CLUSTERING STABILITY AND COLLAPSE IN DEEPER NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

We show that as neural networks grow deeper, their internal representations progressively lose the clustering structure of the input and that this degradation, measured by the Frobenius distance between projection operators, is tightly and consistently associated with a sharp drop in clustering accuracy across architectures and datasets. This geometric characterization leads directly to two principled strategies: a structure-aware regularizer that penalizes projection distance at each layer, and an orthogonal initialization strategy that provides a structurally favorable starting point for training. Across more than 60,000 trials spanning five prevalent architectures and eight real-world datasets, our regularizer delays collapse in performance by an average of 7 layers and sustains high clustering accuracy well past the depth at which all unregularized counterparts have collapsed entirely.

1 INTRODUCTION

Neural Networks are the cornerstone of modern artificial intelligence. However, the mechanisms underlying how these models represent, transform, and generalize complex data remain largely unclear. As consequence, core design choices such as depth, width, connectivity, and regularization are largely guided by empirical heuristics and extensive tuning, rather than by mechanistic insight. Among these open questions, this paper investigates the performance collapse in neural networks as depth increases. In order to mitigate this phenomenon we propose the following regularizer:

$$\mathcal{R}(\Theta) \;:=\; \sum_{\ell} \|\mathbf{P}(\mathbf{X}) - \mathbf{P}(\mathbf{H}^{\ell}_{\Theta})\|_{\mathrm{F}}^{2}, \tag{1}$$

where $\mathbf{X}$ is the data matrix whose columns are the input to the network, $\mathbf{H}^{\ell}_{\boldsymbol{\Theta}}$ represents the hidden representation at the ℓth layer given the parameters Θ, $\mathbf{P}(\cdot)$ denotes the projection operator onto the principal row-space of dimension r, and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm.

Our regularizer is inspired by a recent work showing that *shallow* networks inherently learn parameters that preserve subspace cluster structure without imposing any explicit mechanisms enforcing such behavior (Veeranna Rupashree et al., 2026). This structure is encoded by the projection operator $\mathbf{P}(\mathbf{X})$. Similarly, the subspace cluster structure of the hidden representation at layer ℓ is encoded in $\mathbf{P}(\mathbf{H}^{\ell})$, and the projection distance $\|\mathbf{P}(\mathbf{X}) - \mathbf{P}(\mathbf{H}^{\ell})\|_{\mathrm{F}}$ measures the loss of subspace clustering structure from the input to the ℓth hidden representation. As we will show, performance collapse in deeper networks is tightly coupled with the loss of subspace clustering structure. More precisely, as projection distance rises steadily with depth, clustering accuracy drops in tandem (see Figure 1-Left). Therefore, by penalizing the projection distance between $\mathbf{P}(\mathbf{X})$ and $\mathbf{P}(\mathbf{H}^{\ell})$ at each layer, our regularizer creates a continuous corrective force that prevents intermediate representations from drifting away from the original structure throughout the entire network. In doing so, our regularizer stabilizes performance, effectively extending the range of feasible deep networks (see Figure 1-Right).

Our extensive experiments show that this regularizer can be incorporated into a very wide range of loss function to stabilize deep representations. Figure 2 summarizes over 60,000 trials, showing the average performance of 5 prevalent architectures across 8 canonical datasets, with and without our regularizer (see Figure 4 for an in-depth view of these experiments, layered by dataset and architecture). The dashed lines (unregularized models) show that all architectures face a sharp performance drop as depth increases, some with as few as 4 layers. This is a pattern that repeats

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

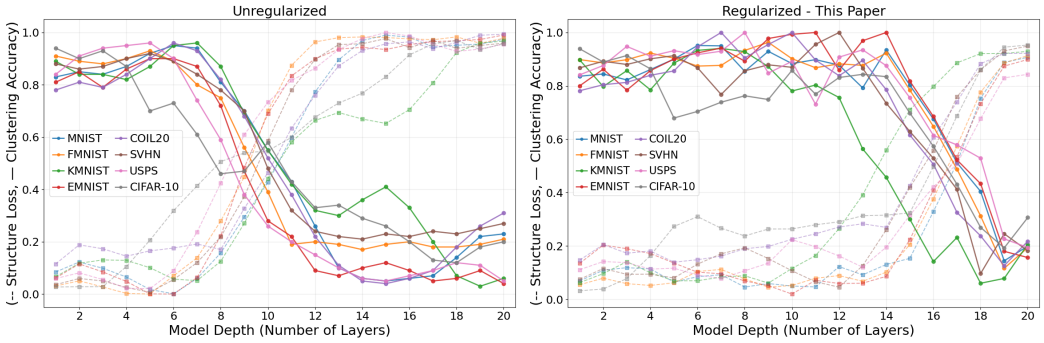


Figure 1: **Left (Unregularized):** Clustering accuracy (solid) and projection distance (dashed) move in opposite directions as depth increases, confirming that performance collapse is directly coupled with the loss of subspace cluster structure. **Right (Regularized, this paper):** By penalizing structural drift at each layer, our regularizer prevents this loss and shifts the collapse depth rightward by an average of 7 layers while sustaining high accuracy.

regardless of dataset size, dimensionality, class count, or even architecture. The solid lines show that our regularizer delays collapse an average of 7 layers, sustaining high accuracies up to layer 20, well past the point where *all* the unregularized counterparts have collapsed entirely.

Naturally, the effectiveness of our regularizer depends on its regularization parameter λ. Across the range we tested, larger λ progressively delayed collapse, and no value degraded performance meaningfully below the unregularized baseline, though very small λ yielded only marginal benefit at large depths (see Figure 3-Right). In other words, our regularizer can improve performance, but does not degrade it. This can be intuitively explained in light of the results in (Veeranna Rupashree et al., 2026): our regularizer guides the network toward the parameters it is inherently driven to discover, namely, those that preserve the underlying subspace clustering structure.

Beyond our regularizer, understanding structure loss as a proxy of performance degradation leads directly to practical interventions. For example, we show that initializing the network parameters according to an orthogonal union of subspaces that minimally distorts the input structure provides a complementary benefit, delaying collapse even further (see Figure 3-Right). Together, these strategies reduce the sensitivity of performance to architectural choices, making depth a far less fragile design decision.

The rest of the paper is organized as follows: in Section 2 we discuss our regularizer in the context of related work. Section 3 details how projection matrices encode subspace cluster structure, and how their distances can be understood as a measure of structure loss. Section 4 establishes the empirical correlation between projection distance and clustering accuracy, showing that structural degradation is consistently associated with performance collapse. Section 5 develops two principled strategies for

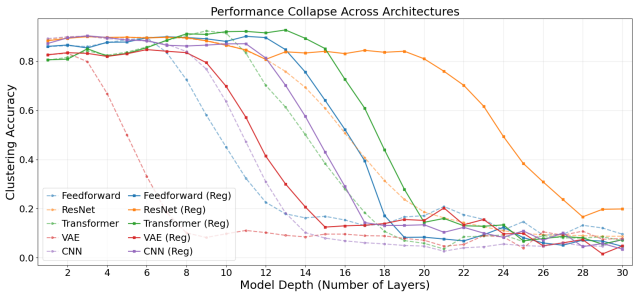


Figure 2: Average clustering accuracy across eight datasets for five architectures. In all cases, unregularized networks (dashed) undergo a sharp and consistent collapse an average of 7 layers before our regularized counterparts (solid).

delaying collapse via structure preservation. Section 6 presents our experimental evaluation across synthetic and real-world datasets and five neural architectures. The goal of this paper is not to improve peak clustering accuracy, which the shallow representations already cluster well, but to preserve that structure as the depth grows, extending the range of depths over which a network remains usable.

2 Related Work

The theoretical understanding of neural networks, due to their high-dimensional and nonlinear transformations, still remains an ongoing challenge. While deep learning models achieve high performance, their interpretability and underlying mechanisms are still not fully understood. Methods such as feature attribution (Lundberg, 2017; Molnar, 2020) and mechanistic interpretability (Bereska & Gavves, 2024) try to explain how neural networks process information, but they often fail to capture the deeper structural and geometric properties that form inside the model. Because of this, many basic questions about how networks build, preserve, and lose clustering structure remain unanswered.

The geometry of learned representations has also been studied through the lens of optimization and implicit bias. Overparameterization can preserve useful feature patterns even when models have far more parameters than required (Elhage et al., 2021). The implicit biases introduced by stochastic gradient descent have likewise been linked to the preservation of clustering structure, showing that training dynamics directly affect how separable learned features become (Soudry et al., 2018). Training behavior such as double descent (Belkin et al., 2019; Nakkiran et al., 2021) has also been connected to how feature geometry evolves at different model sizes. Studies suggest that activation functions shape the structure of the feature space in ways that may help maintain cluster boundaries (Advani et al., 2020; Zhang et al., 2021), while initialization choices such as Xavier and Kaiming (Pan et al., 2022) stabilize gradients early in training and indirectly influence how clusters form and separate. Nonlinearities such as ReLU have also been shown to retain important topological and clustering properties in latent spaces (Xu, 2015).

More recently, several works have begun studying collapse phenomena and their relationship to representation geometry. Classical neural collapse analyses show that deep networks may converge toward highly structured terminal geometries under supervised training (Papyan et al., 2020), while subsequent work has related deep representation evolution to progressive separation and collapse phenomena (He & Su, 2023). Recent theoretical advances further extend collapse analysis beyond multilayer perceptrons, showing that neural collapse can emerge as a globally optimal phenomenon in deep regularized ResNets and transformers (Súkeník et al., 2025), while other recent geometric models relate representation evolution to progressive separation dynamics (Shi et al., 2025). Together, these studies suggest that representation geometry and collapse are closely linked, but they do not explicitly characterize how clustering structure degrades across depth or how this degradation determines collapse behavior.

Recent work from the deep clustering literature is even more closely related to our study. Methods designed to prevent clustering collapse highlight how fragile feature geometry can be, especially in deeper models. Hard regularization strategies, for example, have been shown to delay or prevent collapse in online clustering systems by enforcing stable assignment patterns throughout training (Mahon & Lukasiewicz, 2024). Other methods enforce structural consistency or balance constraints, showing that preventing early representational collapse improves downstream clustering quality. Similarly, studies of deep autoencoder-based clustering show that collapse often emerges as layers become deeper and feature geometry becomes increasingly compressed toward low-rank structures (Xu et al., 2023). These approaches seek to prevent collapse heuristically, but do not explicitly characterize the relationship between structural persistence, collapse depth, and clustering performance.

Across all of these studies, a common theme is that the stability or collapse of clustering structure is related to model performance. However, prior work does not establish a quantitative relationship between how long clustering structure persists across depth, when representational collapse begins, and how that collapse can be systematically delayed through geometric control. This gap motivates the focus of our current work.

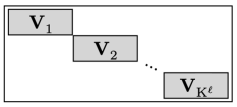
3 Projection Distances as a Measure of Structure Loss

The main insight behind our approach is that performance degradation in deeper layers is tightly coupled to the loss of subspace cluster structure in hidden representations. In this section we show that such structure admits a precise geometric characterization through projection matrices, which in turn can be exploited to mitigate structure loss, and ultimately performance.

To this end, let us first formalize the concept of subspace cluster structure. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ denote the input data matrix. We say $\mathbf{X}$ has subspace cluster structure if its columns lie near a union of low-dimensional subspaces, in which case

$$\mathbf{x}_i = \sum_{k=1}^{K} \mathbb{1}_{\{i \in \boldsymbol{\Omega}_k\}} \mathbf{U}_k \mathbf{v}_i + \boldsymbol{\epsilon}_i, \tag{2}$$

where $\{\boldsymbol{\Omega}_k\}_{k=1}^{K}$ partitions the samples into clusters, $\mathbf{U}_k$ denotes a basis for the kth subspace, $\mathbf{v}_i$ contains coordinates of $\mathbf{x}_i$ in that basis, and $\boldsymbol{\epsilon}_i$ represents deviation from ideal subspace structure. In words, this model assumes there is a collection of K subspaces of collective dimension r, such that each sample lies near one of them. One way to encode this structure is through the coefficient matrix $\mathbf{V}$, defined as the $r \times n$ block-diagonal matrix whose k^{th} diagonal block $\mathbf{V}_k$ contains the coordinates $\{\mathbf{v}_i\}_{i \in \boldsymbol{\Omega}_k}$ of the samples belonging to the kth subspace. That is, if the ith column of $\mathbf{V}$ is nonzero on the kth block, then $\mathbf{x}_i$ belongs to the kth cluster. The clustering structure is therefore embedded in the block-diagonal sparsity pattern of $\mathbf{V}$, as illustrated below.

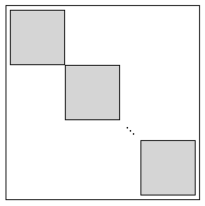
$\mathbf{V} =$ 

.

Unfortunately, $\mathbf{V}$ is not unique, since subspaces admit infinitely many equivalent bases, most of which do not have this group-sparse structure. This motivates pursuing a unique geometric object instead: the projection operator onto the principal row space of $\mathbf{X}$. Orthonormalizing each $\bar{\mathbf{V}}$ we can construct an orthonormal basis $\bar{\mathbf{V}}$ spanning the same row space as $\mathbf{V}$, so that the projection operator onto this space is

$$\mathbf{P}(\mathbf{X}) = \bar{\mathbf{V}}^{\mathsf{T}} \bar{\mathbf{V}}.$$

Notice that $\mathbf{P}(\mathbf{X})$ has a similar block structure as $\mathbf{V}$:

$\mathbf{P}(\mathbf{X}) =$ 
.

That is, if the $(i, j)^{\text{th}}$ entry of $\mathbf{P}(\mathbf{X})$ is nonzero, then $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster. Fortunately, $\mathbf{P}(\mathbf{X})$ can be trivially computed via singular value decomposition. To summarize, the projector operator $\mathbf{P}(\mathbf{X})$ uniquely encodes the same subspace cluster structure of $\mathbf{X}$ represented in $\mathbf{V}^\ell$, while avoiding basis ambiguity. Analogously, $\mathbf{P}(\mathbf{H}^\ell)$ encodes the subspace cluster structure of the ℓth hidden representation $\mathbf{H}^\ell = [\mathbf{h}_1^\ell, \ldots, \mathbf{h}_n^\ell]$.

This motivates the projection distance

$$\|\mathbf{P}(\mathbf{X}) - \mathbf{P}(\mathbf{H}^\ell)\|_F, \tag{3}$$

as a measure of how much of that structure is lost as data propagates through the network. Projection operators can be understood as points in the Grassmannian, the space of all subspaces. Under this interpretation, the Frobenius distance between two projection operators is a principled geometric measure of how far apart two subspaces are. It vanishes when the subspaces are identical and grows monotonically as they diverge. This makes it a natural and interpretable measure of structural drift:

As hidden representations drift away from the input subspace across layers, this distance grows monotonically, and with it, the cluster boundaries that once separated the data begin to erode. A small projection distance indicates that the geometry of the layer representation remains well-aligned with the original input structure; as it grows, the subspace arrangement becomes increasingly distorted. Collapse is therefore not an abrupt or mysterious failure, but rather the inevitable consequence of accumulated geometric drift which can be monitored through the projection distance, and crucially, intervened upon. If collapse is consistently associated with structural drift, then penalizing that drift should delay or mitigate it. Section 5 develops two principled strategies for doing exactly this.

4 Empirical Signatures of Performance Collapse

We now establish the empirical phenomenon motivating our regularizer: the consistent association between performance collapse and the degradation of subspace cluster structure across depth.

Rather than treating collapse as an endpoint observed only after training, we show that it emerges progressively and can be tracked directly through the projection distance defined in Section 3. At shallow depths, clustering accuracy remains high while projection distance remains low, indicating that the network is successfully learning weights that preserve the subspace cluster structure of the input. This is consistent with the findings of Veeranna Rupashree et al. (2026), who use a perturbation analysis and the Davis-Kahan $\sin(\boldsymbol{\Theta})$ Theorem to show that shallow feed-forward networks inherently learn parameters that preserve this structure. However, as depth increases, the perturbation bound degrades with each successive application of the nonlinear activation function resulting in a deteriorated clustering accuracy as projection distances rise.

Architectures. To verify that the same qualitative pattern is observed consistently across models, we evaluate five neural architectures of increasing structural complexity.

- Feedforward networks (Goodfellow et al., 2016) are fully connected ReLU networks composed of L layers, where each layer applies a learned affine transformation followed by a nonlinearity to produce the next hidden representation.
- Residual networks (He et al., 2016) follow the same structure as feedforward networks but augment each layer with a skip connection that adds the layer input directly to its output, helping to preserve information across depth.
- Variational autoencoders (Kingma & Welling, 2019) are encoder-decoder architectures with a stochastic latent bottleneck; we evaluate collapse dynamics on the encoder representations at each depth.
- Convolutional neural networks (Krizhevsky et al., 2012) are composed of convolution-ReLU blocks, where each block is treated as one layer in the collapse analysis and intermediate feature maps are flattened to compute projection operators and clustering metrics.
- Transformer layers (Vaswani et al., 2017) consist of alternating self-attention and feedforward sublayers, where each block is treated as one depth unit in the structural analysis.

Datasets. Similarly, to verify the tandem relation between structure loss and performance collapse across datasets, we use eight real-world datasets spanning image recognition, handwritten character classification, and object recognition, including MNIST (Deng, 2012), CIFAR-10 (Krizhevsky, 2009), Extended MNIST (Cohen et al., 2017), Street View House Numbers (SVHN) (Netzer et al., 2011), Kuzushiji-MNIST (Clanuwat et al., 2018), Fashion-MNIST (Xiao et al., 2017), USPS handwritten digits (Hull, 1994), and COIL20 (Nene et al., 1996). These datasets cover a broad range of dimensionality, cluster complexity, and statistical structure. Full dataset parameters and per-architecture learning rates are summarized in Table 1.

Evaluation Metrics. For each architecture-dataset-depth combination, we train an independently initialized model from scratch over 50 independent trials, ensuring no information is shared across depths or trials. The following two metrics are recorded at each depth.

Clustering accuracy. At each network depth, we apply subspace clustering to the learned representations and compute accuracy by matching predicted cluster assignments to ground-truth labels via optimal permutation according to the Hungarian matching algorithm (Kuhn, 1955).

Dataset	Size	K	d	Rank	LR FF	Res	VAE	CNN	Tr
Coil 20	[1440,32,32,1]	20	12	240	0.005	0.005	0.001	0.001	0.0001
MNIST	[1280,28,28,1]	20	10	200	0.005	0.005	0.001	0.001	0.0001
CIFAR-10	[400,32,32,3]	40	10	400	0.005	0.005	0.001	0.001	0.0001
Extended MNIST	[240000,32,32,1]	10	2	20	0.005	0.005	0.001	0.001	0.0001
Kuzushiji MNIST	[7349,32,32,1]	37	3	111	0.005	0.005	0.001	0.001	0.0001
Fashion MNIST	[507,28,28,1]	10	5	50	0.005	0.005	0.001	0.001	0.0001
SVHN	[303,32,32,3]	5	1	5	0.005	0.005	0.001	0.001	0.0001
USPS	[10299,16,16,1]	6	20	120	0.005	0.005	0.001	0.001	0.0001

Table 1: Parameters of real datasets.

Structure loss is measured using the projection distance equation 3 evaluated at the final layer of each model, quantifying how much subspace cluster structure the network's output representation has lost relative to the input $\mathbf{X}$.

Training Protocol. Each architecture-dataset-depth configuration is trained with per-dataset learning rates as specified in Table 1. To account for variability due to random initialization and stochastic optimization, all experiments are repeated over 50 independent trials and reported results correspond to means across trials.

Implementation Strategy. For each depth, we train a separate model and evaluate its final-layer representation. Predicted cluster assignments are obtained by applying subspace clustering to these representations, and accuracy is computed by matching predictions to ground-truth labels via optimal permutation. Rank parameters are selected as $\mathrm{K} \times \mathrm{d}$ or via singular-value heuristics depending on the dataset, as specified in Table 1. For CNNs and transformer models, intermediate feature maps are reshaped into matrix representations prior to projection-operator computation, ensuring that structural metrics remain consistent across architectures.

Results. Figure 1 illustrates the signature of collapse on a representative real datasets for a feedforward network: at shallow depths clustering accuracy remains high while projection distance remains low, and the two curves intersect close to the collapse depth, beyond which accuracy degrades sharply. The collapse depth thus provides a principled indicator of when additional layers cease to preserve meaningful structure. As shown by the dashed lines in Figure 4, this same qualitative pattern holds consistently across all the remaining four architectures and eight real-world datasets, confirming it is a broader property of deep representation dynamics rather than an artifact specific to a single architecture or dataset.

These observations motivate the central question of this paper: if representational collapse is associated with the degradation of subspace cluster structure, can preserving that structure delay or mitigate collapse? Section 5 develops two principled strategies for doing exactly this.

5 Delaying Collapse via Structure Preservation

The geometric characterization developed in Section 4 suggests a direct route for mitigating representational collapse. If collapse is consistently associated with progressive degradation of subspace cluster structure, then preserving that structure should delay or reduce it, see figure 1. Motivated by this observation, we propose augmenting the training objective with our structure-aware penalty:

$$\mathcal{L}(\mathbf{\Theta}) \;=\; \sum_{\mathrm{i}} \ell(f_{\mathbf{\Theta}}(\mathbf{x}_{\mathrm{i}}), \mathbf{x}_{\mathrm{i}}) \;+\; \lambda \mathcal{R}(\mathbf{\Theta}). \tag{4}$$

Here $f_{\mathbf{\Theta}}$ is the output of the network, the first term is the task-specific loss, $\lambda \geq 0$ controls the strength of the subspace-preservation penalty, and equation 1 penalizes the projection distance between the input and each hidden representation. By penalizing structural drift at every intermediate layer, this regularizer continuously pushes the network toward representations that remain close to the input subspace throughout training. This strategy slows the loss of clustering structure across depth, making it possible to train significantly deeper networks before collapse occurs. In doing so, it reduces the sensitivity of performance to architectural choices such as depth, making model design a far less fragile process.

Besides regularization, weight initialization determines the initial geometry of the embedding transformations and can significantly influence how quickly representational structure degrades with depth. We thus complement our structure-aware penalty with an orthogonal initialization. More precisely, let $\mathbf{\Theta}$ denote the learnable weight parameters of a given layer. For feedforward and residual networks this is a single matrix, for transformer layers this includes the attention and feedforward weights, and for CNNs it collects the convolutional filter banks. In all cases, each constituent matrix $\mathbf{\Theta}$ is initialized such that

$$(\mathbf{\Theta}^{\ell})^{\mathsf{T}}\mathbf{\Theta}^{\ell} = \mathbf{I}.$$

In particular, orthogonal initialization promotes near-isometric transformations, helping to preserve pairwise distances and angular relationships between features. For this reason, orthogonal initialization provides a structurally favorable starting point for training, giving the network the best possible chance of maintaining cluster structure as depth increases.

Together, these two strategies target collapse from complementary angles: the regularizer keeps every intermediate representation close to the input subspace throughout the network, while orthogonal initialization provides a structurally favorable starting point that slows the layer-by-layer accumulation of projection distance, delaying the onset of collapse.

Setup. We demonstrate the effectiveness of our two strategies using synthetic data generated according to equation 2, where the basis matrices $\mathbf{U}_{\mathrm{k}} \in \mathbb{R}^{\mathrm{d}\times r_{\mathrm{k}}}$ are drawn with i.i.d. $\mathcal{N}(0,1)$ entries and subsequently orthogonalized, the coordinate vectors $\mathbf{v}_{\mathrm{i}}$ are drawn with i.i.d. $\mathcal{N}(0,1/\mathrm{d})$ entries, and the noise $\boldsymbol{\epsilon}_{\mathrm{i}}$ is drawn with i.i.d. $\mathcal{N}(0,\epsilon^2)$ entries. For these experiments we use $\mathrm{K}=4$ subspaces of dimension $r_{\mathrm{k}}=5$, $\mathrm{n}=100$ samples per subspace, ambient dimension $\mathrm{d}=100$, and noise level $\epsilon=1$, evaluated on a feedforward network with depth varied from 1 to 20 layers. All experiments based on synthetic data were conducted on a computer with an AMD Ryzen 7 5800H CPU, 16 GB RAM, and an NVIDIA GTX 1660 Ti GPU (6 GB). Having specified the data generation procedure and setup, we now describe the four training configurations used to evaluate the effect of each strategy on collapse depth.

Compared Strategies. The experiments in Figure 3-Left investigate the effect of each of our collapse mitigation strategies. The main strategy used in this paper to sustain the performance with increasing depth involves incorporating structure-aware penalty terms into the training objective. Secondly, we select initialization strategy based on their ability to shape the geometry of representations early in training. For the experiments in this paper, we evaluate four training strategies:

i) Unregularized: Random initialization of $\mathbf{\Theta}^{\ell}$ trained with the task-specific loss $\sum_{\mathrm{i}} \ell(f_{\Theta}(\mathbf{x}_{\mathrm{i}}), \mathbf{x}_{\mathrm{i}})$ only, with no structural penalty.

ii) Orthogonal Initialization: All weight matrices initialized orthogonally, trained with the task-specific loss $\sum_{\mathrm{i}} \ell(f_{\Theta}(\mathbf{x}_{\mathrm{i}}), \mathbf{x}_{\mathrm{i}})$ only.

iii) Regularized: Random initialization trained with the task-specific loss augmented by our subspace preservation penalty $\lambda\mathcal{R}(\Theta)$.

iv) Combined: Orthogonal initialization and subspace-preserving penalty applied jointly.

We note that orthogonality of Θ^{ℓ} constrains only the affine map; it does not by itself guarantee distance preservation after the nonlinear activation, so its role is to supply a favorable initial geometry rather than a structural guarantee.

Results. As shown in Figure 3-Left, the combination of regularization and orthogonal initialization yields the latest collapse depth among all configurations, significantly extending the depth at which meaningful clustering structure is retained. The unregularized baseline collapses earliest regardless of initialization, confirming that the regularizer is the primary driver of collapse resistance.

5.1 Penalty Parameter λ

The regularization hyperparameter λ controls the relative strength of the subspace preservation penalty in equation 4. To understand its influence on collapse, we repeat the synthetic experiment described above using the regularized feedforward baseline for a wide range of λ values and evaluate the resulting collapse depth. As shown in Figure 3-Right, increasing λ progressively delays the

collapse depth, up to a point. Beyond a certain threshold, further increasing λ begins to negatively affect performance, as the regularization term starts to dominate the task-specific loss. Within the tested range, larger λ consistently delayed collapse in deeper networks, while very small λ yielded marginal benefit that in some cases did not exceed the unregularized baseline at large depths.

Monitoring collapse without labels. While selecting depth and λ optimally in a fully unsupervised setting remains open, projection distance provides a label-free signal for monitoring where structure is lost. It indicates the depth beyond which additional layers cease to preserve structure, even when clustering accuracy is unavailable. We do not, however, propose it as a direct objective for tuning λ, since λ already minimizes this quantity and over-penalization degrades performance.

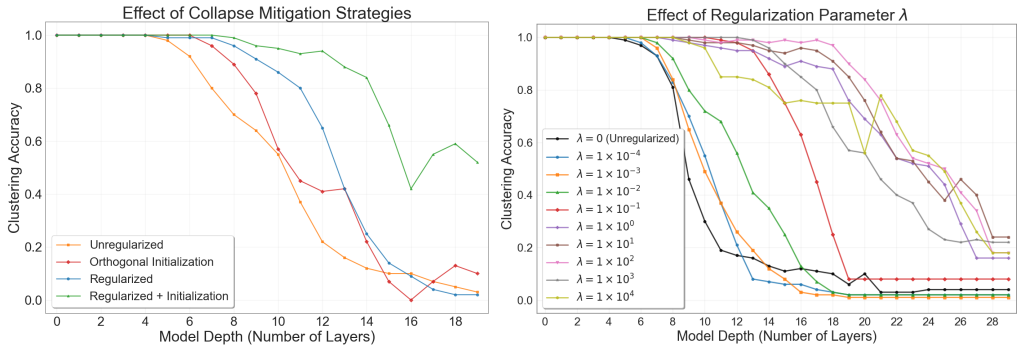


Figure 3: **Left: Comparison of collapse-mitigation strategies.** Orthogonal initialization alone provides a modest delay, our regularizer alone provides a more substantial one, and their combination yields the strongest and latest preservation of clustering structure across all depths. **Right: Effect of the regularization parameter λ on collapse depth.** Increasing λ progressively delays collapse across all values tested. Remarkably, our regularizer never hurts, regardless of the choice of λ, performance is always equal to or better than the unregularized baseline.

6 Experiments

Having established in Section 5 that our regularizer and initialization strategy effectively delay collapse on synthetic data, we now evaluate whether these gains transfer to real-world settings across all five architectures and eight datasets introduced in Section 4.

Across all architectures, depth is varied from 1 to 30 layers or blocks, enabling a consistent comparison of collapse behavior as a function of depth. All layers have a uniform width equal to the input dimensionality of each dataset (see Table 1), ensuring that hidden representations remain in the same ambient space as the input throughout the network. Width is held constant across depths so that observed differences in collapse behavior are attributable solely to depth rather than varying capacity.

Training Protocol. All experiments follow the protocol described in Section 4, with large-scale multi-architecture runs conducted on external high-throughput computing resources to manage the scale of over 60,000 trials. Unless stated otherwise, the unregularized and regularized baselines use identical PyTorch default initialization.

Results. Figure 2 summarizes our results, averaging clustering accuracy across all eight datasets by architecture, and Figure 4 contains the detailed results of each architecture-dataset pair. In all cases, unregularized networks lose performance around depth 7 and fully collapse by depth 15, with accuracy dropping from 0.88 to 0.10. Our approach delays this collapse by over 7 layers on average, maintaining accuracy above 0.80 even at depth 15, where all the unregularized counterparts have collapsed entirely. This pattern holds consistently across all five architectures, confirming that the regularizer's benefits are consistent throughout all these real-world datasets and architecture-agnostic.

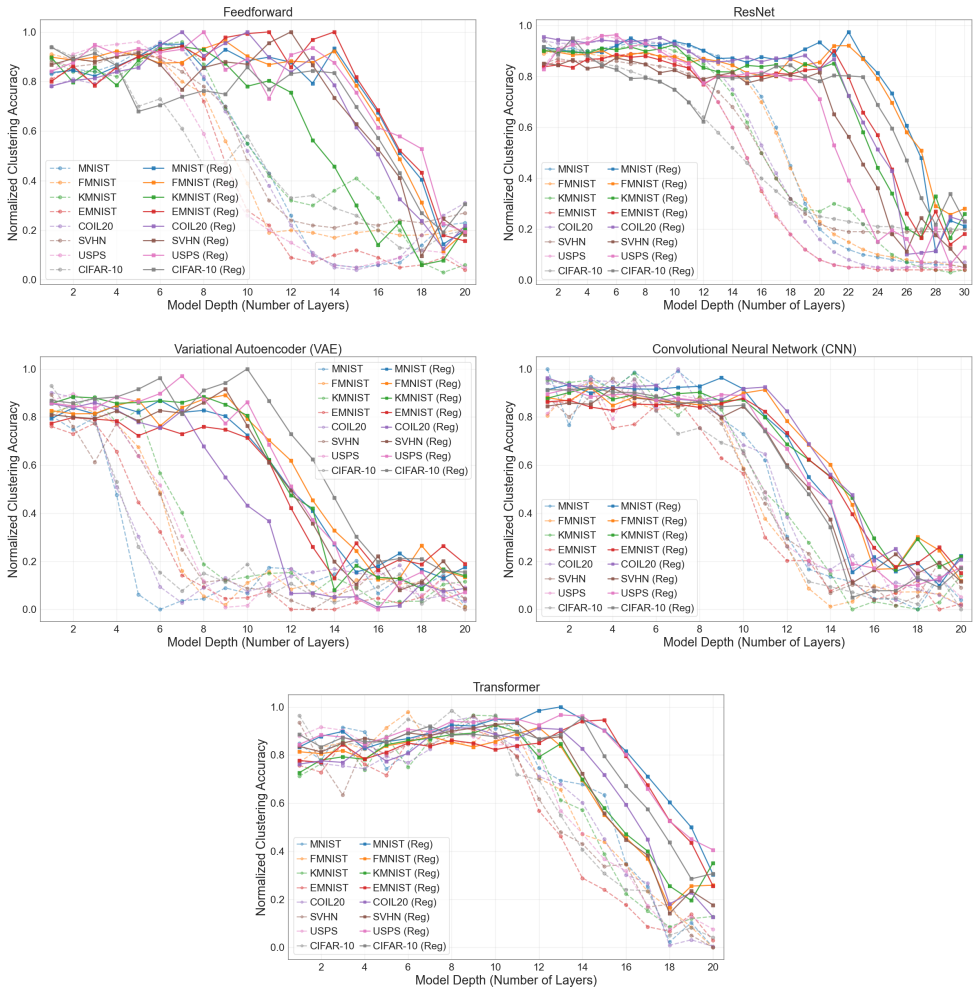


Figure 4: Clustering accuracy across eight datasets for five architectures. The consistent rightward shift of the collapse point across all architectures and datasets confirms the broad effectiveness of our regularizer.

AI USE STATEMENT

In this work, we have not used generative AI tools for Generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. Additionally, we used generative AI tools for create or edit software code, draft parts of a research paper. We have reviewed all AI-assisted work, LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*,

116(32):15849–15854, 2019.

Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. *CoRR*, abs/1702.05373, 2017. URL http://arxiv.org/abs/1702.05373.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Hangfeng He and Weijie J. Su. A law of data separation in deep learning. *Proceedings of the National Academy of Sciences*, 120(36):e2221704120, 2023. doi: 10.1073/pnas.2221704120. URL https://www.pnas.org/doi/abs/10.1073/pnas.2221704120.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. doi: 10.1109/34.291440.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *CoRR*, abs/1906.02691, 2019. URL http://arxiv.org/abs/1906.02691.

Alex Krizhevsky. Learning multiple layers of features from tiny images. pp. 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, Red Hook, NY, USA, 2012. Curran Associates Inc.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Scott Lundberg. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

Louis Mahon and Thomas Lukasiewicz. Hard regularization to prevent deep online clustering collapse without data augmentation, 2024. URL https://arxiv.org/abs/2303.16521.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). 1996.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Yu Pan, Zeyong Su, Ao Liu, Wang Jingquan, Nannan Li, and Zenglin Xu. A unified weight initialization paradigm for tensorial convolutional neural networks. In *International Conference on Machine Learning*, pp. 17238–17257. PMLR, 2022.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020. ISSN 1091-6490. doi: 10.1073/pnas.2015509117. URL http://dx.doi.org/10.1073/pnas.2015509117.

Cheng Shi, Liming Pan, and Ivan Dokmanić. Spring-block theory of feature learning in deep neural networks. *Phys. Rev. Lett.*, 134:257301, Jun 2025. doi: 10.1103/ys4n-2tj3. URL https://link.aps.org/doi/10.1103/ys4n-2tj3.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018. URL http://jmlr.org/papers/v19/18-188.html.

Peter Súkeník, Christoph H. Lampert, and Marco Mondelli. Neural collapse is globally optimal in deep regularized resnets and transformers, 2025. URL https://arxiv.org/abs/2505.15239.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Karan Vikyath Veeranna Rupashree, Siddharth Baskar, and Daniel L. Pimentel-Alarcón. Some neural networks inherently preserve subspace clustering structure. In *International Conference on Learning Representations (ICLR)*, Madison, Wisconsin, USA, 2026. Published as a conference paper.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL https://arxiv.org/abs/1708.07747.

Bing Xu. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Xiao Xu, Haiwei Hou, and Shifei Ding. Semi-supervised deep density clustering. *Appl. Soft Comput.*, 148(C), November 2023. ISSN 1568-4946. doi: 10.1016/j.asoc.2023.110903. URL https://doi.org/10.1016/j.asoc.2023.110903.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

A Appendix

You may include other additional sections here.