Learning to cluster neuronal function

Nina S. Nellen,[1,†,*] Polina Turishcheva,[1,†,*] Michaela Vystrčilová,[1]
Shashwat Sridhar,[2,3] Tim Gollisch,[2–5] Andreas S. Tolias,[6–9] Alexander S. Ecker[1,10,*]

[1] Institute of Computer Science and Campus Institute Data Science, University Göttingen, Germany
[2] University Medical Center Göttingen, Department of Ophthalmology, Göttingen, Germany
[3] Bernstein Center for Computational Neuroscience Göttingen, Göttingen, Germany
[4] Cluster of Excellence "Multiscale Bioimaging: from Molecular Machines to Networks of Excitable Cells" (MBExC), University of Göttingen, Germany
[5] Else Kröner Fresenius Center for Optogenetic Therapies, University Medical Center Göttingen, Germany
[6] Department of Ophthalmology, Byers Eye Institute, Stanford University School of Medicine, CA, US
[7] Stanford Bio-X, Stanford University, Stanford, CA, US
[8] Wu Tsai Neurosciences Institute, Stanford University, CA, US
[9] Department of Electrical Engineering, Stanford University, CA, US
[10] Max Planck Institute for Dynamics and Self-Organization, Göttingen, Germany

[†] Shared contribution
[*]nina.nellen@uni-goettingen.de
[*]{turishcheva,ecker}@cs.uni-goettingen.de

Abstract

Deep neural networks trained to predict neural activity from visual input and behaviour have shown great potential to serve as digital twins of the visual cortex. Per-neuron embeddings derived from these models could potentially be used to map the functional landscape or identify cell types. However, state-of-the-art predictive models of mouse V1 do not generate functional embeddings that exhibit clear clustering patterns which would correspond to cell types. This raises the question whether the lack of clustered structure is due to limitations of current models or a true feature of the functional organization of mouse V1. In this work, we introduce DECEMber – Deep Embedding Clustering via Expectation Maximization-based refinement – an explicit inductive bias into predictive models that enhances clustering by adding an auxiliary t-distribution-inspired loss function that enforces structured organization among per-neuron embeddings. We jointly optimize both neuronal feature embeddings and clustering parameters, updating cluster centers and scale matrices using the EM-algorithm. We demonstrate that these modifications improve cluster consistency while preserving high predictive performance and surpassing standard clustering methods in terms of stability. Moreover, DECEMber generalizes well across species (mice, primates) and visual areas (retina, V1, V4). The code is available at `https://github.com/Nisone2000/DECEMber`, `https://github.com/ecker-lab/cnn-training`.

1 Introduction

Understanding whether neurons form discrete cell types or lie on a continuum is a fundamental question in neuroscience [1]. Previous research has extensively investigated the morphological and electrophysiological properties of neurons in the visual cortex. While discrete anatomical and transcriptomic classifications have been proposed [2–4], recent work on the mouse brain suggests a

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

more continuous organization [5, 6]. Significantly less attention has been devoted to the neurons' functional properties. Each neuron can be characterized by a function that maps high-dimensional sensory inputs to its one-dimensional neuronal response. These functions are highly non-linear, making their analysis complex. Discrete functional cell types are well established in the retina [7] but their existence remains unclear in the mouse visual cortex.

Recently, deep networks showed great potential for predicting neural activity from sensory input [8–15] and also in inferring novel functional properties [16–18]. These networks learn per-neuron vectors of parameters, which are interpreted as neuronal functional embeddings. There were several attempts to use these embeddings to reveal the underlying structure of neuronal population functions through unsupervised clustering [13, 19–21]. However, in none of these studies well-separated clusters emerged, raising the question of whether distinct functional cell types exist among excitatory neurons in the mouse visual cortex. A central challenge is cluster consistency: How reliably are neurons grouped into the same cluster across different model runs? Clustering metrics such as the Adjusted Rand Index (ARI) [22], which evaluates cluster assignment agreement across different seeds or clustering methods and similarity metrics between individual neurons' remained relatively low [13]. These low scores show that clustering results lack the stability and distinctiveness necessary to making strong claims about biological interpretations.

In this work, we incorporate an explicit clustering bias into the training of neuronal embeddings to improve the identifiability of functional cell types, One could view it as model-driven hypothesis testing: if clear functional cell types exist then such bias should improve the model performance, embeddings structure and/or cluster consistencies.

To improve the cluster separability of neuronal embeddings we took inspiration from Deep Embedding Clustering (DEC) [23] and introduced a new clustering loss, which combines updating clusters' locations and shapes along with learning feature representations. We measured the consistency of clustered features across models fitted on different seeds by computing ARI on their clustering results. Additionally, we examined how the clustering loss strength influenced models' performance.

Our contributions are

- We adapted the DEC-loss [23] to allow for non-isotropic multivariate clusters of different sizes by learning a multivariate t mixture model [24].
- We improved cluster consistency while maintaining a state-of-the-art predictive model performance.
- We showed that our method generalizes well, improving cluster consistency across different species, visual areas, and model architectures.

2 Background and related work

Predictive models for visual cortex. In comparison to task-driven networks [25–28], pioneering data-driven population models [10, 29, 30] introduced the core-readout framework, which separates the stimulus-response functions of neurons into a shared nonlinear feature space (core) and per-neuron specific set of linear weights – the readout. The core is shared among all neurons and outputs a nonlinear set of basis functions spanning the feature space of the neuronal nonlinear input-output functions of dimension (height $\times$ width $\times$ feature channels). The early models were extended by including behavioral modulation [17, 31], latent brain state [11, 32] or the perspective transformations of the eye [12]. The core architecture was improved by introducing biological biases such as a rotation-equivariant core [14] to account for orientation selectivity in V1 neurons [33], extending to dynamic models with video input [9, 12, 15, 18, 31] or using transformer architectures [34].

Klindt et al. [35] introduced a factorized readout for each neuron, comprising a spatial mask M_n specifying its receptive field (RF) position and feature weights. This approach was refined by Lurz et al. [36], who proposed the Gaussian readout, replacing the full spatial mask with a pair of coordinates (x_n, y_n) drawn from a learned normal distribution. To predict the neuronal response the model computes the dot product between the neuron's weight vector (per-neuron embedding) and each feature map at the RF location. For later visual layers, like V4, the receptive field location is not necessarily fixed. Therefore, Pierzchlewicz et al. [37] introduced an attention readout, which indicates the most important feature locations for a neuron n depending on the input image.

While different readouts exist, few works have examined their consistency. Turishcheva et al. [13] showed that factorized readouts produced more consistent neuronal clusters than Gaussian readouts,

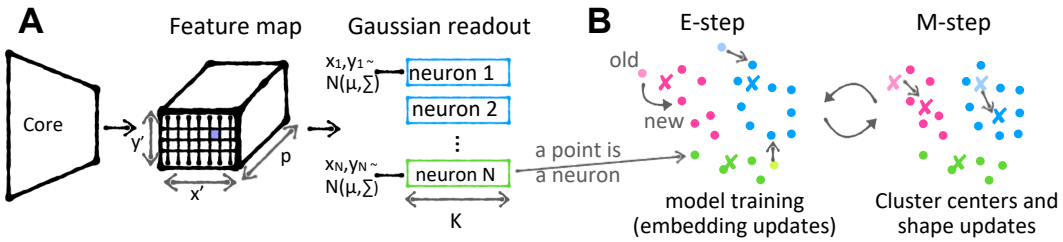

Figure 1: **A: Model architecture:** The model consists of a neuronwise shared core outputting a feature map of size (height $\times$ width $\times$ feature channels) and neuron specific Gaussian readouts. They consist of a receptive field position and a weight vector. The RF position chooses the vector in the feature map which is then combined with the neuron's weight vector by a dot product to get the neuron's response. **B: Clustering procedure:** We're clustering the readouts with an additional loss to incorporate the cluster bias into the features. We update the clustering parameters (cluster centers and scale matrices) with an EM step of a t mixture model as in Alg. 1.

despite lower predictive performance. They addressed this by introducing adaptive log-norm regularization to balance model expressiveness and feature consistency. However, the ARI scores were still not high enough to claim distinct cell types. Moreover, their work involved a rotation-equivariant convolutional core and required a post-hoc alignment procedure [38] to interpret the cluster structures.

Deep embedding clustering. Deep Embedding Clustering (DEC) [23] combines clustering with representation learning. It introduced a clustering loss that simultaneously drives learning the cluster centroids and encourages the feature representation to separate the clusters. After pretraining a deep autoencoder without the clustering loss, the cluster centers μ_j are initialized using k-means [39]. DEC then minimizes a Kullback-Leibler (KL) divergence of soft cluster assignments Q and an auxiliary target distribution P defined as follows:

$$q_{ij} = \frac{\left(1 + \frac{\|z_i - \mu_j\|^2}{\nu}\right)^{-\frac{\nu+1}{2}}}{\sum_{j'} \left(1 + \frac{\|z_i - \mu_{j'}\|^2}{\nu}\right)^{-\frac{\nu+1}{2}}} \qquad (2.1) \qquad p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j'} q_{ij'}^2 / f_{j'}} \quad \text{with } f_j = \sum_i q_{ij}. \quad (2.2)$$

The q_{ij}s are the probabilities of sample z_i belonging to cluster j and are represented by a Student's t-distribution with unit scale and degree of freedom ν being set to 1. The target distribution P (Eq. (2.2)) is chosen such that it:

- **"strengthens predictions."** Original values q_{ij} denote the soft assignment probability of a data point i belonging to cluster j. Squaring q_{ij} and then re-normalizing makes high-confidence assignments more dominant while further diminishing the influence of low-confidence ones.
- **"emphasizes high-confidence data points."** A high q_{ij} dominates q_{ij}^2/f_j, meaning that points strongly associated with a cluster contribute more to p_{ij}.
- **"normalizes loss contribution of each centroid to prevent large clusters from distorting the hidden feature space."** Without f_j, larger clusters could dominate the feature space since they would contribute disproportionately to the loss. By dividing by f_j the impact of each cluster is normalized, ensuring that smaller clusters are not overshadowed by larger ones.

Guo et al. [40] extended this approach by jointly optimizing the clustering objective and the autoencoder's reconstruction loss, enabling the model to learn clusters while preserving the local structure of the feature space.

3 DECEMber – Deep Embedding Clustering via Expectation Maximization-based refinement

DECEMber combines training a predictive model of neuronal responses with the learning of a clustered feature embeddings by optimizing a loss inspired by Deep Embedding Clustering and iteratively updating cluster parameters using the EM algorithm. We now describe our approach (illustrated in Fig. 1, described in Alg. 1).

Algorithm 1 Model Training with clustering loss

Inputs: Degrees of freedom ν, clustering weight β, core parameters θ, neuronal embeddings (readout) Z
Output: Parameters μ_j, Σ_j, θ and Z
Pretraining: Train the predictive model by optimizing L_{model} w.r.t. θ and Z for m epochs
Initialize: Cluster centers μ_j with k-means and diagonal scale matrix Σ_j as within-cluster variance
for epoch $t = 1$ to T **do**
  for minibatch b in dataset **do**
    (1) E-step (Expectation): Compute
    1.1 Soft assignments $q_{ij} = \frac{f_t(z_i;\mu_j,\Sigma_j,\nu)}{\sum_{j'=1}^{J} f_t(z_i;\mu_{j'},\Sigma_{j'},\nu)}$ (3.2)
    1.2 Latent scales $u_{ij} = \frac{\nu+K}{\nu+(z_i-\mu_j)'\Sigma_j^{-1}(z_i-\mu_j)}$ (3.3)
    (2) M-step (Maximization): Update parameters
    2.1 Update $\mu_j = \frac{\sum_{i=1}^{N} q_{ij}u_{ij}\, z_i}{\sum_{i=1}^{N} q_{ij}u_{ij}}$ (3.4)
    2.2 Update $\Sigma_j = \frac{\sum_{i=1}^{N} q_{ij}u_{ij}(z_i-\mu_j)(z_i-\mu_j)'}{\sum_{i=1}^{N} q_{ij}}$ (3.5)
    (3) Gradient step: Optimize predictive model parameters
    3.1 Minimize $L = L_{\text{model}} + \beta KL(Q||P)$ w.r.t θ, Z
    with $p_{ij} = \frac{q_{ij}^2/f_j}{\sum_k q_{ik}/f_k}$ and $f_j = \sum_i q_{ij}$
return μ, Σ, θ, Z

Predictive model for visual cortex. We build on a state-of-the-art predictive model [8] for responses r_i of neurons $i = 1, ..., N$ to visual stimuli $s \in \mathbb{R}^{H' \times W' \times T \times C}$. Here H' and W' are height and width of the input, T time if the input is a video and C is the amount of channels: $C = 1$ for grayscale or $C = 3$ for RGB. For static visual input (images), $T = 1$ and could be ignored. If behavior variables – such as pupil size, locomotion speed, and changes in pupil size – are present, they are concatenated to the stimuli as channels [8]. The model combines a shared convolutional core Φ with neuron-specific Gaussian readouts ψ_i (Fig. 1A). The core outputs a feature space $\Phi(s) \in \mathbb{R}^{H \times W \times K}$. We denote the core's parameters by θ. The readout [36] $\psi_i : \mathbb{R}^{H \times W \times K} \mapsto \mathbb{R}$ first selects the features from Φ at the neuron's receptive field location (x_i, y_i) using bilinear interpolation, which we write with a slight abuse of notation as $\Phi(\tilde{s})_{x_i y_i} \in \mathbb{R}^K$, resulting in a feature vector $\phi_i \in \mathbb{R}^K$. It then computes the predicted neuronal response $\hat{r}_i = z_i^T \phi_i$, where $z_i \in \mathbb{R}^K$ is the neuron-specific readout weight (its functional embedding), overall

$$\hat{r}_i(s) = \psi_i(\Phi(s)) = z_i^T \Phi(\tilde{s})_{x_i y_i}. \tag{3.1}$$

EM step to update cluster parameters. Instead of directly learning the cluster centroids via gradient descent, we updated them after each batch using the EM algorithm applied to the Student's t-mixture model, $f_{\text{TMM}}(z_i; \Theta) = \frac{1}{J}\sum_{j=1}^{J} f_t(z_i; \mu_j, \Sigma_j, \nu)$, [24] where degree of freedom ν controls the probability mass in the tails (if $\nu \to \infty$ the t-distribution becomes Gaussian). The density of the multivariate Student's t-distribution is:

$$f_t(z_i; \mu_j, \Sigma_j, \nu) = \frac{\Gamma\left(\frac{\nu+K}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\nu^{\frac{p}{2}}\pi^{\frac{p}{2}}|\Sigma_j|^{\frac{1}{2}}}\left(1 + \frac{1}{\nu}(z_i-\mu_j)^T\Sigma_j^{-1}(z_i-\mu_j)\right)^{-\frac{\nu+K}{2}} \tag{3.6}$$

$$= \int_0^\infty \mathcal{N}(z_i \mid \mu_j, \tfrac{1}{u}\Sigma_j) \cdot \text{Gamma}\big(u \,\big|\, \tfrac{\nu}{2}, \tfrac{\nu}{2}\big)\, du. \tag{3.7}$$

with the latter being the so-called shape-rate form of the t-distribution [41]. This interpretation is useful because introducing the Gamma-distributed latent variable u allows closed-form M-step updates for μ_j and Σ_j, whereas direct likelihood optimization in a t-mixture model does not generally admit closed-form solutions.

The full procedure is summarized in Alg. 1 and alternates between: (1) E-Step: Compute soft cluster assignments q_{ij} (Eq. (3.2)) – the probability of feature i belonging to cluster j – and the latent scaling factors u_{ij} (Eq. (3.3)). (2) M-Step: Update cluster means μ_j (Eq. (3.4)) and (diagonal) scale matrices

Σ_j (Eq. (3.5)). (3) Gradiet step: Update the parameters of the core and readout via one iteration of stochastic gradient descent.

Clustering loss on readout weights. To encourage a well-clustered structure on the neuron-specific readout weights, we augment the standard model loss with a clustering objective. Specifically, we minimize the KL divergence between soft cluster assignments q_{ij} (Eq. (3.2)) and target distributions p_{ij} (Eq. (2.2)):

$$L_{\text{cluster}} = \text{KL}(Q(Z) \,\|\, P(Z)) = \sum_{i=1}^{N} \sum_{j=1}^{J} p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right). \tag{3.8}$$

This auxiliary loss encourages the embeddings to form J distinct clusters.

Xie et al. [23] use a pretrained autoencoder with well-separated embeddings and model soft cluster assignments using a Student's t-distribution with fixed unit scale. However, this setup is too constrained for our regression setting, where the mean and scale of the embeddings z_i are restricted by the regression loss. By adopting a TMM, where each cluster is characterized by both its center μ_j and scale matrix Σ_j, we allow the clustering structure to adapt more flexibly during training.

4 Experiments

Clustering loss hyperparameters. For the clustering loss, we fixed the degrees of freedom to $\nu = 2.1$, just above the threshold where the variance $\frac{\nu}{\nu-2}\Sigma$ becomes defined (only for $\nu > 2$). To balance model flexibility and robustness, we allowed each cluster to have its own diagonal scale matrix Σ_j, which alloed for different variances per embedding dimension while preventing overfitting. For each dataset, we adjusted the clustering strength $\beta \in \mathbb{R}$ such that it is in the same order of magnitude as the model loss at initialization.

Pretraining and cluster initialization. Before adding the clustering loss, we pretrained the baseline model for m epochs, such that the model could already predict the responses reasonably well. We explored $m = 5, \ldots, 40$ to assess how the length of pretraining (PE) affected our results. We followed Turishcheva et al. [13] for the pretraining procedure, minimizing the following loss:

$$L_{\text{model}} = L_{\text{P}} + L_{\text{reg}} = \frac{1}{N} \sum_{l=1}^{L} \sum_{i=1}^{N} (\hat{r}_{il} - r_{il} \log \hat{r}_{il}) + L_{\text{reg}} \tag{4.1}$$

where L_{P} is the Poisson loss that aligned per-image $l = 1, \ldots, L$ model predictions $\hat{r}_{il}$ with observed neuronal responses r_{il} since neuron's firing rates follow a Poisson process [42], and L_{reg} is the adaptive regularizer that was shown to result in improved embedding consistency [13].

After pretraining, we initialized the cluster centroids μ_j with k-means [39] and the diagonal scale matrices Σ_j as the within-cluster variances. We continued training using $L_{model} + \beta L_{\text{cluster}}$, with $L_{cluster}$ as in Eq. (3.8) and scaled with β.

Evaluation of model performance. Building on previous research [8, 14, 16, 34, 43, 44], we evaluated the model's predictive performance by computing the Pearson correlation (across images in the test set) between the measured and predicted neural responses, averaged across neurons.

Evaluation of embedding consistency. We wanted to assess the relative structure of the embedding space: Do the same groups of neurons consistently cluster together across models fit with different initial conditions? To quantify this notion, we took DECEMber's cluster assignments and measured how often neuron pairs are assigned to the same group using the Adjusted Rand Index (ARI) [22], which quantifies the similarity between two clustering assignments, X and Y. The ARI remains unchanged under permutations of cluster labels. ARI equals one if and only if the two partitions are identical and it equals zero when the partitions agreement is no better than random.

To compare DECEMber with a baseline, we extracted neuronal embeddings from the fully converged default model and fitted Gaussian Mixture Models (GMMs) using the same number of clusters as DECEMber, diagonal covariance, and a regularization of 10^{-6}. We then computed the ARI across three GMM partitions from baseline models initialized with different seeds, using a fixed GMM seed.

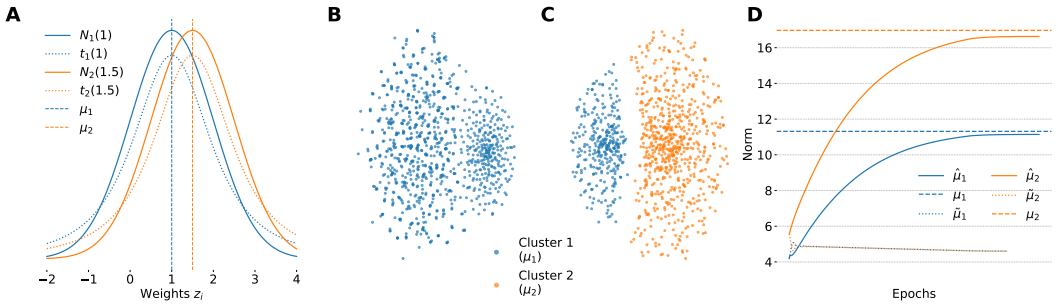

Figure 2: **A:** PDF of z_1 (blue) and z_2 (orange) of the underlying true normal distribution and t-distribution with unit scale estimated by DEC-loss. The two t-distributions as well as the normal distributions are highly overlapping. **B+C** t-SNE projection of toy data after training with DEC loss (B) vs DECEMber (C). We first pretrain a simple linear regression model by minimizing an MSE-loss for 30 epochs. Then we are training jointly with MSE and KL loss. **B:** Visualization of clustering with the DEC loss. All features are assigned to one cluster. Clustered structure is still visible. **C:** Clustering of the learned features with DECEMber. All features get assigned to the right cluster. **D:** Norms of learned cluster centers $\tilde{\mu_1}$ and $\tilde{\mu_2}$ for the DEC-loss. It is clearly visible that the cluster centers collapse after only a few iterations whereas updated cluster centers via DECEMber $\hat{\mu_1}$ and $\hat{\mu_2}$ are converging towards their true mean μ_1 and μ_2, with $\|\mu_1\|_2 = \sqrt{128} \approx 11.3$ and $\|\mu_2\|_2 = \sqrt{128 \cdot 1.5^2} \approx 17$.

Visualization. To visualize the neuronal embeddings, we employed t-SNE [45], following the guidelines of [46]. Specifically, we set the perplexity to $N/100$, the learning rate to 1 and early exaggeration to $N/10$. To be comparable with prior work [13, 20], we randomly sample 2,000 neurons from each of the seven mice in the dataset and used the same neurons across all visualizations.

5 Results

DEC-loss needs learned scale: toy example illustration. To assess whether the DEC-loss provides a useful clustering when applied to model weights that are restricted by a regression loss instead of autoencoder embeddings, we constructed a simple toy example consisting of linear neurons whose responses are given as $y_{ij} = z_i^T x_j + \epsilon_{ij}$, where z_i are the neurons' weights, x_j the stimuli and ϵ_{ij} Gaussian noise. We generated 1100 white noise stimuli, each of the form $x_j \in \mathbb{R}^{128}$ with $x_j \sim \mathcal{N}(0, 1)$. We generated 1000 neuronal embeddings z_i such that they would naturally form 2 clusters. We created 300 weights of the form $z_i \sim \mathcal{N}(I_{128}), I_{128})$ and 700 $z_i \sim \mathcal{N}(1.5I_{128}, I_{128})$. To finally get the neuronal responses we sampled Gaussian noise around 0 with a variance that matches a chosen signal to noise ratio (SNR). In the here shown example we used SNR=2. A detailed discussion about SNR can be found in the Appendix A.1.

We pretrained a linear regression model on this data for 30 epochs by minimizing the MSE of predicted and learned responses. After that we continued training, by jointly minimizing the KL divergence on the learned centers and the MSE using the DEC-loss versus DECEMber. We used early stopping as well as a learning rate scheduler. In theory the model should learn the weights of the two clusters centered at $\mu_1 = (1, \ldots, 1) \in \mathbb{R}^{128}$ and $\mu_2 = (1.5, \ldots, 1.5) \in \mathbb{R}^{128}$ and assign 300 neurons to cluster 1 and 700 neurons to cluster 2.

We found that the vanilla DEC loss fails to identify clusters even in this simple toy example, where cluster weights are well-separable after pretraining. This is because DEC employs a Student's-t distribution with a fixed unit scale parameter for all clusters, which is too large given how close the two weight distributions of clusters 1 and 2 are (Fig. 2A). As the magnitude of the weights is given by the regression problem, the scale of the t distribution needs to be adjusted appropriately during training. When this is not done (as in vanilla DEC), the cluster centroids $\tilde{\mu_1}$ and $\tilde{\mu_2}$ rapidly collapse to a single point after only a few iterations (Fig. 2D). Even though in this toy example the true covariance of the clusters has unit scale after the short pretraining phase the within cluster variance is much lower leading to a collapse of the DEC-centers. This happens because there exists a degenerate optimum of the KL divergence: If all cluster centers are equal $\mu_1 = \ldots = \mu_J = c$, plugging them into Eq. (2.1)

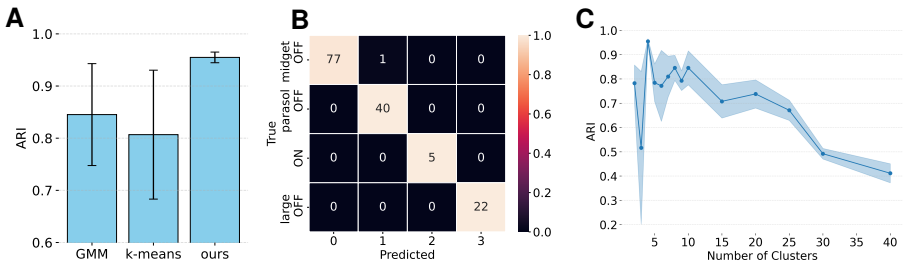

Figure 3: **A**: RI across 3 seeds for GMM, k-means and DECEMber. **B**: DECEMber predictions. Pretraining length: 25 epochs. Corresponding test correlation: 0.805 ± 0.068 (std). **C**: ARI across 3 seeds for DECEMber over different number of clusters. ARI shows a clear peak at 4 clusters.

$q_{ij} = \frac{\left(1+\|z_i-c\|^2\nu^{-1}\right)^{-(\nu+1)/2}}{\sum_{j'}(1+\|z_i-c\|^2\nu^{-1})^{-(\nu+1)/2}} = \frac{1}{J}$ gives us $p_{ij} = q_{ij}^2/(\sum_{j'} q_{ij'}^2) = (1/J^2)/(J \cdot (1/J^2)) = 1/J$ as $f_j = f_{j'}$, (Eq. (2.2)) which means the KL divergence $\mathrm{KL}(P\|Q) = 0$, which of course is not a meaningful solution. In DEC, this minimum is not usually found in practice because clusters are initialized with sufficient separation after pretraining.

To avoid this collapse, we instead used a multivariate t-distribution with (diagonal) scale matrices Σ_j for each cluster, updating both position and scale with an EM step (Alg. 1). On the same toy example, this approach succeeded in finding good cluster separation (Fig. 2C), and the cluster centers $\hat{\mu_1}$ and $\hat{\mu_2}$ converged towards the true underlying locations (Fig. 2D).

DECEMber accurately classifies retinal ganglion cells and outperforms conventional clustering approaches. To check whether DECEMber works on real data, we applied it on marmoset retinal ganglion cells (RGCs) where the existence of discrete cell types is well established [47]. We used data from two male marmoset retinas published by Sridhar et al. [48], where the neural activity was recorded using a micro-electrode array while presenting grayscale natural movies.

As we observed substantial differences between the two retinas' temporal response features (potentially due to temperature variation [49]), we followed Vystrčilová et al. [15] and trained a separate model for each retina to avoid clustering by retina. We trained the model on all reliably responding cells ($N = 235$). However, not all of them corresponded to a known primate RGC type and thus were not assigned a cell type label. When evaluating DECEMber, we only used the labeled cells. The first retina contained responses of four cell types (78 midget-OFF-like cells, 40 parasol-OFF-like cells, 5 ON-like cells, and 22 large-OFF cells, further details on classification are in App. A.2).

We trained a baseline version of a CNN model [15] separately without our proposed clustering loss, using three random seeds. Baseline clustering was then performed post hoc using GMM and k-means. Subsequently, we continued training the model with our clustering loss, again using three seeds.

Applied to marmoset retina data, DECEMber achieved reliable classification across cell types, with high clustering consistency (ARI = 0.96 $\pm$0.01) for 4 clusters while maintaining a high predictive performance of 0.81 $\pm$0.07. It surpassed both GMM and k-means, (Fig. 3A) effectively separating even highly unbalanced groups, such as the ON-cells, resulting in an almost perfect confusion matrix (Fig. 3B). In contrast to k-means, which is sensitive to initialization (Suppl. Fig. 12), DECEMber exhibited greater robustness and aligned more closely with the ground truth labels while the model retained high predictive accuracy. When we tested DECEMber for different amount of clusters we can see a clear peak in ARI at the true number of clusters 4 (Fig. 3C).

DECEMber enhances local structure among embeddings and hurts performance once it dominates the overall model loss. Next, we asked if DECEMber could help to find functional cell types in a visual area without clear known cluster separation. We used SENSORIUM 2022 dataset and baseline architecture to train a model to predict responses of mouse primary visual cortex to grayscale images. for seven mice (more detail on data in App. A.5). Previous work [13, 20] observed density modes in the functional embeddings of mouse V1 neurons (Fig. 4A) and hypothesized that these modes may correspond to discrete functional cell types. To investigate whether these patterns reflect true discrete and distinct cell types, we applied DECEMber (Alg. 1), hypothesizing that if such types exist, DECEMber would help to separate them. As the number of excitatory cell types in the mouse

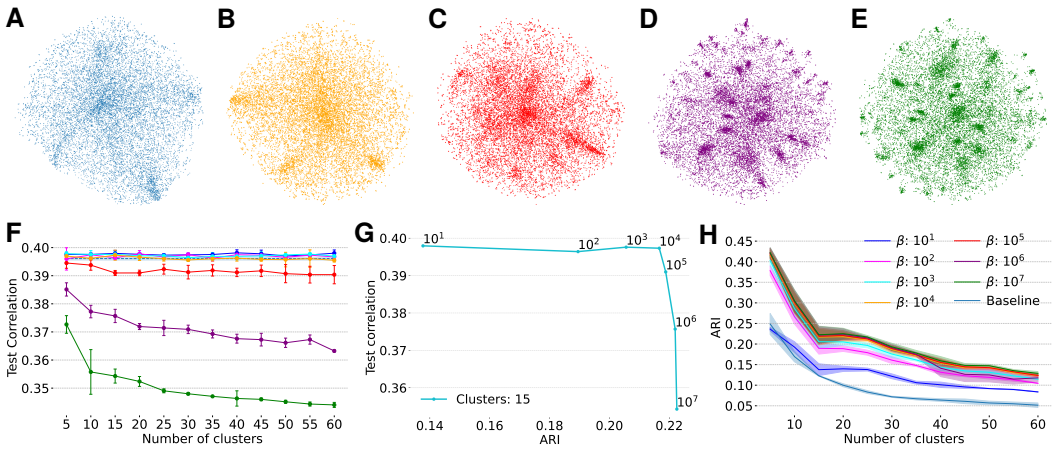

Figure 4: **A**: t-SNE of baseline model without clustering loss. **B-E**: t-SNE projections of our model with clustering bias for different multipliers β and tuned learning rates (lr). All models use 15 clusters, PE = 10 and seed 100. **B**: $\beta = 10^4$ and $lr = 0.008$. **C**: $\beta = 10^5$ and $lr = 0.008$. **D** $\beta = 10^6$ and $lr = 0.007$. **E**: $\beta = 10^7$ and $lr = 0.003$. **F**: Corresponding model performances of the models with clustering bias and differing weights β, tuned learning rates as described in B-D. **G**: Predictive performance vs. ARI for different β. We can see that ARI increases with the increase of β until the model performance drops. After that ARI doesn't increase further. Here we fixed the number of clusters to 15. **H**: ARI for different clustering weights β with optimal learning rates, PE = 10.

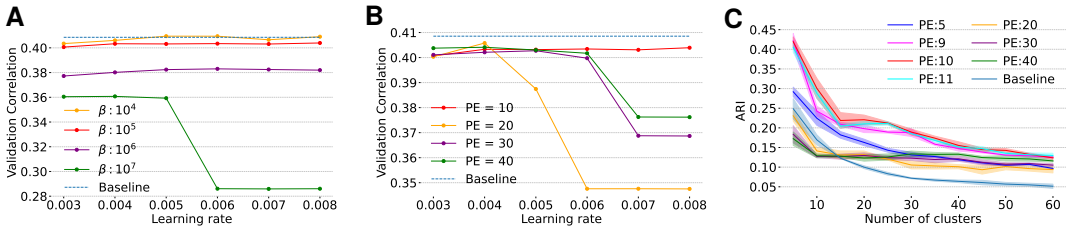

Figure 5: **A–B**: Learning rate tuning for β (A) and length of pretraining (B). We fixed amount of clusters to 15. If the learning rate is too high the clustering loss starts oscillating due to learning rate scheduling leading to a massive drop in performance. **C** ARI for different number of pretraining epochs vs baseline model. For each number of pretraining epochs we used the optimal learning rate and set $\beta = 10^5$ for all experiments. All settings of DECEMber have better cluster structures after 15 clusters at latest. It is visible that 10 pretraining epochs generate the best clustered embeddings.

visual cortex remains unclear, with estimates ranging from 20 to 50 [20, 50], we considered a range of $j = 5, ..., 60$ in increments of 5.

We tested a wide range of loss strengths β, to find the optimal value to balance the clustering a nd the model loss. As β increases, t-SNE vizualization suggests improved qualitative separation of clusters in the embedding space (Fig. 4B–E). However, this comes at a cost: when the clustering loss becomes dominant, the model's predictive performance drops significantly (Fig. 4F). This made us question if the qualitative structure in t-SNE plots is meaningful. To answer this question, we quantified clustering consistency using ARI between three model fits with different seeds and found that clustering consistency noticeably improved compared to the GMM baseline (Fig. 4H).

We see the ARI improvement as long as β does not hurt performance ($\beta \leq 10^4$; Fig. 4G. However, once $\beta > 10^4$, performance starts suffering (Fig. 4F) and the ARI does not improve anymore (Fig. 4G), suggesting that the qualitative structure is created by removing functionally relevant heterogeneity between neurons. While ARI values double compared to the baseline model, there is no clear peak around a certain number of clusters. We would expect ARI to peak noticeably at the "true" number of clusters as shown for the retina ganglion cells (Fig. 3C) if such a structure existed. This suggests that mouse V1 likely lacks discrete functional cell types. Still, the clear improvement indicates meaningful local structure in functional embeddings.

Consistency of embeddings depends on length of pretraining. To validate our conclusions that mouse V1 lacks discrete functional cell types, we performed extensive tuning of DECEMber by using different numbers of pretraining epochs before turning on the clustering loss, different clustering strengths β and tuned learning rates to optimize model predictive performance.

Across all settings, DECEMber achieved higher ARI scores than the baseline, indicating better consistency of embeddings (Fig. 4H, Fig. 5C).

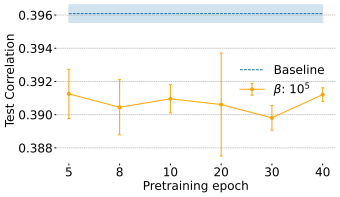

Figure 6: The choice of pretrain epoch doesn't influence performance when we're using an optimal learning rate.

We found that the optimal learning rate varied depending on the number of pretraining epochs (Fig. 5A), and also depended on the clustering loss strength β (Fig. 5B). Importantly, the choice of the number of pretraining epochs had minimal effect on the overall predictive performance if the learning rate was optimally chosen, with differences staying within the standard deviation across runs. We tuned on the validation set (Fig. 5B), and checked on the test set (Fig. 6). However, we observed a distinct peak in ARI for 10 pretraining epochs in the case of the mouse visual cortex (Fig. 5C). While the ARI improved across a range of cluster settings, we did not observe a sharp maximum at any specific cluster count.

DECEMber improves embeddings across different datasets and model architectures. To ensure that DECEMber generalizes across architectures, modalities, and species, we additionally tested it on data from the mouse retina and macaque visual cortex area V4. We did not extensively tune hyperparameters, we only decreased the learning rate (lr) to stabilize the baseline model training for both datasets, and set β as described in Sec. 4 (exact settings in App. A.15). More extensive tuning of the lr, β or the number of pretraining epochs can lead to better results. For both datasets we preserved the performance of the original models (App. A.9).

For the mouse retina we used both the data and the models from Höfling et al. [18]. As for the marmoset retina, we trained a separate model for each retina to account for the temperature differences between retinas. Given the limited availability of cell-type labels, we included all cells in our analysis and evaluated cluster consistency across varying numbers of clusters. For details on dataset averaging and per-dataset analysis, see App. A.10.

For macaque V4 data we used spiking extracellular multi-electrode recorded responses of neurons to gray-scale natural images shown to awake macaque monkeys [51] and the model from Pierzchlewicz et al. [37], which had a different readout architecture – an attention readout instead of the previously used Gaussian readout. We trained the model on 1000 cells and measured the ARI across three model seeds. The embedding consistency doubled using our method (Fig. 7B). This shows that DECEMber is robust not only across different data modalities but also across architectures. For more analysis of the monkey dataset see App. A.11.

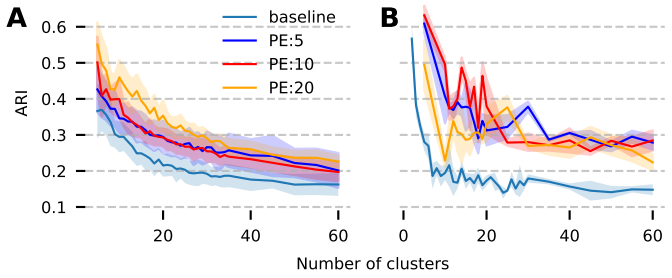

Figure 7: ARI on **A** mouse retina [18], weighted across six models. **B** monkeys V4 [51].

6 Discussion

In this work we introduced DECEMber, an additional training loss with explicit clustering bias for predictive models of neuronal responses. DECEMber enhances cluster consistency, while keeping state-of-the-art predictive performance. It is robust across different data modalities (electrophysiology and calcium imaging), species (mice, primates) and visual areas (retina, V1, V4). We also showed that DECEMber is robust across both static (mouse V1, macaque V4) and dynamic (retinas) cores and multiple readout architectures – the Gaussian readout and the attention readout.

We see DECEMber as a model-driven hypothesis test: if clear functional cell types exist, then incorporating this bias should improve model performance and/or the embedding structure, which we measure as cluster consistency. While improvements are observed across datasets and architectures, our main focus was mouse V1, where the existence of discrete excitatory cell types remains debated. Our results support the idea that excitatory neurons in mouse visual cortex form a functional continuum rather than discrete clusters. This finding is consistent with recent work studying different modalities by Weiler et al. [52], Tong et al. [19], and Weis et al. [6], who independently found no clear boundaries in morphological or electrophysiological features. In line with Zeng [1], we argue that future efforts to define mouse V1 cell types should emphasize multi-modality combining functional, morphological, and genetic data. This approach has proven fruitful in the retina, where functional types alone are coarser than those derived from multiple modalities [7, 21].

Given the generality of our clustering loss, which is model-agnostic and not tied to a specific architecture, we believe it holds promise for use in multi-modal models aiming to define cell types or in broader unsupervised representation learning contexts.

Connection to other works learning neuronal embeddings. There are other works [53–59], which all learn neuronal embeddings in some way, but none of them explicitly enforce or optimize for clustering, which is the main goal of DECEMber. Specifically, Nemo [53], NeurPIR [54] and NuCLR [55] are contrastive methods, DECEMber is not. Nemo [53] and NeurPIR [54] embeddings are functions of input (current activity, autocorrelogram), for DECEMber neuronal embeddings are time- and input- invariant weights of the predictive model (they embed the neuron's full input–output function). Nemo [53], NeuPRINT [56], NetFormer [57] do not model visual stimuli. NeuPRINT, NetFormer, POYO [58], NEDs [59] have time-invariant model weights, but both predict neuronal activity based on masked or previous neuronal activity while the regression model in our paper predicts neuronal activity based on visual stimuli. It might be interesting to integrate DECEMber clustering loss with these works [56–59] but we leave it for future work.

Limitations. DECEMber requires a predefined number of clusters. When this is unknown, multiple runs with varying cluster counts and seeds are necessary in combination with an evaluation ARI-like metric to identify the optimal configuration. This increases the computational cost. Choosing an appropriate clustering strength β is also crucial for balancing ARI and model performance and further work is needed to determine the optimal pretraining duration.

Moreover, operating in high-dimensional feature spaces introduces an additional challenge: the cluster covariance matrices can become large and ill-conditioned, with tiny diagonal values, hitting the limits of numerical stability. We address this issue by clamping small values, though this solution is heuristic rather than principled. Furthermore, high-dimensional settings require a sufficient number of data points to prevent overfitting of the scale matrices. Additionally, we currently assume a t-distributed feature space via a t-mixture model, but this can be adjusted if a more suitable prior over the embeddings is known.

7 Acknowledgments

We thank Suhas Shrinivasan, Max F. Burg, Larissa Höfling, Thomas Zenkel, Konstantin F Willeke, Fabian Sinz.

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project IDs 432680300 (SFB 1456, project B05) and 515774656 – and by European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (Grant agreement No. 101041669). AST acknowledges support from National Institute of Mental Health and National Institute of Neurological Disorders And Stroke under Award Number U19MH114830 and National Eye Institute award numbers R01 EY026927 and Core Grant for Vision Research T32-EY-002520-37. We gratefully acknowledge the computing time granted by the Resource Allocation Board and provided on the supercomputer Emmy/Grete at NHR-Nord@Göttingen as part of the NHR infrastructure. The calculations for this research were conducted with computing resources under the projects nim00010 and nim00012.

References

[1] Hongkui Zeng. "What is a cell type and how to define it?" In: *Cell* 185.15 (2022), pp. 2739–2755.

[2] Javier DeFelipe et al. "New insights into the classification and nomenclature of cortical GABAergic interneurons". In: *Nature Reviews Neuroscience* 14.3 (2013), pp. 202–216.

[3] Marcel Oberlaender et al. "Cell type–specific three-dimensional structure of thalamocortical circuits in a column of rat vibrissal cortex". In: *Cerebral cortex* 22.10 (2012), pp. 2375–2391.

[4] Henry Markram et al. "Reconstruction and simulation of neocortical microcircuitry". In: *Cell* 163.2 (2015), pp. 456–492.

[5] Federico Scala et al. "Layer 4 of mouse neocortex differs in cell types and circuit organization between sensory areas". In: *Nature communications* 10.1 (2019), p. 4174.

[6] Marissa A. Weis et al. "An Unsupervised Map of Excitatory Neuron Dendritic Morphology in the Mouse Visual Cortex". In: *Nature Communications* 16.1 (Apr. 2025), p. 3361. ISSN: 2041-1723. DOI: 10.1038/s41467-025-58763-w. (Visited on 04/10/2025).

[7] Tom Baden et al. "The functional diversity of retinal ganglion cells in the mouse". In: *Nature* 529.7586 (2016), pp. 345–350.

[8] Konstantin F. Willeke et al. *The Sensorium competition on predicting large-scale mouse primary visual cortex activity*. 2022. arXiv: 2206.08666 [q-bio.NC]. URL: https://arxiv.org/abs/2206.08666.

[9] Polina Turishcheva et al. "The dynamic sensorium competition for predicting large-scale mouse visual cortex activity from videos". In: *ArXiv* (2024), arXiv–2305.

[10] Ján Antolík et al. "Model Constrained by Visual Hierarchy Improves Prediction of Neural Responses to Natural Scenes". In: *PLOS Computational Biology* 12.6 (June 2016), pp. 1–22. DOI: 10.1371/journal.pcbi.1004927. URL: https://doi.org/10.1371/journal.pcbi.1004927.

[11] Finn Schmidt et al. *Modeling dynamic neural activity by combining naturalistic video stimuli and stimulus-independent latent factors*. 2024. URL: https://neurips.cc/virtual/2024/101471.

[12] Eric Y. Wang et al. "Towards a Foundation Model of the Mouse Visual Cortex". In: *bioRxiv* (2023). DOI: 10.1101/2023.03.21.533548. eprint: https://www.biorxiv.org/content/early/2023/03/24/2023.03.21.533548.full.pdf. URL: https://www.biorxiv.org/content/early/2023/03/24/2023.03.21.533548.

[13] Polina Turishcheva et al. *Reproducibility of predictive networks for mouse visual cortex*. 2024. arXiv: 2406.12625 [q-bio.NC]. URL: https://arxiv.org/abs/2406.12625.

[14] Alexander Ecker et al. "A rotation-equivariant convolutional neural network model of primary visual cortex". In: (2019). URL: https://iclr.cc/virtual/2019/poster/922.

[15] Michaela Vystrčilová et al. "Convolutional neural network models of the primate retina reveal adaptation to natural stimulus statistics". In: *bioRxiv* (2024). DOI: 10.1101/2024.03.06.583740. eprint: https://www.biorxiv.org/content/early/2024/03/09/2024.03.06.583740.full.pdf. URL: https://www.biorxiv.org/content/early/2024/03/09/2024.03.06.583740.

[16] Edgar Y Walker et al. "Inception loops discover what excites neurons most using deep predictive models". In: *Nature neuroscience* 22.12 (2019), pp. 2060–2065.

[17] Katrin Franke et al. "State-dependent pupil dilation rapidly shifts visual feature selectivity". In: *Nature* 610.7930 (2022), pp. 128–134.

[18] Larissa Höfling et al. "A chromatic feature detector in the retina signals visual context changes". In: *Elife* 13 (2024), e86860.

[19] Rudi Tong et al. "The feature landscape of visual cortex". In: *bioRxiv* (2023). DOI: 10.1101/2023.11.03.565500. eprint: https://www.biorxiv.org/content/early/2023/11/05/2023.11.03.565500.full.pdf. URL: https://www.biorxiv.org/content/early/2023/11/05/2023.11.03.565500.

[20] Ivan Ustyuzhaninov et al. "Digital twin reveals combinatorial code of non-linear computations in the mouse primary visual cortex". In: *bioRxiv* (2022), pp. 2022–02.

[21] Max F. Burg et al. *Most discriminative stimuli for functional cell type clustering*. 2024. URL: https://iclr.cc/virtual/2024/poster/19293.

[22] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2 (1985), pp. 193–218.

[23] Junyuan Xie, Ross Girshick, and Ali Farhadi. "Unsupervised Deep Embedding for Clustering Analysis". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 478–487. URL: https://proceedings.mlr.press/v48/xieb16.html.

[24] Geoffrey J. McLachlan and David Peel. "Robust mixture modelling using the t distribution". In: *Statistical Science* 15.1 (2000), pp. 1–19. URL: https://people.smp.uq.edu.au/GeoffMcLachlan/pm_sc00.pdf.

[25] Daniel LK Yamins et al. "Performance-optimized hierarchical models predict neural responses in higher visual cortex". In: *Proceedings of the national academy of sciences* 111.23 (2014), pp. 8619–8624.

[26] Charles F Cadieu et al. "Deep neural networks rival the representation of primate IT cortex for core visual object recognition". In: *PLoS computational biology* 10.12 (2014), e1003963.

[27] Santiago A Cadena et al. "Deep convolutional models improve predictions of macaque V1 responses to natural images". In: *PLoS computational biology* 15.4 (2019), e1006897.

[28] Galen Pogoncheff, Jacob Granley, and Michael Beyeler. "Explaining V1 properties with a biologically constrained deep learning architecture". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 13908–13930.

[29] Eleanor Batty et al. "Multilayer Recurrent Network Models of Primate Retinal Ganglion Cell Responses". In: *International Conference on Learning Representations*. 2016. URL: https://api.semanticscholar.org/CorpusID:39002941.

[30] Lane T. McIntosh et al. "Deep Learning Models of the Retinal Response to Natural Scenes". In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016, pp. 1369–1377.

[31] Fabian Sinz et al. "Stimulus domain transfer in recurrent models for large scale cortical population prediction on video". In: *Advances in neural information processing systems* 31 (2018).

[32] Mohammad Bashiri et al. "A flow-based latent state generative model of neural population responses to natural images". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 15801–15815. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/84a529a92de322be42dd3365afd54f91-Paper.pdf.

[33] Andrew Y. Tan et al. "Orientation Selectivity of Synaptic Input to Neurons in Mouse and Cat Primary Visual Cortex". In: *Journal of Neuroscience* 31.34 (2011). Erratum in: J Neurosci. 2011 Oct 12;31(41):14832, pp. 12339–12350. DOI: 10.1523/JNEUROSCI.2039-11.2011.

[34] Bryan M Li et al. "V1t: large-scale mouse v1 response prediction using a vision transformer". In: *arXiv preprint arXiv:2302.03023* (2023).

[35] David Klindt et al. "Neural system identification for large populations separating "what" and "where"". In: *Advances in neural information processing systems* 30 (2017).

[36] Konstantin-Klemens Lurz et al. "Generalization in data-driven models of primary visual cortex". In: *bioRxiv* (2020). DOI: 10.1101/2020.10.05.326256. eprint: https://www.biorxiv.org/content/early/2020/10/07/2020.10.05.326256.full.pdf. URL: https://www.biorxiv.org/content/early/2020/10/07/2020.10.05.326256.

[37] Pawel Pierzchlewicz et al. "Energy guided diffusion for generating neurally exciting images". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 32574–32601.

[38] Ivan Ustyuzhaninov et al. "Rotation-invariant clustering of neuronal responses in primary visual cortex". In: *International Conference on Learning Representations*. 2019.

[39] James MacQueen. "Some Methods for Classification and Analysis of Multivariate Observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. 1967, pp. 281–297.

[40] Xifeng Guo et al. "Improved deep embedded clustering with local structure preservation". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. IJCAI'17. Melbourne, Australia: AAAI Press, 2017, pp. 1753–1759. ISBN: 9780999241103.

[41] Geoffrey J McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2000.

[42] Peter Dayan and Laurence F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, 2005.

[43] Brett Vintch, J Anthony Movshon, and Eero P Simoncelli. "A convolutional subunit model for neuronal responses in macaque V1". In: *Journal of Neuroscience* 35.44 (2015), pp. 14829–14841.

[44] Max F Burg et al. "Learning divisive normalization in primary visual cortex". In: *PLoS computational biology* 17.6 (2021), e1009028.

[45] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[46] George C. Linderman and Stefan Steinerberger. *Clustering with t-SNE, provably*. 2019. DOI: 10.1137/18M1216134. URL: https://doi.org/10.1137/18M1216134.

[47] Rania A Masri et al. "Survey of retinal ganglion cell morphology in marmoset". In: *Journal of Comparative Neurology* 527.1 (2019), pp. 236–258.

[48] Shashwat Sridhar and Tim Gollisch. *Dataset - Marmoset retinal ganglion cell responses to naturalistic movies and spatiotemporal white noise*. en. Apr. 2025. DOI: 10.12751/G-NODE.3DFITI. URL: https://doi.gin.g-node.org/10.12751/g-node.3dfiti (visited on 05/16/2025).

[49] Zhijian Zhao et al. "The temporal structure of the inner retina at a single glance". In: *Scientific reports* 10.1 (2020), p. 4399.

[50] Nathan W Gouwens et al. "Classification of electrophysiological and morphological neuron types in the mouse visual cortex". In: *Nature neuroscience* 22.7 (2019), pp. 1182–1195.

[51] KF Willeke et al. "Deep learning-driven characterization of single cell tuning in primate visual area V4 unveils topological organization. bioRxiv. 2023". In: *doi. org/10.1101/2023.05* 12 (2023).

[52] Sebastian Weiler et al. "Functional and structural features of L2/3 pyramidal cells continuously covary with pial depth in mouse visual cortex". In: *Cerebral Cortex* 33.7 (2023), pp. 3715–3733. DOI: 10.1093/cercor/bhac303.

[53] Han Yu et al. "In vivo cell-type and brain region classification via multimodal contrastive learning". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=10JOlFIPjt.

[54] Wei Wu et al. "Neuron Platonic Intrinsic Representation From Dynamics Using Contrastive Learning". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=vFanHFE4Qv.

[55] Vinam Arora et al. "Know Thyself by Knowing Others: Learning Neuron Identity from Population Context". In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2025. URL: https://openreview.net/forum?id=zt3RKc6VBp.

[56] Lu Mi et al. "Learning Time-Invariant Representations for Individual Neurons from Population Dynamics". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=EcN3l6Xmnx.

[57] Ziyu Lu et al. "NetFormer: An interpretable model for recovering dynamical connectivity in neuronal population dynamics". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=bcTjW5kS4W.

[58] Mehdi Azabou et al. "A Unified, Scalable Framework for Neural Population Decoding". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=sw2Y0sirtM.

[59] Yizi Zhang et al. "Neural Encoding and Decoding at Scale". In: *Forty-second International Conference on Machine Learning*. 2025. URL: https://openreview.net/forum?id=vOdz3zhSCj.

[60] Shashwat Sridhar et al. "Modeling spatial contrast sensitivity in responses of primate retinal ganglion cells to natural movies". In: *bioRxiv* (2025). DOI: 10.1101/2024.03.05.583449. eprint: https://www.biorxiv.org/content/early/2025/04/09/2024.03.05.583449.full.pdf. URL: https://www.biorxiv.org/content/early/2025/04/09/2024.03.05.583449.

[61] Edward B Fowlkes and Colin L Mallows. "A method for comparing two hierarchical clusterings". In: *Journal of the American statistical association* 78.383 (1983), pp. 553–569.

[62] Andrew Rosenberg and Julia Hirschberg. “V-measure: A conditional entropy-based external cluster evaluation measure”. In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*. 2007, pp. 410–420.

[63] Alexander Strehl and Joydeep Ghosh. “Cluster ensembles—a knowledge reuse framework for combining multiple partitions”. In: *Journal of machine learning research* 3.Dec (2002), pp. 583–617.

A Appendix

A.1 Signal to noise ratio for toy example

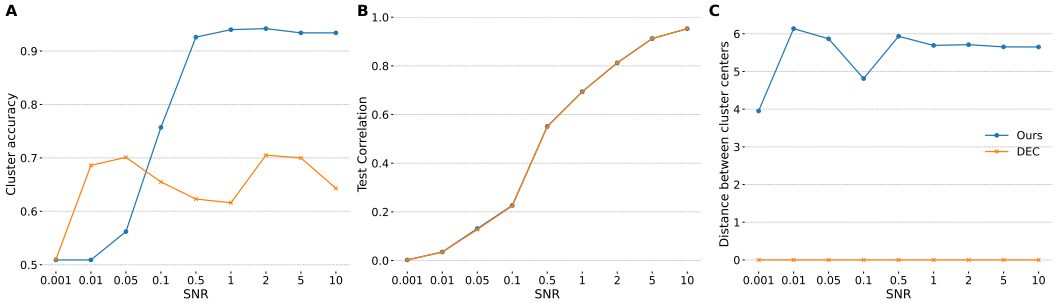

Figure 8: **A**: Clustering accuracy for predicted vs. ground-truth label for DEC vs. DECEMber for different Signal-to-Noise ratios. **B**: Predictive performance on test set. **C**: The distance of the norms of the 2 cluster centroids $\|\mu_1\| - \|\mu_2\|$ for DEC vs. DECEMber. The ground truth distance is $\sqrt{128 \cdot 1.5^2} - \sqrt{128} \approx 5.66$.

Data in mouse V1 is recorded using 2-photon calcium imaging which is known to be noisy. To investigate how robust DECEMber is towords noise we varied the SNR in the toy example setting. As described in the toy example we generated clean responses as the dot product of the stimuli and the created network weights. To simulate noisy observations, we added Gaussian noise independently for each neuron and stimulus but with the same variance. The noisy responses are thus given by $y_{ij} = z_i^T x_j + \epsilon_{ij}$, where z_i are the neurons' weights, x_j the stimuli and ϵ_{ij} Gaussian noise. Since both the input and the noise are mean-centered, the Signal-to-Noise Ratio of the neuronal population of N neurons and M stimuli is defined as $SNR = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\mathrm{Var}\left[z_i^T x_j\right]}{\mathrm{Var}\left[\epsilon_{ij}\right]}$.

In this setup, we vary the SNR by adjusting the noise variance, thereby controlling the noise power in the simulation. We varied the SNR between 0.001 and 10 and trained in the same way as before. We calculated the Pearson correlation on a left out test set. Both DEC and DECEMber converge to similar performance since the MSE drives the learning of the model's weights (Fig. Fig. 8B). We see that even for low SNR DECEMber successfully separates the clusters (Fig. Fig. 8A), though the distance is not ideal before SNR is above 0.1. However, for DEC cluster collapse happens independently of SNR (Fig. Fig. 8C). Since the weights (e.g. neurons) are generated based on predefined clusters (means of the Gaussians), we know the ground truth label for each neuron. To evaluate the cluster accuracy of DEC and DECEMber we measure the proportion of neurons that are assigned to the correct cluster.

A.2 Retina gagnlion cells

To select reliable cells from the marmoset RGCs dataset [48], we used the same reliability assessment of each cell's responses to visual stimuli as in [60]; only reliable cells were used for model training. The model architecture was also taken from [15]. For clustering evaluations using DECEMber, k-means, and GMM, we considered only cells for which cell-type labels were available.

The dataset contains recordings from two different retinas of male marmosets. The second retina (not analyzed in the main part of this paper) includes 38 parasol-OFF and 35 parasol-ON cells which are well separable (Fig. 9B). We trained our models on all reliable cells from

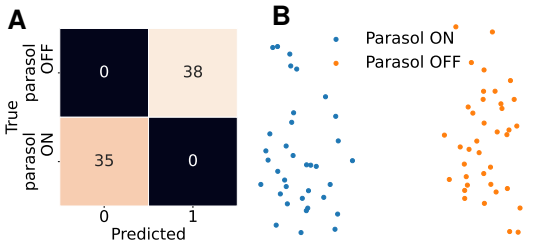

Figure 9: **A**: DECEMber predictions. Pretraining length: 20 epochs. Same predictions for GMM and k-means. All methods have ARI 1. **B**: t-SNE projections of the corresponding cells.

this retina as well and tested DECEMber with
varying pretraining lengths (which did not affect cluster consistency). All three clustering methods—GMM, k-means, and DECEMber—successfully and robustly identified the two cell types, as visualized in Fig. 9A with ARI=1.

Cell type labels We used the same cell-type classification procedure as in [60] (Methods section 4.5), clustering the cells using the KMeans++ algorithm on features extracted from receptive-field estimates obtained using spike-triggered averaging from responses to spatiotemporal white-noise, and from autocorrelograms computed on responses to white-noise and naturalistic movies. The cell-type labels in our analysis differ from the ones used in the original publication because we did not exclude cells that violated the tiling of spatial receptive fields.

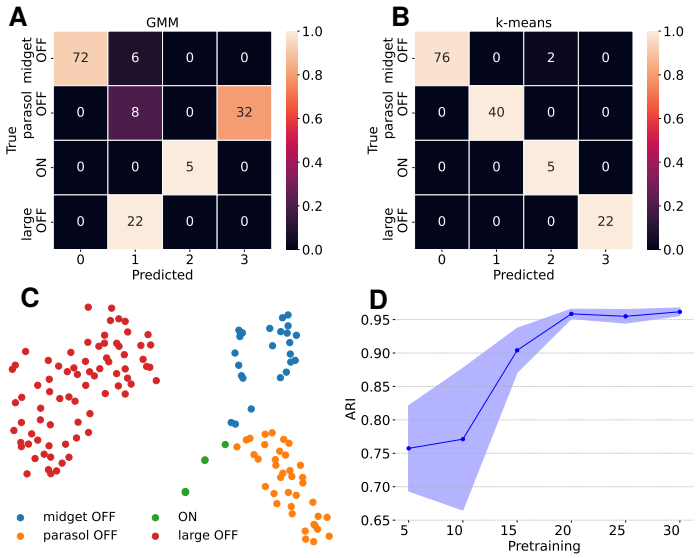

Figure 10: All plots show evaluations of seed 4 of the trained marmoset RGC model [15] for retina 1 used in the main part of this paper. **A**: GMM predictions. **B**: k-means. **C**: t-SNE projections of the corresponding cells. **D**: ARI for different length of pretraining. Longer pretraining seems to be beneficial with ARI stabilizing after pretraining of 20 epochs.

A.3 ARI-stability for k-means and GMM on marmoset RGC

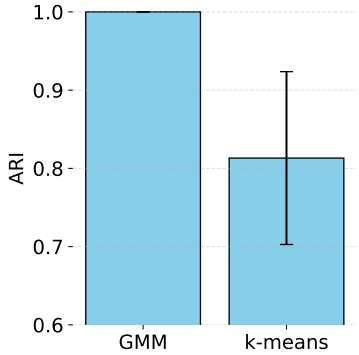

Figure 11: **Clustering stability of k-means and GMM on retina 1 for marmoset RGC.** We started with a single baseline RGC model of retina 1 (seed 2) and performed k-means and GMM clustering (4 clusters each), varying the random seed (42, 10, 100) for both algorithms. Clustering was done on all cells the model was trained on, but ARI was calculated using only labeled cells.

A.4 ARI stability for GMM on mouse V1

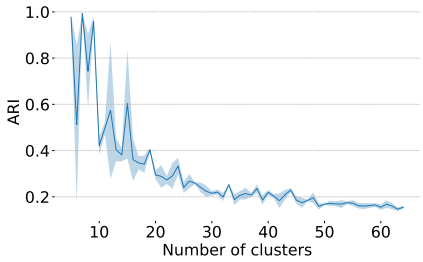

Figure 12: **Clustering stability of GMM on mouse V1.** We trained a baseline model of mouse V1 for one seed (42). We then did GMM for clusters ranging from 5 to 60 as the ground truth is not known varying just the seed for the initialization of the GMM. It's clearly visible that GMM becomes unstable if the amount of clusters is large.

A.5 Sensorium data details.

The model was trained on the SENSORIUM 2022 dataset [8], which contains neural responses to natural images recorded from seven mice (a total of 54,569 neurons). Recordings were made from excitatory neurons in layer 2 and 3 of the primary visual cortex using two-photon calcium imaging. In addition to neural activity, the dataset includes five behavioral variables: locomotion speed, pupil size, the instantaneous change in pupil size (estimated via second-order central differences), and horizontal and vertical eye position, all of which are incorporated into the model. Three of them – locomotion speed, pupil size, and the instantaneous change in pupil size – were appended to the grayscale images and are used as input to the core, while pupil horizontal and vertical position were used as input to the shifter – a model part shifting the readout receptive field locations depending on where the mouse is looking. The validation set contains responses to roughly 500 and test set to 5000 images per mouse.

A.6 Additional clustering metrics show qualitatively consistent results with ARI

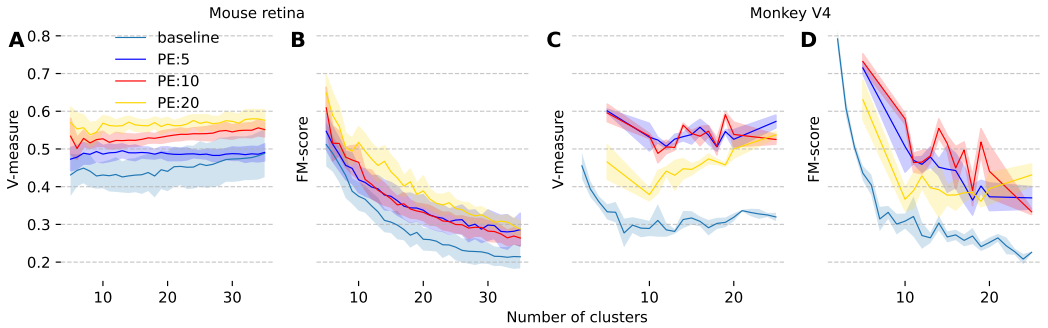

Figure 13: Different clustering consistency metrics for monkey V4 and mouse retina datasets. Same as in the main paper, mouse retina is weighted across six models using all neurons, monkey V4 model is trained on a subset of 1000 neurons. The order of lines is same as for ARI, confirming its results qualitatively. V-measure is biased towards bigger amount of clusters due to the set-based nature.

Clustering quality can be evaluated using metrics beyond ARI. ARI measures the similarity between two clusterings by checking whether pairs of points are assigned to the same cluster in both. The Fowlkes-Mallows index [61] (Fig. 14) also compares two partitions but does not adjust for chance; it is the geometric mean of precision and recall, based on how consistently point pairs are clustered together.

Other common metrics – homogeneity, completeness, and V-measure [62] – are asymmetric and compare one clustering against a reference (typically ground truth). Homogeneity measures whether each cluster contains only members of a single class, while completeness checks whether all members of a given class are assigned to the same cluster. Swapping the roles of predicted and true labels interchanges homogeneity and completeness. V-measure, equivalent to normalized mutual information (NMI [63]) and it is the harmonic mean of the two. As in our case we do not have ground truth, we

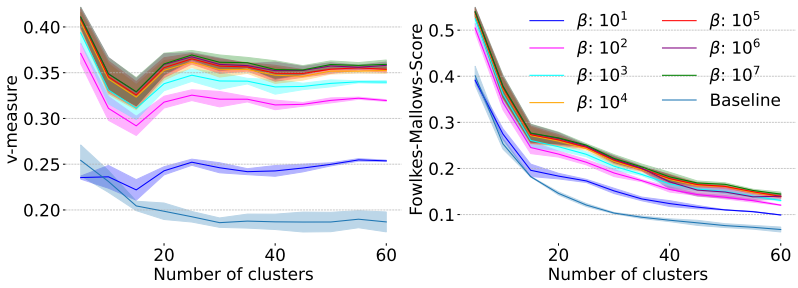

Figure 14: V-measure and Fowlkes-Mallows-score for PE 10, 15 clusters.

compute the metrics with all possible seed pairs, which leads to homogeneity, completeness, and V-measure being equivalent (Fig. 14).

A.7 Influence of Degree of freedom as a hyperparameter

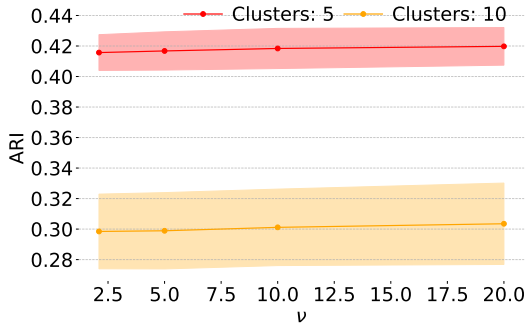

Figure 15: DECEMber, with PE = 10, lr=0.008, $\beta = 10^4$ with different degrees of freedom ν.

We've tested the influence of the degree of freedom on ARI for both 5 and 10 clusters 15 and can't really see a difference.

A.8 Comparison with the rotation equivariant baseline

Turishcheva et al. [13] is the only work to date that specifically addresses neuronal embedding consistency, and thus serves as our baseline for comparison. We use the $\gamma_{\text{lognorm}} = 10$ condition from their paper and compare it to our consistency results in Fig. 16. Our approach achieves comparable consistency levels while eliminating the need for a rotation-equivariant core, thereby removing the post-hoc alignment step and improving predictive performance from $\approx 38.1\%$ (Fig. 3 A in [13]) to $\approx 39.5\%$ (Fig. 4F).

A.9 Models performances on mouse retina and macaque V4 data

	Baseline	PE 5	PE 10	PE 20
Mouse retina	0.4727 ± 0.0008	0.4695 ± 0.0009	0.4732 ± 0.0008	0.4727 ± 0.0009
Macaque V4	0.308 ± 0.004	0.308 ± 0.006	0.304 ± 0.005	0.305 ± 0.003

Table 1: Performances on mouse retina and macaque V4 data for the models reported in the main paper (Sec. 5). Mouse retina is weighted as described in App. A.10.All performances are on the held-out test set. The values are averaged across all cluster counts. Seeds were 42, 101 and 7607. For GMM baseline the seed was 42.

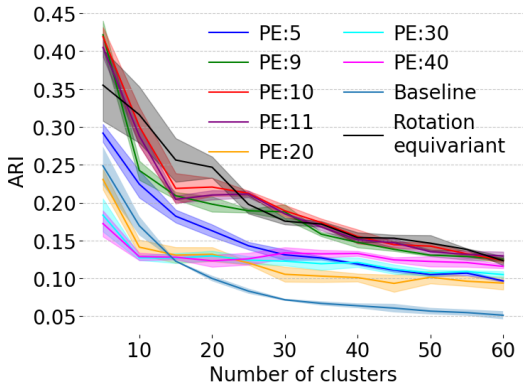

Figure 16: DECEMber, with PE = 10; DECEMber cluster consitency matches the rotation-equivariant model from Turishcheva et al. [13].

A.10 Further analysis of mouse retina data

Averaging across datasets For the mean ARI across datasets we weighted ARI lines like $\mu_{\text{total}} = \sum_i w_i \mu_i$, where $w_i = n_{\text{cur}}/n_{\text{total}}$ with n_{cur} - the number of neurons in the current model, n_{total} is the number of neurons in all six models, and μ_i is the average ARI score across three seeds for the current model. We used the law of total variance and computed the variance as $\sigma^2_{\text{total}} = \sum_i w_i \left[\sigma_i^2 + (\mu_i - \mu_{\text{total}})^2\right]$, where the first term captures within-dataset ARI variability and the second term captures between-dataset ARI variability. Fig. 17 shows the ARIs per models. We can see that the fewer neurons were present in the models the less the improvement was.

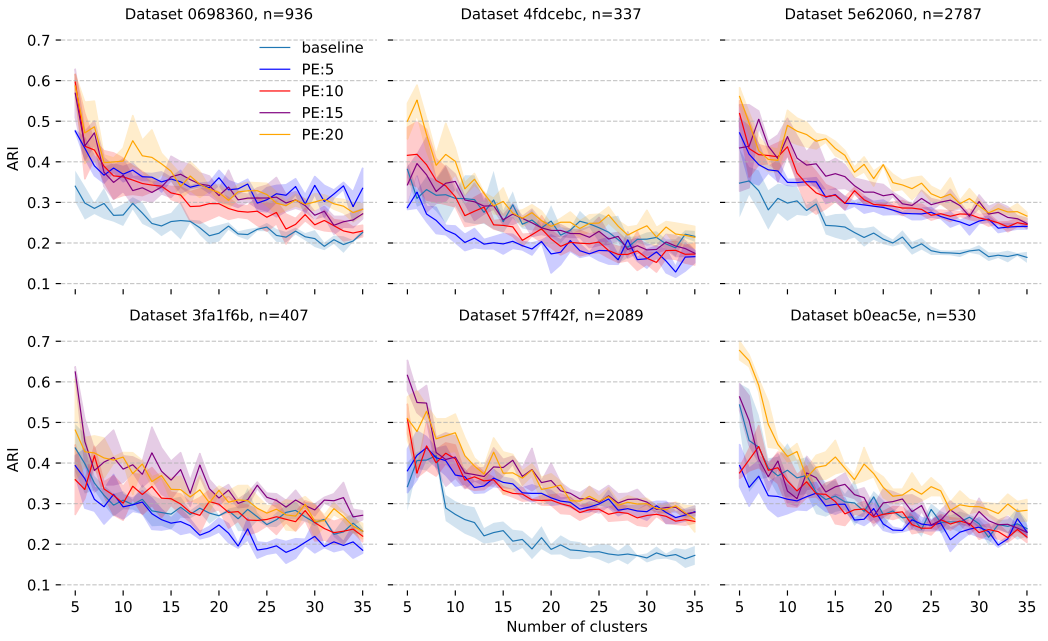

Figure 17: ARI per retina. n is the number of neurons in the model

A.11 Further analysis of monkeys data

For monkey V4, we trained models for 5, 10 to 20 and 25 clusters, as original work reported 12 clusters for 1000 cells. For 144 cells there were no labels and 100 cells ahd a "not properly clustered" label. Therefore, we decided to use only the 1000 labeled cells. For results of models trained on all

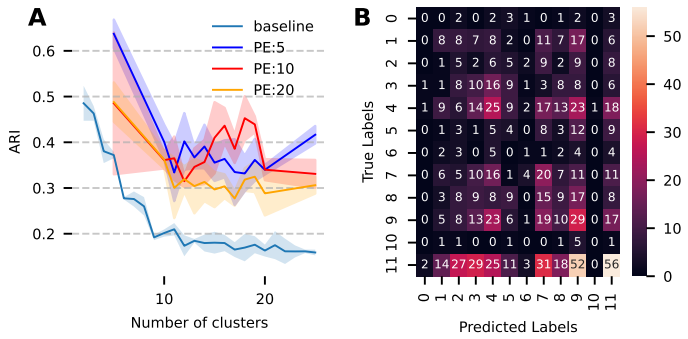

Figure 18: **A** ARI between models seeds for models trained on all 1244 neurons.
B Confusion matrix between predictions of the model trained on 1000 neurons and labels suggested in Willeke et al. [51]

cells see Fig. 18A. While the trends and values are qualitatively similar to the model trained only on a 1000 neurons subset, the standard deviation corridor seems to be wider, likely due to some of the "not properly clustered" neurons being in between the distinct groups. Please note that the labels from Willeke et al. [51] are rather a suggestion but not ground truth as they were not verified using independent biological measurements. For the confusion matrix of our labels and labels from Willeke et al. [51] see Fig. 18A. Same as for Burg et al. [21], our labels do not perfectly match the ones proposed in Willeke et al. [51].

A.12 Bayesian inference criterion BIC

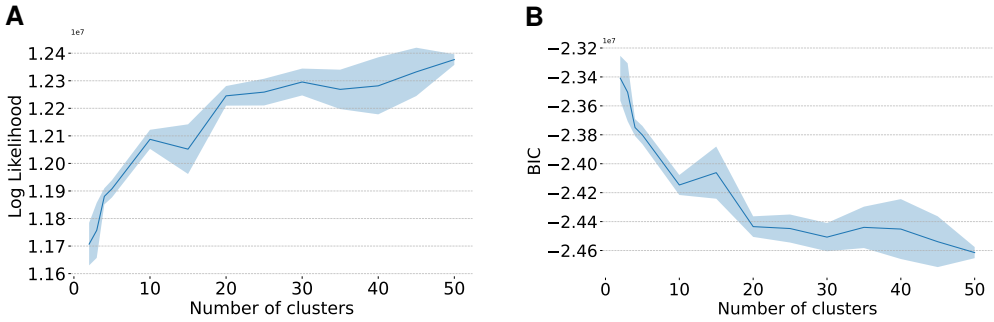

Figure 19: **A**: Log likelihood estimation. **B**: Calculation of BIC.

As suggested by a reviewer, BIC could be an alternative way to find the underlying number of clusters. We used a model with 10 pretraining epochs and $\beta = 10^4$.

After training the model, we calculated the log-likelihood of the neurons

$$\log \mathcal{L} = \sum_{i=1}^{N} \log \frac{1}{K} \sum_{k=1}^{K} t_\nu(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where $t_\nu(\cdot \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the multivariate t-distribution with mean $\boldsymbol{\mu}_k$, covariance $\boldsymbol{\Sigma}_k$, and degrees of freedom ν.

We computed the BIC as $\text{BIC} = -2 \log \mathcal{L} + (2 * K * d + 1) \log(N)$,

where N is the number of neurons, d is the dimensionality of the embeddings, K is the number of clusters.

The likelihood grows with the number of clusters whereas BIC falls with the number of clusters (Fig Fig. 19) indicating the lack of a clear cluster peak, which agrees with ARI (Fig Fig. 4H). BIC also scales with the number of model parameters which in our case increase a lot when the number of clusters increases making it not the most suitable measure in our case.

A.13 Compute requirements

All of our models can be considered light-weight in terms of compute by modern deep learning model standards. A single mouse retina model requires less the 10Gb of GPU memory and trains under 20 minutes of walltime. A single mouse V1 model requires $\approx$ 12Gb of GPU memory and trains for under 2 hours of walltime. A single marmoset retina model uses 40Gb GPU and trains for under 16 hours of walltime. A single monkey model requires 24Gb of memory and trains for under 2 hours of walltime.

We use a local infrastructure cluster with 8 NVIDIA RTX A5000 GPUs with 24Gb of memory each for mouse experiments. For mouse retina, marmoset retina, and monkey V4 we used 40Gb NVIDIA A100.

A.14 Broader impact

Our work contributes to building more reproducible models, which are more suitable for making biologically meaningful statements. It is even more related to derive a functional taxonomy of cell types in the primary visual cortex, which can enhance our understanding of brain function and support the development of treatments for neurodegenerative diseases.

A.15 Experimental settings

For marmoset RGC dataset, we used the three layer CNN described in [15]. We trained it for a maximum of 1000 epochs, stopping early if validation correlation did not improve for 20 epochs. The learning rate of both pretraining and training with the clustering loss was initially 0.005 and reduced during training using the `ReduceLROnPlateau` learning rate scheduler, patience 10 and minimal learning rate $1e^{-8}$.
For SENSORIUM 2022, we used their model and training hyperparameters for the baselines training. Pretraining duration, learning rates and clustering strength β is reported in every experiment. For mouse retina, we followed Hofling et al. [18] model and training hyperparameters, changing only learning rate from 0.01 to 0.005 to improve baselines stability. Clustering strength was set to 0.001 across all experiments. For monkey V4 data we followed model and training hyperparameters from [37], again only changing the learning rate from $3 \cdot 10^{-4}$ to $5 \cdot 10^{-5}$ to improve baselines stability. Clustering strength was set to 0.001 across all experiments. Changing learning rate in both cases did not impacted performance in more than std boundaries.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The introduction clearly state the contributions as an bullet point list at the end. Limitations as discussed in Sec. 6 as a separate paragraph.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes, we have a **limitations** paragraph in the Discussion (Sec. 6).

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide code, experiments parameters and use open-source available datasets (three out of four). In order to access monkey dataset [51], please contact the original paper authors.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We provide the code for repeating experiment in the data and we use open-source available data for our experiments. The only exception is data from Willeke et al. [51], which could be available upon request from the original authors.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: App. A.15

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report std corridors or error bars in the vast majority all of our plots.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: App. A.13

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes] ,

   Justification: Our research does not include any human data or experiments. We believe our research does not have any immediate societal impact and potential harmful consequences as we are interested in a fundumental biological question and models reproducibility. We also believe that the research process was done in a harmless way. All the animal datasets used in this paper were collected primarily for different studies.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Section App. A.14

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite the original papers that produced the code package or dataset (Höfling et al. [18] for mouse retina data and models, Willeke for monkey V4 data [51], Pierzchlewicz et al. [37] for monkey V4 model, Sridhar et al. [48] for marmoset RGC data and Vystrčilová et al. [15] for the marmoset RGC models.). Openly available assets are Pierzchlewicz et al. [37] under the CC-BY-NC 4.0 license, which allows non-commercial use, and Höfling et al. [18] data is under CC BY-NC-SA 4.0 license, again allowing us to use data and models for research purposes.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We release the code for the mouse V1 experiments, which includes our additional loss. As our code builds upon SENSORIUM 2022 codebase [8], it is well documented and straightforward to use.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.