Developing Autoencoder: Incremental Bottleneck Expansion Leads to an Informed Latent Space

David Vogenauer[1,2,3,*], **Deyue Kong**[1,2,4,*], **Jonas Elpelt**[1,2,*], **Markos Genios**[2], **Matthias Kaschube**[1,2]

[1]Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany
[2]Department of Computer Science and Mathematics, Goethe University, Frankfurt am Main, Germany
[3]Technical University of Munich, Munich, Germany
[4]International Max Planck Research School for Neural Circuits, Frankfurt am Main, Germany
* These authors contributed equally

Abstract

Representation learning models, such as autoencoders (AEs), can effectively extract meaningful and generalizable features from natural image data. However, the learned latent features are often mixed or distributed across all bottleneck units, making interpretation difficult. Previous work has sought to address this by explicitly optimizing for feature separation or ordering.

We propose a biologically inspired progressive learning scheme, the Developing Autoencoder (Dev-AE), which incrementally expands the representational capacity. Increasing the size of the bottleneck layer over training epochs forces the Dev-AE to first learn compressed, low-dimensional representations before expanding into progressively higher-dimensional feature spaces. Comparing the latent space organization in Dev-AEs with that in standard AEs and PCA-initialized AEs (PCA-AE), we observe improved feature ordering and higher activation sparsity. Moreover, Dev-AEs show better classification performance based on the learned encodings, with units added in the final increment contributing the most.

Our findings indicate that an incremental latent space expansion fosters ordered, sparse, and more diverse representations, leading to more efficient use of representational capacity and improved classification accuracy, thereby offering a promising route toward interpretable and compact encodings.

Introduction

Nonlinear representation learning models, such as autoencoders, have achieved remarkable success across a wide range of machine learning tasks, demonstrating powerful feature extraction capabilities (Hinton and Salakhutdinov 2006). However, understanding the structure of the learned latent space remains a major challenge. In contrast to linear methods like Principal Component Analysis (PCA), which yield ordered and orthogonal components, the latent dimensions of autoencoder-like models are often intermixed, with information distributed across multiple units. This lack of organization in the latent space limits our understanding of how learned features are utilized for downstream tasks. Although numerous methods have been proposed to enhance the interpretability of such models, they often rely on explicitly enforcing statistical independence or orthogonality between latent features in the objective function (Burgess et al. 2018; Ladjal, Newson, and Pham 2019). This challenge raises the broader question of how meaningful and structured representations can emerge without imposing explicit constraints, a question also faced by biological learning systems.

Studies in developmental neuroscience and psychology have found that humans and animals undergo a gradual refinement of visual perception during early development (Cassia et al. 2002; Chandna 1991; Navon 1977). This refinement is accompanied by changes at the single-neuron level, shown by a refinement of receptive fields (Malone, Kumar, and Ringach 2007; Tschetter et al. 2018), and at the population and columnar activity level, shown by an increase in dimensionality and a decrease in correlation of activity patterns (Golshani et al. 2009; Powell et al. 2025, 2024). Inspired by these observations, we hypothesize that in a system whose latent space capacity grows incrementally during training, meaningful representations can emerge naturally. Such an incremental learning process could have several advantages: it may facilitate more efficient learning by initially constraining the parameter space, promote the formation of ordered features, and enhance generalization by encouraging low-dimensional representations in the early stages. Additionally, a progressive increase in complexity could lead to the emergence of more diverse features, progressing from simpler to more specialized representations over time.

Here, we investigate whether such a developmental learning scheme can lead to a more interpretable latent space and examine its potential functional advantages. Focusing on autoencoder models, we initially restricted the size of the bottleneck layer, thereby limiting the model's ability to capture fine-grained details. Then, we systematically increased the bottleneck size during training, allowing the model to learn increasingly complex representations. We examined whether this training scheme encourages more separated and ordered feature representations than standard training methods and PCA-initialization, a method known to promote ordered representations (Phan et al. 2025). We tested this training procedure across two architectures and datasets: fully connected networks (MLPs) trained on MNIST and convolutional networks (CNNs) trained on CIFAR-10. In this paper, we present results only for CNNs trained on CIFAR-10.

The results for the MLP experiments are qualitatively similar and are available in the supplement. Our findings demonstrate that this incremental training scheme promotes the formation of a more informative latent space, leading to improved classification performance and providing a novel perspective on neuro-inspired, structured representation learning.

Related work

Many approaches have been proposed to enhance the interpretability of AE models. A major line of work focuses on disentanglement, aiming to learn independent factors of variation within the latent space. Models such as the β-Variational Autoencoder (β-VAE) encourage statistical independence between latent variables, leading to more interpretable and transferable representations (Burgess et al. 2018). However, while these approaches promote independence, they do not typically impose any ordering or hierarchy among the latent features.

Ordered representations

A complementary direction explores ordered latent representations, inspired by the structure of PCA. The close connection between PCA and AE representations has long been recognized: early work demonstrated that simple linear neural networks can perform PCA (Baldi and Hornik 1989; Bourlard and Kamp 1988; Oja 1982), and subsequent studies extended this relationship to nonlinear settings (Scholz 2002). These PCA-like approaches indeed improved the organization of AE latent spaces. For instance, nested dropout has been shown to induce ordered representations in autoencoders, achieving a form of equivalence with PCA and enabling faster data retrieval using optimized encodings (Rippel, Gelbart, and Adams 2014). Moreover, PCA-based weight initialization of autoencoders has been found to improve convergence time and downstream classification compared to random initialization (Al-Digeil et al. 2022; Phan et al. 2025; Seuret et al. 2017). Our work builds on these insights but takes a different approach. Rather than using initialization techniques or objective-level constraints, we investigate how structured, ordered representations can emerge naturally with a neuroscience-inspired training procedure. In contrast to prior "PCA-like" autoencoders (Ladjal, Newson, and Pham 2019), we allow all existing connections to remain plastic as new units are added, enabling the latent representation to evolve dynamically. Unlike methods that retrain separate encoders for each latent dimension (Pham, Ladjal, and Newson 2022) or explicitly sort units by activity (Bertens 2016), our framework maintains a unified network and examines the emergent ordering induced purely by gradual representational growth.

Neurogenesis in ANNs

Neurogenesis describes the biological process by which the brain generates new functional neurons, occurring primarily during development and also in adulthood. It has been observed in various species and different brain regions (Chapouton, Jagasia, and Bally-Cuif 2007; Eriksson et al. 1998; Kaslin, Ganz, and Brand 2008). Our approach, which incrementally expands the latent space during training, can be viewed as a form of artificial neurogenesis, where new computational units in artificial neural networks are progressively added. The idea of dynamically growing architectures dates back to the 1980s (Ash 1989; Fahlman and Lebiere 1989) and has since reappeared in modern deep learning, where adaptive network growth has been applied to improve capacity and performance in supervised settings (Appolinary et al. 2024; Evci et al. 2022; Mitchell et al. 2024). Nevertheless, neurogenesis in deep neural networks remains underexplored compared to the far more established literature on neuron removal and pruning strategies (Laakom et al. 2022; Maile et al. 2022).

Parallel lines of research have examined developmental learning curricula, where hierarchical, coarse-to-fine label structures or dynamic expansion of network units facilitate continual learning and generalization (Bengio et al. 2009; Draelos et al. 2017; Stretcu et al. 2021; Vogelsang et al. 2018; Yoon et al. 2018). However, these approaches typically depend on externally structured data schedules or supervision to guide learning progression. In contrast, our method uses a self-supervised feature learning approach, maintaining a consistent input distribution and objective throughout training. Rather than enforcing structural or loss-based constraints, as in approaches such as Kusupati et al. (2022), we demonstrate that ordered, increasingly structured latent representations can emerge solely through incremental architectural expansion.

Emerging sparsity

Our training framework also yields sparse representations, consistent with the principle of sparse coding, in which information is captured by the activity of only a small subset of neurons (Olshausen and Field 1996; Ranzato, Boureau, and Cun 2007). Sparse coding has been viewed as a hallmark of biological neural systems, in which sensory cortices efficiently encode stimuli through selective, energy-saving activation patterns (Vinje and Gallant 2000). Such sparsity reduces metabolic cost and interference while enhancing memory capacity by engaging only task-relevant neurons. While prior work has often achieved these benefits by explicitly enforcing sparsity (Geadah et al. 2024; Makhzani and Frey 2014), our model produces sparsity as an emergent property.

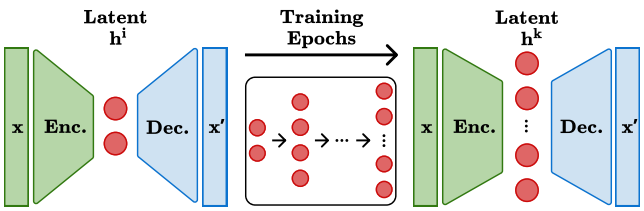

Figure 1: Schematics of a Developing Autoencoder. Training process of a Developing Autoencoder. The model is initialized with a small latent space h^i, which progressively grows to h^k throughout its training.

Methods

Developing Autoencoder

Building upon standard Autoencoders (AEs), we propose a novel training procedure that reflects the gradual expansion of representational capacity during development. The bottleneck size of the Dev-AE increases incrementally over training epochs (Fig. 1). When new bottleneck units are inserted into the network, new weights connecting these units to the encoder and decoder are drawn from a Gaussian distribution. To summarize:

1. Preservation of existing weights:

$$w_i^{(t+1)} = w_i^{(t)} \quad \text{for} \quad i = 1, ..., N^{(t)} \tag{1}$$

where $w_i^{(t)}$ denotes the weight at index i at epoch t.

2. Initialization of new weights:

$$\begin{aligned} &w_i^{(t+1)} = n_i \quad \text{for} \quad N^{(t)} < i \leq N^{(t+1)}, \\ &\text{where} \quad n_i \sim \mathcal{N}\big(\mu(w^{(t)}),\, \sigma(w^{(t)})\big) \end{aligned} \tag{2}$$

In this way, we can prevent a complete disruption of the existing network structure and thus avoid catastrophic forgetting. Meanwhile, previously learned weights can be further adjusted with added noise to the new weights, helping prevent redundant connectivity paths in the network. For a given bottleneck size, all weights are trained until convergence (training loss decrease is less than 0.1%), before the bottleneck size is increased by a certain percentage of the existing size.

Implementation: AE As a baseline control to Dev-AEs, we implemented AEs using PyTorch (Paszke et al. 2019)[1]. To demonstrate the transferability of our results, we used two different AE architectures trained on different datasets: an MLP with 3 fully connected encoder layers and 3 fully connected decoder layers, trained on the MNIST dataset, and a CNN with 5 convolutional encoder layers and 5 convolutional decoder layers, trained on the CIFAR-10 dataset. The dataset was split into a training set (4/6), a validation set (1/6), and a test set (1/6). All hidden layers used ReLU activation functions. The models were trained using stochastic gradient descent (SGD) with a batch size of 128 and a learning rate of 0.1 (tuned via grid search). Backpropagation was used for gradient calculation. We repeated experiments over 40 randomly initialized models. We include only the results of CNNs trained on the higher-dimensional, multi-colored CIFAR-10 dataset, due to space constraints. Qualitatively similar results were observed with the MLP architecture on MNIST and are available in the supplement.

Implementation: Dev-AE In all Dev-AE experiments presented here, we used the same architecture as AE controls. The bottleneck began with 6 units and was increased by approximately 70% at each growth stage, providing a practical balance between training speed and performance. This choice, however, did not qualitatively affect the results. Specifically, the developmental autoencoder expanded its bottleneck sequentially from 6 (6 epochs) $\rightarrow$ 10 (6 epochs) $\rightarrow$ 17 (7 epochs) $\rightarrow$ 29 (7 epochs) $\rightarrow$ 50 (8 epochs) $\rightarrow$ 85 (8 epochs) $\rightarrow$ 128 units (18 epochs). In total, each Dev-AE was trained for 60 epochs. In contrast, the standard autoencoder maintained a fixed bottleneck size of 128 and was trained for 60 epochs. While the number of neurons in the bottleneck provides an upper-bound of the possible dimensionality of the latent space, we explicitly estimated the intrinsic dimensionality of the bottleneck neuron activities by the nonlinear TwoNN method (Facco et al. 2017).

[1]https://github.com/davidvgr/developing-autoencoders

PCA-Autoencoder

While the standard AE serves as a baseline for analyzing similarities and differences arising from incremental bottleneck expansion, we also aimed to examine whether similar effects occur in a simpler, more easily implemented architecture. For this, we used a PCA-initialized autoencoder (PCA-AE). The PCA-AE builds on the same fixed bottleneck AE architecture but differs in the encoder's final layer and its training (Phan et al. 2025). We implemented the PCA-AE by initializing a regular AE, feeding training data through the model, and collecting the feature activations entering the final encoding layer. We then performed PCA with n components equal to the latent dimension. The PCA components were then set to the weights (W) of the encoder's last layer, and the bias was set to $-W\mu$, where μ is the mean of the features, for centering. During the first 20 training epochs, the encoder's final layer remained frozen, and the optimizer was updated only with the parameters of all other layers (lr=0.1). For the remaining 40 epochs, the layer was unfrozen, and its parameters were added to the optimizer.

Dimensionality

To validate whether incremental increases in the bottleneck size yield meaningful representational expansion, we estimated the intrinsic dimensionality of the bottleneck activations. We employed the Two Nearest Neighbors (TwoNN) estimator (Facco et al. 2017), which infers intrinsic dimensionality from local distance statistics. For each data point, TwoNN computes the ratio between the distances to its first and second nearest neighbors and fits the empirical distribution of these ratios to a theoretical model, providing an estimate of the underlying manifold dimensionality. We applied this method to the bottleneck activations of the test set.

Receptive fields

We defined the receptive field of a neuron as the input image that maximally activated that neuron. In our trained AEs and Dev-AEs, the receptive fields of the bottleneck layer units were estimated using the 'Activity Maximization' toolbox (Yosinski et al. 2015). We first initialized the input to a random image across all three color channels and then iteratively maximized the activation of a single bottleneck unit by performing gradient ascent on the input image with L_2 regularization.

We computed the power spectra by taking the 2D Fourier transform of the gray-scaled receptive fields and angularly averaging their squared moduli in the frequency domain.

We estimated the color bias of the neurons based on their receptive field structure. For this, we computed the Jensen-Shannon Divergence (JSD) across the distributions of values for each color channel (red, green, blue). A high JSD would indicate a stronger bias towards certain colors, to which the neurons would respond maximally. A low JSD would show low color selectivity and more color-invariant features.

Stability

To quantify how receptive fields of bottleneck neurons change over training, we computed the unsigned angular distance between each neuron's receptive field after each training epoch and its receptive field in the final epoch.

Input perturbations

To better understand the learned representations, we added different types of noise to the input images and tested how these noises affect the model encodings (as measured by the activity of the bottleneck units).

PC noise To identify how changes in input affect the encoding of each neuron group, we added noise to different input Principal Components (PCs) and assessed the activation in bottleneck units. We first computed PCs of all input images and then added Gaussian noise to selected groups of PCs. The PC group sizes matched the sizes in which bottleneck neurons were introduced during training of the Dev-AE. The noisy PCs were then transformed back into the original image space using the inverse PCA transform. Finally, we passed these noisy images to the encoder network and computed the change in activation compared to the original images for each bottleneck unit. We ranked the activation difference between PC groups, and this rank served as a measure of the unit's relative sensitivity to the manipulated PC group.

Frequency noise To assess the model's encoding robustness to input noise at different spatial frequencies, we evaluated classification accuracy after adding low- or high-frequency noise to the input images. A 10-class logistic regression classifier was trained on bottleneck-layer encodings to predict the class labels of the input images. We generated band-pass noise patterns by first performing a Fourier transform on a white noise image. A band-pass filter was then applied to isolate specific frequency ranges (defined by spectral diameter): ranges [0, 3] for low-frequency noise, [4, 7] for medium-frequency noise, and [8, 16] for high-frequency noise. An inverse Fourier transform was performed to generate the corresponding noise in the spatial domain, which was then added to the input images. To ensure a fair comparison, each noise image was normalized to have the same L_2 norm before being added to the original image. This allowed us to isolate the effect of different spatial frequencies on the model's performance.

Classification

To assess how well the encodings of the noisy images can be classified, a logistic regression classifier was trained on each model's encodings of clean input data. The classifier yielded classification accuracies for clean images (control) and for low-, medium-, and high-frequency noise. A two-tailed, paired-samples t-test was then used to assess the significance of the difference in accuracy between the AE and Dev-AE. We computed the absolute weights of the classifier and averaged them across all classes for each neuron, these averages were then normalized by their L_1 norm. These values represent the importance of each neuron for classification.

Sparsity

To measure sparsity, the activations of the neurons in the bottleneck were measured as the test set was passed through. Specifically, we assessed two metrics: the percentage of images that left each neuron inactive (which we then aggregated by neuron group), and the total number of inactive neurons per individual image. Because the bottleneck consists of linear units without an immediately following ReLU activation, which is applied only after the subsequent reshaping step, the latent units do not naturally output exact zeros. Therefore, a neuron was considered inactive for a specific image if its absolute activation was less than 1% of the highest absolute activation of all neurons for that image.

Results

Dev-AE training

During training, the loss of standard AE models decreased rapidly, converging after approximately 50 epochs. PCA-initialized AEs showed a similar trend, with an even faster initial decrease of loss (Fig. 2A). The Dev-AE showed a similar rapid initial decline in loss, reaching early convergence with a small bottleneck, but continued to improve when new units were added (Fig. 2A). Both models were trained for the same total number of epochs, and achieved comparable image reconstruction performance at the final epoch (Fig. 2A, C). To verify that our developmental training procedure indeed leads to a gradual expansion of representational capacity, we quantified the dimensionality of bottleneck activations over training epochs. The Dev-AE showed a steady increase in dimensionality across training epochs and eventually reached a higher dimensionality than that of standard AEs and PCA-AEs (Fig. 2B). These results confirm that the incremental training scheme enables a progressive expansion of representational capacity without compromising reconstruction quality.

Receptive field structure

To characterize the representational structure of the latent space, we first analyzed the receptive fields of units in the bottleneck layer. In the standard AE, these receptive fields showed similar frequency spectra across units, indicating a largely homogeneous representation (Fig. 3A, B). However, in the Dev-AE, the receptive fields were organized in a hierarchy: bottleneck units added earlier to the network were tuned to coarse, low-frequency image features, whereas later-added units appeared to detect more fine-scale structures in the input images (Fig. 3A, B). PCA-AEs showed a mild frequency ordering, though less pronounced

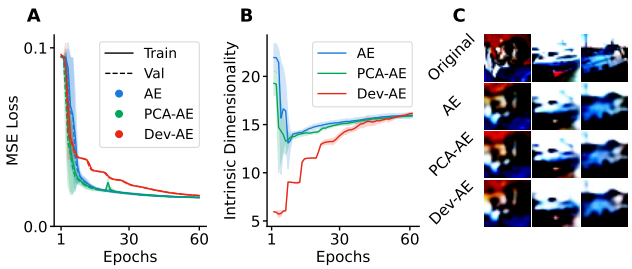

Figure 2: All models reach comparable performance and latent space dimensionality after training convergence. A) Training and validation loss over training epochs. Upon convergence of the validation loss, the Dev-AE's bottleneck layer dimensionality expands by approx. 70%. The shaded area shows the mean $\pm$ standard deviation. B) Intrinsic dimensionality (Facco et al. 2017) throughout the training process in the bottleneck layer. C) Representative examples of original images and their reconstructions.

than in the Dev-AE. In general, the receptive fields continuously changed over training epochs, but later added bottleneck units did not significantly disrupt earlier ones (Fig. 4). Thus, the structure of the earlier learned receptive fields remained relatively stable during training and robust against newly added latent units, and as a consequence, receptive fields were ordered by the time of insertion into the network. Moreover, AE and PCA-AE receptive fields exhibited a strong bias towards a certain color (Fig. 3D), while receptive fields of the Dev-AE had smaller and more equally distributed values over the three color channels (Fig. 3C). These results indicate that the incremental training scheme fosters a more ordered and diverse latent representation that captures different image features.

Coding

To examine how the learned ordered representation in the Dev-AE influences information coding, we analyzed the contribution of individual bottleneck units to the encoding of the input data. We added noise to selected subsets of image PCs and measured its effect on bottleneck activations. In Dev-AEs, noise added to the first few image PCs predominantly affected units added early in training, whereas noise applied to later PCs primarily influenced units inserted at later stages (Fig. 5A). Consistent with our receptive fields analysis above, this result suggests that an intrinsic ordering emerges in Dev-AE bottleneck units: earlier units learn components with larger input variance, and later units learn subsequent components that explain smaller input variance. Accordingly, the Dev-AE exhibited a clear alignment between its latent units and the principal components of the input data. In contrast, standard AEs showed no such structure, with PC-dependent noise affecting bottleneck activations more uniformly, and PCA-AEs displayed only weak alignment compared to the Dev-AE (Fig. 5A). Next, we evaluated how efficiently the learned latent representations support downstream classification. Encodings from the Dev-AE achieved higher classification accuracy than those from the standard AE or the PCA-AE (Fig. 5B). When input images were corrupted with noise at varying spatial frequencies, performance decreased in all models, yet the Dev-AE remained markedly more robust to noise of low and high frequency ranges, but not for medium frequency noise (Fig. 5B). These results suggest that different classes are better separated in Dev-AE latent space without being specifically optimized for classification. We then looked into which bottleneck units were more important for classification by comparing the L_1 normalized weights of the linear classifier. We found that in Dev-AEs, later neuron groups were heavily weighted, a phenomenon not shown in standard AEs and less so in PCA-AEs (Fig. 5C). Our findings suggest that in Dev-AEs, latent units are specialized and ordered in their coding functions, with earlier units encoding large-variance components (important for reconstruction) and later units encoding fine features (important for classification).

Sparsity

To assess how the developmental training paradigm affects bottleneck unit activity, we quantified the sparsity of the latent representations by measuring the proportion of input images that do not activate a neuron and the count of inactive bottleneck neurons per image. In Dev-AEs, earlier neuron groups had a lower percentage of non-active neurons, and later neuron groups tended to be more silent (Fig. 6A). Overall, the Dev-AE bottleneck produced sparser activations than those of standard AE and PCA-AE, consistent with more efficient, biologically plausible coding (Fig. 6B). Neurons in the Dev-AE bottleneck exhibited a gradient in sparsity levels between early and late units. These observations suggest that incremental expansion gives rise to a sparse coding scheme reminiscent of the efficient coding observed in biological neural systems.

Discussion

In this work, we demonstrate that an incremental increase in latent dimensionality during unsupervised learning facilitates the formation of disentangled, rank-ordered representations, in which different groups of bottleneck neurons encode distinct aspects of the input features, leading to improved performance in classification tasks. Our findings are in line with previous studies showing that imposing constraints during training can be beneficial for learning well-structured and informative latent codes (Bertens 2016; Ladjal, Newson, and Pham 2019; Rippel, Gelbart, and Adams 2014). While the specific nature of the constraints may vary across approaches, our results reinforce the general principle that early restrictions can dynamically guide the learning process towards more desirable representational properties.

Unlike the hierarchical organization observed in CNNs, where a progression from simple receptive fields to more complex features emerges across successive layers during training (Zeiler and Fergus 2013), our Dev-AE exhibits an ordered latent code of global and local features within the same layer, providing insights into how receptive fields can be organized within a single layer. Specifically, our results

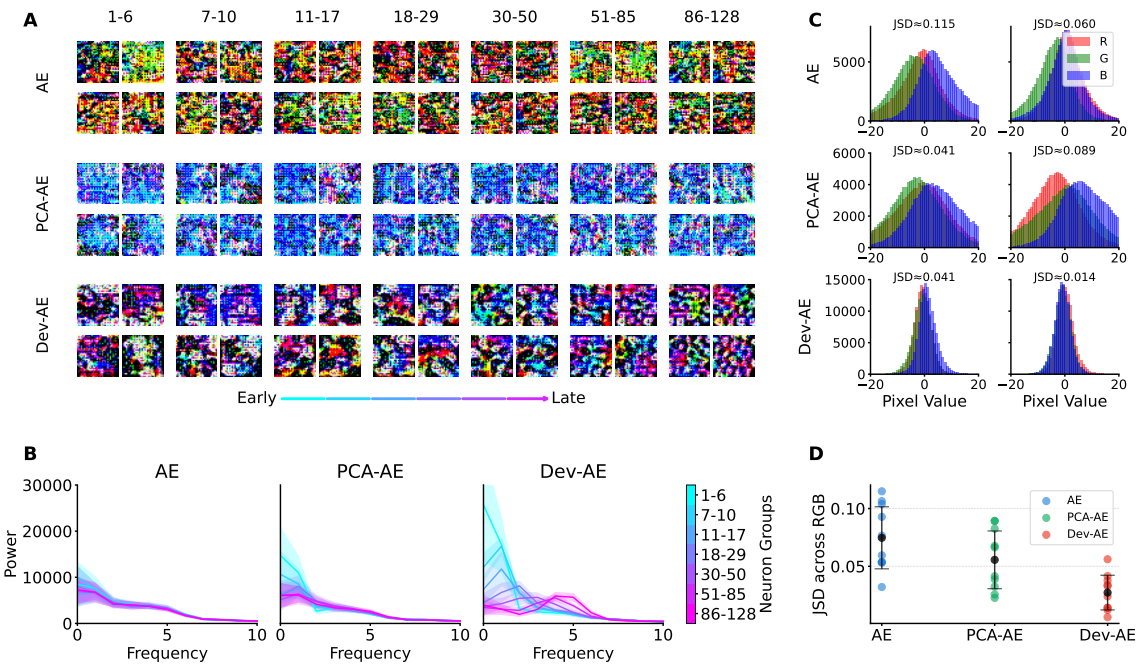

Figure 3: Receptive fields in Dev-AEs are ordered by developmental progression. A) Example receptive fields of bottleneck units in an AE (top), a PCA-AE (middle), and a Dev-AE (bottom). In the Dev-AE, the example receptive fields are shown from early-added (left) to late-added (right) units. B) Power spectra of AE (left), PCA-AE (middle), and Dev-AE receptive fields (right). Colors indicate the neuron groups shown in (A). Mean and standard deviation are shown across all model initializations. C) Distribution of receptive field values in different color channels. D) Quantification of the difference in distributions of different color channels, using Jensen-Shannon Divergence.

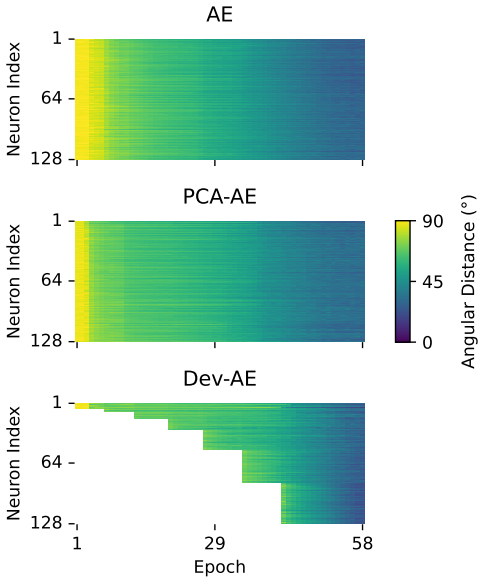

Figure 4: Stability of receptive fields over training. The unsigned angular distance between the receptive fields after each epoch and the receptive fields of the final epoch per neuron is shown.

suggest that a dynamic relaxation of constraints, implemented by progressively increasing the representational capacity of the latent space, plays a critical role in guiding this organization, allowing for the simultaneous representation of both large and small spatial scales within the same level of the hierarchy.

The emerging specialization of latent units in the Dev-AE enhances downstream class separability while maintaining robustness to input noise, demonstrating that capacity expansion during learning can yield representations that are more compact, interpretable, and structured. Importantly, the observed increasing sparsity across subsequently added groups of latent units suggests an emergent coding hierarchy that mirrors principles of efficient and energy-conserving representation found in biological systems (Olshausen and Field 1996; Vinje and Gallant 2000). Early units, with higher activity and broader receptive fields, provide a coarse yet stable encoding backbone, while later, sparser units refine the representation through selective activation. This pattern implies that gradual capacity increases do not merely expand representational space but actively shape how information is distributed and utilized within it.

This gradual increase in bottleneck capacity may also serve as a mechanism for estimating the dataset's intrinsic dimensionality (Pope et al. 2021). By progressively increasing the size of the bottleneck, the optimal capacity can be determined at which the reconstruction error shows no further

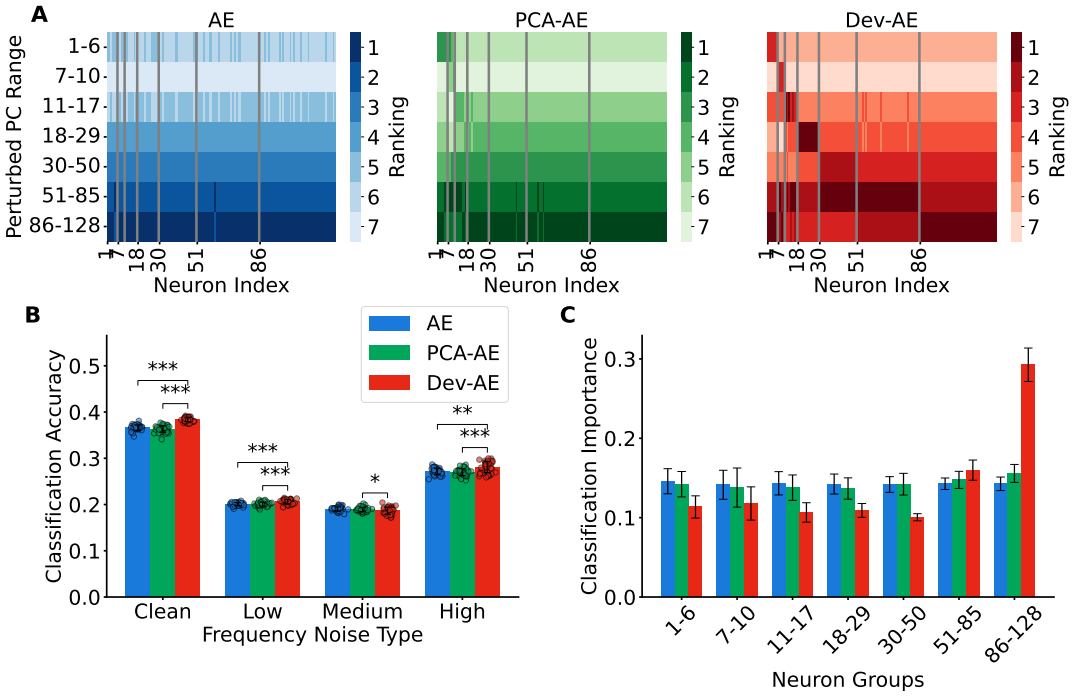

Figure 5: Bottleneck units are functionally ordered. A) Image-PC noise analysis reveals an alignment of PC range and neuron index in the bottleneck of the Dev-AE (right), but only slightly in PCA-AE (middle), and not in standard AE (left). Colors indicate the ranking of the activation difference of noise-perturbed inputs relative to all neuron groups. Gray vertical lines separate different neuron groups. B) Classification accuracy for original (clean) and noise-corrupted images. Noise was added at different spatial frequency bands. Classification used all bottleneck units after training. Dev-AEs achieve significantly higher accuracy for both clean and noisy inputs across low- and high-frequency ranges (t-test, p-values Bonferroni-corrected). For medium-frequency noise, the difference is less pronounced, with the Dev-AE classification accuracy slightly lower than that of the PCA-AE. C) Classifier weights indicate higher importance of later added neural units. Mean and standard deviation are shown across all model initializations in (B) and (C).

significant improvement (Ho, Zhao, and Wandelt 2025). By starting with a low-dimensional representation that captures only the most salient features, the model is initially constrained to avoid overfitting the training data, particularly in the early stages when the model is learning the coarse statistics of the input. As dimensionality increases, the model can incorporate finer-grained details and nuances in the data. We show that such strategy can lead to the development of a more robust and generalizable code that is less susceptible to noise and variations in the input, effectively guiding the model to form informative latent representations.

In comparison to previous work on autoencoders (Ladjal, Newson, and Pham 2019), we found that it was not necessary to introduce strict constraints on the network (e.g., keep the weights of previous units fixed when adding new latent axes or impose regularization or additional loss terms). Our work provides evidence that learned latent features are not perturbed much when progressively increasing the bottleneck capacity. Lastly, our results indicate that the incremental increase of units within the bottleneck layer facilitates higher sparsity in the learned features, enabling better classification performance of the encodings. Future work could explore strategies for determining optimal scheduling of growth stages during learning, as well as on dynamic capacity adjustments across other layers (Maile et al. 2022; Mitchell et al. 2024).

Overall, our work contributes to ongoing research on interpretable representation learning and proposes a comparatively simple, biologically inspired mechanism for creating informed latent spaces in autoencoder models by incrementally expanding their representational capacity. This approach reveals the benefits of incorporating ideas from developmental biology into neuro-inspired artificial neural network architectures.

Acknowledgements

We would like to thank Gordon Smith for helpful discussions and Santiago Galella for helpful feedback on the manuscript. This work was supported by research grant Deutsche Forschungsgemeinschaft DIP "Neurobiology of Forgetting" (J.E., M.K.) and Bundesministerium für Bildung und Forschung (BMBF 01GQ2002) (D.K.,M.K.), DFG Research Unit FOR 5368 ARENA (D.K.,M.K.).

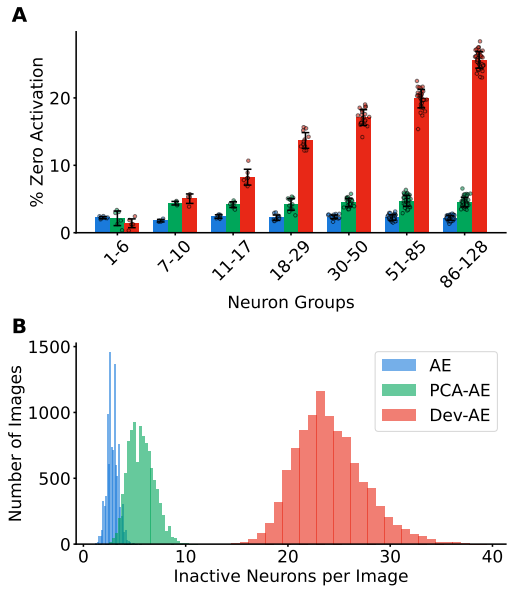

Figure 6: Sparsity of bottleneck neural activation. A) The mean percentage of images not triggering a response for units in each neuron group. Late units of the Dev-AE show higher sparsity. Mean and standard deviation are shown across all model initializations. B) Distribution of inactive neurons per image. The histogram shows the frequency of images that yield a specific count of inactive neurons. Neurons in the Dev-AE show higher sparsity, with a distribution centered around 23 inactive neurons. In contrast, the AE and PCA-AE models show much lower inactivity, peaking at approximately 3 and 6 inactive neurons per image, respectively.

References

Al-Digeil, M.; Grinberg, Y.; Melati, D.; Dezfouli, M. K.; Schmid, J. H.; Cheben, P.; Janz, S.; and Xu, D.-X. 2022. PCA-Boosted Autoencoders for Nonlinear Dimensionality Reduction in Low Data Regimes. ArXiv:2205.11673.

Appolinary, B.; Deaconu, A.; Yang, S.; Qingze; and Li. 2024. Self Expanding Convolutional Neural Networks. ArXiv:2401.05686.

Ash, T. 1989. Dynamic Node Creation in Backpropagation Networks. *Connection Science*, 1(4): 365–375.

Baldi, P.; and Hornik, K. 1989. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1): 53–58.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 41–48. Montreal Quebec Canada: ACM. ISBN 9781605585161.

Bertens, P. 2016. Rank Ordered Autoencoders. ArXiv:1605.01749.

Bourlard, H.; and Kamp, Y. 1988. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4-5): 291–294.

Burgess, C. P.; Higgins, I.; Pal, A.; Matthey, L.; Watters, N.; Desjardins, G.; and Lerchner, A. 2018. Understanding disentangling in β-VAE. ArXiv:1804.03599.

Cassia, V. M.; Simion, F.; Milani, I.; and Umiltà, C. 2002. Dominance of global visual properties at birth. *Journal of Experimental Psychology: General*, 131(3): 398–411. Place: US Publisher: American Psychological Association.

Chandna, A. 1991. Natural history of the development of visual acuity in infants. *Eye (London, England)*, 5 (Pt 1): 20–26.

Chapouton, P.; Jagasia, R.; and Bally-Cuif, L. 2007. Adult neurogenesis in non-mammalian vertebrates. *BioEssays*, 29(8): 745–757.

Draelos, T. J.; Miner, N. E.; Lamb, C. C.; Cox, J. A.; Vineyard, C. M.; Carlson, K. D.; Severa, W. M.; James, C. D.; and Aimone, J. B. 2017. Neurogenesis deep learning: Extending deep networks to accommodate new classes. In *2017 International Joint Conference on Neural Networks (IJCNN)*, 526–533. ISSN: 2161-4407.

Eriksson, P. S.; Perfilieva, E.; Björk-Eriksson, T.; Alborn, A.-M.; Nordborg, C.; Peterson, D. A.; and Gage, F. H. 1998. Neurogenesis in the adult human hippocampus. *Nature Medicine*, 4(11): 1313–1317.

Evci, U.; Merriënboer, B. v.; Unterthiner, T.; Vladymyrov, M.; and Pedregosa, F. 2022. GradMax: Growing Neural Networks using Gradient Information. ArXiv:2201.05125.

Facco, E.; d'Errico, M.; Rodriguez, A.; and Laio, A. 2017. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1): 12140.

Fahlman, S.; and Lebiere, C. 1989. The Cascade-Correlation Learning Architecture. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

Geadah, V.; Barello, G.; Greenidge, D.; Charles, A. S.; and Pillow, J. W. 2024. Sparse-Coding Variational Autoencoders. *Neural Computation*, 36(12): 2571–2601.

Golshani, P.; Gonçalves, J. T.; Khoshkhoo, S.; Mostany, R.; Smirnakis, S.; and Portera-Cailliau, C. 2009. Internally Mediated Developmental Desynchronization of Neocortical Network Activity. *Journal of Neuroscience*, 29(35): 10890–10899.

Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science (New York, N.Y.)*, 313(5786): 504–507.

Ho, M.; Zhao, X.; and Wandelt, B. D. 2025. Ordered embeddings and intrinsic dimensionalities with information-ordered bottlenecks. *Mach. Learn. Sci. Technol.*, 6(3): 035010.

Kaslin, J.; Ganz, J.; and Brand, M. 2008. Proliferation, neurogenesis and regeneration in the non-mammalian vertebrate brain. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1489): 101–122.

Kusupati, A.; Bhatt, G.; Rege, A.; Wallingford, M.; Sinha, A.; Ramanujan, V.; Howard-Snyder, W.; Chen, K.; Kakade, S.; Jain, P.; and Farhadi, A. 2022. Matryoshka Representation Learning. In *Advances in Neural Information Processing Systems*.

Laakom, F.; Raitoharju, J.; Iosifidis, A.; and Gabbouj, M. 2022. Reducing Redundancy in the Bottleneck Representation of the Autoencoders. ArXiv:2202.04629.

Ladjal, S.; Newson, A.; and Pham, C.-H. 2019. A PCA-like Autoencoder. ArXiv:1904.01277.

Maile, K.; Rachelson, E.; Luga, H.; and Wilson, D. G. 2022. When, where, and how to add new neurons to ANNs. In *Proceedings of the First International Conference on Automated Machine Learning*, 18/1–12. PMLR.

Makhzani, A.; and Frey, B. 2014. k-Sparse Autoencoders. ArXiv:1312.5663 version: 2.

Malone, B. J.; Kumar, V. R.; and Ringach, D. L. 2007. Dynamics of Receptive Field Size in Primary Visual Cortex. *Journal of Neurophysiology*, 97(1): 407–414. Publisher: American Physiological Society.

Mitchell, R.; Menzenbach, R.; Kersting, K.; and Mundt, M. 2024. Self-Expanding Neural Networks. ArXiv:2307.04526.

Navon, D. 1977. Forest before trees: The precedence of global features in visual perception. *Cognitive Psychology*, 9(3): 353–383.

Oja, E. 1982. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3): 267–273.

Olshausen, B. A.; and Field, D. J. 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583): 607–609.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. ArXiv:1912.01703.

Pham, C.-H.; Ladjal, S.; and Newson, A. 2022. PCA-AE: Principal Component Analysis Autoencoder for Organising the Latent Space of Generative Networks. *Journal of Mathematical Imaging and Vision*, 64(5): 569–585.

Phan, N.; Nguyen, T.; Dang, U.; Halvorsen, P.; and Riegler, M. A. 2025. Principal Components for Neural Network Initialization. ArXiv:2501.19114.

Pope, P.; Zhu, C.; Abdelkader, A.; Goldblum, M.; and Goldstein, T. 2021. The Intrinsic Dimension of Images and Its Impact on Learning. ArXiv:2104.08894.

Powell, N. J.; Hein, B.; Kong, D.; Elpelt, J.; Mulholland, H. N.; Holland, R. A.; Kaschube, M.; and Smith, G. B. 2025. Developmental maturation of millimeter-scale functional networks across brain areas. *Cerebral Cortex*, 35(2): bhaf007.

Powell, N. J.; Hein, B.; Kong, D.; Elpelt, J.; Mulholland, H. N.; Kaschube, M.; and Smith, G. B. 2024. Common modular architecture across diverse cortical areas in early development. *Proceedings of the National Academy of Sciences*, 121(11): e2313743121. Publisher: Proceedings of the National Academy of Sciences.

Ranzato, M. a.; Boureau, Y.-l.; and Cun, Y. 2007. Sparse Feature Learning for Deep Belief Networks. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Rippel, O.; Gelbart, M. A.; and Adams, R. P. 2014. Learning ordered representations with nested dropout. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, II–1746–II–1754. JMLR.org.

Scholz, M. 2002. Nichtlineare Hauptkomponentenanalyse auf Basis neuronaler Netze. Diploma thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II.

Seuret, M.; Alberti, M.; Liwicki, M.; and Ingold, R. 2017. PCA-Initialized Deep Neural Networks Applied to Document Image Analysis. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, 877–882. ISSN: 2379-2140.

Stretcu, O.; Platanios, E. A.; Mitchell, T. M.; and Póczos, B. 2021. Coarse-to-Fine Curriculum Learning. ArXiv:2106.04072.

Tschetter, W. W.; Govindaiah, G.; Etherington, I. M.; Guido, W.; and Niell, C. M. 2018. Refinement of Spatial Receptive Fields in the Developing Mouse Lateral Geniculate Nucleus Is Coordinated with Excitatory and Inhibitory Remodeling. *The Journal of Neuroscience*, 38(19): 4531–4542.

Vinje, W. E.; and Gallant, J. L. 2000. Sparse Coding and Decorrelation in Primary Visual Cortex During Natural Vision. *Science*, 287(5456): 1273–1276.

Vogelsang, L.; Gilad-Gutnick, S.; Ehrenberg, E.; Yonas, A.; Diamond, S.; Held, R.; and Sinha, P. 2018. Potential downside of high initial visual acuity. *Proceedings of the National Academy of Sciences*, 115(44): 11333–11338.

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2018. Lifelong Learning with Dynamically Expandable Networks. ArXiv:1708.01547.

Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; and Lipson, H. 2015. Understanding Neural Networks Through Deep Visualization. ArXiv:1506.06579.

Zeiler, M. D.; and Fergus, R. 2013. Visualizing and Understanding Convolutional Networks. ArXiv:1311.2901.

Supplementary Material

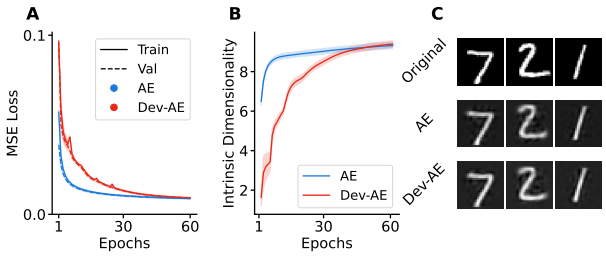

Figure 7: Performance and training dynamics of MLP models on MNIST. Both standard AE and Dev-AE models reach comparable final performance and latent space dimensionality upon convergence. A) Training and validation loss over training epochs. B) Intrinsic dimensionality (Facco et al. 2017) throughout the training process. C) Representative examples of original images and their reconstructions. These trends align with observations from CNN models trained on CIFAR-10 (Fig. 2).

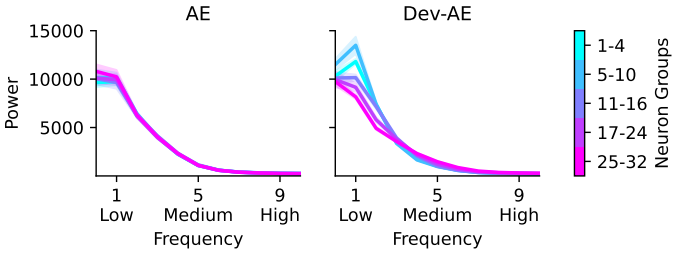

Figure 8: Power spectra of receptive fields for AE and Dev-AE (MNIST). Plots display the spectral properties of receptive fields for AE (left) and Dev-AE (right). While distinct neuron groups show spectral separation in Dev-AEs, the effect is less pronounced compared to the more complex CIFAR-10 dataset (Fig. 3B).

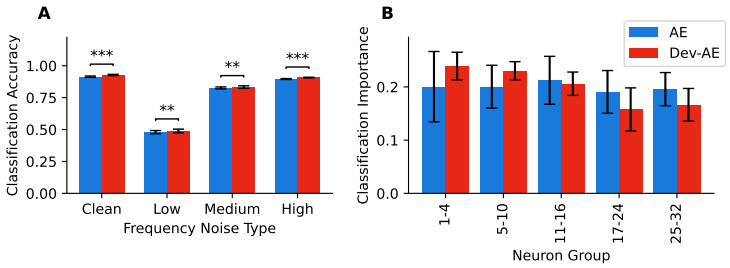

Figure 9: Classification accuracy and feature importance in MLP models (MNIST). A) Classification accuracy across frequencies. The Dev-AE models demonstrate a small but statistically significant accuracy advantage across all frequencies. B) Contribution of neuron groups to classification decisions. While AE models rely equally on all groups, Dev-AE models show higher importance in early neuron groups. Note that this trend opposes that of CNNs on CIFAR-10, where later groups dominate (Fig. 5C).