Neurocomputing 701 (2026) 134496

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Beyond topography: Topographic regularization improves robustness and reshapes representations in convolutional neural networks

Nhut Truong, Uri Hasson*

Center for Mind/Brain Sciences (CIMeC), University of Trento, Rovereto, Italy

ARTICLE INFO

Communicated by L. Zhu

Keywords:
Topographic neural networks
Robustness
Representation learning
Spatial regularization

ABSTRACT

Topographic convolutional neural networks (TCNNs) are computational models that can simulate aspects of the brain's spatial and functional organization. However, it is unclear whether and how different types of topographic regularization shape robustness, representational structure, and functional organization during end-to-end training. We address this question by studying TCNNs trained with two local spatial losses applied to a penultimate-layer topographic grid: *i)* Weight Similarity (WS), whose objective penalizes differences between neighboring units' incoming weight vectors, and *ii)* Activation Similarity (AS), whose objective penalizes differences between neighboring units' activation patterns over stimuli. Relative to matched non-topographic controls, both regularizers generally improved robustness to input perturbations, reduced activation sparsity, and altered unit-level tuning profiles. In a larger-scale setting, topographic regularization also improved robustness to adversarial attacks. WS, but not AS, consistently increased robustness to weight perturbations and produced stronger signatures of functional localization. Both AS and WS also changed orientation tuning, symmetry sensitivity, and eccentricity profiles relative to control models. These findings show that local topographic regularization can improve robustness during end-to-end training while systematically reshaping representational structure, and that these effects depend strongly on how similarity is imposed.

1. Introduction

Topographic models are computational models that provide an analogy to the cortical organization of the brain by instantiating a cortical-like map. In these models, each unit is assigned a position in a 2D grid [1], which defines a notion of distance between grid units. A topographic loss term can then be defined, which encourages spatially local similarity or smoothness in units' responses (or parameters) across the grid. It has been shown that when a task loss (e.g., cross-entropy) is optimized jointly with a topographic loss, both shallow [2] and deep [3–12] topographic networks can produce competitive task performance, while producing spatially organized responses that resemble cortical functional maps. For example, they produce angular and orientation preferences such as those observed in early visual cortex, as well as category-selective clusters analogous to those reported in higher-level visual cortex [e.g.,3–9]. These models can also capture spatial biases in human behavior, recognizing objects more accurately when they appear in locations where they are most frequently experienced [5]. Outside the vision domain, topographic modeling has reproduced tonotopic organization in auditory cortex [12] and higher-level language organization [e.g.,10,11]. Topographic networks can also be more strongly pruned, suggesting a sparser distribution of weight values [1,8].

The correlated activation patterns produced by topographic models carry parallels to brain activity, where nearby neurons often fire together, with correlations sometimes exceeding $r = 0.4$ in certain cortical circuits [e.g.,13,14]. Although such correlations reduce the degrees of freedom and representational capacity in biological neural populations [13], they may also offer benefits such as robustness, as redundant neurons can compensate for one another [15]. A similar trade-off has been discussed for computational models: on the one hand, correlated units encode overlapping information and can impair decoding performance [16], but on the other, moderate correlations can act as a form of regularization, producing more compact representations or giving priority to more informative features [1]. In summary, correlations constrain capacity but may provide computational advantages.

While topographic models are a topic of emerging interest in computational and systems neuroscience [3–12,17–29], there is a substantial

* Corresponding author.
Email addresses: leminhnhut.truong@unitn.it (N. Truong), uri.hasson@unitn.it (U. Hasson).

https://doi.org/10.1016/j.neucom.2026.134496
Received 12 January 2026; Received in revised form 22 April 2026; Accepted 12 July 2026
Available online 14 July 2026

gap in our understanding of the computational consequences of imposing topographic regularization. This is an important open question not only for computational neuroscience, where topographic networks are mainly studied for their ability to reproduce brain-like spatial organization of responses, but also for machine learning, where the impacts of these locally induced correlations are not well understood.

Going beyond a focus on spatial organization, we therefore ask whether these correlations provide any computational advantages, such as improved robustness to noise, and how they, in turn, shape the representations learned by topographic networks. To our knowledge, neither of these issues has been systematically examined in controlled comparisons to date. Specifically, we address these questions using two related criteria:

1. **Network robustness**: We introduce internal noise by perturbing the readout weight matrix connecting the penultimate (topographic) layer to the classification layer [30–32]. We also evaluate robustness under external noise by corrupting input images during the testing phase, including both adversarial and non-adversarial noise. We test how resilient topographic models are compared to non-topographic models in terms of both representational stability and classification accuracy.
2. **Representational properties**: We compare topographic models to size-matched non-topographic control models. We quantify unit-level sparsity and entropy, similarity of weights and activations, the dimensionality of the latent space, and the tendency of models to produce category-selective "expert units". We further characterize the spatial organization of representations by measuring the smoothness of topographic maps and the extent to which activity is functionally localized (i.e., whether similarly responding units are spatially clustered). Finally, we test whether topographic regularization changes functional organization by quantifying orientation, eccentricity, and angular tuning in topographic and non-topographic networks.

To understand the computational effects of topographic regularization, we use local spatial losses and end-to-end training. Prior work has often relied on global spatial losses that include a distance-dependent term over all unit pairs. These global losses directly encourage high activation similarity between nearby units as well as low activation similarity between distant ones [e.g.,1,4]. They therefore effectively *impose* functional localization by suppressing long-range correlations. This makes localized clusters an outcome of the training objective rather than an emergent property. We therefore restrict the regularizer to local interactions only [e.g.,5]. In addition, in two studies we train topographic networks end-to-end, because post hoc projection methods operating on frozen pretrained features [e.g., self-organized mappings,7,9,18,21] cannot address whether topographic regularization impacts feature learning itself. In a third study, where a larger-scale architecture is evaluated, we allow fine-tuning a non-frozen backbone.

Regarding the technical implementation, correlated activations in topographic models are often produced using an objective that encourages similarity between neighboring units' activation patterns [e.g.,1,4,10,17]. An alternative is to impose similarity constraints on incoming weights [5], which may be more biologically plausible: correlated activity should emerge due to shared afferent connectivity instead of being directly enforced at the level of activations. For this reason, we also avoid explicit lateral connections, as correlations in the brain often reflect shared afferent input [33]. To compare these two alternatives, we contrast two local topographic losses (Fig. 1): **Activation Similarity (AS)**, whose objective is to produce correlated activations in adjacent units, and **Weight Similarity (WS)**, whose objective is to produce similar afferent weight vectors in adjacent units. In both cases, we consider only local neighbors.

We find that AS and WS training produced qualitatively different computational outcomes. Compared with AS and control (non-topographic) models, WS training produced representations that were more robust to perturbations of the readout weight matrix and stronger functional localization, with similarly responding units positioned at closer spatial distances. AS training produced representations that were more robust to degraded inputs, but showed weaker functional localization than WS. Furthermore, in one case, AS produced highly saturated connectivity over the entire grid, and while WS produced a compression of the latent space, AS did not. In addition, as compared to control models, AS and WS produced different emphases on angular, orientation, and symmetry tuning. While this does not mean that topography instantiates a homologous functional organization, it shows that the regularizers change the representational structure of the network. Taken together, our findings suggest that topographic regularization does not simply reorganize the weight matrix to maintain classification, but produces a markedly different feature space.

2. Related work

2.1. Topographic regularization in artificial neural networks (ANNs)

Several studies have examined how spatially organized response patterns, often accompanied by increased local correlations, can be induced in ANNs by applying topographic objectives. One approach is end-to-end topographic training, which introduces a topographic loss term directly into the training objective. Typically, a joint loss combines a task loss (e.g., cross-entropy for classification) with an additional topographic spatial term. Early work [2] introduced a spatial loss in small-scale,

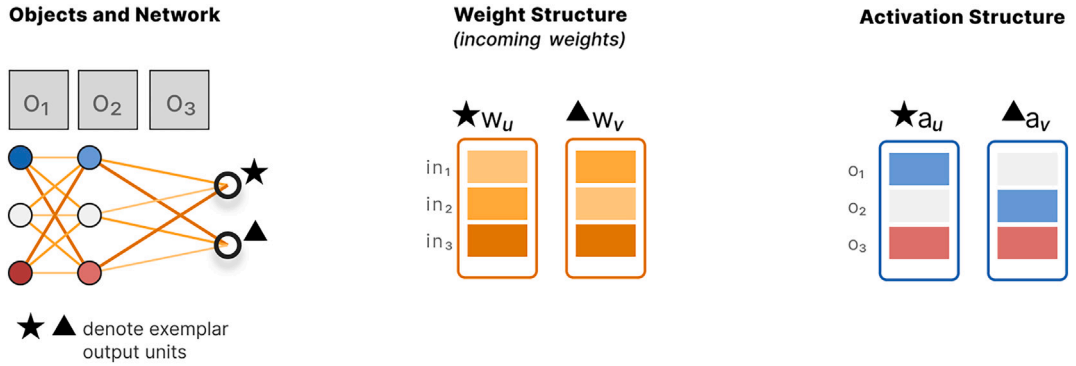


Fig. 1. Overview of main concepts. Objects (O_1–O_3) are passed to a neural network. Two example output units ($\star = u$ and $\blacktriangle = v$) are indicated. For each example unit, an *activation vector* ($\mathbf{a}_u, \mathbf{a}_v$) summarizes that unit's responses across objects, and a *weight vector* ($\mathbf{w}_u, \mathbf{w}_v$) summarizes the incoming weights from the preceding layer. Activation similarity is computed by correlating activation vectors across objects. Weight similarity is computed as the Euclidean distance between weight vectors. Training optimizes a joint objective that minimizes the similarity loss while preserving task performance.

fully connected networks, penalizing the weight magnitude proportionally to the physical distances between the pre- and post-synaptic units' assigned positions. As a result, adjacent units showed similar tuning. Others [e.g.,1] introduced losses that enforce an inverse relationship between the activation similarity of spatially arranged convolutional filters and their distances. This improved robustness to pruning while maintaining task performance. Related approaches that produce cortical-like spatial organization include penalizing connections between spatially remote units [3], introducing lateral pooling between neighboring filters [6], and adding topographic structure to variational autoencoders [20]. Topography can also emerge without an explicit spatial loss when kernels in deeper layers are defined as averages of partially overlapping kernels from a preceding layer, producing smooth transitions across kernel maps [6,26].

While some approaches employ a learning objective that aims at an inverse correlation between the similarity of unit activations and their distance, several studies use local constraints alone, requiring that a unit or feature map be similar to its immediate neighbors. Most generally, global objectives tend to penalize cases where highly correlated units are positioned far apart. They therefore discourage the formation of multiple, disconnected clusters of similarly responding units, and instead produce spatially contiguous regions of similarly responding units. This is not required by local objectives. In Lu et al. [5], the authors examine whether a fully topographic end-to-end architecture can reproduce signatures of cortical organization, using a local weight-similarity constraint applied across multiple layers. Another study [6] has a similar goal, but uses an activation-smoothing method based on lateral averaging of entire feature maps, with no supervised similarity loss. In both cases, the focus is on whether the resulting models reproduce biologically plausible spatial organization. In contrast, we use a single topographic bottleneck inserted into a standard architecture, with regularization applied only at that stage. This allows us to compare AS and WS regularizers under matched settings that are compatible with typical deep-learning pipelines.

Other work used a hybrid approach [4], in which unit locations are first optimized based on activation similarity computed from a pretrained model, and then these locations are held fixed during subsequent training. This model reproduced spatial properties of both early and high-level visual cortical features and outperformed standard networks on biological benchmarks.

Finally, there exist topographic models that are based on post hoc projection of pretrained representations, for example, via self-organizing maps (SOMs). In these approaches, a fixed, pretrained feature space is mapped onto a two-dimensional grid with the objective of producing spatial clusters of similarly responding units [7,9,18,21]. Conceptually, these methods learn a spatial embedding of *existing* representations: they operate on frozen features and therefore cannot address how topographic regularization shapes feature learning during end-to-end task training [34].

2.2. Impact of topography on representational structure

Beyond producing correlated responses, topographic regularization also influences the representational structure of ANNs. Several studies have reported that such regularization reduces the effective dimensionality of latent representations [4,6,8] and improves robustness to pruning [1,8] and adversarial attacks [6,25,26].

At the same time, inter-unit correlations are sometimes considered detrimental in machine learning research. For example, the Barlow Twins architecture [35] maximizes agreement between paired views (original and distorted images) while penalizing correlated activity across units computed from the off-diagonal terms of the batch-wise cross-correlation matrix. The study demonstrates that reducing correlations improves classification accuracy. In addition, similarity between incoming weight vectors is often considered a negative computational property, as minimizing such correlations improves classification accuracy [36–39]. Reducing redundancy across convolutional filters has been reported to produce similar benefits [40].

3. Methods

3.1. Models and datasets

MNIST. The model used for MNIST [41] training was a relatively shallow CNN. It consisted of two convolutional layers: the first with 32 output channels and the second with 64 channels, each using a 3×3 kernel. Both convolutional layers were followed by a ReLU activation function and 2×2 max-pooling. After the second convolutional layer (`conv2`), global average pooling was applied to each of the 64 feature maps. The resulting vector was then fed into a fully connected layer, `fc1`, which mapped it to 121 units. The output of `fc1` was connected to a second fully connected layer, `fc2`, which produced the final 10 logits for classification, corresponding to the 10 MNIST classes. To reduce overfitting, dropout with a rate of 0.5 was applied after `fc1`.

CIFAR-10. We used the standard CIFAR-10, a 10-class dataset of 32×32 images [42]. The CNN model used for image classification consisted of four convolutional layers with batch normalization applied after each. The number of channels in the first three layers was 32, 64, and 128, each followed by max-pooling with $stride = 2$. The fourth convolutional layer consisted of 256 channels and was followed by a global average pooling layer ($n = 256$ values). The last two layers were fully connected. The first (`fc1`) consisted of 121 units, with dropout (0.3), and the second (`fc2`) mapped feature activations to the 10 output classes.

The exact same model definitions were used for the topographic models and the control models, for CIFAR-10 and MNIST. The only difference was that in the topographic models, the 121 `fc1` units were arranged on an 11×11 grid to which a spatial loss function could be applied as described below.

3.2. Spatial loss: Weight-similarity and activation-similarity

Weight similarity. For training with a weight-similarity objective, we used a joint loss function, combining the standard cross-entropy loss term, $\mathcal{L}_{\text{CE}} = \text{cross-entropy}(\text{output}, \text{target})$, and a spatial loss term $\mathcal{L}_{\text{spatial}}$. For weight-similarity, the spatial loss term was designed to increase similarity among immediately adjacent weight vectors in the 11×11 grid structure. To compute $\mathcal{L}_{\text{spatial}}$, we indexed the 121 incoming weight vectors of `fc1` on an 11×11 grid. Each grid position corresponds to one `fc1` unit with an incoming weight vector of dimension 64 (MNIST) or 256 (CIFAR-10). For each of the 121 grid cells, the immediate neighbors were identified. Then, for each cell, the L_2 norm (Euclidean distance) was computed between the weight vector of that cell and those of each neighboring cell. These distances were summed across all cells and divided by the number of neighboring cells, producing a single value indicating the average pairwise distance across the grid. The joint loss function was therefore $\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{spatial}}$, where λ is a weighting factor that sets the contribution of the spatial loss. We evaluated the impact of the spatial loss term under six weighting levels, with $\lambda \in \{0.1, 0.3, 0.5, 1, 2, 3\}$.

Activation similarity. The activation-similarity spatial loss term produced a single value indicating the similarity of each unit to its immediate neighbors. For AS, similarity was defined as $1 - r$, where r is Pearson's correlation coefficient between the two units' activation vectors across minibatch samples. Here too `fc1` was the topographic layer, defined by reshaping 121 units to an 11×11 grid.

Training parameters. MNIST models were trained using the Adam optimizer with a learning rate of $\eta = 0.001$ for 15 epochs. Initial evaluations showed that all models converged to a training-set accuracy of approximately 97% under moderate regularization strength. The CIFAR-10 model was trained using the same optimizer parameters for 30 epochs,

reaching a similar training accuracy of around 93% across the three model conditions.

We trained 10 independently initialized models for each λ level under both WS and AS regularization, resulting in 60 WS models and 60 AS models in total. Additionally, we trained 10 control models without topographic regularization, which are equivalent to $\lambda = 0$. In all analyses, mean statistics were computed from each set of 10 models from the same λ.

3.3. Robustness tests

Robustness of representational geometry. After training the control, AS, and WS models on MNIST and CIFAR-10, we extracted the weight matrix connecting the topographic penultimate layer to the classification layer (a 10×121 matrix in both cases). We consider each of this matrix's 10 row vectors as a *class weight vector; (CWVs)*. From these CWVs, we computed a class-weight representational similarity matrix (RSM), which is a 10×10 matrix of pairwise similarity between CWVs [43–46]. This was treated as the baseline representational geometry.

To evaluate the robustness of this representation, we conducted several perturbation tests in which Gaussian noise (four intensity levels) was added to the 10×121 class-weight matrix, and the RSM was recomputed. We refer to these as perturbed RSMs. Each analysis was repeated 100 times, and we report the mean results.

We determined the impact of noise using two metrics. First, we computed the second-order similarity as the cosine similarity between the upper triangles of the unperturbed and perturbed RSMs. This indicates how much the representational geometry was affected by noise. This was computed separately for the WS, AS, and control models. Second, we evaluated the drop in classification accuracy caused by the addition of noise, relative to the unperturbed models.

Robustness to input corruption. For both WS and AS models, we evaluated their performance under various corruptions, with the noise applied to test-set images (training images were not corrupted). In all cases, corrupted images were normalized to the mean and standard deviation of MNIST and CIFAR-10 training sets. The noise interventions consisted of adding white noise, pink noise, and salt-and-pepper noise. White noise was introduced by adding to each pixel a random value from the standard normal distribution. Pink (1/f) noise was generated such that the power spectral density of the signal is inversely proportional to its frequency. Salt-and-pepper noise converted a proportion of randomly chosen image pixels to either black or white. Examples of each type are given in Appendix Fig. A.4. Each noise intervention was applied at five different intensities.

3.4. Orientation and eccentricity tuning

After training on MNIST and CIFAR-10, we evaluated the responses of the trained models on a standard stimulus set typically used for retinotopic mapping of the human occipital cortex. To study angular and orientation tuning we presented the trained networks with a rotating wedge (36 positions, angle extent 10°, radius = 14 pixels). To study eccentricity tuning, we presented the network with ring images (13 different radius levels). For each unit in the grid this produced a 36-element series for the wedge angle and a 13-element series for ring eccentricity.

Orientation analysis. To describe angular tuning in topographic units, for each unit we measured responses to the 36 wedge stimuli. We applied a Fast Fourier Transform to each unit's 36-point response profile and extracted the spectral power for the first five harmonics (cycles 1–5). These components reflect different angular periodicities: cycle 1 indicates preference for a single direction, cycle 2 for 180° symmetry consistent with orientation tuning, and cycle 4 indicates fourfold (90°) angular periodicity. We defined the *dominant harmonic* for each unit as the harmonic (cycles 1–5) with the largest spectral power (excluding the DC component; i.e., the mean value of the signal). These dominant harmonic labels were assigned to the 11×11 grid.

Eccentricity analysis. To study eccentricity response profiles, we analyzed each unit's response to the 13 ring stimuli of increasing eccentricity by fitting a linear model to the 13 response values. Units with a Pearson correlation coefficient of $|r| > 0.8$ were labeled as `increasing` or `decreasing` depending on the sign of r. For units not showing a linear profile, we evaluated whether the response was selective to a particular eccentricity, indicating a band-pass response. To test this, we fitted a four-parameter Gaussian function (baseline, amplitude, center, width) to the unit's 13-point response profile. The quality of fit was determined using the coefficient of determination (R^2), and a unit was categorized as showing a `bandpass` response when $R^2 > 0.5$. Profiles that did not meet any of the above criteria were labeled `flat`.

3.5. Functional localization

For each trained model, we measured localization to understand how closely located the correlated units were. We first computed the correlation between every pair of units' activations across the test images. Given a chosen correlation threshold, we marked a pair of units as connected if their correlation exceeded that threshold, producing a binary connectivity matrix (connected vs. not connected). Next, for each unit, we identified all other connected units and computed the mean Euclidean distance to them. In the last step, we averaged these per-unit distance values across all units in the grid to obtain a single localization score for the model at that threshold. Smaller values indicate that strongly correlated units are more spatially clustered on the grid.

3.6. Weight correlations and activation correlations

After WS and AS training, we computed, for each unit in the 11×11 grid, its average incoming-weight correlation with neighboring units. Incoming weights refer to the weight matrix from the preceding global-average pooling layer to the 121-unit grid layer. The correlation $r_{ij}^{(\mathrm{In})}$ between the incoming weight vectors of units i and j was the Pearson correlation of the two vectors:

$$r_{ij}^{(\mathrm{In})} = \frac{\sum_k (w_{ik} - \bar{w}_i)(w_{jk} - \bar{w}_j)}{\sqrt{\sum_k (w_{ik} - \bar{w}_i)^2 \ \sum_k (w_{jk} - \bar{w}_j)^2}},$$

where w_{ik} and w_{jk} denote the k-th afferent weight into units i and j, respectively, and $\bar{w}_i$ and $\bar{w}_j$ are the means of their incoming weight vectors (dimension 64 for MNIST; 256 for CIFAR-10). For each unit i, we then computed the mean correlation over its immediate (Moore) neighborhood S_i. The neighborhood size $|S_i|$ was 3, 5, or 8 for corner, edge, and interior units, respectively:

$$R_i^{(\mathrm{In})} = \frac{1}{|S_i|} \sum_{j\in S_i} r_{ij}^{(\mathrm{In})}$$

We evaluated activation correlations using the same logic described above for computing incoming weight correlations. The difference was that for each unit-pair, we computed the correlation of their activation profiles for a minibatch of images. This allowed us to compute, for each unit, its average neighborhood correlation. It also allowed us to study the entire distribution of pairwise correlations.

3.7. Expert unit analysis

Following prior work [e.g.,47], we define expert (class-selective) units as those whose activations reflect one-vs-rest discrimination of a target class. For each unit in the 11×11 topographic grid layer and each class, we quantified discriminability using the area under the receiver operating characteristic curve (AUC), computed from the unit's

post-ReLU activations over the test set. The AUC estimates the probability that the unit's activation for an input from the target class exceeds activations for inputs from any other class.

We use two AUC thresholds. Units with $AUC > 0.70$ were classified as moderately class-selective; they produce higher activation to the target class in more than 70% of random class–nonclass comparisons. Units with $AUC > 0.90$ were classified as strongly class-selective. For each condition, we computed the proportion of units meeting these criteria. Condition-level values were obtained by averaging the corresponding measures across the 10 independently trained models for each condition.

To quantify the distribution of expert units over classes, each expert unit was assigned to the class for which it achieved its maximal AUC. This produced a distribution of expert units for each model family. We then computed the entropy of this distribution and normalized it by the maximum entropy for ten classes ($log10$). We refer to this quantity as *expertise balance*; it equals 1.0 when expert units are uniformly distributed across classes and is lower otherwise.

Finally, we quantified the selectivity of expert units by computing, for each unit, the difference between its highest AUC (corresponding to the most discriminative class) and its second-highest AUC across classes. This AUC gap shows how exclusively a unit discriminates its preferred class relative to alternative classes. We refer to this as *expertise selectivity*.

3.8. ResNet34 extension

Overview. To evaluate scalability, we extended the main study to a ResNet34-based DNN. As for the MNIST and CIFAR-10 experiments, the topographic and control models had identical feedforward structures except for the presence or absence of the topographic regularizer (AS, WS), using a joint loss function. The main difference was that instead of training a small CNN from scratch, we fine-tuned a pretrained ResNet34 on ImageNet100 [48], which consists of 50,000 images (500 images for each of 100 classes). The topographic bottleneck was a 16×16 layer (256 units) receiving the 512-dimensional global average pooling representation. The incoming weight vectors were therefore substantially larger than in the other studies (512 vs. 64). Model selection was based on a fixed stratified 10% validation split. Because this setting was substantially more computationally expensive, we evaluated only $\lambda \in \{0.3, 1.0\}$ and trained 5 models per condition. The architecture operated on natural variable-size images which were cropped during inference to 224×224.

Architecture and training. We replaced the original ResNet34 classification head with an identity mapping, producing a 512-dimensional global average pooling representation. This was followed by a $512 \rightarrow 256$ linear layer interpreted as a 16×16 topographic grid, ReLU nonlinearity, dropout, and a final linear classifier. The ResNet34 backbone was not frozen and was fine-tuned jointly with the topographic head. The control condition used the same 256-unit bottleneck and classifier but no topographic regularization. The held-out ImageNet100 split was used for clean testing. A fixed validation split was sampled once from the training pool and reused across runs. Across the 5 runs per condition, the pretrained backbone initialization and the train/validation/test partition were fixed; stochasticity was implemented using random initialization of the added layer, minibatch order, dropout, and data augmentation.

Analyses shared with the smaller-scale experiments. Several analyses matched those used for MNIST and CIFAR-10. Robustness to weight perturbation was evaluated on the last weight matrix (100×256). Input corruption was implemented using white Gaussian noise, pink noise, and salt-and-pepper noise. We quantified Moran's I, functional localization, unit-wise response entropy, and post-ReLU percentage of zero firing (PoZ), all computed from activity in the topographic layer.

ResNet-specific analyses. The larger images used and larger number of classes allowed several additional analyses. First, we performed representational similarity analyses on a set of 100 class prototypes to quantify how fine-tuning and topographic regularization altered representational geometry at both the ResNet backbone and topographic bottleneck levels. Second, we added a circular center-mask corruption to quantify sensitivity to central information in the higher-resolution images. Finally, we evaluated robustness to adversarial attacks using the fast gradient sign method (FGSM) attacks on the prototype set and fooling-image generation. The first evaluates the stability of decision boundaries around the natural images, and the second assesses how easy it is to drive the model to produce target-category predictions for synthetic images. Full details of all ResNet34-related methods are given in the Appendix.

4. Results

4.1. Accuracy

Control models were generally associated with the highest accuracy, followed by AS (Fig. 2). For MNIST, WS slightly outperformed the control models at the lowest $\lambda = 0.1$ (Hedges' $g = 0.7$), and AS outperformed at $\lambda = 0.1 - 0.5$ ($g = 1.04, 1.51, 0.94$ respectively). For CIFAR-10, the accuracy of the topographic models was often lower than that of control. Stronger topographic regularization produced a drop in accuracy of up to 2%, and more strongly for WS. This drop in accuracy was also observed in previous end-to-end topographic models [4,5,10]. We measured Hedges' g effect size, which is a preferred choice over Cohen's d when the sample size is small (10 in our case), for all comparisons between control accuracies and those of AS and WS.

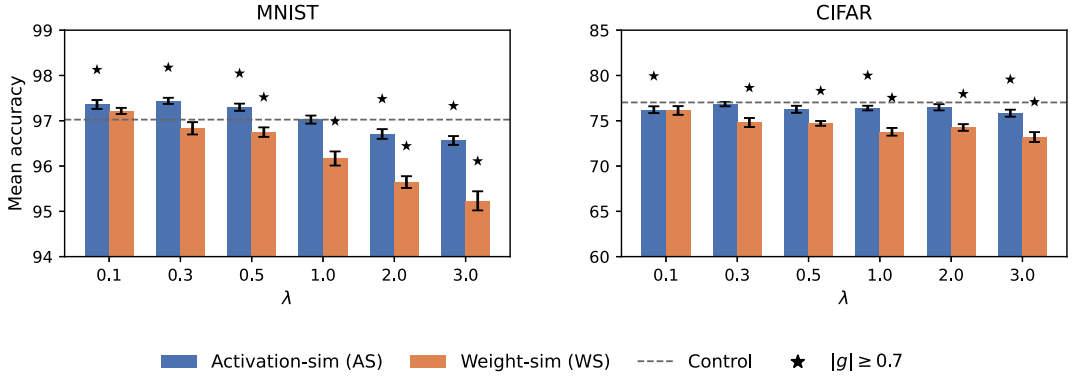


Fig. 2. Test accuracy as a function of topographic regularization strength λ. Mean classification accuracy for control (dashed), AS (blue), and WS (orange) models is shown for the different values of λ. Error bars indicate ± s.e.m. Star symbols indicate comparisons in which the differences between control and AS/WS had moderate to high effect size (Hedges' $g \geq 0.7$).

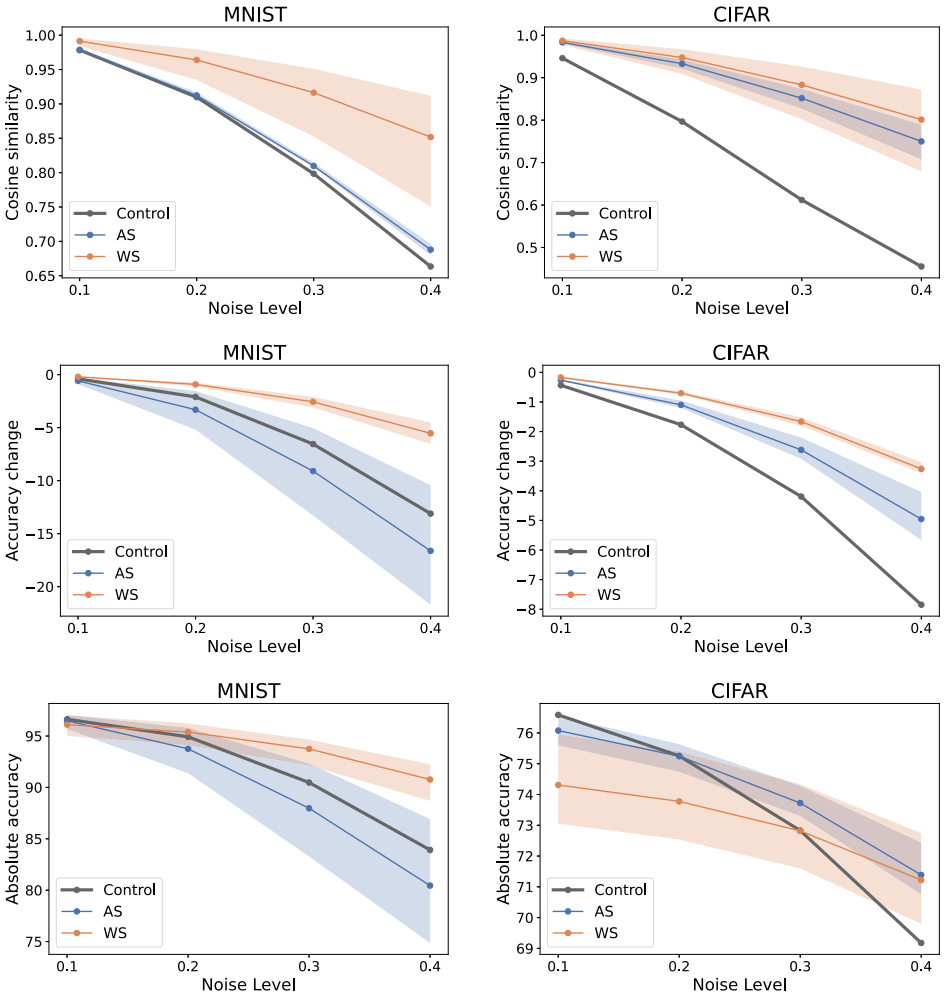


Fig. 3. Robustness to weight perturbations as a function of noise levels. Robustness is evaluated by changes in representational geometry (top row), and test accuracy changes (i.e., noise accuracy minus original accuracy; middle row) for increasing levels of additive weight noise. Raw test accuracy (bottom row) is provided for reference. Representational geometry is defined as the representational similarity matrix (RSM) computed from class weight vectors (CWVs) in the readout layer. Cosine similarity between original and perturbed representations is shown in the top row; corresponding changes in test accuracy relative to baseline (non-perturbed) models are shown in the middle row. Shaded regions indicate the min–max range across λ levels. Values computed separately for each λ level and model are reported in Appendix Fig. A.3.

4.2. Robustness

Robustness of representational geometry. We added Gaussian noise of varying magnitudes to the weight matrix connecting the topographic and classification layers, then evaluated the resulting representation by constructing a representational similarity matrix (RSM) from the class-weight vectors. The intervention showed that WS training resulted in a more robust representational geometry than that of AS and control models (see Fig. 3). First, the second-order similarity between the baseline and perturbed RSMs was consistently higher for WS models, for both MNIST and CIFAR-10. WS models, particularly when trained with larger λ, were consistently top-ranked, showing less degradation in representational geometry. The control models showed the least robustness.

Robustness of representation was also evident in the accuracy drop statistics (Fig. 3, middle): for MNIST, WS models showed a relatively small drop of up to 5%, but AS models showed larger drops of up to 20%. A similar pattern was observed for CIFAR-10, where the accuracy drop for AS was consistently larger than for WS. In both datasets, WS outperformed the control models as well. These results suggest that WS training stabilizes representations under perturbation. In certain cases this also translated into overall better classification performance for WS under noise corruption (see absolute accuracies in Fig. 3).

Robustness to image corruption. Fig. 4 shows, for each level and type of noise, which model family (AS, WS, control) was most robust to input corruption. For both datasets, AS tended to be the most robust. (Fig. A.5 presents the full breakdown, showing accuracy changes compared to baselines by noise type, noise level, and λ).

We note that at the lowest regularization strength ($\lambda = 0.1$) in MNIST, AS and WS showed stronger robustness to input corruption and stronger representational robustness than control models, but similar or even slightly higher classification accuracy. As reported below, the same pattern was reproduced for ResNet34/ImageNet100. Prior studies have tried training for robustness, finding that this typically reduces accuracy [49–52]. Our findings suggest that moderate topographic regularization can achieve a similar aim without a strong trade-off.

4.3. Activation entropy, sparsity and effective dimensionality

The variance or entropy of a unit's activations is often treated as an indicator of its functional importance within an ANN. Higher entropy reflects greater sensitivity to input variation, and lower-entropy elements are targets for pruning [5,53–56]. Similarly, a unit's tendency to remain inactive across inputs, quantified as PoZ (Percentage of Zeros), is used as a pruning criterion, with higher PoZ indicating less functional importance [57]. We analyzed the entropy and PoZ of the grid units in

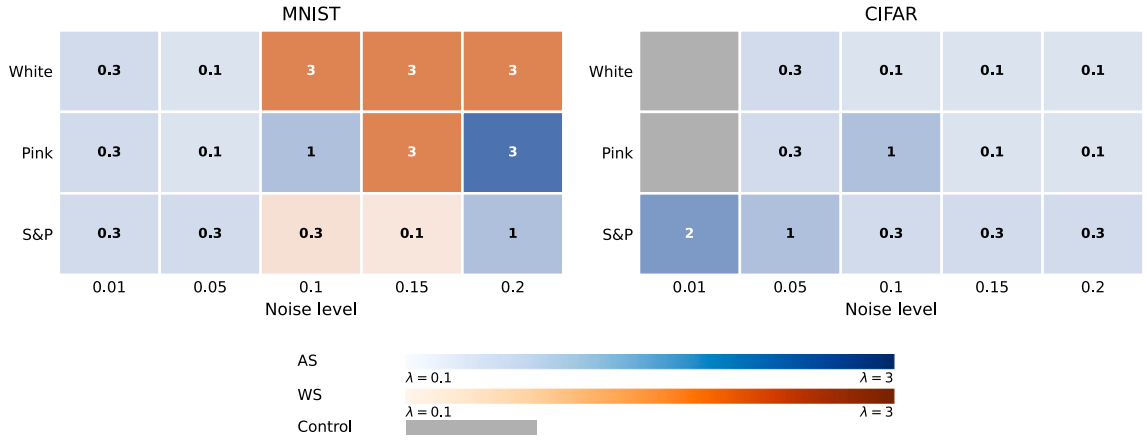


Fig. 4. Model-group robustness under input corruption. For each dataset and corruption type (white, pink, and salt-and-pepper), the most robust model group is defined as the one showing the smallest drop in test accuracy relative to the uncorrupted baseline. Each cell reports the λ value associated with the winning group. A full breakdown of accuracy drops across λ and corruption intensities is reported in Appendix Fig. A.5.

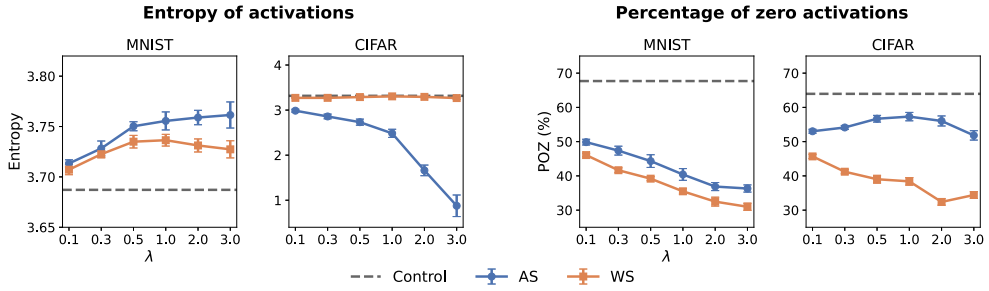


Fig. 5. Unit-level entropy and Percentage-of-Zero activations as a function of topographic regularization strength (λ). The two left plots show the average entropy of pre-ReLU unit activations for MNIST and CIFAR-10 across values of λ. The two right plots show the average Percentage-of-Zero (PoZ) of post-ReLU unit activations. Error bars indicate ± s.e.m.

AS, WS, and control models. Entropy was computed from each unit's pre-ReLU activations across the 10,000 test images. PoZ was computed post-ReLU, reflecting the fraction of inputs for which a unit's activation was zero. We computed the average entropy and PoZ across units within each grid. As shown in Fig. 5, for MNIST, WS models showed slightly lower entropy than AS models, but for CIFAR-10 they were significantly higher. Regarding PoZ, WS models show the lowest across all λ in both datasets, and both AS and WS showed PoZ below that of control. This suggests that incorporating similarity in either weights or activations during training leads to less sparse models. The data suggest consistent effects of topographic regularization on PoZ, but dataset-dependent effects on Entropy. We return to this issue when discussing the ResNet34 findings.

To determine how these unit-level properties translate into the organization of information in the latent space, we examined the Effective Dimensionality (ED) of model activations [e.g.,4,6,8]. Using the activations evoked by the 10,000 test images, we computed ED from the covariance matrix of all images. In addition, we computed within-class ED by isolating activations for each class and averaging ED across classes. For MNIST, AS and WS models produced similar effects on the dimensionality of latent representations (Fig. 6), with larger λ levels producing a gradual reduction in both overall and within-class effective dimensionality. The results for CIFAR were mixed, with a reduction in ED in within-class variance but not in overall effective dimensionality.

4.4. Functional localization metrics

4.4.1. Functional co-localization

To evaluate to what extent units with similar firing patterns were positioned closely on the topographic grid, we defined two units as belonging to the same functional network if their activation patterns exceeded a correlation threshold α. We then computed the average Euclidean distance between connected units (see Methods). Fig. 7 shows the results for WS and AS (for simplicity, the figure presents the mean and $\pm 1SD$ over responses of α; details are presented in Appendix Fig. A.6). As the figure shows, for both MNIST and CIFAR-10, WS was associated with smaller distances between correlated units. For AS, increased levels of λ tended to produce an increase in connection distance. At the highest λ level, AS connectivity approached the expected distance value associated with an all-to-all connectivity grid (5.8 units). As we detail later, this has to do with the fact that at higher levels of λ, the AS condition produces saturated connectivity matrices with very high all-to-all correlation values.

4.4.2. Spatial autocorrelation of unit activations

To understand the spatial organization produced by WS and AS training, we quantify activation smoothness, weight similarity, and activation correlations among neighboring and non-neighboring units on the topographic grid. We first quantified the spatial smoothness of activation maps using Moran's I, which is a spatial smoothness statistic [58], previously used in related work [e.g.,10]. Positive values indicate smoother transitions among neighboring units, negative values indicate spatial dispersion (e.g., high-low activation transitions between neighbors), and zero indicates randomness. We applied this metric to the pre-ReLU activation maps produced by each image in each model, and averaged within AS, WS, and control models. Fig. 8 presents the results for MNIST, and similar findings were observed for CIFAR-10 (see Appendix Fig. A.7).

As Fig. 8(C) shows, the topographic models produced consistently positive Moran's I values, which increased monotonically with λ. The

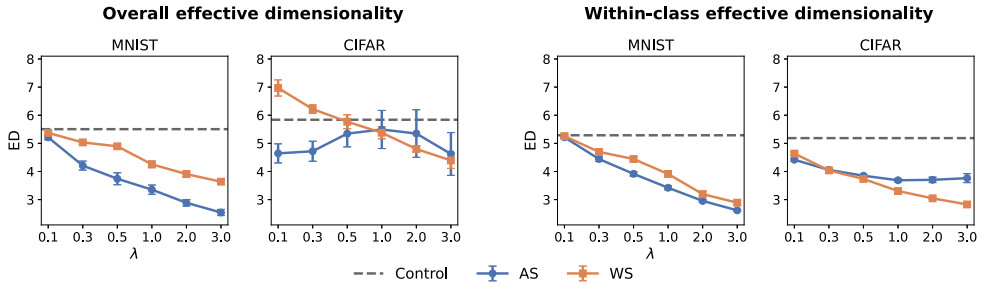


Fig. 6. Effective dimensionality of topographic representations. Overall effective dimensionality (ED; left) and within-class effective dimensionality (right) of the fc1 layer as a function of the spatial regularization strength λ. Overall ED is computed from the activation covariance across all images; within-class ED is computed for each class separately and then averaged across classes. Error bars denote s.e.m, and dashed lines indicate control models.

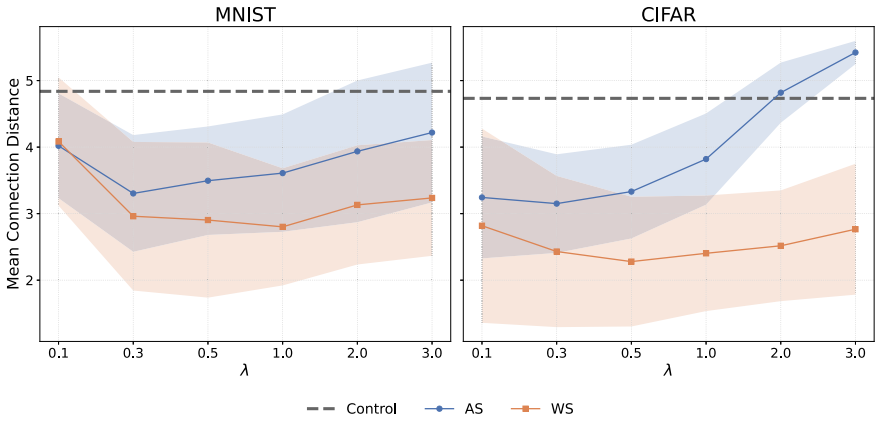


Fig. 7. Correlated-unit distance in the topographic grid as a function of topographic regularization strength (λ). Mean spatial distance between pairs of units whose activity-correlation exceeds a threshold $\alpha \in \{0.1, 0.3, 0.5, 0.6, 0.7, 0.8\}$. Solid lines indicate distances averaged across correlation thresholds α; shaded regions denote ± 1 SD across α thresholds. Distances computed separately for each correlation threshold and model are reported in Appendix Fig. A.6.

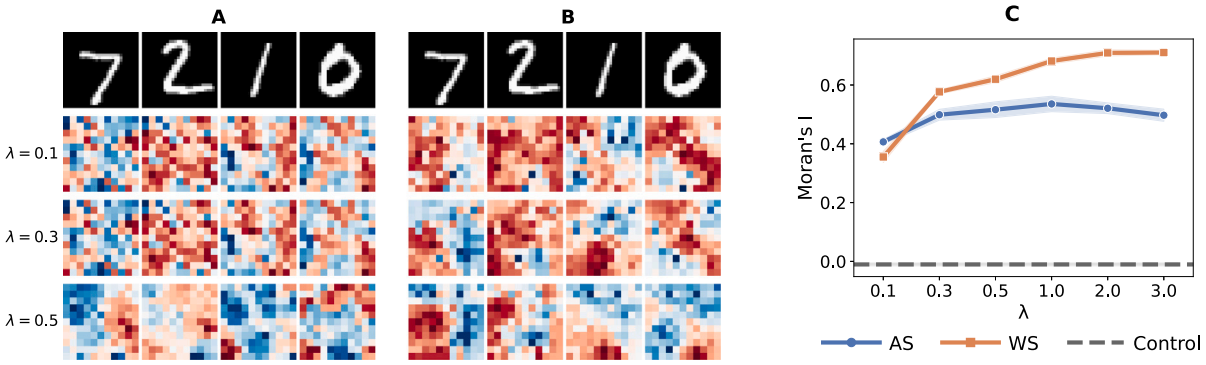


Fig. 8. Spatial smoothness of activation maps under activation- and weight-similarity training. Sample activation maps from a topographic grid layer are shown for WS-trained models (Panel A) and AS-trained models (Panel B) at three values of λ. Panel C shows Moran's I, which quantifies spatial autocorrelation of grid activations. Similar patterns are observed for CIFAR-10 (Appendix Fig. A.7).

control models approximated zero as expected. To visually appreciate the distribution of local activation similarity, Fig. 8(A),B shows activation maps from three randomly selected WS and AS models trained under a regularization strength of $\lambda = 0.1, 0.3, 0.5$. It is evident that both WS and AS training produce substantial spatial autocorrelation even at $\lambda = 0.1$.

To understand these patterns, we analyzed activation correlations at both local and global scales. At the neighborhood level, WS training produced slightly stronger correlations between adjacent units than AS across all values of λ, even though its objective did not encourage activation similarity (Fig. 9, A–B). The control models show approximately zero correlations. At the level of all unit pairs (Fig. 9, C–D), AS and WS training produced qualitatively different correlation structures: AS produced a heavy-tailed distribution with many units showing near-perfect correlation ($r \approx 1$). This means that the AS objective was achieved by strongly coupling a subset of units while leaving others weakly correlated. WS, in contrast, produced a flatter distribution with higher correlations across a broader range of unit pairs. Both AS and WS increased local similarity among incoming weight vectors relative to control (Appendix Fig. A.8), though more strongly for WS as would be expected from its training objective.

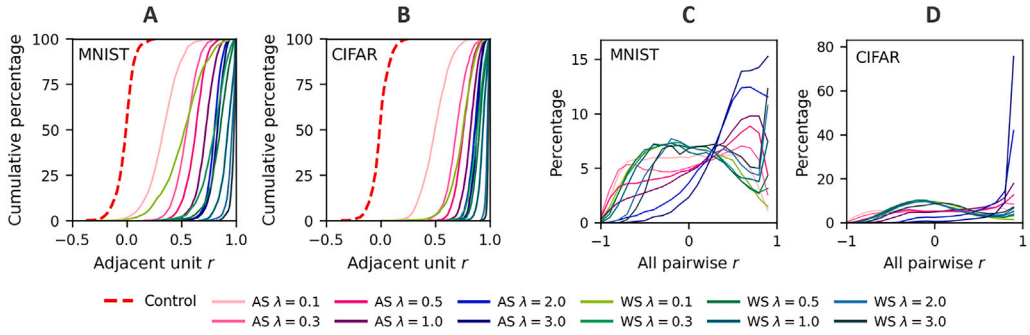


Fig. 9. Different patterns of correlated activations under activation- and weight-similarity training. Cumulative distributions of pairwise activation correlations between adjacent units are shown for MNIST (Panel A) and CIFAR-10 (Panel B). Distributions of correlations computed across all unit pairs are shown for MNIST and CIFAR-10 (Panels C and D).

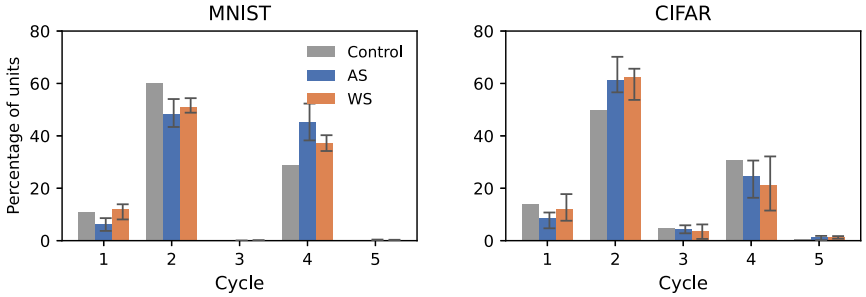


Fig. 10. Distribution of angular tuning profiles relative to control models under topographic training. Percentage of units associated with each angular sensitivity profile (Cycles 1–5) is shown. Cycle 1: classic polar-angle tuning; cycle 2: orientation-like tuning; Cycles 3–5: higher angular frequencies. Bars indicate mean percentages averaged across the six values of the regularization parameter λ; vertical whiskers indicate the full min–max range across λ. In both datasets, orientation-like tuning (Cycle 2) is the most prevalent profile, and AS and WS models differ from control models in the relative distribution of units across profiles. Full cycle-response data, shown separately for each value of λ, are reported in Fig. A.10.

4.5. Reorganization of angular and eccentricity tuning under topography

Relative to control models, topographic regularization resulted in changes to tuning profiles. AS and WS tended to alter tuning in similar ways, though in a dataset-dependent manner (see Fig. 10). For MNIST, they increased the proportion of symmetry-tuned units (`cycle = 4`) and decreased the proportion of orientation-tuned units (`cycle = 2`). For CIFAR-10, the pattern was reversed. Thus, topographic training reweighted angular harmonics, but this depended on dataset properties (MNIST vs. CIFAR-10). Appendix Fig. A.9 shows the spatial distribution of angular tuning profiles in the topographic grid for WS-trained models.

A reorganization was also observed for eccentricity tuning (Fig. 11). Here, tuning profiles reflect radial gain functions: *decreasing* responses correspond to filters that weight central locations more strongly; *increasing* responses weight peripheral locations more strongly. The most salient effect was that, in both datasets, both AS and WS strongly reduced centrally preferring (decreasing) responses as compared to control. In MNIST, this reduction was accompanied by an increase in monotonically increasing responses, indicating greater sensitivity to peripheral locations.

4.6. Expert units

All models developed expert units that discriminated specific categories. We defined two levels of expertise: moderate-expertise units that systematically showed higher activity for one category in more than 70% of random category–non-category comparisons, and high-expertise units, corresponding to a 90% threshold. The main finding, as shown in Fig. 12, was that for both MNIST (Panel A) and CIFAR-10 (Panel D), AS models produced a larger proportion of high-expertise units as compared to WS and control. Regarding moderate-expertise, AS and WS were on par with control as all conditions showed a ceiling response.

We next examined how expert units were distributed across categories by quantifying the balance of expertise (Fig. 12, B, E). Overall, in both datasets the balance decreased for stronger λ. For moderate expertise, both MNIST and CIFAR-10 showed slightly higher balance for WS, whereas for high expertise, the pattern was inconsistent. Relatedly, the specific association of expert units with different categories was impacted by regularization (see Appendix Fig. A.12). For MNIST, in control models most experts were dedicated to the digit *1*. For AS, a greater proportion of experts was allocated to *3, 8*, while for WS they were more balanced over *3, 4, 6, 7, 8*. For CIFAR-10, in control models most experts were allocated to the *automobile* category. For the topographic networks, this category was still prominent, but in AS the category *ship* had more expert units than in WS.

Finally, we evaluated the selectivity of expert units by isolating them and quantifying the difference between the response to the most preferred category and the second-most preferred category (Fig. 12, panels C and F). Here, the topographic effects were consistent within but not across datasets. For MNIST, both topographic conditions produced less selectivity than control, whereas the converse held for CIFAR-10. Taken together, the results show that topographic training strongly impacted the proportion and distribution of expert units. However, this depended on an interaction of dataset properties (MNIST vs. CIFAR-10) and the topographic loss itself.

5. Extension to ResNet34 architecture

5.1. Summary of model, data and training

We next tested whether the findings observed in the smaller-scale experiments transferred to a larger architecture and dataset. We used ResNet34 [59] as the backbone, initialized from the ImageNet1K_V1 pretrained weights, and fine-tuned it on ImageNet100 [48], a 100-class subset of ImageNet containing 50K images in total (500 per class). We

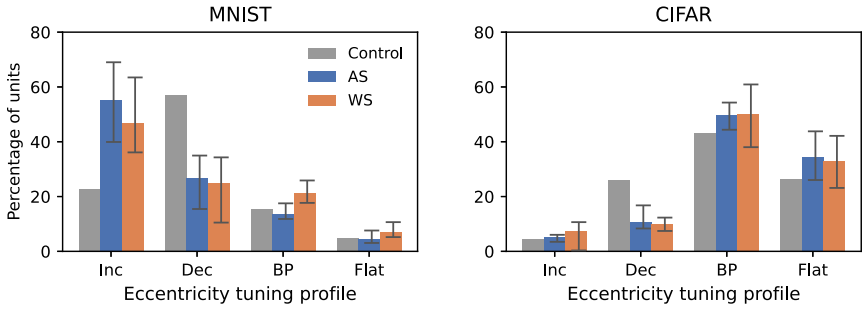


Fig. 11. Distribution of eccentricity tuning profiles relative to control models under topographic training. Percentage of units showing increasing (Inc), decreasing (Dec), band-pass (BP), or flat eccentricity tuning profiles is shown. Bars indicate mean percentages averaged across values of the regularization parameter λ; vertical whiskers indicate the full min–max range across λ. In both datasets, AS and WS models differ from control in the relative distribution of units across eccentricity tuning profiles. Eccentricity data, shown separately for each value of λ, are reported in Fig. A.11.

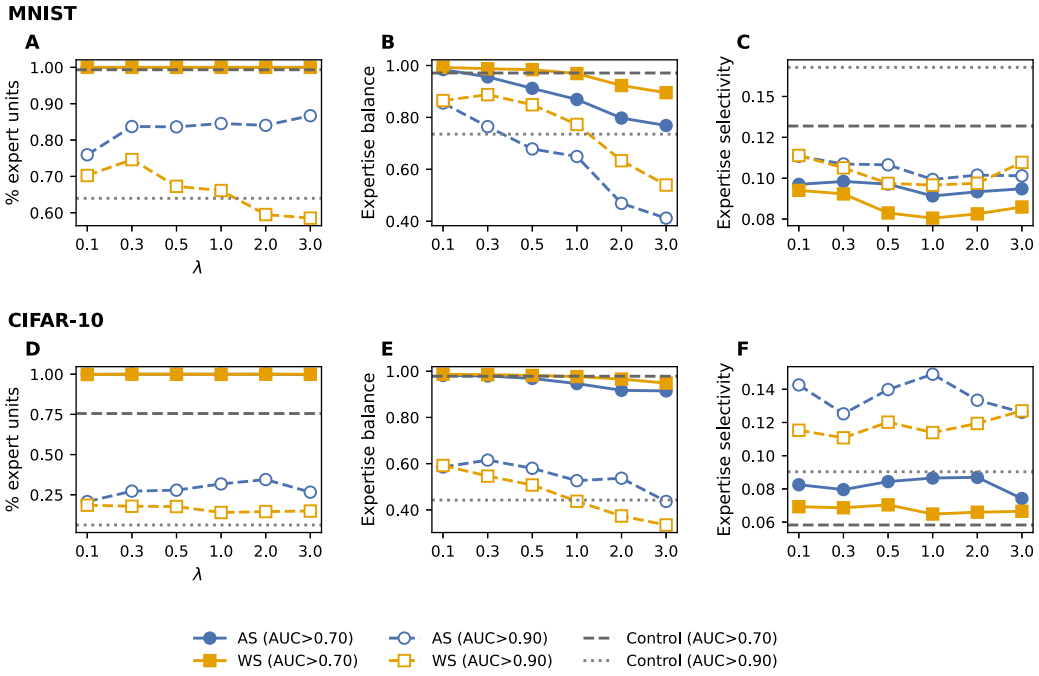


Fig. 12. Expert unit prevalence, balance, and selectivity. Expert unit prevalence, defined as the proportion of units whose category discriminability exceeds a threshold (AUC > 0.70 or AUC > 0.90), is shown for MNIST (Panel A) and CIFAR-10 (Panel D). Expertise balance, quantified as the normalized entropy of expert units across categories, is shown in Panels B and E. Expertise selectivity, defined as the difference between the highest and second-highest AUC values for each unit, is shown in Panels C and F. See Methods for formal definitions of all measures.

used the official held-out split that contained 50 images in each of the 100 classes (5000 images total) and was used only for final held-out testing. We replaced the original ResNet34 classification head with an identity mapping, producing a 512-dimensional global average pooling (GAP) feature vector. The topographic head consisted of a 512→256 linear layer treated as a 16×16 grid, followed by a ReLU, dropout, and a linear classifier operating on the post-ReLU grid activity (see Appendix *ResNet Methods* for full details).

The backbone was not frozen and was fine-tuned jointly with the topographic head. The control condition used the same 256-unit bottleneck (effectively a dimensionality-reduction layer with ReLU output) but no topographic regularizer. Because this setting is much more computationally expensive, we evaluated AS and WS only at $\lambda \in \{0.3, 1.0\}$, training five models per condition (producing 25 checkpoints in all). For model selection, we constructed a fixed 10% validation split from the training data and kept this split fixed across runs.

5.2. Matched accuracy under topographic regularization

Performance was essentially matched across conditions. Test accuracy, evaluated on the held-out test split using the checkpoint with the lowest validation loss, was 0.892 ± 0.005 for control, 0.889 ± 0.004 and 0.890 ± 0.003 for AS 0.3/1.0, and 0.893 ± 0.003 and 0.893 ± 0.001 for WS 0.3/1.0, respectively. Best validation cross-entropy also showed only small differences: control at 0.291 ± 0.007, AS 0.3/1.0 at 0.300 ± 0.012 and 0.300 ± 0.007, WS 0.3 showed the lowest mean at 0.288 ± 0.007, and WS 1.0 reached 0.294 ± 0.011. We conclude that topographic regularization did not harm performance, matching that of control on both validation and test metrics.

Control models reached the best validation cross-entropy earliest (27.6 ± 2.1 epochs), AS 0.3/1.0 reached it at 33.6 ± 7.3 and 39.4 ± 12.0 epochs, and WS 0.3/1.0 at 38.8 ± 10.1 and 42.8 ± 7.9 epochs. The difference in training time is expected given the need to satisfy a joint loss.

Table 1
Second-order RDM similarity comparisons identify representational change in the ResNet34 experiments. Row 1: shift of each model backbone from the pretrained backbone. Row 2: comparison of regularized model backbones to the control backbone. Row 3: changes in representation induced by the 256-unit topographic bottleneck relative to its own backbone. Similarity was operationalized as the Spearman rank correlation between the upper triangles of any two RDMs.

Comparison	Control	AS 0.3	AS 1.0	WS 0.3	WS 1.0
Backbone vs. pretrained backbone	0.57	0.41	0.43	0.57	0.55
Backbone vs. control backbone	–	0.53	0.54	0.71	0.63
Topographic layer vs. own backbone	0.93	0.70	0.70	0.88	0.76

5.3. Smaller shifts in network representations under WS regularization

We identified 100 prototypical images by selecting those closest to each class centroid, based on embeddings extracted from a pretrained standard architecture (defined independently of any training performed here). Using these images, we computed 100×100 first-order representational dissimilarity matrices (RDMs) from the image embeddings, and compared their second-order similarity across model layers (Spearman rank correlation) to identify how fine-tuning and topographic regularization impacted representational geometry. We first evaluated if fine-tuning shifted the pretrained ResNet34 representation ('pretrained'). At the GAP layer, all conditions shifted away from the pretrained geometry, but the effect was stronger for the AS models (see Table 1). Thus, fine-tuning for the 100-class task changed the backbone representation in all cases, but the shift was larger for AS than for WS.

We next compared the regularized models to the control backbone at the same GAP layer, as this isolates the impact of the topographic loss on learned feature maps. As shown in the Table 1, AS maintained only moderate similarity to control, with WS showing stronger similarity. Finally, we isolated the contribution of the 256-unit topographic bottleneck itself to representation shift. In the control condition, the bottleneck left the representation almost unchanged (similarity to its own backbone: 0.93). For AS, the bottleneck induced a stronger transformation, with a more subtle effect for WS. Overall, the data show that AS had a stronger impact on changing the representational geometry of the prior layer, with WS and control showing a largely preserved backbone geometry at the topographic layer.

5.4. Topographic organization and wiring

Wiring, entropy, and PoZ were all strongly impacted by topographic regularization. Both AS and WS produced a spatial organization that was absent in the control model, but they did so in different ways (see Fig. 13). Moran's I was near zero in the control condition, as expected. It increased to approximately 0.2–0.3 for AS 0.3/1.0, and further increased to approximately 0.40 and 0.72 for WS 0.3/1.0. This indicates substantially stronger local spatial autocorrelation for WS.

AS and WS produced markedly different unit entropy values: AS reduced average unit entropy from the control level ($\approx$ 3.95) to $\approx$ 0.43 and $\approx$ 0.15, suggesting that the responses were almost completely localized into one response-magnitude bin. In contrast, WS produced values matching the Control condition ($\approx$ 3.95 at both λ values). Post-ReLU PoZ was around half that of Control for both regularizers, dropping from the Control level of $\approx$ 35% to $\approx$ 18% and below, with the lowest found for the WS conditions. Together, these results show that both AS and WS increased spatial smoothness and reduced PoZ, but WS did so while preserving the original higher-entropy code found for Control.

The analysis of response-correlation graphs (Fig. 14) shows an important difference between AS and WS. Namely, that for AS, connectivity was effectively saturated: almost all units remained connected even at the highest correlation threshold (T). The mean connection distance for AS was high, approaching the expectation values under all-to-all coupling for a grid of this size (expectation is 8.3 units). WS showed a qualitatively different pattern: higher thresholds were associated with smaller fractions of connected units and shorter connection distances.

These results demonstrate that functional localization, which is a core property of local cortical circuits, was only found for WS. AS, in contrast, produced global redundancy. Finally, for the control condition we found limited connectivity only at the 0.1 and 0.3 thresholds.

5.5. Effective dimensionality

As in the other experiments, we computed effective dimensionality (ED) using $ED = \frac{(\sum_i e_i)^2}{\sum_i e_i^2}$, where e_i are the eigenvalues of the covariance spectrum obtained by PCA applied to the raw unit-activation matrices. As shown in Table 2, and consistent with the very strong inter-unit correlations in Fig. 14, AS models showed extremely low ED when computed from the raw embeddings. We hypothesized that this dominance of a single variance mode could be produced if many units in the grid tracked a common image-dependent signal, with each unit expressing that signal with its own offset and scaling. If so, this would mean that the near-saturated correlations in AS should *not* be taken to indicate a loss of class-relevant information or loss of representational capacity (which would be inconsistent with the high classification accuracy of AS). Instead, it would indicate that the optimizer learns a solution where many units come to share a common response profile across images while maintaining sufficient representational capacity using other variance directions. To evaluate this possibility, we examined the effect of two normalization approaches before computing ED. The first was image-level mean-centering, which subtracts, for each image, the

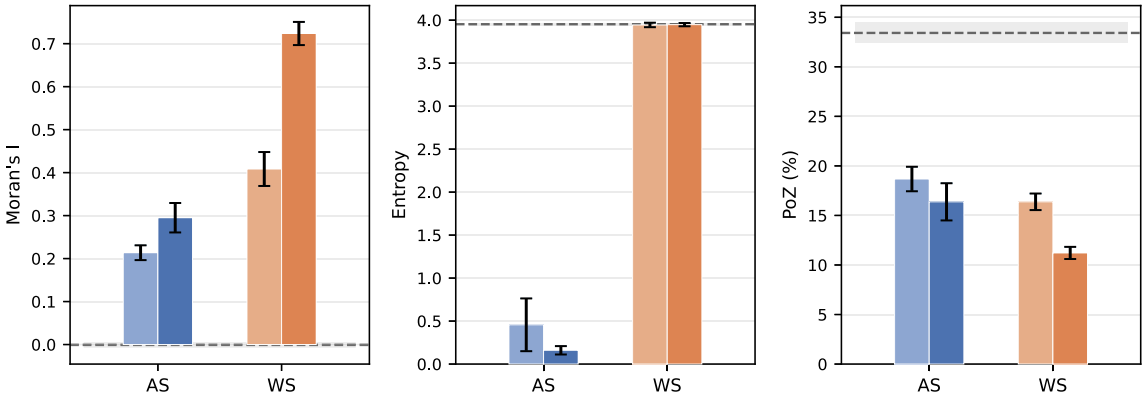


Fig. 13. ResNet34 experiment: AS and WS topographic regularizers produce spatial structure, but in different ways. WS produced stronger local spatial autocorrelation (Moran's I), preserved entropy at near-control levels, and produced lower post-ReLU percentage of zero firing.

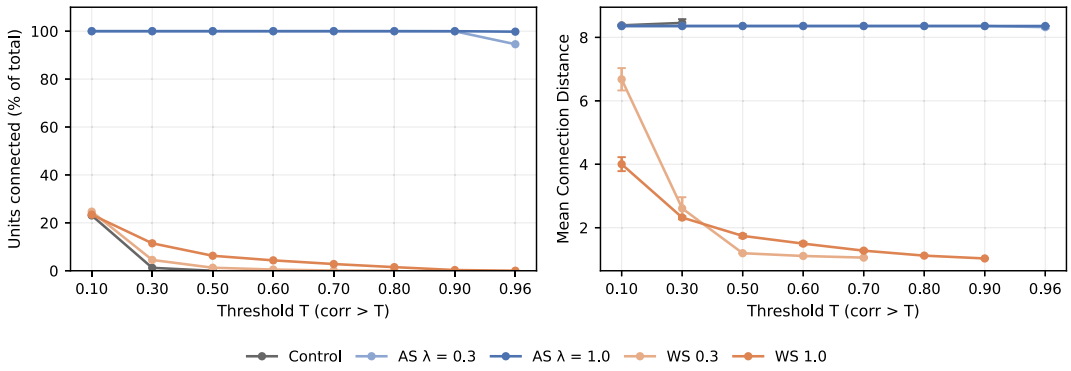


Fig. 14. ResNet34 experiment: Functional localization is found only with WS topographic regularization. Analyzing the response-correlation graph, we find that the AS regularizer produces near-maximal connectivity across thresholds. In contrast, WS shows fewer strongly correlated units and shorter connection distances as the correlation threshold increases.

mean activation across units. This removes per-image common offsets across the population. The second was image-level L2 normalization, which rescales each image's activation vector to unit norm. This removes between-image differences in overall population magnitude and can test if the low raw ED reflects image-wise common gain across the population response pattern.

As shown in the Table 2, the normalization results were quite informative about these issues. They suggest that the very low raw ED in the AS conditions *cannot* be taken to reflect a reduced diversity of representation in the DNN. First, mean-centering only slightly increases ED, suggesting that the correlations were not driven by additive unit offsets. Second and most importantly, image-level L2 normalization strongly increased ED in the AS models, bringing it to near-control levels. This suggests that the apparent reduction in ED seen for AS is due to the presence of a strong, image-level common-gain signal that is shared across units. Within-class ED showed a similar trend. All this suggests that for AS, the relevant information is in fact present in the raw embeddings, but it has a weak contribution in relation to the total covariance spectrum quantified by the ED formula. To evaluate whether the classifier effectively ignores embedding norm, we conducted a pilot analysis where we used the trained AS networks but passed L2-normalized embeddings to the classifier. Accuracy dropped by only about 2%, suggesting that embedding norm provides a limited contribution to classification and that the classifier mainly relies on the relative pattern of activations across units.

For WS, the results were quite different, indicating that the regularizer did substantially compress the dimensionality of the network, and that the WS results cannot be explained in the same way as for AS. Specifically, the impact of the normalization schemes was minor, and in all cases, ED was well below control, both when computed on the global level and within category. Our findings suggest a different perspective on prior reports of topographic networks producing reduced ED: the reported reduction might or might not indicate compression of representational geometry, and more detailed analyses are required to address this issue.

5.6. Robustness

Classifier-weight noise indicates greater robustness for WS regularization. The analysis of classifier-weight noise showed a clear advantage for WS (see Fig. 15). As noise increased in the class-weight vectors, WS 0.3 and WS 1.0 maintained the original class-weight similarity structure better than Control or AS. WS also demonstrated a smaller drop in classification accuracy as compared to AS and Control. We conclude that the gain in robustness to weight perturbation in ResNet34 was limited to WS regularization. These findings correspond to what was found for the MNIST dataset, where WS but not AS produced an advantage over Control.

Table 2
Effective dimensionality (ED) for the different conditions in the ResNet34 experiment when computed using different normalization schemes. Values are mean ± sem across 5 checkpoints per condition.

Section	Condition	Raw	Mean-centered	L2	Centered + L2
Global ED					
	Control	52.93 ± 0.53	52.31 ± 0.54	51.56 ± 0.52	51.65 ± 0.51
	AS 0.3	1.03 ± 0.00	2.24 ± 0.14	49.19 ± 0.58	49.26 ± 0.57
	AS 1.0	1.01 ± 0.01	1.61 ± 0.33	44.32 ± 0.99	44.74 ± 1.07
	WS 0.3	38.07 ± 0.43	37.31 ± 0.45	37.19 ± 0.41	37.16 ± 0.41
	WS 1.0	21.18 ± 0.35	20.43 ± 0.36	21.15 ± 0.33	21.44 ± 0.32
Within-class ED					
	Control	6.99 ± 0.11	7.68 ± 0.11	10.96 ± 0.13	11.19 ± 0.13
	AS 0.3	6.76 ± 0.08	7.65 ± 0.10	10.39 ± 0.11	10.57 ± 0.11
	AS 1.0	6.82 ± 0.10	7.73 ± 0.08	10.03 ± 0.08	10.14 ± 0.09
	WS 0.3	5.52 ± 0.06	5.73 ± 0.05	7.55 ± 0.05	7.96 ± 0.06
	WS 1.0	3.77 ± 0.05	3.93 ± 0.05	5.34 ± 0.05	5.66 ± 0.05

Topographic regularization improves robustness for most but not all corruptions. For input perturbations, we applied the same corruptions used in the CIFAR-10 and MNIST analyses: white Gaussian noise, pink noise, and salt-and-pepper noise. In addition, because these models operated on larger images, we also applied a corruption based on a circular center mask with radii from 12 to 60 pixels, which tests sensitivity to centrally available information. In all cases, we measured accuracy drop relative to the unperturbed images. The analysis (see Appendix Table A.1) indicated an advantage for the topographic models for white noise, pink noise, and circular masking, with the best performing models often being WS regularized. However, we did not find an advantage for salt-and-pepper noise, for which the control model was most robust in four of the five severity levels. Thus, the robustness gains were consistently found for some corruption types, but not others.

Topographic regularization improves robustness against adversarial attacks. When studying larger models that process high-resolution images, adversarial robustness is also an informative metric. We therefore evaluated two attack families: fast gradient sign method (FGSM) attacks on real images [60], and fooling images, which are synthetic images optimized from scratch to produce a high-confidence prediction for a target class [61]. FGSM provides information about the local geometry of the decision boundary by identifying what shift in input space is sufficient to move an image outside the original category boundary. In contrast, fooling-images test the difficulty of having the model strongly predict a target category even when the driving image is of a very different type (e.g., abstract shapes) from the manifold of natural images on which the model was trained.

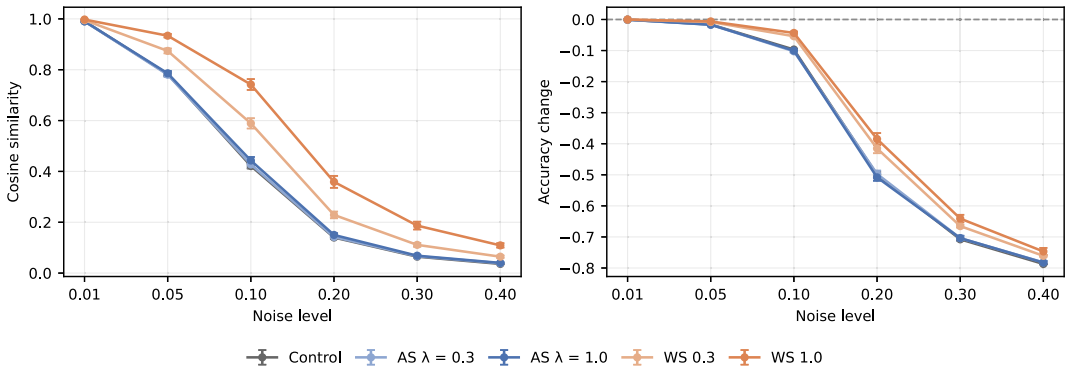


Fig. 15. ResNet34 experiment: WS, but not AS topographic regularization, improves robustness to classifier-weight noise. For increasing noise levels, WS better preserved the original class-weight similarity structure. It also showed a smaller drop in accuracy, with AS remaining close to Control.

For FGSM targets, we used the 100 prototypical images, which were classified by all checkpoints with near-ceiling accuracy. For each image, we asked (i) whether any tested perturbation magnitude (ϵ) could at all change the predicted class, and (ii), if so, at what ϵ the first class switch occurred. We applied this analysis to all 25 checkpoints (5 per condition).

We found a strong effect of topographic regularization on the fraction of prototypes that were never fooled within the tested ϵ range: 0.25 ± 0.01 for control, 0.29 ± 0.01 and 0.36 ± 0.03 for AS 0.3/1.0, and 0.40 ± 0.02 and 0.50 ± 0.03 for WS 0.3/1.0. In other words, on this measure, WS 1.0 was approximately twice as robust as control.

For those images that were successfully attacked, the switch ϵ was highly similar across conditions (median switch ϵ was in a limited range between 0.0039 and 0.0047). This means that when successful, FGSM attacks were equally easy across conditions. This suggests that the advantage for the topographic conditions was due to a larger subset of prototypes that were located in stable positions in decision regions, and were therefore difficult to perturb.

The fooling-image analysis was less informative because attack success (for the three classes chosen) was extremely high for all conditions (0.960–0.993; lower indicates greater robustness). Control reached 0.987 ± 0.008, which was slightly less robust than WS 0.3 and WS 1.0 (0.960 ± 0.032 and 0.973 ± 0.012, respectively). AS was mixed: AS 0.3 was slightly less robust than control at 0.993 ± 0.007, whereas AS 1.0 was somewhat more robust at 0.960 ± 0.012. We conclude that the fooling-image test did not reveal a strong separation between conditions, but here too the topographic models showed a small advantage over control, and certainly not poorer robustness.

6. Discussion

We investigated how topographic regularization affects robustness and representational properties of CNNs, both during end-to-end training of shallow models and during fine-tuning of a standard deep CNN. We focused on two local topographic losses: Activation Similarity (AS), which encourages Pearson correlation between neighboring units' activation vectors (computed across inputs), and Weight Similarity (WS), which encourages similarity between adjacent units' incoming (afferent) weight vectors. Our main objectives were to evaluate (1) robustness to readout-weight perturbations and input corruption, and (2) representational properties including the entropy, sparsity, similarity profiles of the weights and activations, the effective dimensionality, the emergence of class-selective (expert) units, and the spatial organization of responses on the 2D grid including smoothness and functional localization.

The following main findings emerge: (1) Overall, AS and WS models produced greater robustness to input corruption as compared to control models; (2) WS models were more robust to perturbations of the readout (classifier) weight matrix, accompanied by lower activation sparsity (lower PoZ); (3) Both AS and WS models produced smooth topographic maps, but WS did so while producing stronger functional localization; (4) AS could satisfy the regularization objective through a shortcut-like solution where a shared norm is applied across the grid, though it carries little discriminative information.

6.1. Topographic regularization improves robustness to noise and adversarial attacks

One key finding emerging from our study is that topographic regularization can produce models that are more robust to both weight perturbations and input corruption. This is important because several approaches for achieving robustness against noisy inputs have used training on corrupted data [e.g.,50,62]. In prior work, robustness against parameter perturbations [30–32] typically relies on specifically tailored loss terms [63]. In contrast, we find that robustness can be a byproduct of the topographic regularization itself. This suggests that a biologically inspired regularizer can be a viable strategy for improving robustness (see [64] for a review of robustness methods).

Specifically, for ResNet34/ImageNet100, AS and WS were consistently more robust than the control for white noise, pink noise and center-masking corruptions. However, they were frequently less robust against salt-and-pepper corruption. For CIFAR-10 and MNIST, AS and WS also tended to show greater robustness to input corruption. A separate finding emerged from the analysis of robustness to weight perturbations, where for MNIST, CIFAR-10 and ResNet34/ImageNet100 we find that WS was consistently the most robust, showing greater stability of representational similarity matrices and weaker degradation of classification accuracy. Finally, we found that both AS and WS were more robust to FGSM adversarial attacks, but WS showed a particular advantage. These results extend prior work showing that certain types of topographic architectures are more robust against adversarial attacks [6,25,26].

Importantly, we observe a non-monotonic relationship between robustness and model performance. For ResNet34/ImageNet100, the WS condition showed higher accuracy than Control, while at the same time showing greater robustness to adversarial attacks, to weight perturbations, and to multiple types of image corruption. For MNIST, the AS condition with $\lambda = 0.1$ showed better performance than Control, as well as greater robustness to weight perturbations (Fig. A.3) and image corruption. These patterns contrast with prior work reporting that improved accuracy is associated with reduced robustness, particularly in adversarial-robustness settings [51,52]. Our findings point to a potentially important exception, where moderate topographic regularization can, under some settings, improve both classification performance and robustness.

Our findings also show that the relative accuracy of topographic models and non-topographic controls depends on regularization strength

(λ). In fact, previous studies report mixed findings, with some reporting reduced accuracy [4,5,10] or slightly improved performance [1,8,10,65] depending on the setting. These spatial loss functions differ in their topographic objective, including local similarity between unit weights or activations [5,8], and global distance-weighted similarity penalties on activations [1,4] or weight magnitudes [2,65]. While the literature on this point is mixed, our findings suggest that the strength of topographic regularization is a key determinant, and we find multiple cases where topographic networks match or exceed the accuracy of non-topographic controls.

6.2. Topographic regularization reshapes model representations

As noted in the introduction, in machine-learning practice, the presence of correlations among unit activations or among incoming weight vectors is often discouraged due to reduced feature diversity [35,39] and because models often aim to decorrelate representations [36–38]. This emphasis differs from the role of correlations in biological systems, where correlated activity can serve to amplify specific signals [e.g.,33]. In artificial networks, however, similar effects can be achieved without correlated populations, for instance, by using high downstream readout weights. Still, some machine-learning approaches intentionally produce redundancy, for example, by developing methods to cluster similar weights or activations into groups, enforcing within-group similarity, with the intention of providing a basis for subsequent model compression [66–69]. Our results are relevant to this literature, as they indicate that inter-unit similarity should be interpreted with caution and that correlations are not necessarily a sufficient guide for clustering or pruning. Specifically, because the strong correlations in AS can reflect a shared common component, correlation structure alone may not be a good basis for clustering or pruning. While we have not specifically studied the pruning or compression potential of topographic models, several studies have already shown that topographic models are more amenable to pruning [1,5,8].

The changes we find in representational organization were accompanied by changes to single-unit response properties. In all cases, AS and WS reduced the percentage of zero (PoZ) firing as compared to control. For CIFAR-10 and ResNet34/ImageNet100, unit entropy for WS and control was almost identical, with AS showing lower entropy. This suggests that for naturalistic images, WS maintains the capacity of units to respond to multiple types of images. For MNIST, the pattern was different, with both AS and WS showing slightly higher entropy than control. However, we note that for that dataset, all entropy values were within a limited range ($3.7 - 3.75$), suggesting that any impact of the regularizer was quite limited.

Consistent with prior reports [4,6,8], we found that the topographic regularizer could reduce the effective dimensionality of the representations as λ increases (Fig. 6). Both AS and WS training tended to produce lower effective dimensionality than control models. However, not all topographic conditions were associated with lower overall effective dimensionality; specifically, for CIFAR-10, WS showed higher dimensionality than control at two levels of regularization, and AS models did not follow a monotonic decreasing trend. The topographic regularization mainly reduces within-class variation. The results for ResNet34/ImageNet100 were most decisive as they indicated that the results of ED computations should be interpreted with caution, as the network can learn to satisfy the AS regularizer by inducing common image-level norms that produce an apparent reduction in dimensionality (when computed using the typical formula), but a weak effective reduction (as seen in the high accuracy and in the recovery of effective dimensionality after L2 normalization). In contrast, for WS, the reduction in ED appeared to indicate true compression of the representation space.

Lower-dimensional representations have been linked to increased robustness, as they are more resilient to various types of perturbations [70,71]. Related findings in non-topographic models also suggest that encouraging similarity in learned features or activations improves robustness [72,73]. Importantly, our results (Fig. 6) suggest that robustness is not necessarily accompanied by lower dimensionality: in some cases, topographic regularization was associated with both higher overall effective dimensionality and greater robustness. We note that the effective dimensionality results for MNIST differed from those for CIFAR-10 and ResNet34/ImageNet100. Given that these experiments differed strongly in the model used, image complexity, class structure, and variation, it could be expected that even the same regularizer could produce different outcomes.

When analyzing the spatial organization of activity, we found that both AS and WS models produced high spatial smoothness (Moran's I). However, only WS additionally produced functional localization, indicating that highly correlated units tended to be located closer to each other on the grid. In fact, for ResNet34/ImageNet100, the AS condition produced no signature of functional localization as a result of the saturated connectivity matrix. AS and WS also shifted angular and eccentricity tuning, often in similar ways. For MNIST, both increased the expression of a tuning profile in which more eccentric locations were associated with higher weights. A similar, though weaker profile was found for CIFAR-10. For both MNIST and CIFAR-10, this was accompanied by a reduction in the expression of a tuning profile associated with greater weights at central locations. Given that these experiments used simple models and simple stimuli belonging to few classes, the networks could arrive at good performance by learning relatively coarse discriminative features. For this reason, we treat these representations as specific to these training contexts, rather than as an analogue of biological visual-cortical organization.

As in Lu et al. [5], we find that weight-similarity regularization is sufficient for producing smooth activation maps. One difference is that our WS regularizer uses Euclidean distance between neighboring incoming weight vectors, whereas Lu et al. use a cosine-based similarity term. The two may lead to different optimization pressures. Cosine similarity constrains vector directions but allows neighboring units to differ in overall norm. In contrast, Euclidean distance penalizes differences in both direction and magnitude. Furthermore, the two losses reflect slightly different biological considerations: the Euclidean form stays closer to the idea that neighboring units receive similar overall afferent drive, whereas cosine similarity allows for different response gain. For this reason, we cannot say whether our WS results will generalize to other weight-similarity objectives.

Regarding the emergence of expert units, we find that topographic regularization influences unit-level specialization by changing the number and distribution of class-selective expert units. At low levels of λ, both regularizers increase the proportion of strong experts ($AUC > 0.90$) and improve the balance of experts, indicating a more even distribution. A qualitative analysis suggests that the control condition tends to concentrate expertise in fewer classes.

6.3. Practical implications and future directions

An important implication for machine-learning applications is that moderate topographic regularization strengths can improve both task performance and robustness to noise. This was most strongly seen in the ResNet34/ImageNet100 results, where the WS condition displayed accuracy that matched or even slightly exceeded that of the control condition, while showing increased robustness on multiple metrics. This suggests that modulating local correlations may be beneficial in some settings, even when brain-like spatial smoothness is not an objective in itself. Practically, for performance-oriented users interested in improving robustness while maintaining performance, based on our experiments we recommend selecting λ empirically by starting from low values and increasing them while monitoring validation performance relative to the matched control condition (one with a topographic bottleneck but without a similarity loss). While the training process will likely be longer, validation loss may be similar, and this strategy can identify a strong usable topographic constraint that is compatible with

the required task performance. By contrast, for researchers whose main interest is in studying the spatial statistics introduced by the regularizer, weaker values may already be sufficient, because we find that even light regularization produces spatial smoothness in our studies. In that case, post-training validation of spatial organization, for example using Moran's *I*, is especially useful.

A direction for future research is to evaluate pruning in different types of topographic models [1,3,5,8,65]. This can be done not only to achieve compression, but also to identify subnetworks that maintain high performance. Another possibility is to evaluate the existence of "winning tickets" as suggested by the lottery ticket hypothesis [74], to test whether topographic networks are associated with sparser winning tickets. Extending topographic regularization to additional layers, including convolutional ones, could improve robustness to adversarial inputs, as suggested in [25,26]. In parallel, scaling topographic training to large-scale datasets could help facilitate comparisons of noise robustness between models and neural data [75,76].

Finally, we note that we studied topographic regularization in a supervised classification setting, where the spatial losses were jointly optimized with cross-entropy. More generally, topographic regularizers can be combined with many sorts of objectives, including unsupervised and self-supervised losses. An important direction for future research is to understand whether the effects observed here generalize beyond supervised training.

CRediT authorship contribution statement

Nhut Truong: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation. **Uri Hasson:** Writing – review & editing, Writing – original draft, Validation, Software, Project administration, Methodology, Conceptualization.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work the authors used OpenAI's ChatGPT in order to assist with copyediting. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank Gerrit Sander for his comments on the manuscript.

Appendix A. Training dynamics

Accuracy. To evaluate how AS and WS regularization impacted training, we evaluated the cross-entropy loss and spatial-loss trajectory over the training epochs for $\lambda = 0.1$. Fig. A.1 shows the results for MNIST, and Fig. A.2 shows similar findings for CIFAR-10. For both MNIST and CIFAR-10, the trajectory of cross-entropy reduction (and accuracy) was highly similar, for all three types of models. This suggests that the different models learn the differentiation between classes at similar rates. With respect to the spatial loss terms, in MNIST, both AS and WS showed a strong increasing trend in the first few epochs, then the spatial losses dropped towards the end of training. In CIFAR-10, the AS-loss monotonically decreased, while WS-loss only decreased in the latter half of the training process. For both datasets, accuracy remained at a relatively high level (~50% of the initial level) and began to slow down when training accuracy was already high. These data suggest that the inclusion of AS and WS regularization did not strongly impact the dynamics of classification accuracy or those of cross entropy loss during training. They also suggest that both AS and WS loss functions can produce an initial trade-off between the spatial and cross-entropy objectives, perhaps because the topographic regularization harms the feature learning required for classification. Once the features are learned, the spatial objective is more easily satisfied.

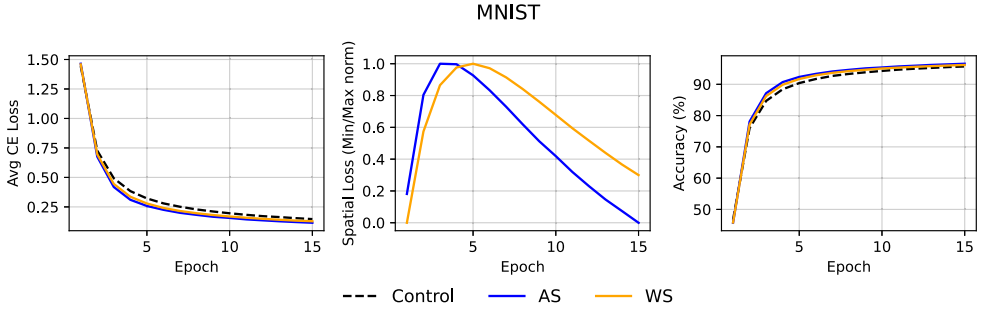


Fig. A.1. MNIST Training stats: Training-set accuracy and loss terms.

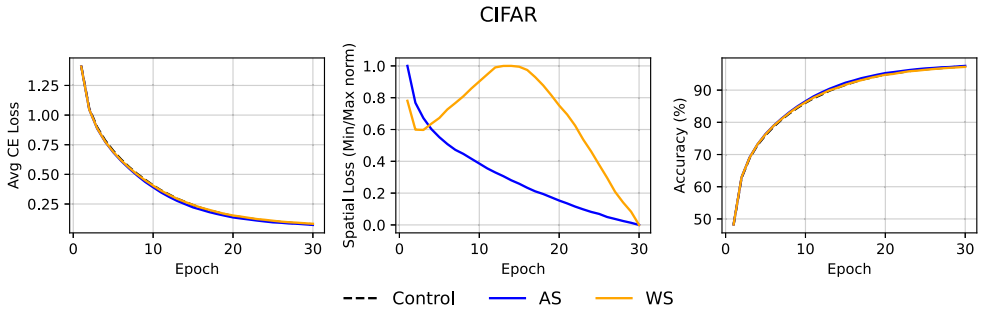


Fig. A.2. CIFAR-10 Training stats: Training-set accuracy and loss terms.

A.1. Appendix figures

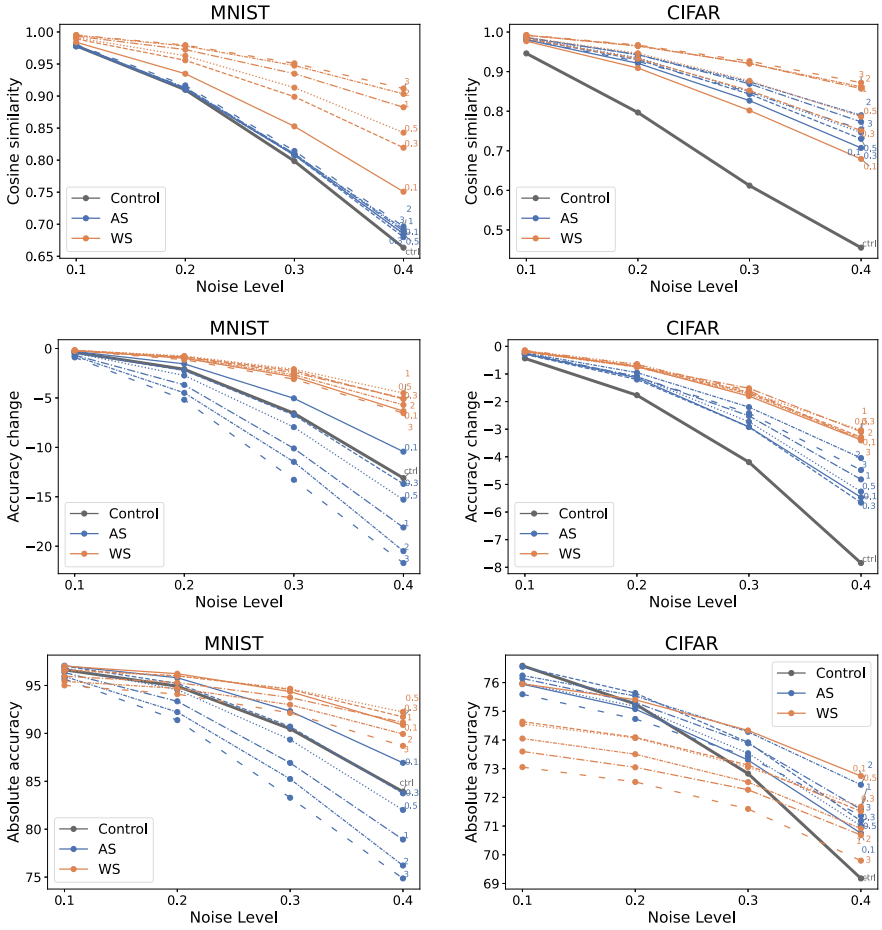
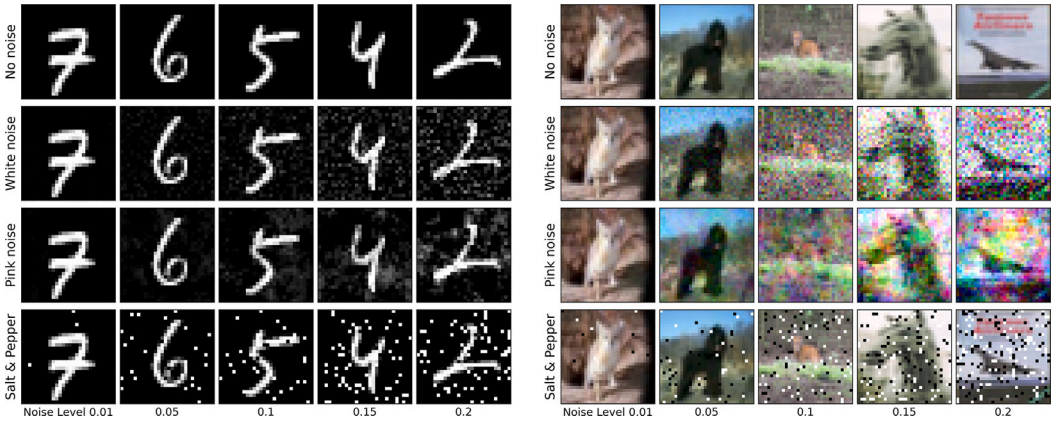


Fig. A.3. Robustness to weight perturbations evaluated across individual values of λ. Robustness is assessed by changes in representational geometry (top row), and drop in test accuracy under increasing levels of additive noise applied to the final classification-layer weights. Raw test accuracy (bottom row) is provided for reference. Representational geometry is defined as the 10×10 cosine similarity matrix computed from category-level weight vectors, and robustness is quantified as the similarity between the original and perturbed matrices. Results are shown separately for MNIST (left) and CIFAR-10 (right), and for control, AS, and WS models at individual values of the regularization parameter λ (indicated by line style).



(a) MNIST (b) CIFAR-10

Fig. A.4. Examples of noise types and noise levels. Representative MNIST (a) and CIFAR-10 (b) images are shown under different noise types (white noise, pink noise, and salt-and-pepper noise) and increasing noise levels. Rows: noise types; columns: noise levels. For reference, noise-free inputs are presented in the top row.

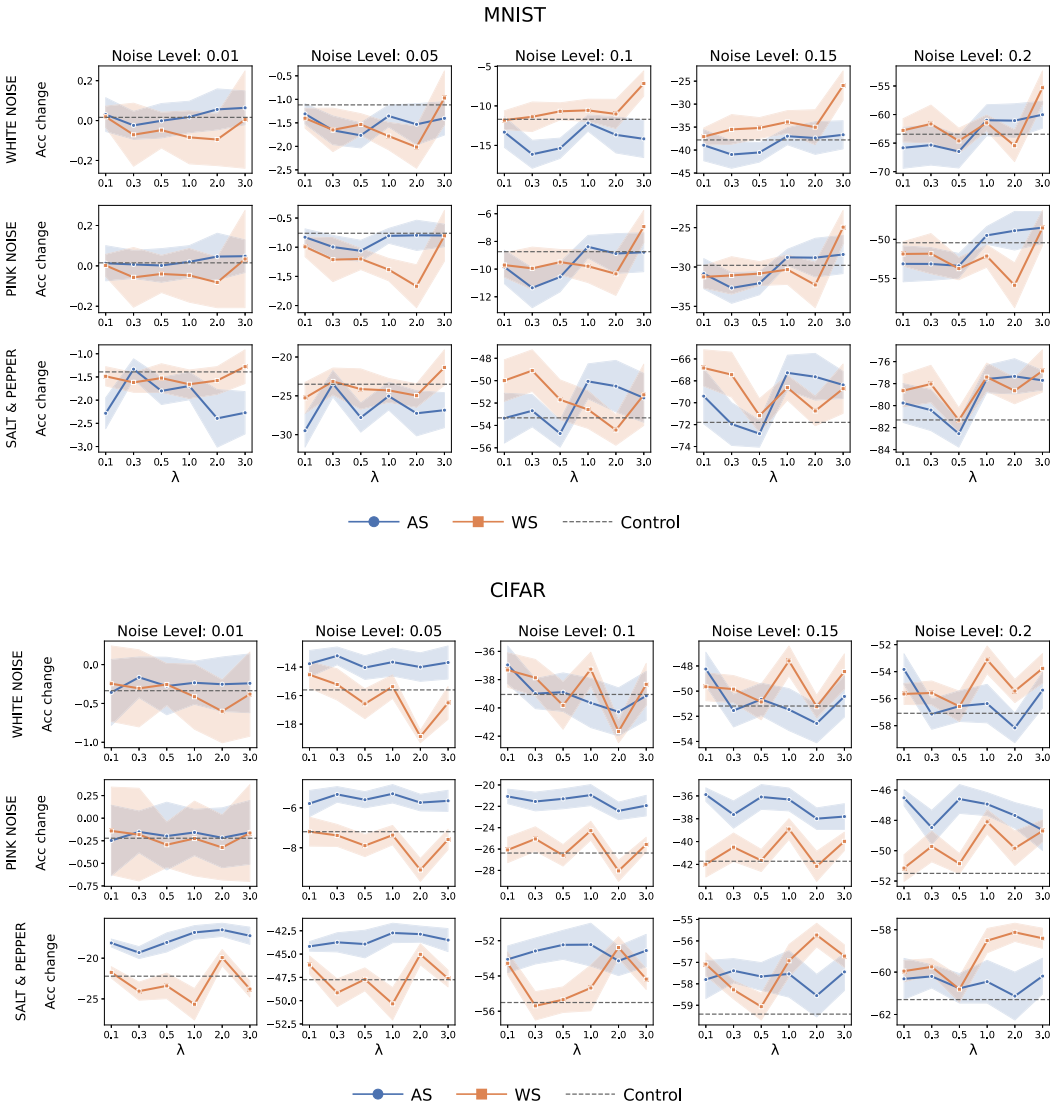


Fig. A.5. Accuracy changes under input corruption across noise type, noise level, and topographic regularization. Changes in classification accuracy compared to baseline (non-perturbed) models are shown as a function of λ for models trained with AS, WS, and control objectives. Results are shown for MNIST (top) and CIFAR-10 (bottom), for three types of corruption (white noise, pink noise, and salt-and-pepper; rows), and for increasing noise levels (columns). Accuracy values are compared to the corresponding noise-free baseline. Shaded regions indicate $\pm s.e.m$ across runs.

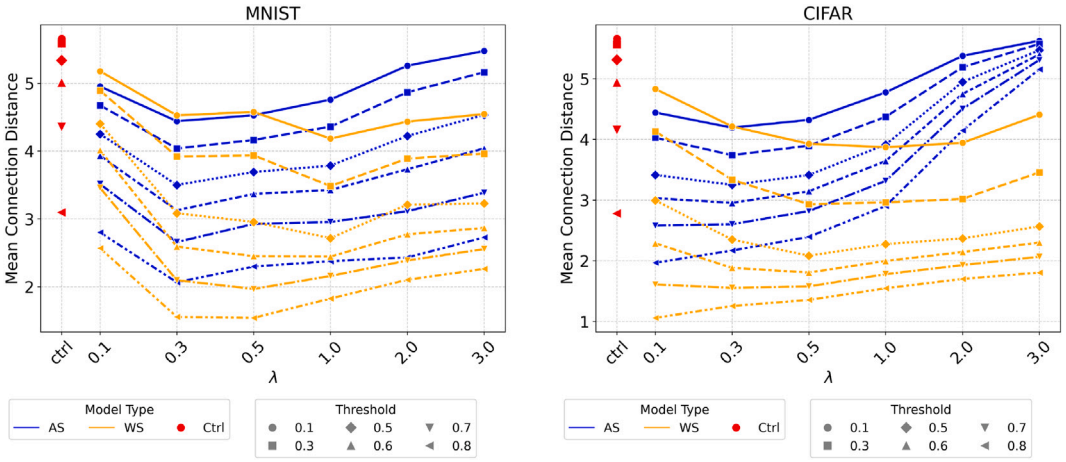


Fig. A.6. Correlated-unit distances across individual correlation thresholds. The mean spatial distance between pairs of units whose activity correlation exceeds a threshold $\alpha \in \{0.1, 0.3, 0.5, 0.6, 0.7, 0.8\}$ is shown for MNIST (left) and CIFAR-10 (right). Separate curves correspond to individual correlation thresholds.

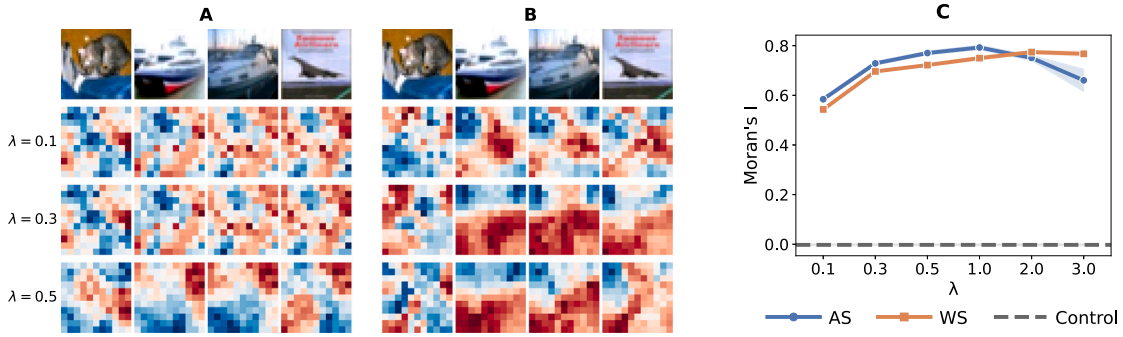


Fig. A.7. Spatial smoothness of activation maps under activation- and weight-similarity training for CIFAR-10.

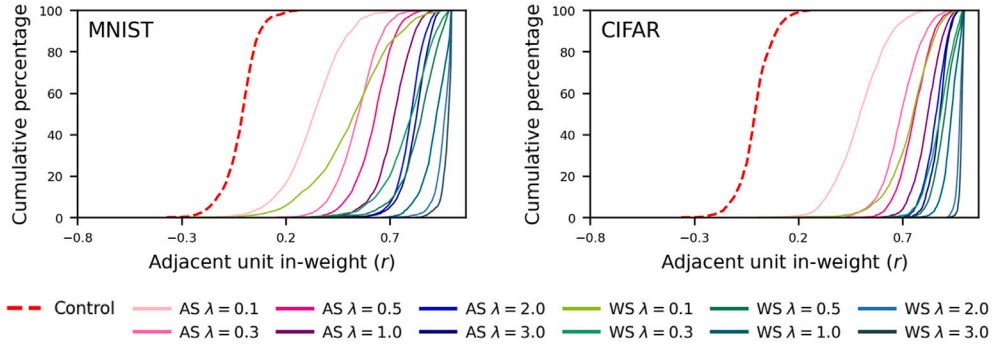


Fig. A.8. Incoming weight correlations between adjacent units. Cumulative distributions of average pairwise correlations between incoming weight vectors of adjacent units. For each unit, the mean correlation is computed from all adjacent neighbors. Distributions are shown for control, AS, and WS models across values of the regularization parameter λ.

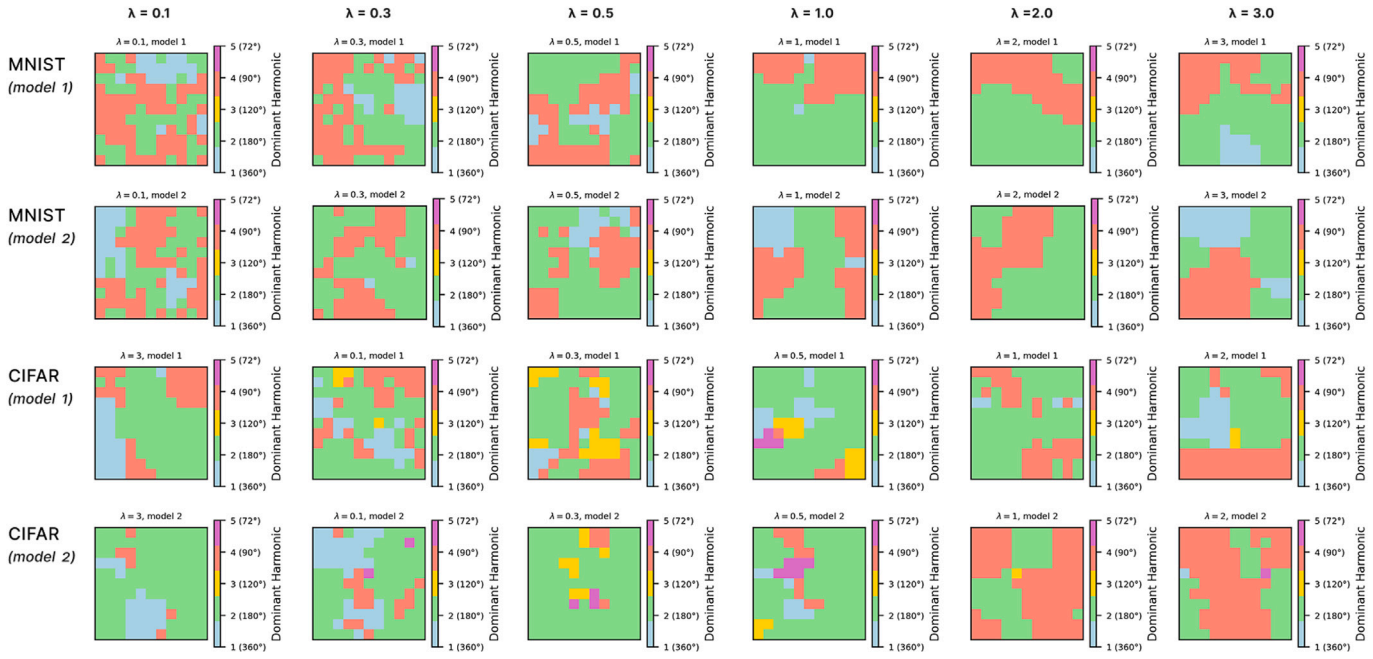


Fig. A.9. Angular response properties under weight-similarity training. Topographic maps show the dominant angular response type for units in the grid across values of the regularization parameter λ under WS training. Angular response types correspond to five harmonic components with periodicities of 360°, 180°, 120°, 90°, and 72° (Cycles 1–5), indicated by color. Results are shown for MNIST and CIFAR-10. Two randomly trained models are shown per dataset. Columns correspond to values of λ.

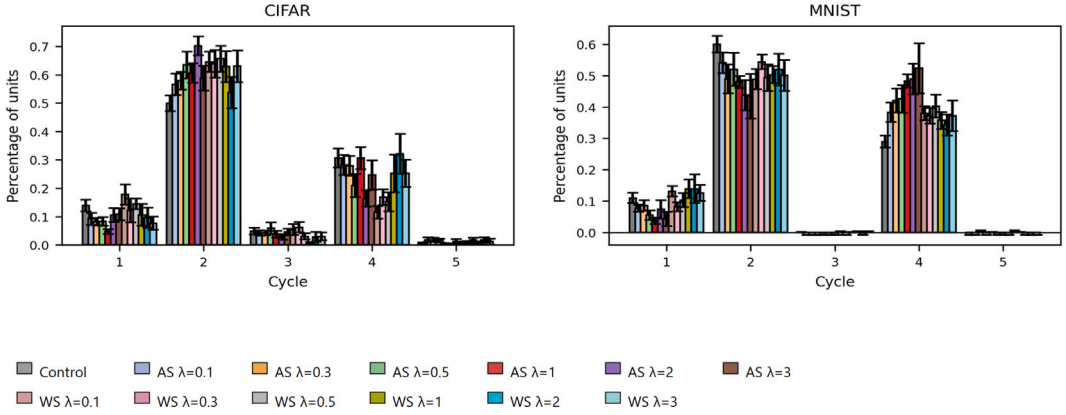


Fig. A.10. Angular tuning profiles across training conditions and regularization strengths (λ). Percentage of units assigned to each angular tracking profile (Cycles 1–5) is shown. Cycles correspond to harmonic response types with periodicities of 360°, 180°, 120°, 90°, and 72°, respectively. Within each cycle, bar order is control, followed by AS and WS, at increasing values of λ. Bars indicate mean percentages across runs; Error bars denote variability across runs.

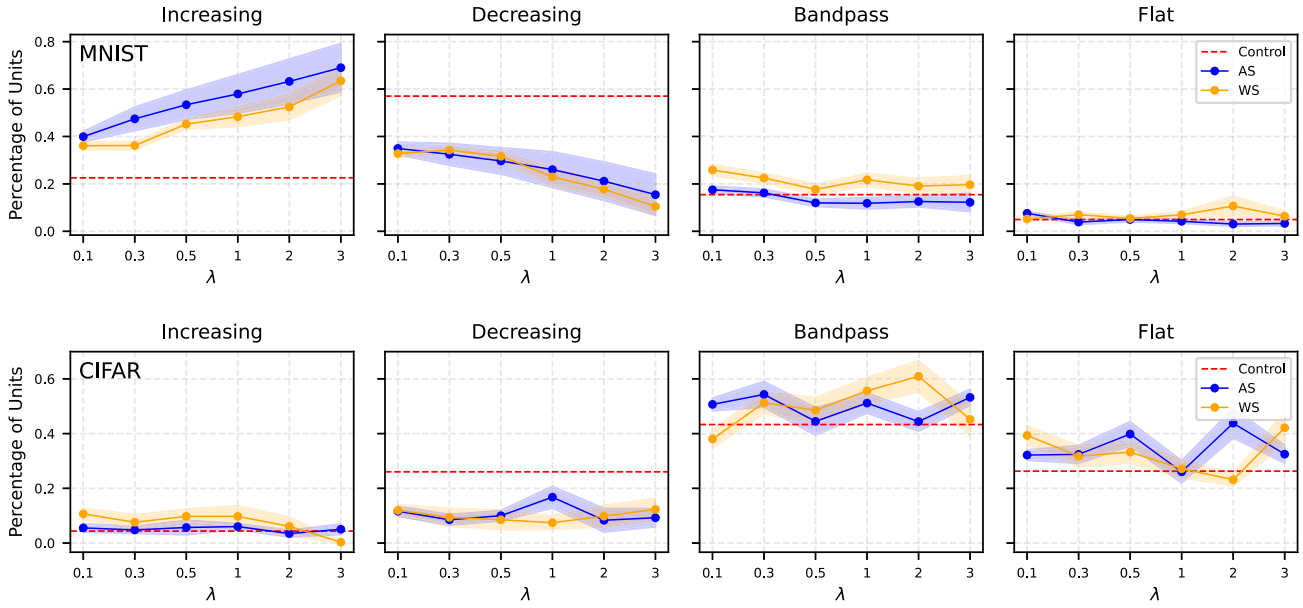


Fig. A.11. Eccentricity tuning profiles across training conditions and regularization strengths. Percentage of units showing increasing, decreasing, band-pass, or flat eccentricity tuning profiles. Each panel corresponds to one eccentricity tuning profile, as defined in the main text. *Increasing* and *decreasing* profiles correspond to monotonic changes in response magnitude with eccentricity and reflect units with positive or negative radial gain, respectively.

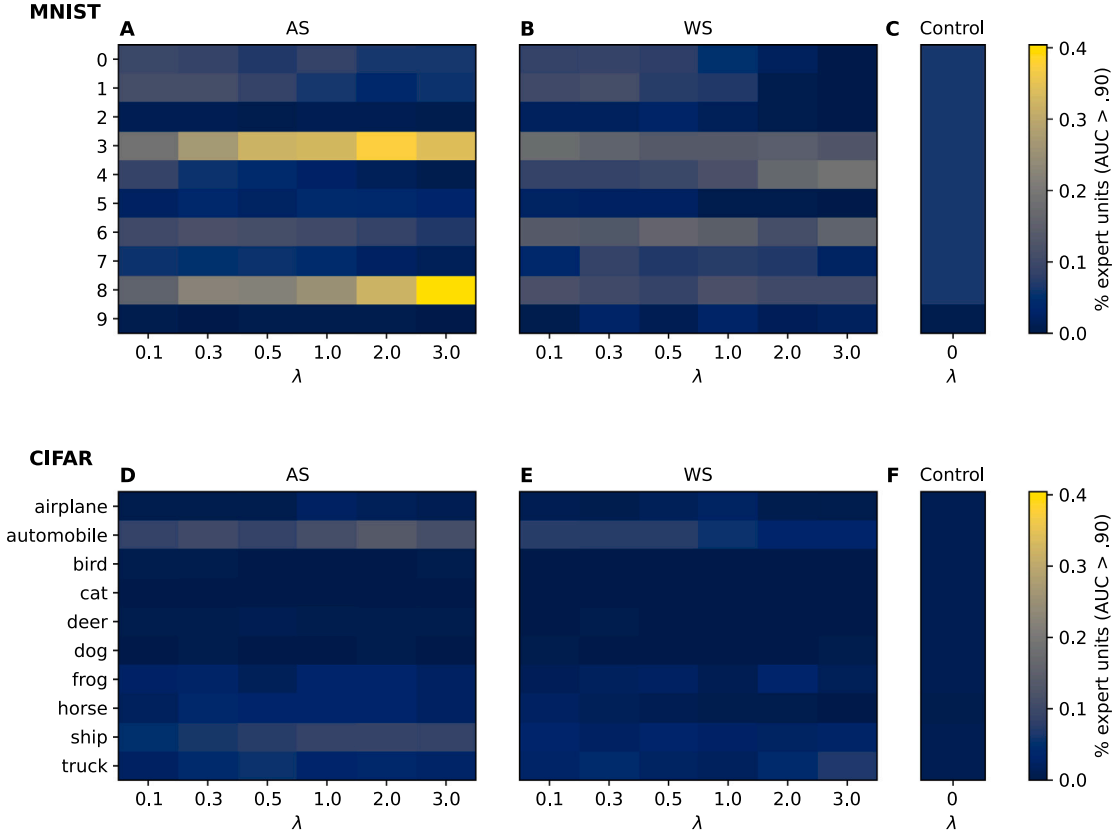


Fig. A.12. Category-wise distribution of high-expertise units across training conditions. Heatmaps show the percentage of units with high category discriminability (AUC > 0.90) for each category as a function of the regularization parameter λ. For AS and WS models, columns correspond to values of λ; control models are shown as $\lambda = 0$. The color scale indicates the proportion of high-expertise units within each category.

Table A.1
Accuracy drop (%) caused by input perturbations in relation to the control condition. Control indicates raw accuracy drop in the control condition. For all other conditions, the entries indicate *control minus condition*; positive values indicate greater robustness for topographic conditions. Bold indicates the best-performing condition per noise treatments (i.e., lowest accuracy drop).

Perturbation type	Severity	Control	AS 0.3	AS 1.0	WS 0.3	WS 1.0
White Gaussian	0.010	0.44	+0.18	+0.14	**+0.23**	+0.13
	0.030	3.13	+0.13	+0.45	**+0.50**	+0.41
	0.050	6.67	−0.22	+0.21	**+0.71**	+0.57
	0.080	13.68	−0.13	−0.27	+0.83	**+1.18**
	0.120	26.09	−1.00	−1.51	+0.48	**+0.65**
Pink noise	0.010	0.50	+0.12	+0.17	**+0.19**	+0.15
	0.030	3.84	−0.09	+0.22	**+0.58**	+0.48
	0.050	9.04	+0.36	+0.58	+0.49	**+0.84**
	0.080	19.61	−0.22	−0.19	+0.91	**+1.08**
	0.120	39.05	−0.52	−1.71	+0.14	**+1.33**
Salt-and-pepper	0.005	**18.22**	−1.08	−0.65	−0.64	−0.66
	0.010	**29.98**	−2.40	−0.93	−1.52	−2.43
	0.020	**40.50**	−2.05	−1.03	−0.33	−1.81
	0.050	49.96	−1.82	−1.85	**+0.36**	+0.05
	0.100	**62.40**	−3.40	−2.41	−0.56	−0.88
Center circular mask	12	5.79	+0.14	**+0.45**	+0.44	+0.38
	24	13.29	+0.83	+1.07	+1.00	**+1.53**
	36	22.90	**+0.86**	+0.78	+0.82	−0.07
	48	34.13	+0.79	**+1.04**	+0.19	−0.77
	60	47.41	**+1.85**	+0.87	−0.89	−1.73

Appendix B. ResNet34 extension methods

ResNet34 topographic architecture. We used torchvision ResNet34 initialized from ImageNet1K_V1 pretrained weights. The original fully connected classification layer was replaced with an identity mapping, resulting in a 512-dimensional global average pooling (GAP) layer. This served as input to a topographic bottleneck. The topographic head itself consisted of a $512 \rightarrow 256$ linear layer. When computing topographic losses these were further interpreted as a 16×16 grid, followed by a ReLU nonlinearity, dropout, and a 100-way linear classifier operating on the post-ReLU grid activity. The ResNet34 backbone was not frozen; all layers were fine-tuned jointly. The control condition used the same 256-unit bottleneck and classifier, but no topographic regularization.

Training objective and topographic regularizers. The total training objective was a joint loss function consisting of a cross-entropy and topographic-loss component weighted with a λ term. Because the ResNet34/ImageNet100 setting was substantially more computationally expensive than the smaller-model setting, we only used $\lambda \in \{0.3, 1.0\}$. For AS, the regularizer processed pre-ReLU bottleneck activity with a loss identical to that used for MNIST and CIFAR-10, the only difference being that 256 units were arranged on a 16×16 grid. For weight similarity (WS), the regularizer operated on the bottleneck weight matrix $W \in \mathbb{R}^{256\times512}$. That is, each unit was associated with a 512-value weight vector that was pushed to be similar to those in its neighborhood.

Optimization and model selection. Training images were augmented with `RandomResizedCrop(224)` and random horizontal flips (probability 0.5). Validation and test images were processed using `Resize(256)` followed by `CenterCrop(224)`. All images were normalized using standard ImageNet channel means and standard deviations. Models were trained with AdamW, using a learning rate of 1×10^{-4}, weight decay, batch size 128, and dropout probability 0.3. The learning rate was reduced when validation loss plateaued, with a factor of 0.5, patience of 4, and a minimum learning rate of 10^{-6}. Training was set to a maximum of 100 epochs, with early stopping if validation loss did not improve by at least 10^{-4} for 10 consecutive epochs. The checkpoint with the lowest validation loss was selected for all further analyses.

Across runs, the pretrained backbone weights and the train/validation/test partition were fixed. Stochasticity was introduced using random initialization of the topographic head, minibatch order, dropout masks, and data augmentation. Five independently trained checkpoints were obtained for each condition.

Prototype construction. Analyses of representational similarity and adversarial analyses were based on one prototype image per class. Prototype images were identified from the held-out ImageNet100 split using GAP-layer embeddings from a pretrained, vanilla ResNet34. For each class, all candidate images were embedded, the class mean embedding was computed, and the prototype was defined as the image whose cosine similarity to the class mean was highest.

Representational similarity analysis. Representational geometry was analyzed using 100×100 first-order representational dissimilarity matrices (RDMs) computed from the prototype set. For each model and each layer of interest, we embedded the prototypes and computed pairwise dissimilarities between all prototype images. RDMs were averaged within each condition and compared using second-order similarity. This allowed us to quantify (i) how far each trained backbone moved from the pretrained ResNet34 backbone, (ii) how much each regularized backbone differed from the control backbone, and (iii) how much the 256-unit bottleneck changed the representation relative to its own backbone. The main layers analyzed were the GAP representation and the 256-unit bottleneck representation.

Spatial topography metrics. To quantify topographic organization, we reshaped the resulting 256-dimensional activity vector into a 16×16 grid for each image. We computed three metrics per checkpoint. *Moran's I* quantified local spatial autocorrelation on each prototype activation map using a lattice-based neighborhood definition over the 16×16 grid. Values were averaged across the 100 prototype maps to obtain a checkpoint-level estimate. *Unit Entropy* quantified the variance of each unit's response distribution across the 100 prototypes. For each unit, a histogram of response magnitudes was computed across the prototype set, and Shannon entropy was computed from that histogram. The mean entropy across the 256 units was the recorded value. Because entropy was computed from per unit assessment across images, low values indicate highly concentrated responses at the unit level, not high sparsity at the population level. Post-ReLU *PoZ* (percentage of zero firing) was

computed as the fraction of prototype responses that produced no firing after the ReLU nonlinearity, averaged across units.

Thresholded response-correlation graphs. To assess functional localization more directly, we computed thresholded response-correlation graphs from activity matrices for the prototypes, computed on the topographic grid. For each checkpoint, pairwise correlations were computed between the 256 units across prototypes. For a set of thresholds

$$T \in \{0.1, 0.3, 0.5, 0.6, 0.7, 0.8, 0.9, 0.96\},$$

we computed an undirected graph by considering a unit pair as connected if it had a correlation greater than T. For each threshold T, two summary measures were extracted: (i) the mean number of connected units per unit, and (ii) the mean Euclidean connection distance on the 16×16 grid. These measures were then averaged within conditions across the 5 checkpoints.

Classifier-weight noise robustness. To evaluate the robustness of the learned readout, we added Gaussian noise to the final classifier parameters, without modifying upstream features. For each checkpoint, noise was added to the classifier weights at levels

$$\sigma \in \{0.01, 0.05, 0.10, 0.20, 0.30, 0.40\},$$

with repeated noise sampled ($n = 20$ times) per checkpoint and noise level. For each noisy realization, we measured (i) classification accuracy on the held-out validation split, and (ii) similarity to the original class-weight similarity structure. The latter was quantified by computing a 100×100 cosine similarity matrix across class-weight vectors, vectorizing its upper triangle, and measuring cosine similarity between the original and noisy vectors.

Image corruption robustness. Input corruption robustness was evaluated on the held-out ImageNet100 split. We applied four types of perturbations with the following severity levels:

White Gaussian noise: {0.01, 0.03, 0.05, 0.08, 0.12}
Pink noise: {0.01, 0.03, 0.05, 0.08, 0.12}
Salt-and-pepper noise: {0.005, 0.01, 0.02, 0.05, 0.10}
Center-mask radii: {12, 24, 36, 48, 60} pixels.

For each checkpoint and each severity level, we measured classification accuracy on the perturbed images and reported the *drop* as compared to the clean baseline.

FGSM adversarial robustness. FGSM robustness was evaluated on the 100 prototypes. For each checkpoint, we first identified the prototypes that were classified correctly under clean input, and attacked those. The epsilon grid was

$$\epsilon \in \left\{0, \frac{1}{255}, \frac{2}{255}, \frac{4}{255}, \frac{8}{255}, \frac{12}{255}, \frac{16}{255}\right\}.$$

For each prototype, we recorded if any tested ϵ caused a class switch and, if so, the minimal ϵ that produced the first switch. Reported FGSM summaries were aggregated over the 5 checkpoints per condition.

Fooling-image generation. Fooling-image robustness was evaluated using a CPPN-based optimization procedure (Compositional Pattern-Producing Networks; [61]). The analysis utilized all 25 checkpoints, with three target classes selected randomly. We used 10 optimization seeds for each model-class pair. CPPN optimizations were allowed 700 steps with a learning rate of 0.02. The loss function was a joint loss that maximized the target-class logit while penalizing total variation and image magnitude. The CPPN network used a hidden width of 64, a depth of 3, tanh activations, and coordinate inputs consisting of x, y, radius, and bias.

Aggregation and reporting. All condition-level summaries for the ResNet34 extension were based on five independently trained checkpoints per condition. For those analyses that required additional stochasticity (classifier-weight noise and fooling-image generation), random draws were aggregated within each checkpoint and then averaged across checkpoints.

Data and code availability

The code used to generate the results in this study is publicly available at: https://github.com/tlmnhut/weight_vs_act_toponet. All datasets used in this study are publicly available: MNIST [41], CIFAR-10 [42], and ImageNet100 (a subset of ImageNet). No new datasets were generated. All main analyses can be reproduced using the provided code and publicly available datasets.

References

[1] M. Poli, E. Dupoux, R. Riad, Introducing topography in convolutional neural networks, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2023, pp. 1–5.
[2] R.A. Jacobs, M.I. Jordan, Computational consequences of a bias toward short connections, J. Cogn. Neurosci. 4 (4) (1992) 323–336.
[3] N.M. Blauch, M. Behrmann, D.C. Plaut, A connectivity-constrained computational account of topographic organization in primate high-level visual cortex, Proc. Natl. Acad. Sci. 119 (3) (2022) e2112566119.
[4] E. Margalit, H. Lee, D. Finzi, J.J. DiCarlo, K. Grill-Spector, D.L. Yamins, A unifying framework for functional organization in early and higher ventral visual cortex, Neuron 112 (14) (2024) 2435–2451.
[5] Z. Lu, A. Doerig, V. Bosch, B. Krahmer, D. Kaiser, R.M. Cichy, T.C. Kietzmann, End-to-end topographic networks as models of cortical map formation and human visual behaviour, Nat. Hum. Behav. (2025) 1–17.
[6] X. Qian, A.O. Dehghani, A.B. Farahani, P. Bashivan, Local lateral connectivity is sufficient for replicating cortex-like topographical organization in deep neural networks, Nat. Commun. 17 (2026) 4042.
[7] F.R. Doshi, T. Konkle, Cortical topographic motifs emerge in a self-organized map of object space, Sci. Adv. 9 (Jun 2023) eade8187.
[8] M. Deb, M. Deb, N. Murty, TopoNets: high performing vision and language models with brain-like topography, arXiv preprint arXiv:2501.16396, 2025.
[9] Y. Zhang, K. Zhou, P. Bao, J. Liu, A biologically inspired computational model of human ventral temporal cortex, Neural Netw. 178 (2024) 106437.
[10] N. Rathi, J. Mehrer, B. AlKhamissi, T. Binhuraib, N.M. Blauch, M. Schrimpf, TopoLM: brain-like spatio-functional organization in a topographic language model, arXiv preprint arXiv:2410.11516, 2024.
[11] T. Binhuraib, G. Tuckute, N. Blauch, Topoformer: brain-like topographic organization in transformer language models through spatial querying and reweighting, arXiv preprint arXiv:2510.18745, 2025.
[12] H. Al-Tahan, M. Deb, J. Feather, N. Murty, End-to-end topographic auditory models replicate signatures of human auditory cortex, arXiv preprint arXiv:2509.24039, 2025.
[13] E. Zohary, M.N. Shadlen, W.T. Newsome, Correlated neuronal discharge rate and its implications for psychophysical performance, Nature 370 (6485) (1994) 140–143.
[14] B.J. Hansen, M.I. Chelaru, V. Dragoi, Correlated variability in laminar cortical circuits, Neuron 76 (3) (2012) 590–602.
[15] K.D. Harris, T.D. Mrsic-Flogel, Cortical connectivity and sensory coding, Nature 503 (7474) (2013) 51–58.
[16] L.F. Abbott, P. Dayan, The effect of correlated variability on the accuracy of a population code, Neural Comput. 11 (1) (1999) 91–101.
[17] H. Lee, E. Margalit, K.M. Jozwik, M.A. Cohen, N. Kanwisher, D.L.K. Yamins, J.J. DiCarlo, Topographic deep artificial neural networks reproduce the hallmarks of the primate inferior temporal cortex face processing network, Biorxiv (Jul 2020. Preprint.
[18] V. Krug, R.K. Ratul, C. Olson, S. Stober, Visualizing deep neural networks with topographic activation maps, in: HHAI 2023: Augmenting Human Intellect, IOS Press, 2023, pp. 138–152.
[19] T. Hannagan, Reset networks: emergent topography by composition of convolutional neural networks, Biorxiv (2021).
[20] T.A. Keller, Q. Gao, M. Welling, Modeling Category-Selective Cortical Regions with Topographic Variational Autoencoders, arXiv, Oct 2021. Preprint, v2: Dec 19, 2021; SVRHM workshop @ NeurIPS 2021.
[21] D. Jiang, T. Wang, S. Tang, T. Lee, Computational Constraints Underlying the Emergence of Functional Domains in the Topological Map of Macaque V4, bioRxiv, Nov 2024. Preprint, version 1.
[22] A. Dehghani, X. Qian, A. Farahani, P. Bashivan, Credit-based self organizing maps: training deep topographic networks with minimal performance degradation, in: The Thirteenth International Conference on Learning Representations, 2024.
[23] D. Finzi, E. Margalit, K. Kay, D.L.K. Yamins, K. Grill-Spector, A Single Computational Objective Drives Specialization of Streams in Visual Cortex, bioRxiv, 2023.
[24] J. Achterberg, D. Akarca, D. Strouse, J. Duncan, D.E. Astle, Spatially embedded recurrent neural networks reveal widespread links between structural and functional neuroscience findings, Nat. Mach. Intell. 5 (12) (2023) 1369–1381.

[25] D. Zhou, Y. Fang, Z. Wang, R. Xu, TDSNNs: competitive topographic deep spiking neural networks for visual cortex modeling, arXiv preprint arXiv:2508.04270, 2025.
[26] P. Bashivan, R. Bayat, A. Ibrahim, A. Dehghani, Y. Ren, Learning adversarially robust kernel ensembles with kernel average pooling, Expert Syst. Appl. 266 (2025) 126017.
[27] K.M. Jozwik, H. Lee, N.G. Kanwisher, J.J. DiCarlo, First steps in using topographic deep artificial neural network models to generate hypotheses about not-yet-detected functional neural clusters in the ventral stream, in: 2023 Conference on Cognitive Computational Neuroscience, 2023.
[28] J. Mehrer, C.J. Spoerer, N. Kriegeskorte, T.C. Kietzmann, Individual differences among deep neural network models, Nat. Commun. 11 (1) (2020) 5725.
[29] D. Cortinovis, N. Truong, H. Op de Beeck, S. Bracci, Investigating action topography in visual cortex and deep artificial neural networks, Nat. Commun. 17 (2026) 1094.
[30] N. Cheney, M. Schrimpf, G. Kreiman, On the robustness of convolutional neural networks to internal architecture and weight perturbations, arXiv preprint arXiv:1703.08245, 2017.
[31] A.P. Arechiga, A.J. Michaels, The effect of weight errors on neural networks, in: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2018, pp. 190–196.
[32] A.G. Savva, T. Theocharides, C. Nicopoulos, Robustness of artificial neural networks based on weight alterations used for prediction purposes, Algorithms 16 (7) (2023) 322.
[33] M.N. Shadlen, W.T. Newsome, The variable discharge of cortical neurons: implications for connectivity, computation, and information coding, J. Neurosci. 18 (10) (1998) 3870–3896.
[34] T.N. Aflalo, M.S.A. Graziano, Possible origins of the complex topographic organization of motor cortex: reduction of a multidimensional space onto a two-dimensional array, J. Neurosci. 26 (23) (2006) 6288–6297.
[35] J. Zbontar, L. Jing, I. Misra, Y. LeCun, S. Deny, Barlow twins: self-supervised learning via redundancy reduction, in: International Conference on Machine Learning, PMLR, 2021, pp. 12310–12320.
[36] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, D. Batra, Reducing overfitting in deep networks by decorrelating representations, arXiv preprint arXiv:1511.06068, 2015.
[37] P. Rodríguez, J. Gonzalez, G. Cucurull, J.M. Gonfaus, X. Roca, Regularizing CNNS with locally constrained decorrelations, arXiv preprint arXiv:1611.01967, 2016.
[38] G. Jin, X. Yi, L. Zhang, L. Zhang, S. Schewe, X. Huang, How does weight correlation affect generalisation ability of deep neural networks? Adv. Neural Inf. Process. Syst. 33 (2020) 21346–21356.
[39] Z. Wang, C. Xiang, W. Zou, C. Xu, Mma regularization: decorrelating weights of neural networks by maximizing the minimal angles, Adv. Neural Inf. Process. Syst. 33 (2020) 19099–19110.
[40] H. Zhang, H. Hu, D. Zhou, X. Zhang, B. Cao, Compact CNN module balancing between feature diversity and redundancy, Neural Netw. 188 (2025) 107456.
[41] Y. LeCun, The MNIST database of handwritten digits, 1998, http://yann.lecun.com/exdb/mnist/ (Accessed 10 January 2026).
[42] A. Krizhevsky, Learning Multiple Layers of Features From Tiny Images, tech. rep., University of Toronto, 2009.
[43] B.M. Lake, W. Zaremba, R. Fergus, T.M. Gureckis, Deep neural networks predict category typicality ratings for images, in: Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 37, 2015.
[44] G.K. Nayak, K.R. Mopuri, V. Shaj, V.B. Radhakrishnan, A. Chakraborty, Zero-shot knowledge distillation in deep networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 4743–4751.
[45] K. Filus, J. Domańska, Extracting coarse-grained classifiers from large convolutional neural networks, Eng. Appl. Artif. Intell. 138 (2024) 109377.
[46] K. Filus, J. Domańska, What is the doggest dog? Examination of typicality perception in ImageNet-trained networks, Neural Netw. 188 (2025) 107425.
[47] M. Fedzechkina, E. Gualdoni, S. Williamson, K. Metcalf, S. Seto, B.-J. Theobald, ExpertLens: activation steering features are highly interpretable, arXiv preprint arXiv:2502.15090, 2025.
[48] ImageNet100, Kaggle dataset page, 2026, https://www.kaggle.com/datasets/ambityga/imagenet100 (accessed 7 April 2026).
[49] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, arXiv preprint arXiv:1903.12261, 2019.
[50] R.G. Lopes, D. Yin, B. Poole, J. Gilmer, E.D. Cubuk, Improving robustness without sacrificing accuracy with patch Gaussian augmentation, arXiv preprint arXiv:1906.02611, 2019.
[51] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, A. Madry, Robustness may be at odds with accuracy, arXiv preprint arXiv:1805.12152, 2018.
[52] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, Y. Gao, Is robustness the cost of accuracy?–A comprehensive study on the robustness of 18 deep image classification models, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 631–648.
[53] A. Polyak, L. Wolf, Channel-level acceleration of deep face representations, IEEE Access 3 (2015) 2163–2175.
[54] J. Wang, T. Jiang, Z. Cui, Z. Cao, Filter pruning with a feature map entropy importance criterion for convolution neural networks compressing, Neurocomputing 461 (2021) 41–54.
[55] J.-H. Luo, J. Wu, An entropy-based pruning method for CNN compression, arXiv preprint arXiv:1706.05791, 2017.
[56] Z. Liao, V. Quétu, V.-T. Nguyen, E. Tartaglione, Can unstructured pruning reduce the depth in deep neural networks? in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 1402–1406.
[57] H. Hu, G. Penna, V. Monga, Network trimming: a data-driven neuron pruning approach towards efficient deep architectures, arXiv preprint arXiv:1607.03250, 2016.
[58] P.A. Moran, Notes on continuous stochastic phenomena, Biometrika 37 (1/2) (1950) 17–23.
[59] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[60] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572, 2014.
[61] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: high confidence predictions for unrecognizable images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 427–436.
[62] J. Sietsma, R.J. Dow, Creating artificial neural networks that generalize, Neural Netw. 4 (1) (1991) 67–79.
[63] Y.-L. Tsai, C.-Y. Hsu, C.-M. Yu, P.-Y. Chen, Formalizing generalization and adversarial robustness of neural networks to weight perturbations, Adv. Neural Inf. Process. Syst. 34 (2021) 19692–19704.
[64] J. Liu, Y. Jin, A comprehensive survey of robust deep learning in computer vision, J. Autom. Intell. 2 (4) (2023) 175–195.
[65] X.-J. Zhang, J.M. Moore, T.-T. Gao, X. Zhang, G. Yan, Brain-inspired wiring economics for artificial neural networks, PNAS Nexus 4 (1) (2025) pgae580.
[66] S. Han, H. Mao, W.J. Dally, Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding, arXiv preprint arXiv:1510.00149, 2015.
[67] D. Zhang, H. Wang, M. Figueiredo, L. Balzano, Learning to share: simultaneous parameter tying and sparsification in deep learning, in: International Conference on Learning Representations, 2018.
[68] J.O. Neill, G.V. Steeg, A. Galstyan, Compressing deep neural networks via layer fusion, arXiv preprint arXiv:2007.14917, 2020.
[69] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 499–515.
[70] A. Sanyal, V. Kanade, P.H. Torr, P.K. Dokania, Robustness via deep low-rank representations, arXiv preprint arXiv:1804.07090, 2018.
[71] P. Awasthi, H. Jain, A.S. Rawat, A. Vijayaraghavan, Adversarial robustness via robust low rank representations, Adv. Neural Inf. Process. Syst. 33 (2020) 11391–11403.
[72] M.R. Nassar, D. Scott, A. Bhandari, Noise correlations for faster and more robust learning, J. Neurosci. 41 (31) (2021) 6740–6752.
[73] S.K. Gourtani, N. Meratnia, Improving robustness of compressed models with weight sharing through knowledge distillation, in: 2024 IEEE 10th International Conference on Edge Computing and Scalable Cloud (EdgeCom), IEEE, 2024, pp. 13–21.
[74] J. Frankle, M. Carbin, The lottery ticket hypothesis: finding sparse, trainable neural networks, arXiv preprint arXiv:1803.03635, 2018.
[75] H. Jang, D. McCormack, F. Tong, Noise-trained deep neural networks effectively predict human vision and its neural responses to challenging images, PLOS Biol. 19 (12) (2021) e3001418.
[76] H. Jang, F. Tong, Improved modeling of human vision by incorporating robustness to blur in convolutional neural networks, Nat. Commun. 15 (1) (2024) 1989.

Author biography

Dr. Nhut Truong received his Ph.D. from the University of Trento in 2026. His research interests include topographic neural networks, representation learning, NeuroAI and computational neuroscience.

Dr. Uri Hasson is faculty at the University of Trento. His research interests include cognitive neuroscience, neural representations, and computational models of the human brain and behavior.