H} > \mathrm{ID_Y}$ to $\mathrm{ID_H} < \mathrm{ID_Y}$ between $\lambda_{\mathrm{WD}} = 10^{-3}$ and 5×10^{-3}; on Cheetah-run, 2-NN and MLE place it between 5×10^{-4} and 10^{-3}, while PHD places it one step earlier because the $\lambda_{\mathrm{WD}} = 5 \times 10^{-4}$ model falls marginally below $\mathrm{ID_Y}$ (6.00 vs. 6.03). In particular, the best-performing Cheetah-run model ($\lambda_{\mathrm{WD}} = 10^{-3}$; lowest test MSE and highest return) lies next to the $\mathrm{ID_H} \approx \mathrm{ID_Y}$ transition under every estimator.

Table 6: Intrinsic dimension estimated by 2-NN, MLE, and PHD along weight-decay sweeps. Columns correspond to weight decay λ_{WD}. Each estimator row reports $\mathrm{ID_H}$ for every model. The last column reports the corresponding estimate of $\mathrm{ID_Y}$, computed on the 20K training set, which is hence slightly different from the 80K value reported in Section 4 for Cheetah-run. Test MSE, return, and NRC1 do not depend on the estimator. Bold numbers mark the best value of each performance metric across the sweep.

Halfcheetah (20K)

	λ_{WD}							
	0	1E-5	1E-4	5E-4	1E-3	5E-3	1E-2	$\mathrm{ID_Y}$
Test MSE	0.281	0.271	**0.265**	0.293	0.352	0.574	0.703	–
NRC1	0.251	0.235	0.166	0.059	0.034	0.005	0.001	–
2-NN	7.89	7.97	7.86	6.96	6.17	3.89	3.50	5.27
MLE	7.78	7.77	7.84	6.75	5.98	4.40	4.02	5.40
PHD	7.28	7.23	7.29	6.28	5.61	4.06	3.69	5.18

Cheetah-run (20K)

	λ_{WD}								
	0	1E-5	5E-5	1E-4	5E-4	1E-3	5E-3	1E-2	$\mathrm{ID_Y}$
Test MSE	3.352	3.241	3.093	3.001	2.377	**2.210**	2.435	2.725	–
Return	28.9	40.1	30.8	33.0	34.3	**55.4**	27.8	14.4	–
NRC1	0.645	0.665	0.673	0.648	0.021	0.023	0.014	0.007	–
2-NN	9.79	9.85	10.04	10.00	6.43	5.45	4.79	3.76	6.14
MLE	10.32	10.69	10.96	11.13	6.47	5.44	4.82	4.06	6.43
PHD	9.10	9.40	9.67	9.67	6.00	4.94	4.31	3.79	6.03

C.3 Intrinsic Dimension versus Spectral Metrics

Metrics computed from the eigenvalue spectrum of the feature covariance are popular proxies for representation quality. In deep reinforcement learning, the effective rank of the features is widely used to diagnose implicit under-parameterization and loss of plasticity (Kumar et al., 2021; Lyle et al., 2022). In self-supervised learning, RankMe (Garrido et al., 2023), an entropy-based effective rank of the feature matrix, and the power-law decay of the eigenspectrum (Ghosh et al., 2022) are used to select encoders with diverse features. NRC1 and the explained variance ratio (EVR) of the leading principal components belong to the same family. In neural regression collapse, however, such metrics

saturate: Andriopoulos et al. (2024, Figure 3) show that, after a short period of training, the variance of the last-layer features concentrates on the first n principal components, with little or no variance in the remaining ones. From then on, a spectral metric can change only through the relative magnitudes of the top-n eigenvalues and is blind to the nonlinear structure of the features within the collapsed subspace. Spectral metrics therefore indicate *whether* collapse has occurred, but not the geometry *within* the collapsed subspace, which, according to our analysis, is where generalization is decided.

This limitation is intrinsic to any metric that depends on the data only through its covariance. Consider X uniformly distributed on the unit circle in $\mathbb{R}^2$ and $Y \sim \mathcal{N}(0, \frac{1}{2}\mathbf{I}_2)$. Writing $X = (\cos\theta, \sin\theta)$ with θ uniform on $[0, 2\pi)$, both distributions are centered and satisfy $\mathbb{E}[XX^\top] = \frac{1}{2}\mathbf{I}_2 = \mathbb{E}[YY^\top]$. Their spectra are identical, so every covariance-based metric, including effective rank, assigns them the same value (for instance, RankMe equals 2 in both cases). Their intrinsic dimensions, however, differ: $\mathrm{ID}(X) = 1$ and $\mathrm{ID}(Y) = 2$. This is precisely the configuration of collapsed regression features, which lie in a low-dimensional linear subspace while occupying a curved, lower-dimensional manifold within it (Figures 2 and 5).

Table 7 confirms this empirically by additionally comparing RankMe with ID_H along the weight-decay sweep on Cheetah-run. RankMe saturates exactly where NRC1 does: from $\lambda_\mathrm{WD} = 5 \times 10^{-4}$ onward, it stays within 3.16–3.63, whereas ID_H continues to decrease from 6.43 to 3.76, crossing $\mathrm{ID}_\mathrm{Y} = 6.14$ into the over-compressed regime. Yet this saturated range is where model quality is decided: test MSE traces its U-shape and return varies from 14.4 to 55.4. Selecting the model with the highest rank, as RankMe prescribes, would favor the non-collapsed models ($\lambda_\mathrm{WD} \leq 10^{-4}$), all of which have higher test MSE and lower return than the best model. In contrast, the two models closest to ID_Y attain the two lowest test MSEs and include the best-performing policy.

Table 7: Spectral rank (RankMe) versus intrinsic dimension along the weight-decay sweep on Cheetah-run (20K). Columns correspond to weight decay λ_WD. The RankMe and ID rows are computed on the last-layer features of each model and the last column instead evaluates both metrics on the targets Y. Bold numbers mark the best value of each performance metric across the sweep.

	λ_WD								
	0	1E-5	5E-5	1E-4	5E-4	1E-3	5E-3	1E-2	Y
Test MSE	3.352	3.241	3.093	3.001	2.377	**2.210**	2.435	2.725	–
Return	28.9	40.1	30.8	33.0	34.3	**55.4**	27.8	14.4	–
NRC1	0.645	0.665	0.673	0.648	0.021	0.023	0.014	0.007	–
RankMe	73.10	76.76	77.25	69.89	3.54	3.63	3.57	3.16	3.53
ID (2-NN)	9.79	9.85	10.04	10.00	6.43	5.45	4.79	3.76	6.14

D Evolution of Intrinsic Dimension during Training

To further understand the behavior of collapsed models and their counterparts, we track the evolution of the intrinsic dimension throughout training and provide insights. Figure 10 provides illustrative examples for a collapsed and a non-collapsed model:

- For both the collapsed and non-collapsed models, the intrinsic dimension of the last-layer features generally decreases over training and stabilizes at convergence.
- For the collapsed model, the deeper the layer in the network, the lower the intrinsic dimension at the end of training. ReLU activations cause a mild reduction in intrinsic dimension in comparison with the reduction in intrinsic dimension between consecutive layers (ignoring ReLU). Notably, the final intrinsic dimension of the output layer, which gives the actual vector-valued predictions, can be significantly lower than ID_H.
- For non-collapsed models, we usually see — but not always — ID_H decrease monotonically as we move from shallow to deep layers. Furthermore, we observe that during training, the intrinsic dimension of the output layer hugs the intrinsic dimension of the targets. Thus, tracking the intrinsic dimension of the output layer provides yet another criterion for discriminating between collapsed and non-collapsed models; see Appendix E.

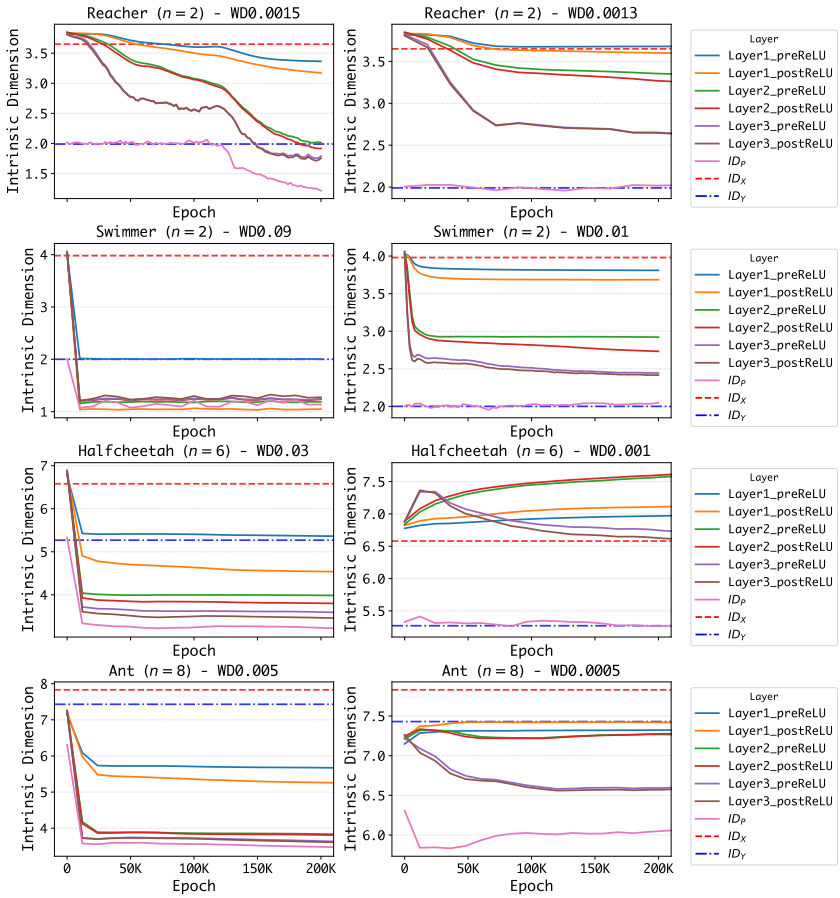


Figure 10: Intrinsic dimension of input, output, and hidden layers over training epochs for a collapsed (left) and a non-collapsed model (right) for MuJoCo datasets. Each subfigure shows the evolution of intrinsic dimension across layers with red dashed, blue dash-dotted, and pink lines denoting the intrinsic dimension of inputs, targets, and predicted outputs, respectively.

E Intrinsic Dimension and Output layer

We consider here the intrinsic dimension of the outputs (equivalently, the final predictions), $\mathrm{ID_P}$. We will see that here too the relationship between intrinsic dimension and generalization exhibits key differences between classification and regression.

With respect to the output layer, a structural constraint arises from the classification setting. Specifically, the intrinsic dimension of the output layer necessarily satisfies

$$\log_2 C \leq \mathrm{ID_P} \leq C,$$

where C is the number of classes. Empirical results consistently show $\mathrm{ID_P}$ equals the lower bound of this inequality if the model generalizes well. We refer the reader to the discussion in Section 3.1 of Ansuini et al. (2019). Conversely, saturation of the upper bound, i.e., $\mathrm{ID_P} \approx C$, is associated with poor generalization performance, suggesting that maximal output layer dimensionality corresponds to overfitting in classification tasks, see Section 3.5 in Ansuini et al. (2019).

In contrast, for neural multivariate regression, the structure of the output leads to the trivial bound

$$1 \leq \mathrm{ID_P} \leq n,$$

where n is the number of output variates. Interestingly, our empirical findings reveal a departure from the classification setting. As shown in the middle column in Figures 11-12, when the test MSE is low, the intrinsic dimension of the output layer, $\mathrm{ID_P}$ satisfies $\mathrm{ID_P} \approx \mathrm{ID_Y}$, which can be close to n, saturating the upper bound of the inequality above. Notably, unlike in classification, this saturation

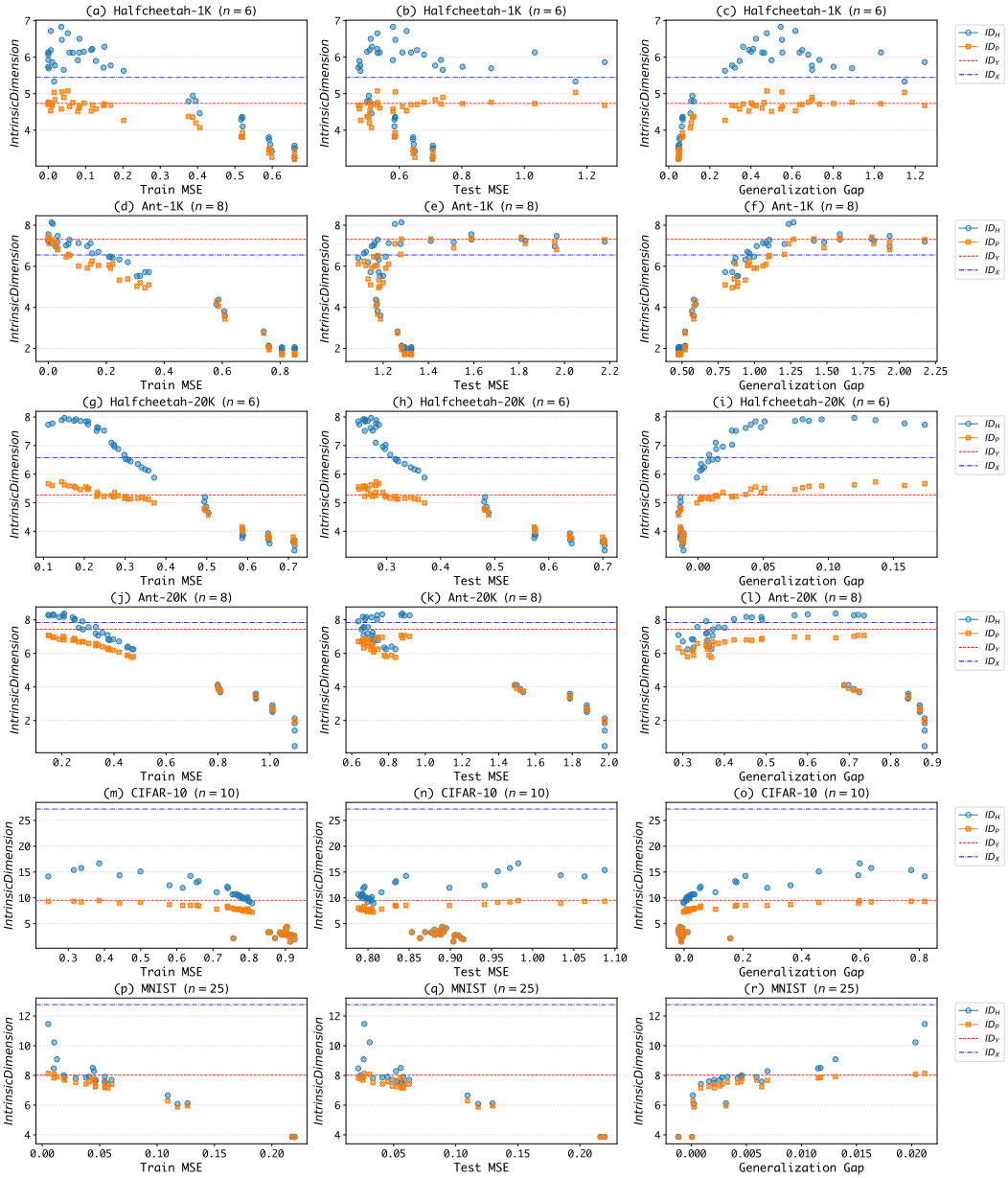


Figure 11: Comparison between $\mathrm{ID_H}$ and $\mathrm{ID_P}$ for Halfcheetah, Ant, CIFAR-10, and MNIST datasets

is associated with improved test performance. By contrast, when $\mathrm{ID_P}$ falls below $\mathrm{ID_Y}$, test MSE performance deteriorates, see Figures 11-12.

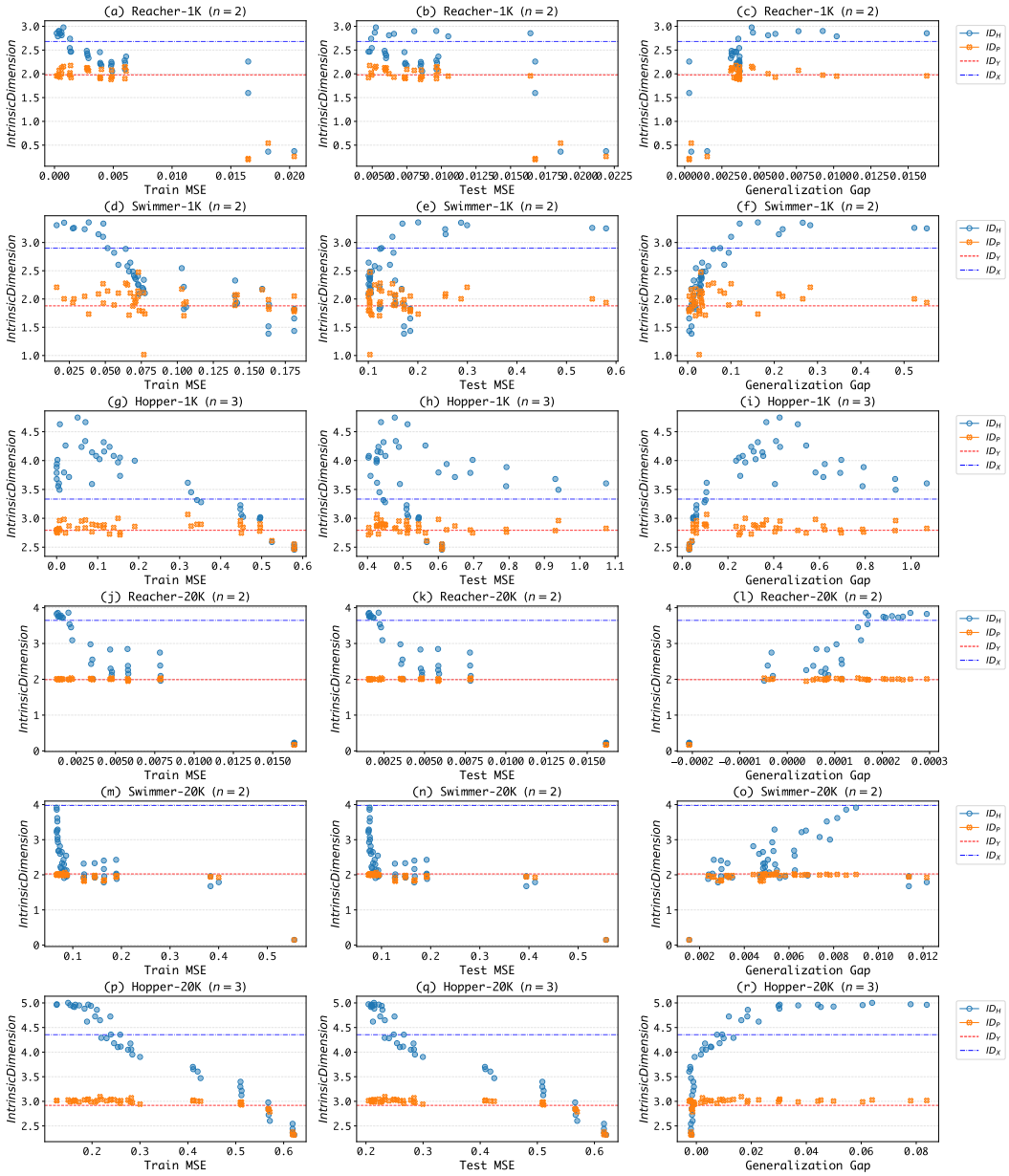


Figure 12: Comparison between $\mathrm{ID_H}$ and $\mathrm{ID_P}$ for Reacher, Swimmer and Hopper datasets

F Additional Experiments on Generalization

This section lists additional results that complement the experiments in the main body for all considered datasets. Figure 13 and 14 shows how generalization power correlates with ID_{H}. Our key takeaways are summarized in Table 2.

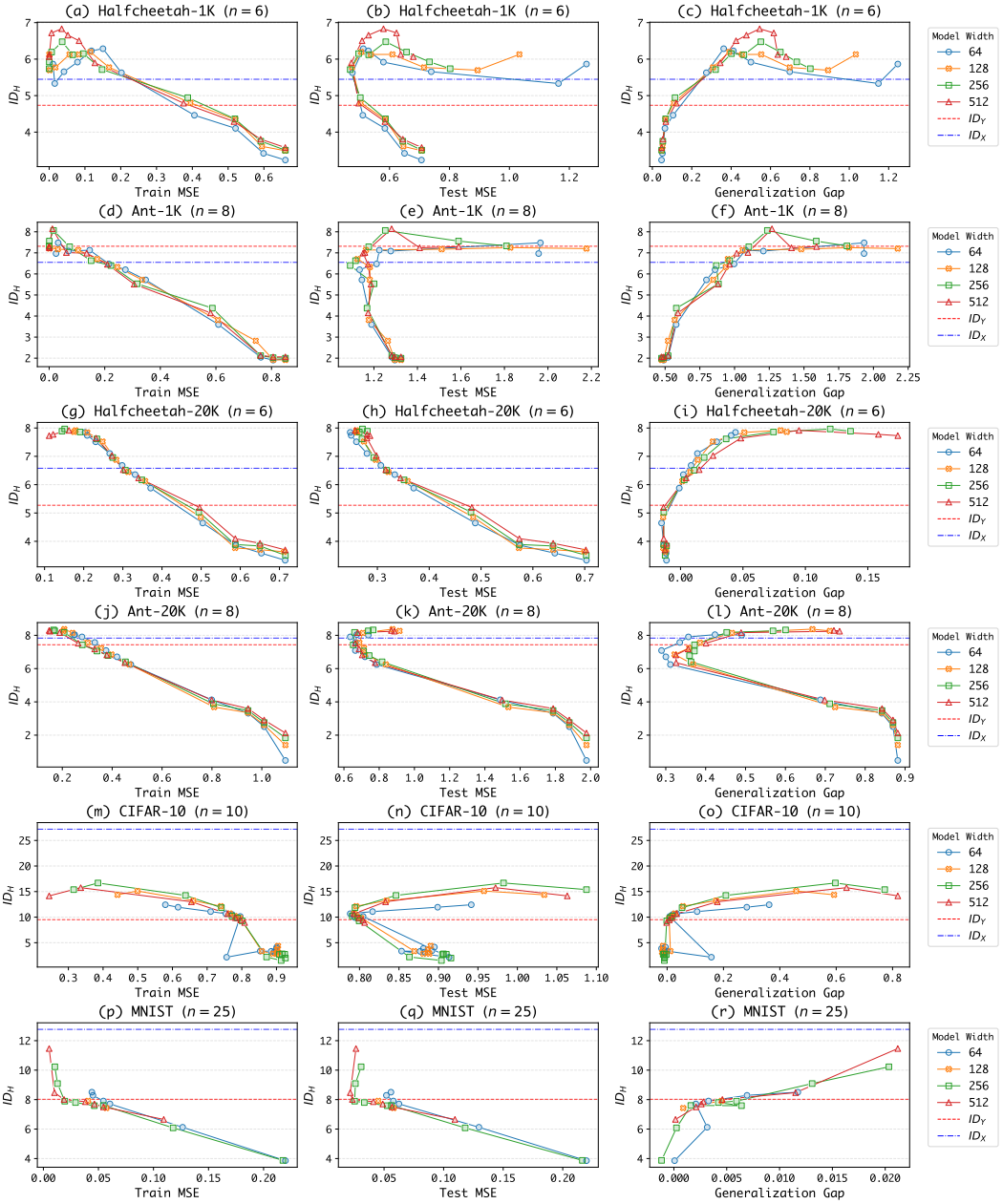


Figure 13: Generalization ability and Intrinsic Dimension for Halfcheetah, Ant, CIFAR-10, and MNIST datasets.

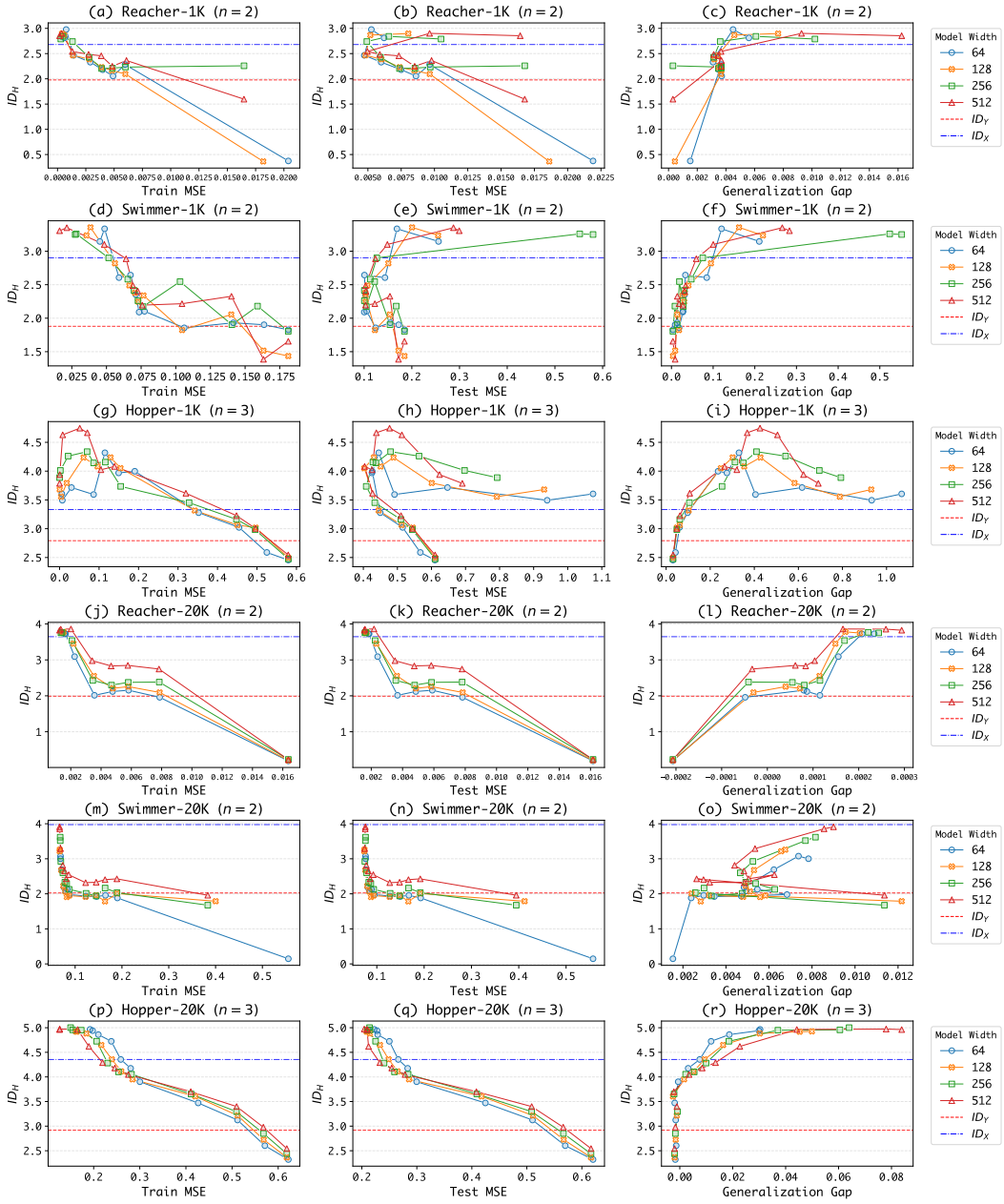


Figure 14: Generalization ability and Intrinsic Dimension for Reacher, Swimmer, and Hopper datasets.

G How does regularization affect NRC and Intrinsic Dimension?

G.1 Weight Decay

Weight decay is a canonical and widely adopted model regularization technique for preventing large models from overfitting to data. Real-world applications include but are not limited to (1) regularizing transformer backbones for large language models (Wolf et al., 2020) and robotic generalist policies (Chen et al., 2021a); (2) participating, by default, in the common Pytorch implementation of AdamW optimizer (Loshchilov & Hutter, 2019; PyTorch Contributors, 2025) with `weight_decay=0.01`; (3) improving sample efficiency of online reinforcement learning algorithms (Liu et al., 2021; Li et al., 2023b); (4) and facilitating research in model plasticity in deep learning (Lyle et al., 2023; Nauman et al., 2024; Ceron et al., 2024).

Figure 15 investigates NRC1 for values of the weight decay parameter λ_{WD}. We see that when λ_{WD} is zero or small, there is no neural regression collapse; but if we increase the weight decay, the NRC1 geometric structure quickly emerges during training. This matches the empirical observation in Andriopoulos et al. (2024).

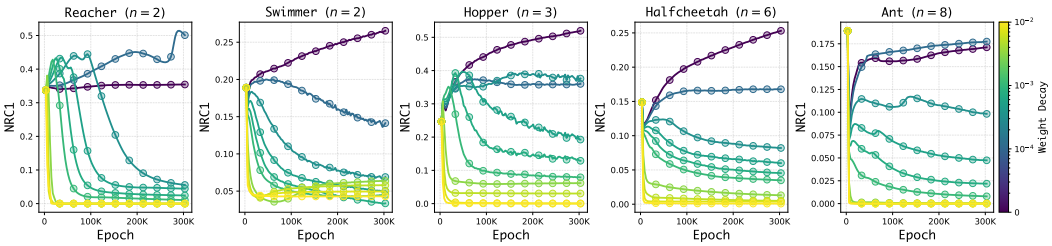


Figure 15: NRC1 decreases as weight decay becomes stronger, leading to model collapse.

G.2 Dropout Regularization

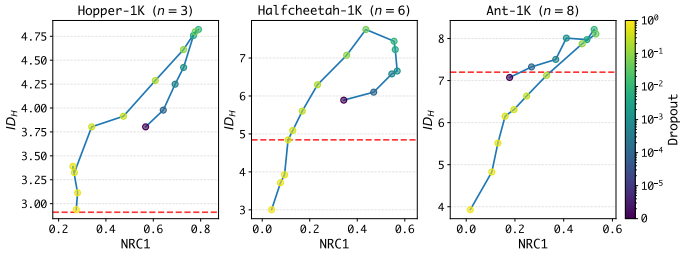
Modern implementations, across deep learning domains, continue to rely on dropout regularization (Srivastava et al., 2014) to mitigate overfitting, underscoring its persistent role in practical training, such as computer vision (Dosovitskiy et al., 2021), NLP (Devlin et al., 2019; Wolf et al., 2020), and reinforcement learning (Hiraoka et al., 2022). We empirically analyze how dropout regularization influences neural regression collapse by varying its strength from a large range ($\in \{0, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$). *No* weight decay is applied. In this section, datasets include Hopper, Halfcheetah, and Ant with two sizes, 1K (Figure 16) and 20K (Figure 17). The horizontal red dashed line represents ID_{Y}.

Figure 16(a) and Figure 17(a) show the relationship between ID_{H} and NRC1. We first confirm the same conclusion as made in Section 5, despite the new regularization. ID_{H} provides a more refined geometric structure than NRC1. Collapsed models with near-zero NRC1 values have varying ID_{H} below or in the vicinity of $\text{ID}_{\text{Y}} < n$, while non-collapsed models with non-trivial NRC1 maintain their ID_{H} to be above ID_{Y} and to be positively correlated with $NRC1$. Interestingly, dropout regularization differs from weight decay in that mild dropout (e.g., ≤ 0.01) can effectively prevent models from collapse by increasing both NRC1 and ID_{H}. This observation sheds light on a geometric interpretation of the effectiveness of mild dropout in reinforcement learning as proposed by Hiraoka et al. (2022).

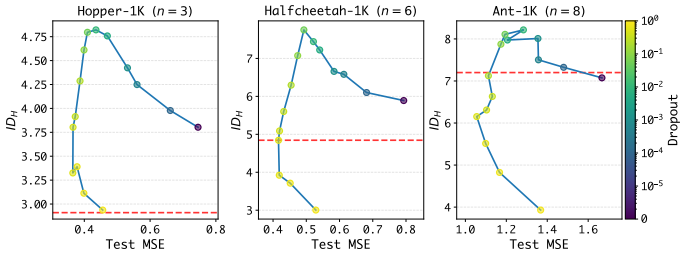
Figure 16(b) and Figure 17(b) show the relationship between ID_{H} and test MSE. The results again verify the three regimes discussed in Section 6. For both data sizes, models over-compress features when $\text{ID}_{\text{H}} < \text{ID}_{\text{Y}}$ and thus lead to increasing test MSE (and thus poor generalization). Then, for small datasets with 1K samples, $\text{ID}_{\text{H}} \approx \text{ID}_{\text{Y}}$ identifies the sweet spot where test MSE tends to be the lowest and exhibits the 'U-shape' plots. Note that models trained with Hopper datasets have not collapsed yet, so they only exhibit the upper part of the 'U-shape'. Finally, with more samples, e.g., 20K, $\text{ID}_{\text{H}} \gg \text{ID}_{\text{Y}}$ achieves the best generalization with under-compressed models.

In summary, dropout regularization offers an alternative approach to adjusting the degree of model collapse, while all conclusions drawn from the main body remain intact and inclusive. In addition,

mild dropout regularization is more effective than weight decay regularization in increasing NRC1 and $\mathrm{ID_H}$ metrics for the under-compressed regime.

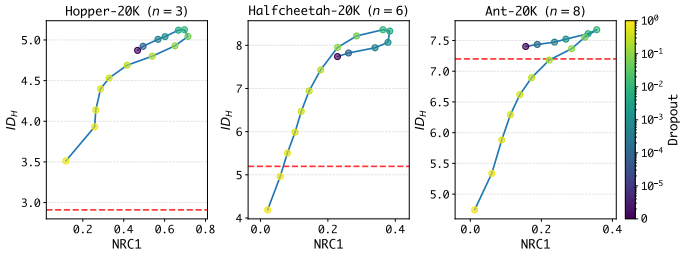


(a) NRC1 - $\mathrm{ID_H}$

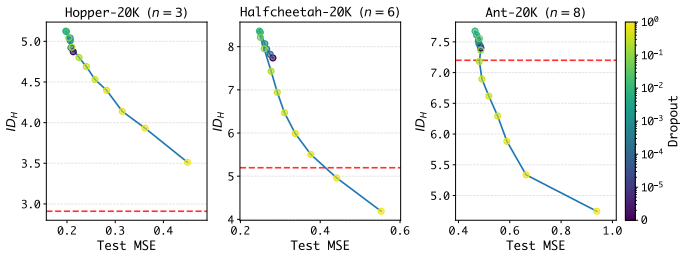


(b) Test MSE - $\mathrm{ID_H}$

Figure 16: Relationship between $\mathrm{ID_H}$ and NRC1 and Test MSE for Hopper-1K, Halfcheetah-1K, and Ant-1K datasets, when applying model dropout regularization. The horizontal red dashed line represents $\mathrm{ID_Y}$.



(a) NRC1 - $\mathrm{ID_H}$



(b) Test MSE - $\mathrm{ID_H}$

Figure 17: Varying model dropout regularization, relationship between $\mathrm{ID_H}$ and NRC1 and Test MSE for Hopper-20K, Halfcheetah-20K, and Ant-20K datasets. The horizontal red dashed line represents $\mathrm{ID_Y}$.

G.3 Model Depth

With mild weight decay regularization, we find that increasing model depth leads to smaller NRC1 and $\mathrm{ID_H}$ and thus to more collapsed features. In Figure 18, we examine the relationship between $\mathrm{ID_H}$

and NRC1 for Hopper-20K and Halfcheetah-20K datasets. We fix the model width to be 256 and vary the model depth ($\in \{2, 3, 4, 5\}$). For each model depth, we show five mild weight decay values: 0.0001, 0.0003, 0.0005, 0.0007, 0.001. The figure shows that increasing model depth gradually pushes points to the region on the bottom left, where both NRC1 and $\mathrm{ID_H}$ are small. For example, Halfcheetah-20K with a depth of 5 can result in collapsed models with $\mathrm{ID_H} < \mathrm{ID_Y}$.

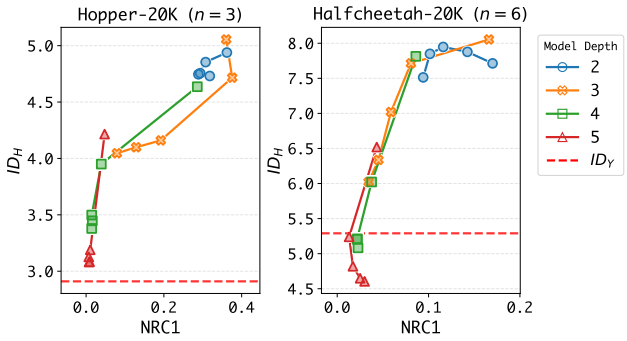


Figure 18: Relationship between $\mathrm{ID_H}$ and NRC1 for Hopper-20K and Halfcheetah-20K datasets, when varying model depth. All models have a hidden size of 256. The horizontal red dashed line represents $\mathrm{ID_Y}$.

H Empirical Analysis on More Challenging Tasks

We extend our empirical analysis to more challenging tasks with varying data sizes, increased intrinsic dimensions, and visual inputs.

Table 8: Overview of additional datasets employed in this section.

Dataset	**Data Size**	**Input Type**	**Input Dim** (D)	**Input ID** ($\mathrm{ID_X}$)	**Target Dim** (n)	**Target ID** ($\mathrm{ID_Y}$)
Humanoid	500,000	raw state	348	11.02	17	9.85
Relocate	500,000	raw state	39	6.90	30	19.82
Cheetah_run	80,000	RGB image	$84 \times 84 \times 9$[3]	8.96	6	6.00
Humanoid_walk	100,000	RGB image	$84 \times 84 \times 9$	9.41	21	14.84

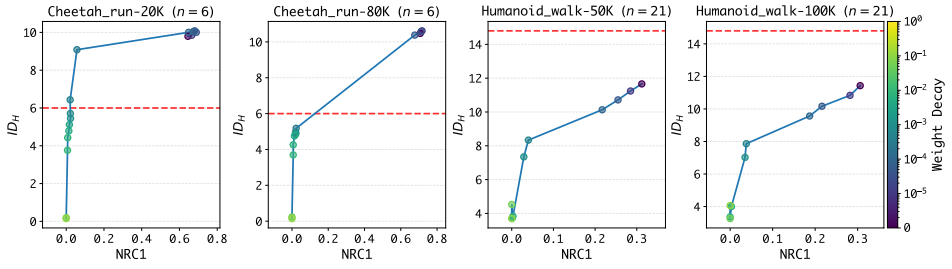
Humanoid (MuJoCo locomotion) The Humanoid dataset (Younis et al., 2024) is generated from the MuJoCo physics simulator, introduced in the main body and Appendix A.1. Each state consists of high-dimensional proprioceptive information, and the corresponding targets are the expert control torques applied at each joint. The goal is to enable the humanoid to run forward stably while maintaining balance, which is substantially more difficult than all previously considered MuJoCo tasks due to its high degrees of freedom and complex contact dynamics. Among all MuJoCo environments, Humanoid is widely regarded as the most challenging.

Relocate (Adroit manipulation) The Relocate dataset (Fu et al., 2020) comes from the Adroit suite (Rajeswaran et al., 2018) of dexterous manipulation tasks. Adroit uses a simulated 24 degrees of freedom (24-DoF) robotic hand, combined with an arm of up to 6-DoF, with rich contact and articulation dynamics. In the Relocate task, the state includes joint positions, velocities, hand pose information, and kinematic information about the ball and target. The action corresponds to the joint torques for the 24 actuators and to the arm movement. The objective is to grasp a small object and relocate it to a specified target position. It is a long-horizon manipulation task requiring precise coordination and contact control. Among the four Adroit tasks in the D4RL benchmark (Fu et al., 2020), Relocate is widely considered the most difficult due to its combination of dexterity, precision, and exploration complexity.

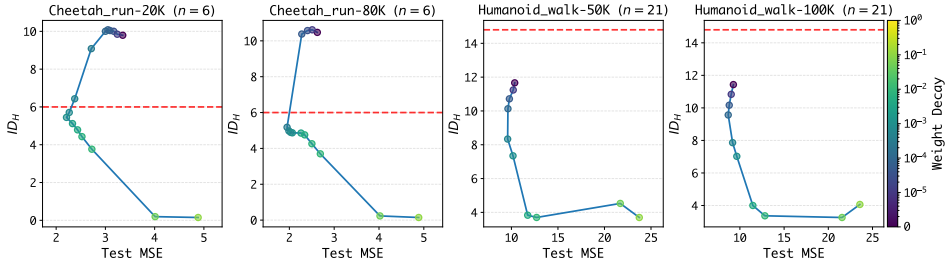
[3]Frame stack is commonly applied for visual control tasks. A single observation is of shape $84 \times 84 \times 3$. A frame stack of 3 is used in our experiments, resulting in visual inputs of dimension $84 \times 84 \times 9$.

Cheetah_run & Humanoid_walk (Visual continuous control) The Cheetah_run and Humanoid_walk datasets (Lu et al., 2023) are visual control benchmarks constructed from demonstrations generated in the DeepMind Control Suite (Tassa et al., 2018). Inputs consist of raw image observations (e.g., 84×84x3 RGB frames), and targets correspond to the continuous control commands. In this section, the Cheetah_run dataset consists of expert demonstrations, as is the case for all previous datasets, while the Humanoid_walk dataset contains some noisy suboptimal behavior in addition to the expert demonstrations ('medium-expert dataset'). This adds difficulty in extracting a good policy by imitating the dataset's behavior. We use a CNN image encoder (consisting of 4 `conv2d` layers and ReLU activation), followed by a 3-layer MLP policy network.

Visual control is particularly interesting and challenging because each frame provides only partial information about the system state, effectively forming a Partially Observed Markov Decision Process (POMDP) (Yarats et al., 2022). As a result, the model must infer underlying transition dynamics and identify salient visual features directly from high-dimensional pixel inputs. Within the visual D4RL tasks (Lu et al., 2023), Humanoid_walk is the most challenging due to the humanoid's instability and high-dimensional dynamics, whereas Cheetah_run is comparatively easier but still nontrivial.



(a) NRC1 - $\mathrm{ID_H}$



(b) Test MSE - $\mathrm{ID_H}$

Figure 19: Relationship between $\mathrm{ID_H}$ and NRC1 and Test MSE for Cheetah_run-20K, Cheetah_run-80K, Humanoid_walk-50K and Humanoid_walk-100K. The horizontal red dashed line represents $\mathrm{ID_Y}$.

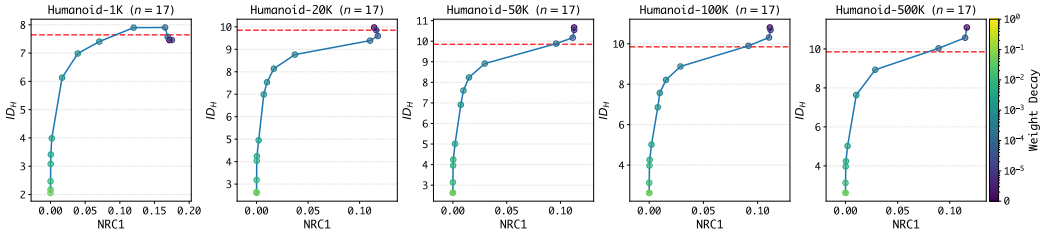
H.1 Visual Control Tasks

Figure 19(a) shows the relationship between $\mathrm{ID_H}$ and NRC1 for the visual control datasets with varying sizes. Consistent with the conclusions in Section 5, $\mathrm{ID_H}$ provides a more refined geometric structure than NRC1. Collapsed models with small NRC1 values have varying $\mathrm{ID_H}$ below or in the vicinity of $\mathrm{ID_Y} < n$, while non-collapsed models with non-trivial NRC1 maintain their $\mathrm{ID_H}$ to be above $\mathrm{ID_Y}$ and to be positively correlated with NRC1. Notably, for the most challenging Humanoid_walk task, which has a substantially higher $\mathrm{ID_Y}$ due to its complex high-dimensional dynamics, models trained with *zero* weight decay initially exhibit neural regression collapse with $\mathrm{ID_H} < \mathrm{ID_Y}$. This explains the extremely large test MSE observed for Humanoid_walk in Figure 19(b) and its negative correlation with $\mathrm{ID_H}$, which matches the over-compressed regime ($\mathrm{ID_H} < \mathrm{ID_Y}$) summarized in Table 2. In comparison, the two Cheetah_run datasets exhibit the 'U-shape' in the relationship between $\mathrm{ID_H}$ and test MSE. This emphasizes the sweet spot with $\mathrm{ID_H} \approx \mathrm{ID_Y}$, which approaches the lowest test MSE in noisy and challenging tasks.

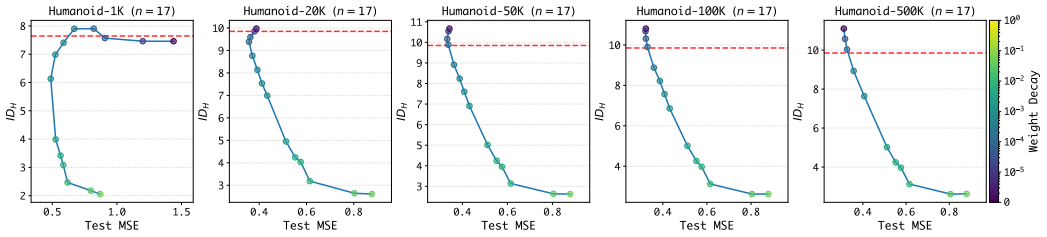
H.2 Relocate & Humanoid

Humanoid In Figure 20, we show 'NRC1 - $\mathrm{ID_H}$' and 'Test MSE - $\mathrm{ID_H}$' plots for Humanoid datasets with varying sizes. In Figure 20(a), we observe that $\mathrm{ID_H}$ provides a more refined geometric structure than NRC1. Collapsed models with near-zero NRC1 values have varying $\mathrm{ID_H}$ below or in the vicinity of $\mathrm{ID_Y} < n$, while non-collapsed models with non-trivial NRC1 maintain their $\mathrm{ID_H}$ to be above $\mathrm{ID_Y}$ and to be positively correlated with NRC1. Then, Figure 20(b) again verifies the three regimes discussed in Section 6. For all data sizes, models over-compress features when $\mathrm{ID_H} < \mathrm{ID_Y}$ and thus lead to increasing test MSE (and thus poor generalization). Then, for small datasets with 1-50K samples, $\mathrm{ID_H} \approx \mathrm{ID_Y}$ identifies the sweet spot where test MSE tends to be the lowest and exhibits the 'U-shape' plots. Finally, with more samples, e.g., 100-500K, $\mathrm{ID_H} > \mathrm{ID_Y}$ achieves the best generalization with under-compressed models.

Relocate For the Relocate task with varying data sizes, which have the highest $\mathrm{ID_Y}$ due to its complex task dynamics, models trained with *zero* weight decay initially exhibit a severe collapse with both NRC1 ≈ 0 and $\mathrm{ID_H} \ll \mathrm{ID_Y}$. Unsurprisingly, Test MSE remains large for all data sizes and decay values, and it also negatively correlates with $\mathrm{ID_H}$, uncovering the over-compressed regime.

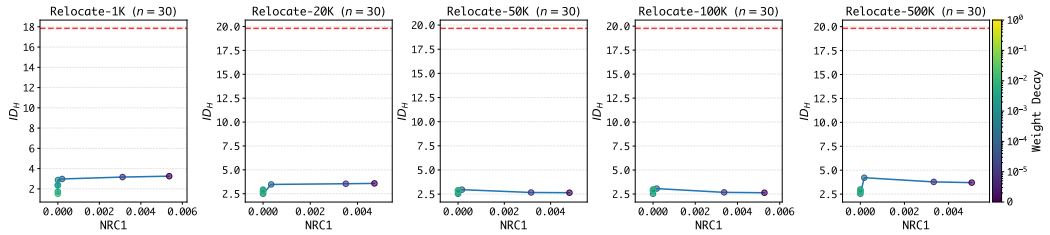


(a) NRC1 - $\mathrm{ID_H}$

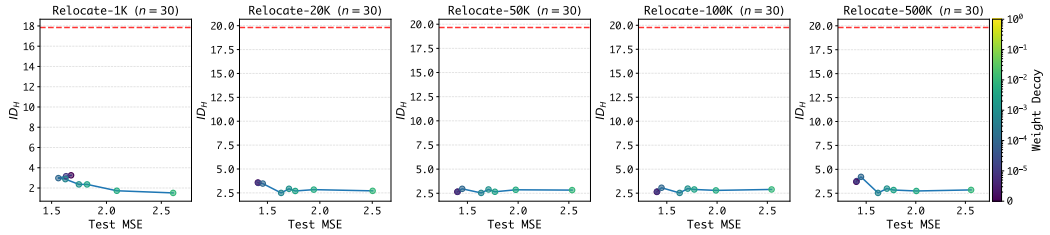


(b) Test MSE - $\mathrm{ID_H}$

Figure 20: Relationship between $\mathrm{ID_H}$ and NRC1 and Test MSE, for Humanoid locomotion task, when varying data size ($\in \{1\mathrm{K}, 20\mathrm{K}, 50\mathrm{K}, 100\mathrm{K}, 500\mathrm{K}\}$). The red dashed line represents $\mathrm{ID_Y}$.



(a) NRC1 - $\mathrm{ID_H}$



(b) Test MSE - $\mathrm{ID_H}$

Figure 21: Relationship between $\mathrm{ID_H}$ and NRC1 and Test MSE, for Relocate manipulation task, when varying data size ($\in \{1\text{K}, 20\text{K}, 50\text{K}, 100\text{K}, 500\text{K}\}$). The red dashed line represents $\mathrm{ID_Y}$.

I Mathematical Analysis of Regression Collapse

I.1 Why Weight Decay Leads to Collapse: Analysis via the Unconstrained Feature Model

In this section, we provide a theoretical explanation for why weight decay causes neural regression collapse through the lens of the Unconstrained Feature Model (UFM). The UFM is a simplified mathematical abstraction that does not capture all aspects of practical neural networks, but provides insight into the geometric mechanisms by which regularization constrains learned representations.

The UFM abstracts the neural regression problem by treating the feature extractor $h_\theta(\cdot)$ as producing arbitrary feature vectors $\mathbf{h}_i \in \mathbb{R}^d$ for each input $\mathbf{x}_i$, collected into a feature matrix $H \in \mathbb{R}^{d\times M}$. The final prediction is obtained via a linear map $W \in \mathbb{R}^{n\times d}$ and bias $\mathbf{b} \in \mathbb{R}^n$, giving predictions $\hat{Y} = WH + \mathbf{b}\mathbf{1}^\top$. The UFM objective is:

$$\min_{W,H,\mathbf{b}} \frac{1}{2M}\|WH + \mathbf{b}\mathbf{1}^\top - Y\|_F^2 + \frac{\lambda_W}{2}\|W\|_F^2 + \frac{\lambda_H}{2}\|H\|_F^2 \tag{2}$$

The model is "unconstrained" because H can be any matrix in $\mathbb{R}^{d\times M}$, unlike in actual neural networks, where H is constrained by the input data and network architecture. In the UFM, we typically have $d \gg n$, mirroring the overparameterized regime where feature dimension greatly exceeds target dimension.

The optimal bias is $\mathbf{b}^* = \bar{\mathbf{y}}$ where $\bar{\mathbf{y}} = \frac{1}{M}\sum_{i=1}^M \mathbf{y}_i$. Defining the centered target matrix $\tilde{Y} = Y - \bar{\mathbf{y}}\mathbf{1}^\top \in \mathbb{R}^{n\times M}$, the problem reduces to:

$$\min_{W,H} \frac{1}{2M}\|WH - \tilde{Y}\|_F^2 + \frac{\lambda_W}{2}\|W\|_F^2 + \frac{\lambda_H}{2}\|H\|_F^2 \tag{3}$$

We denote the empirical covariance by $\Sigma = \frac{1}{M}\tilde{Y}\tilde{Y}^\top \in \mathbb{R}^{n\times n}$, and assume Σ is full rank with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$. A key quantity is $c = \lambda_W\lambda_H$.

We first restate the characterization of global minimizers when regularization is present, adapted from Theorem 4.1 of Andriopoulos et al. (2024). We focus on the regime where $c < \lambda_n$ since we are interested in the limiting behavior when $\lambda_H \to 0$ and $\lambda_W \to 0$.

Theorem I.1 (Regularized UFM Solution, adapted from Andriopoulos et al. (2024))**.** *Suppose $0 < c < \lambda_n$. Define $A = \Sigma^{1/2} - \sqrt{c}I_n$. Then any global minimizer (W_λ, H_λ) of equation 3 can be expressed as:*

$$W_\lambda = \left(\frac{\lambda_H}{\lambda_W}\right)^{1/4} A^{1/2}R \tag{4}$$

$$H_\lambda = \left(\frac{\lambda_W}{\lambda_H}\right)^{1/4} R^\top A^{1/2}(\Sigma^{1/2})^{-1}\tilde{Y} \tag{5}$$

where $R \in \mathbb{R}^{n\times d}$ is any matrix satisfying $RR^\top = I_n$.

Proof. This follows from Theorem 4.1 of Andriopoulos et al. (2024) with $c < \lambda_n$ implying $j^* = n$. □

The regularized solution has a specific dimensional structure. Since $A^{1/2}(\Sigma^{1/2})^{-1}\tilde{Y} \in \mathbb{R}^{n\times M}$ and $R^\top \in \mathbb{R}^{d\times n}$, the matrix $H_\lambda \in \mathbb{R}^{d\times M}$ has columns lying in the column space of $R^\top$, which is at most n-dimensional. Even though H_λ lives in a d-dimensional ambient space with $d \gg n$, its columns are confined to an n-dimensional subspace.

When regularization is absent, the problem has infinitely many global minimizers, characterized by the following theorem from Andriopoulos et al. (2024).

Theorem I.2 (Unregularized UFM Solutions, from Andriopoulos et al. (2024))**.** *When $\lambda_W = \lambda_H = 0$, a pair (W, H) is a global minimizer of equation 3 if and only if $WH = \tilde{Y}$ and W has full row rank. For any such W, the corresponding global minimizers in H are:*

$$H_{unreg} = W^\dagger\tilde{Y} + (I_d - W^\dagger W)Z \tag{6}$$

where $W^\dagger$ is the Moore–Penrose pseudoinverse and $Z \in \mathbb{R}^{d\times M}$ is arbitrary.

Proof. See Theorem 4.3 of Andriopoulos et al. (2024). □

The structure in equation 6 decomposes solutions into two orthogonal components: $W^\dagger \tilde{Y}$ lies in the row space of W (dimension at most n), while $(I_d - W^\dagger W)Z$ lies in the null space of W (dimension exactly $d - n$). The Frobenius norm satisfies $\|H_{\text{unreg}}\|_F^2 = \|W^\dagger \tilde{Y}\|_F^2 + \|(I_d - W^\dagger W)Z\|_F^2$ by orthogonality. The minimum-norm solution is achieved when $Z = 0$, eliminating the $(d - n)$-dimensional null space component. When $Z \neq 0$, the null space allows H to span up to d dimensions, whereas $Z = 0$ confines H to at most n dimensions.

We now investigate what happens to the regularized solution as weight decay vanishes, examining the limit $\lambda_W, \lambda_H \to 0$ while maintaining $\lambda_H/\lambda_W = k > 0$.

Lemma I.3 (Limiting Reconstruction)**.** *Suppose* $\lim_{\lambda_H \to 0, \lambda_W \to 0}(\lambda_H/\lambda_W) = k > 0$ *and let* (W_λ, H_λ) *be as in Theorem I.1. Then:*

$$\lim_{\lambda_W, \lambda_H \to 0} W_\lambda H_\lambda = \tilde{Y} \tag{7}$$

Proof. Since $c = \lambda_W \lambda_H \to 0$, Theorem I.1 applies for sufficiently small λ_W, λ_H. Multiplying:

$$W_\lambda H_\lambda = \left(\frac{\lambda_H}{\lambda_W}\right)^{1/4} A^{1/2} R \cdot \left(\frac{\lambda_W}{\lambda_H}\right)^{1/4} R^\top A^{1/2} (\Sigma^{1/2})^{-1} \tilde{Y} \tag{8}$$

$$= A^{1/2}(RR^\top)A^{1/2}(\Sigma^{1/2})^{-1}\tilde{Y} = A(\Sigma^{1/2})^{-1}\tilde{Y} \tag{9}$$

Substituting $A = \Sigma^{1/2} - \sqrt{c} I_n$:

$$W_\lambda H_\lambda = (\Sigma^{1/2} - \sqrt{c} I_n)(\Sigma^{1/2})^{-1}\tilde{Y} = (I_n - \sqrt{c}\Sigma^{-1/2})\tilde{Y} \tag{10}$$

As $c \to 0$, this yields $\tilde{Y}$. □

Theorem I.4 (Limiting Solution Structure)**.** *Under the assumptions of Lemma I.3, define:*

$$W_0 = \lim_{\lambda_W, \lambda_H \to 0} W_\lambda = k^{1/4} \Sigma^{1/4} R \tag{11}$$

$$H_0 = \lim_{\lambda_W, \lambda_H \to 0} H_\lambda = k^{-1/4} R^\top \Sigma^{-1/4} \tilde{Y} \tag{12}$$

Then (W_0, H_0) *is a global minimizer of the unregularized problem with:*

$$H_0 = W_0^\dagger \tilde{Y} \tag{13}$$

In particular, H_0 *has no null space component (i.e.,* $Z = 0$ *in Theorem I.2).*

Proof. From Lemma I.3, $W_0 H_0 = \tilde{Y}$, so (W_0, H_0) is a global minimizer. Since $W_0 = k^{1/4}\Sigma^{1/4}R$ has full row rank, its pseudoinverse is:

$$W_0^\dagger = W_0^\top (W_0 W_0^\top)^{-1} = (k^{1/4}\Sigma^{1/4}R)^\top (k^{1/2}\Sigma^{1/2})^{-1} = k^{-1/4} R^\top \Sigma^{-1/4} \tag{14}$$

Thus $W_0^\dagger \tilde{Y} = k^{-1/4} R^\top \Sigma^{-1/4} \tilde{Y} = H_0$, confirming $Z = 0$. For any H satisfying $W_0 H = \tilde{Y}$, we have $H = W_0^\dagger \tilde{Y} + (I_d - W_0^\dagger W_0)Z$ with:

$$\|H\|_F^2 = \|W_0^\dagger \tilde{Y}\|_F^2 + \|(I_d - W_0^\dagger W_0)Z\|_F^2 \geq \|H_0\|_F^2 \tag{15}$$

by orthogonality, with equality if and only if $Z = 0$. Thus H_0 is the minimum-norm solution. □

These results explain why weight decay causes dimensional collapse. Without regularization, H can utilize the full d-dimensional space through the arbitrary null space component $(I_d - W^\dagger W)Z$, where the null space has dimension $d - n$. With any positive regularization, Theorem I.1 shows H_λ is confined to an at most n-dimensional subspace. Theorem I.4 proves that even as $\lambda_W, \lambda_H \to 0^+$, the limiting solution has $Z = 0$, eliminating the $(d - n)$-dimensional null space component. This demonstrates that even infinitesimally small weight decay induces dimensional collapse by biasing toward the minimum-norm solution of the unregularized problem, which lies entirely in the n-dimensional row space of W. Since typically $d \gg n$, this represents a massive dimensional reduction from the ambient feature space to a low-dimensional subspace determined by the target structure.

Remark I.5 (Breakdown of Affine Congruence in the Over-Compressed Regime.). Theorem I.1, adapted from Theorem 4.1 of Andriopoulos et al. (2024), characterizes global minimizers of the regularized UFM objective in terms of the parameter

$$c := \lambda_W \lambda_H$$

and the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n > 0$ of the target covariance Σ. When $0 < c < \lambda_n$, the theorem yields $j^* = n$, and the solution satisfies

$$H_\lambda = \left(\frac{\lambda_W}{\lambda_H}\right)^{1/4} R^\top A^{1/2} (\Sigma^{1/2})^{-1} \tilde{Y}, \quad A = \Sigma^{1/2} - \sqrt{c}\, I_n,$$

with $A \succ 0$. In this regime, $A^{1/2}$ is full rank and the map relating H_λ and $\tilde{Y}$ is an invertible affine transformation on the support of $\tilde{Y}$. Consequently, the learned features and targets are affine congruent, implying preservation of intrinsic dimension:

$$\mathrm{ID_H} = \mathrm{ID_Y}.$$

In contrast, when $c \geq \lambda_n$, Theorem 4.1 of Andriopoulos et al. (2024) yields $j^* < n$, and the solution involves truncated matrices $[A^{1/2}]_{j^*}$ and $[\Sigma^{1/2}]_{j^*}$ obtained by retaining only the eigenspaces corresponding to eigenvalues $\lambda_i > c$. In this case, the resulting feature matrix satisfies

$$\mathrm{rank}(H_\lambda) \leq j^* < n,$$

and the affine map relating H_λ and $\tilde{Y}$ is no longer invertible on the support of $\tilde{Y}$. As a consequence, affine congruency is not guaranteed to hold. This regime allows for a reduction in intrinsic dimension,

$$\mathrm{ID_H} < \mathrm{ID_Y},$$

which is precisely what we observe empirically in the over-compressed region. This observation highlights the diagnostic value of intrinsic dimension: deviations between $\mathrm{ID_H}$ and $\mathrm{ID_Y}$ signal departure from the regime in which the assumptions underlying affine congruency hold. Although the parameter c is not directly observable in practice, as it arises from the UFM abstraction rather than explicit training dynamics, intrinsic dimension provides an empirical proxy for this regime. In particular, the transition from $\mathrm{ID_H} \approx \mathrm{ID_Y}$ to $\mathrm{ID_H} < \mathrm{ID_Y}$ indicates that the effective regularization has crossed the threshold $c = \lambda_n$, placing the model in the over-compressed regime and violating the conditions under which affine congruency is expected.

I.2 Why Collapsed Models Fail to Generalize?

The proof of Theorem 7.2 follows directly from Sard's theorem.

Theorem I.6. *Let $\mathcal{M}$ be a smooth m-dimensional manifold and $\mathcal{N}$ be a smooth n-dimensional manifold, with $m < n$. A smooth map $g : \mathcal{M} \to \mathcal{N}$ cannot be surjective, i.e., $g(\mathcal{M}) \neq \mathcal{N}$.*

Proof. Let $g : \mathcal{M} \to \mathcal{N}$ be a smooth map where $\dim(\mathcal{M}) = m$ and $\dim(\mathcal{N}) = n$, under the condition $m < n$. Consider an arbitrary point $p \in \mathcal{M}$. The differential of the map at this point, $dg_p : T_p\mathcal{M} \to T_{g(p)}\mathcal{N}$, is a linear transformation from the m-dimensional tangent space of $\mathcal{M}$ at p to the n-dimensional tangent space of $\mathcal{N}$ at $g(p)$.

By the rank-nullity theorem, the rank of dg_p is bounded by the dimension of its domain, so it holds that $\mathrm{rank}(dg_p) \leq m$. Given that $m < n$, it follows that $\mathrm{rank}(dg_p) < n$. A linear map is surjective if and only if its rank equals the dimension of its codomain; thus, dg_p is not surjective.

As the choice of p was arbitrary, this holds for all $p \in \mathcal{M}$. By definition, a point is critical if its differential is not surjective. Therefore, every point in the domain $\mathcal{M}$ is a critical point of g. The image of the set of critical points is the set of critical values. In this case, the set of critical values is the entire image of the map, $g(\mathcal{M})$.

By Sard's Theorem, the set of critical values of a smooth map has Lebesgue measure zero in the codomain. It follows that the image $g(\mathcal{M})$ has measure zero in $\mathcal{N}$. However, a smooth n-dimensional manifold (for $n \geq 1$) has positive Lebesgue measure. Since a set of measure zero cannot be equal to a set of positive measure, it must be that $g(\mathcal{M}) \neq \mathcal{N}$.

Therefore, the map g is not surjective. □

This theorem provides the geometric foundation for understanding the failure of collapsed models. In our regression context, the learned features $\{\mathbf{h}_\theta(\mathbf{x})\}$ form a feature manifold $\mathcal{M}_H$ of dimension $m = \mathrm{ID}_\mathrm{H}$, while the targets $\{\mathbf{y}\}$ lie on a target manifold $\mathcal{N}_Y$ of dimension $n = \mathrm{ID}_\mathrm{Y}$. The final layer of the network constitutes a smooth map from the feature manifold to the target space.

When a model is in the over-compressed regime ($\mathrm{ID}_\mathrm{H} < \mathrm{ID}_\mathrm{Y}$), the theorem's condition ($m < n$) is met. The direct consequence is that this smooth map cannot be surjective. This means the image of the feature manifold—the set of all possible predictions the model can generate—is a proper subset of the target manifold. Geometrically, there will always be points on the target manifold that lie outside the model's predictive reach. A perfect reconstruction is therefore impossible, as the model is fundamentally incapable of generating the full range of target data, leading to an unavoidable error.

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 three contributions listed in the introduction are directly supported by Section 5, Sections 6–7, and Section 6.1, respectively.

   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: Section 8 provides a detailed discussion of limitations and outlines directions for future work.

   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: Appendix I includes full details about our theoretical results.

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: Appendix A lists all experimental details for reproduction.

   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: [Yes]

Justification: All datasets are publicly available, and our code is released at `https://github.com/Konokiii/neural_regression_collapse`.

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: Full training and evaluation details, including all hyperparameter grids, are provided in Appendix A and summarized in Tables 4 and 5.

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: We sweep over model widths and weight decay values to demonstrate robustness of trends, and RL policy evaluations are averaged over 100 episodes.

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: Appendix A reports the GPU type, memory, number of CPU workers, and average training time per model.

   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: We have reviewed and adhered to the NeurIPS Code of Ethics throughout this work.

   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: [N/A]

    Justification: This work is foundational research on the geometry of learned representations, with no direct pathway to negative societal applications.

    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 pretrained models or new datasets that pose misuse risks.

    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 datasets (MuJoCo/Gymnasium, D4RL, Visual D4RL) and pretrained models are properly cited, and each is released under an open-source license.

    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: This paper does not introduce new datasets, models, or benchmarks.

    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: This work does not involve crowdsourcing or 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: This work does not involve human subjects research.

    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: LLMs were used solely for writing assistance and editing; they played no role in the core methodology or experiments.

    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.