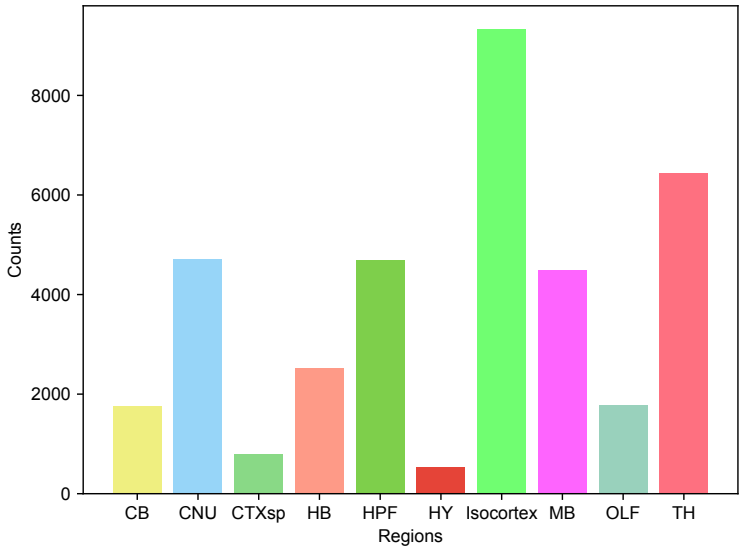
A IBL BRAIN-WIDE-MAP DATASET

The full IBL Brain-Wide-Map dataset contains 699 insertions. Among those, only 675 successfully went through the pre-processing, with 37017 units that pass the IBL spike-sorting quality control. The region distribution of those units are shown in Supplementary Figure 1.



Supplementary Figure 1: Region distribution of the IBL Brain-Wide-Map dataset. The dataset is very imbalanced and has a small number of units in hypothalamus and cortical subplates.

B BASELINES AND HYPERPARAMETERS

B.1 MODEL PARAMETERS

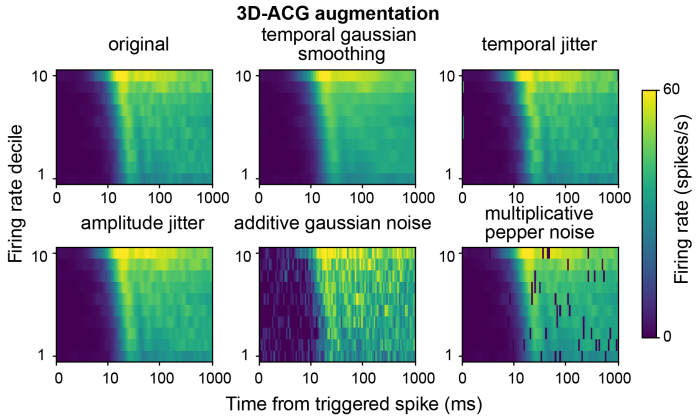
We apply the model augmentations in table 1 on the training data during NEMO. p is the probability an augmentation gets applied to an instance. We use the Adam optimizer with learning rate 0.0005 and Cosine Annealing scheduler with $T_0 = 20$. Other model hyperparameters are in table 2.

Supplementary Table 1: Model Augmentations

Augmentation	**Description**	**Type**	**p**
Gaussian noise	Gaussian noise with mean 0 and std 0.1×std of WVF	WVF	0.3
Temporal Gaussian smoothing	Smooths an ACG using a Gaussian filter along the temporal axis with $\sigma = 2$	ACG	0.5
Temporal jittering	Jitters the temporal axis of an ACG by a random integer between -3 and 3 inclusive	ACG	0.5
Amplitude scaling	Scales the amplitude of an ACG by a random number between 0.9 and 1.1	ACG	0.5
Additive Gaussian noise	Adds Gaussian noise with mean 0 and std 0.1×maximum of ACG	ACG	0.5
Additive pepper noise	Each value in the ACG has a 5% of being set to 0	ACG	0.5

B.2 MLP FINE-TUNING

For cell-type classification, we fix the hyperparameters of the MLP classifier for all methods to those used in Beau et al. (2024). This is because we have very few labels for evaluation and it is challenging to hold out any for hyperparameter tuning. For the brain region classification experiments, we tune the optimizer, learning rate, model size and dropout using the validation set of insertions.



Supplementary Figure 2: Illustration of the augmentations on spiking activity ACGs used by NEMO.

Supplementary Table 2: Model Hyperparameters

Parameter	Description	Value
Epochs	Numbers of epochs to run	6000
Batch Size	Number of items processed in a single operation	1024
Learning Rate	Learing rate of the model	0.0005
Log every n steps	Save model and cross validate results every n epochs	100
dim_embed	Latent dimension size	512
Temperature	τ in the formula for contrastive loss	0.5

B.3 SIMCLR-BASED NEMO

For the SimCLR-based NEMO, we randomly apply the same set of augmentation methods to each data point, generating two correlated augmented views. We then train each encoder and a linear projection layer to maximize agreement using contrastive learning. Compared to other methods, we use stronger augmentations for SimCLR. This is because SimCLR focuses on the differences between augmentations from the same data point, whereas CLIP-based learning strategies compare augmentations across different modalities. Stronger augmentations improved SimCLR's performance.

Both methods use contrastive learning to maximize similarity between augmented views of the same neuron, while keeping views of different neurons distinct. The main difference between the SimCLR-based method and the CLIP-based method is that CLIP is multimodal and SimCLR is unimodal. For the SimCLR-based method, we use the contrastive objective from (Chen et al., 2020). Let z_{view1} and z_{view2} be the normalized projections of two augmented views from the same modality. For a batch B, the objective is as follows:

$$\mathcal{L} = -\frac{1}{2|B|}\sum_{i=1}^{|B|}[\log \frac{\exp(z_{\text{view1}_i} \cdot z_{\text{view2}_i}/\tau)}{\sum_{j=1}^{|B|} \mathbb{1}_{\{j\neq i\}} \exp(z_{\text{view1}_i} \cdot z_{\text{view2}_j}/\tau) + \sum_{j=1}^{|B|} \mathbb{1}_{\{j\neq i\}} \exp(z_{\text{view1}_i} \cdot z_{\text{view1}_j}/\tau)} + \log \frac{\exp(z_{\text{view2}_i} \cdot z_{\text{view1}_i}/\tau)}{\sum_{j=1}^{|B|} \mathbb{1}_{\{j\neq i\}} \exp(z_{\text{view2}_i} \cdot z_{\text{view1}_j}/\tau) + \sum_{j=1}^{|B|} \mathbb{1}_{\{j\neq i\}} \exp(z_{\text{view2}_i} \cdot z_{\text{view2}_j}/\tau)}] \tag{2}$$

For SimCLR, we use the same types of augmentation as CLIP with increased strength: for the EAPs, we applied Gaussian noise with standard deviation 1 with probability 0.5, rather 0.1 standard with a probability of than a 0.3. For the ACG images, we keep the augmentation methods consistent across all models.

In our SimCLR-based method, we use the pretrained encoder to obtain separate embeddings for each modality, which are then combined to train downstream tasks such as cell-type and brain region classification.

C Different projection techniques for SimCLR

We explored different projection techniques for SimCLR. One approach was to use a SimCLR model with a smaller projection size than the representation, reducing the projection size from 512 to 128. Another approach was to use a 2-layer MLP with a 512-dimensional hidden layer and a 128-dimensional projection size instead of a linear projection layer. However, our original linear projector still achieved the highest validation metrics, as shown in Supplementary Table 3. Therefore, we continued to use the original linear projector for all experiments.

Supplementary Table 3: Linear brain region classification for IBL dataset

Model	**Acc**	**F1**
original	0.388	0.338
128-dim linear projection	0.371	0.326
mlp-based projection	0.377	0.328

D Dataset split and validation strategy

Given the limited quantity of ground-truth labels for the NP Ultra dataset, we evaluate the performance of each model using a 5-fold cross-validation of the labeled cells with 10 repeats. Since we do not have a separate validation set while training NEMO, we utilize a nested cross-validation approach to choose an evaluation checkpoint for NEMO. In other words, for each cross-validation fold, we perform a nested 5-fold cross-validation on the *training* folds with a linear classifier to choose the best checkpoint for NEMO. We then use this NEMO checkpoint to perform the evaluation on the heldout fold of the original 5-fold cross-validation. This checkpoint choosing procedure is done only on the training folds and does not use any information from the testing fold. Without this procedure, we are still able to achieve high performance.

For the IBL dataset, we use a standard 70-10-20 split for the training, validation, and test sets, respectively. During training, we monitor the performance of the model using a linear classifier trained on the training set and validated on the validation set. We then compute our evaluation metrics on the test set using the checkpoint with the highest validation F1.

E Multichannel

Using multi-channel templates has the potential to be more informative for identifying cell-types and classifying brain regions. In the following subsections, we demonstrate that using multi-channel templates does improve these downstream tasks, however, more work is needed to generalize this paradigm to different probe geometries.

E.1 IBL MULTICHANNEL

For the IBL multichannel experiments, we use 25 channel templates centered on the peak channel if possible (this is hard for edge spikes). We add two additional template augmentations: amplitude jitter (Vishnubhotla et al., 2024) and electrode dropout. Amplitude jitter rescales a channel's amplitude by a uniform value between 0.9 and 1.1 with $p = 0.3$. Electrode dropout zeros out a channel with $p = 0.1$. If all channels are zeroed out, we leave the peak channel. Our model architecture is the same as NEMO with the only difference being the input size of the waveform encoder.

We tune hyperparameters for both the linear and MLP classification models. For the linear model, we tune the inverse of the regularization strength between $1e-5$ and $1e4$ using the python module optuna (Akiba et al., 2019). For the MLP model, we do a grid search over the dropout probability$(0.1, 0.2, 0.3, 0.4)$, learning rate$(1e-4, 1e-5, 1e-6)$, number of layers$(1, 2, 3)$ and layer size$(128, 256, 512)$.

Supplementary Table 4: Multi-channel, single-unit brain region decoding for the IBL dataset

Model	**Linear**		**MLP**		**MLP fine-tuned**	
	Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**
NEMO	0.42 ± 0.00	0.40 ± 0.00	0.45 ± 0.01	0.42 ± 0.00	0.47 ± 0.01	0.44 ± 0.01
NEMO (25-channel)	**0.45 ± 0.01**	**0.42 ± 0.00**	**0.47 ± 0.01**	**0.44 ± 0.01**	**0.48 ± 0.00**	**0.45 ± 0.01**

Supplementary Table 5: Multi-unit, multi-channel brain region classification for the IBL dataset

Model	**Linear**		**MLP**		**MLP fine-tuned**	
	Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**
NEMO	0.48 ± 0.00	0.45 ± 0.00	0.50 ± 0.00	0.48 ± 0.00	0.51 ± 0.00	**0.50 ± 0.00**
NEMO (25-channel)	**0.50 ± 0.00**	**0.47 ± 0.00**	**0.52 ± 0.00**	**0.49 ± 0.00**	**0.52 ± 0.00**	**0.50 ± 0.00**

E.2 NP-ULTRA MULTICHANNEL

For NP-Ultra multichannel, we tested using 9 or 25 channels centered on the peak channel if possible. We use the same augmentations and encoding model architecture as IBL multichannel and we do not finetune the hyperparameters in the MLP or linear model.

Supplementary Table 6: Multichannel cell-type classification for the NP-Ultra dataset

Model	**Linear**		**MLP (5-fold)**		**MLP fine-tuned(5-fold)**	
	Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**
NEMO (500d rep)	0.833 ± 0.008	0.833 ± 0.007	0.839 ± 0.007	0.839 ± 0.008	0.878 ± 0.005	0.878 ± 0.005
NEMO (25-channel)	0.853 ± 0.006	0.85 ± 0.006	0.866 ± 0.007	0.862 ± 0.007	0.874 ± 0.006	0.874 ± 0.006
NEMO (9-channel)	**0.854 ± 0.013**	**0.851 ± 0.014**	**0.869 ± 0.004**	**0.865 ± 0.005**	**0.887 ± 0.003**	**0.884 ± 0.003**

F PICKING PARAMETERS FOR IBL UNIT REPRESENTATION CLUSTERING

We used the Python-implemented UMAP package (McInnes et al., 2018b) and the Python-Louvain package (Aynaud, 2020) for our clustering analyses. For our clustering analysis, we aim to find the most informative clustering with the smallest number of clusters. There are two parameters to tune: 1) the size of local neighborhood used for the UMAP graph manifold approximation resolution that controls the community size in Louvain clustering ($n_{neighbor}$), and 2) the resolution γ that determines the size of the communities. We tuned $n_{neighbor}$ while keeping the resolution to be 1.0 and tracked the final number of clusters. We picked the 'elbow' that has the smallest $n_{neighbor}$ (200). The resolution was chosen by maximizing the modularity index Q of Louvain clustering with $n_{neighbor} = 200$. The modularity index of a graph is defined as:

$$Q = \sum_{c=1}^{n} \left[\frac{L_c}{m} - \gamma \left(\frac{k_c}{2m} \right)^2 \right]$$

where each c represents a community in the graph, m is the number of edges, L_c is the number of intra-community links for community c, k_c is the sum of degrees of the nodes in community c, and γ is the resolution parameter.

G PARAMETERS FOR CLASSIFIERS AND FINE-TUNING METHODS

Supplementary Table 7: Linear probe best hyperparameters

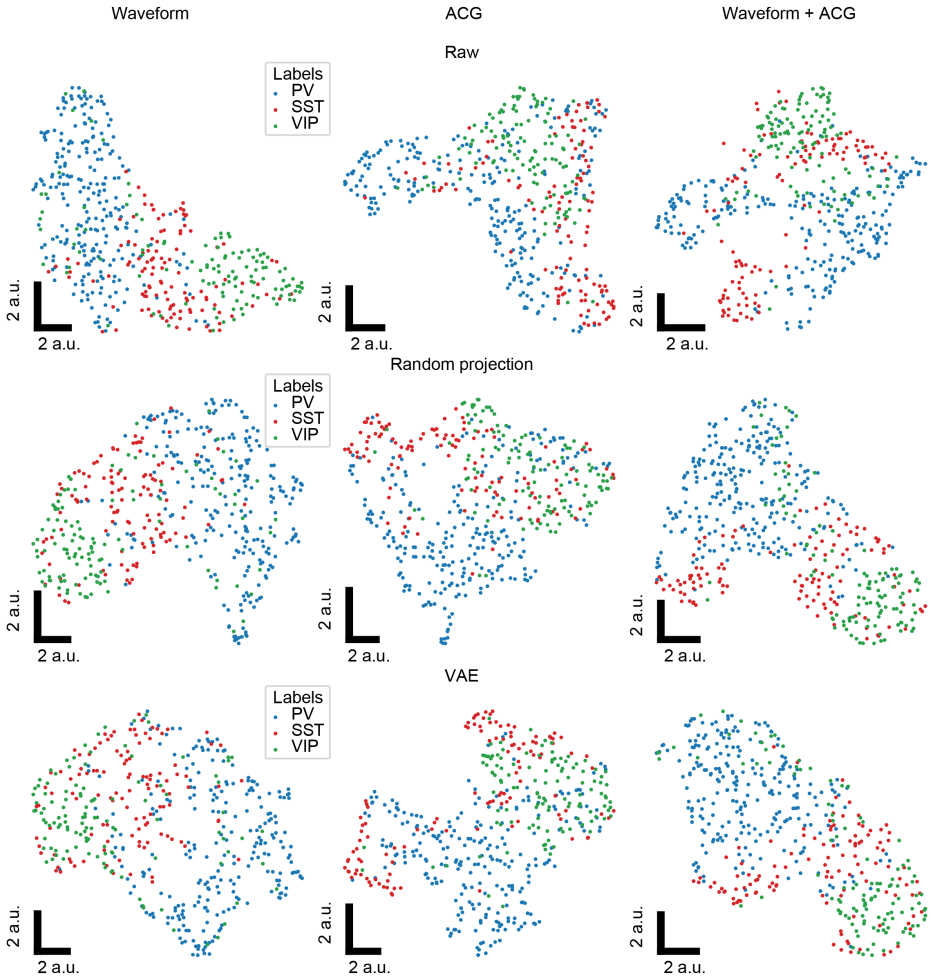
Hyperparameter	**Value**
max_{iter}	1000
tol	1e-5
NP-Ultra Celltype c	0.02
IBL NEMO joint c	0.02
IBL VAE c	0.001
IBL supervise c	0.001
IBL NEMO independent c	2.5

Supplementary Table 8: IBL MLP hyperparameters

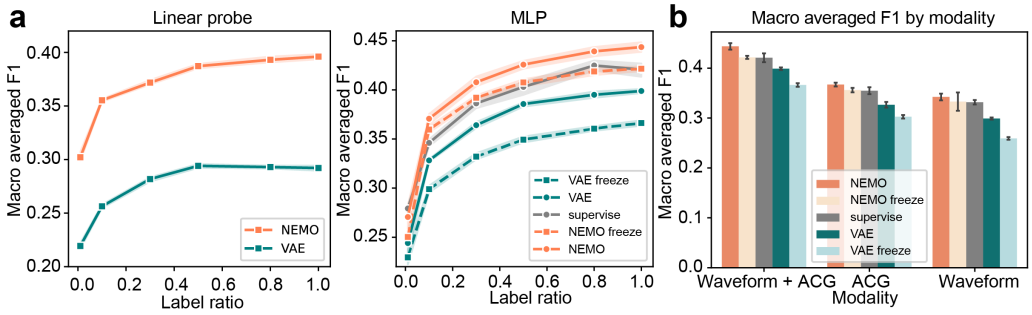
Hyperparameter	**Value**
n_{layer}	1
layer0 size	256
Dropout rate	0.2
Supervised scheduler	CosineAnnealingWarmRestarts
Supervised T_0	20
Supervised T_{mult}	1
Other models' schedular	StepLR
StepLR $step_{size}$	200
StepLR γ	0.8
Supervised lr	1×10^{-4}
Other models' lr	1×10^{-5}

G.1 UMAP EMBEDDINGS WITH RAW FEATURES, RANDOM PROJECTION AND VAE

We visualize the UMAP embeddings of the raw features, randomly initialized encoder projection and the VAE projection in Supplementary figure 3. We find that NEMO representations are visually more structured.



Supplementary Figure 3: UMAP visualizations with raw features, random projections, and from the VAE. The random projections are representations which are passed through randomly initialized encoders with the same architecture as NEMO.

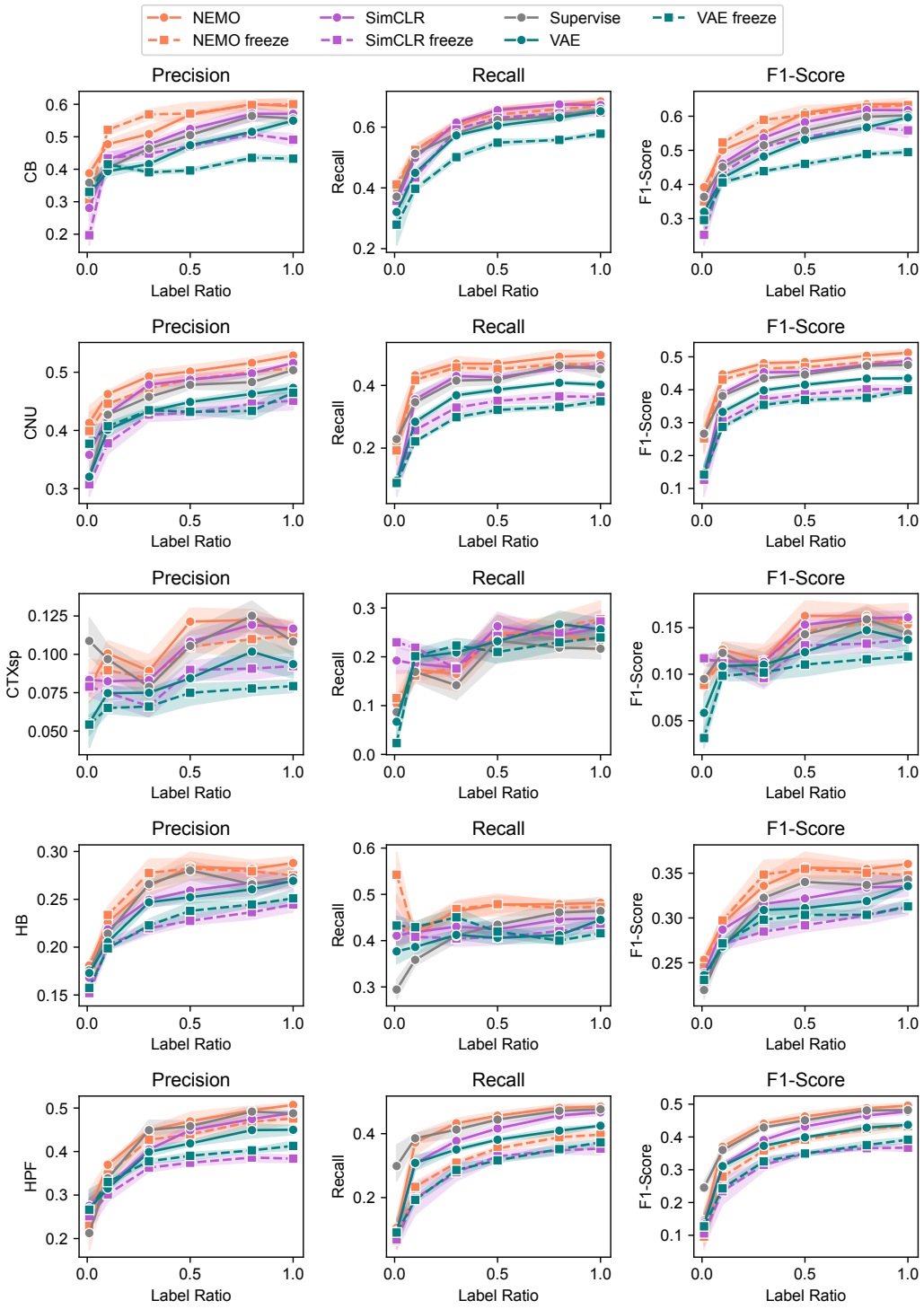


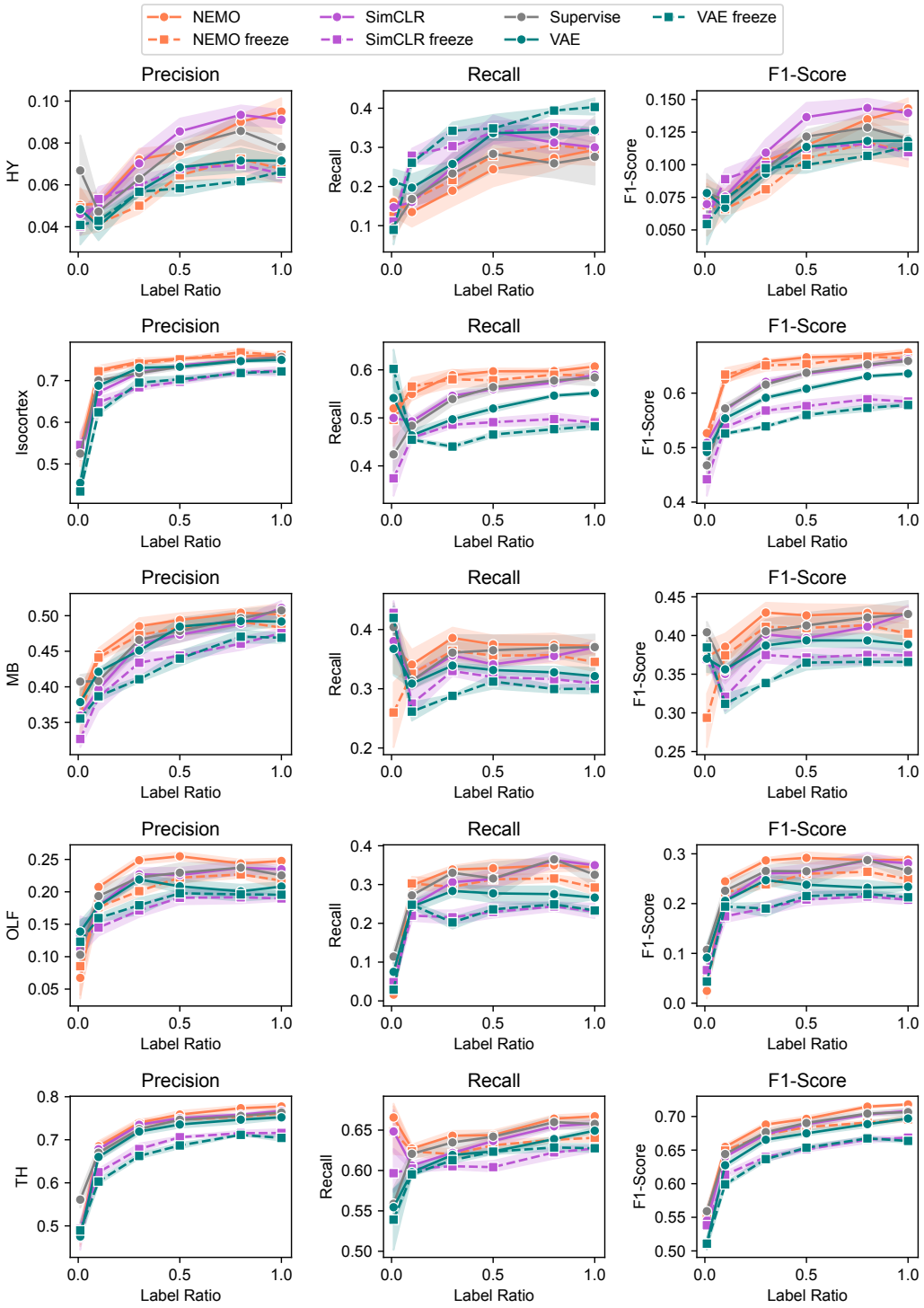
Supplementary Figure 4: Left and middle, macro-averaged F1 scores for linear and MLP-based single-neuron classification of brain region by label ratio, as in Fig. 3(d), replacing accuracy with F1 score. Right, single-neuron MLP-classification balanced F1 scores for uni- and bimodal models, as in Fig. 3(e), replacing accuracy with F1 score.

G.2 IBL REGION CLASSIFICATION WITH LABEL RATIO SWEEP

In Supplementary figure 4, we show the macro-averaged F1 scores for single neuron classification of brain region by label ratio as complementary to Figure 3.

In Supplementary Figure 5, we study the effect of varying the ratio of labeled data used to train the brain region classifier. Fully supervised methods used the same labeled examples, but were not pretrained. Means and standard-deviation bands are computed over five random initializations for each label ratio.





Supplementary Figure 5: Detailed comparison of NEMO and SimCLR for region classification performance on IBL data with varying label ratio. We show the precision, recall and F1 score for each class. NEMO shows superior performance in most of the classes, except hypothalamus (HY), which has a small sample size compared to other classes.

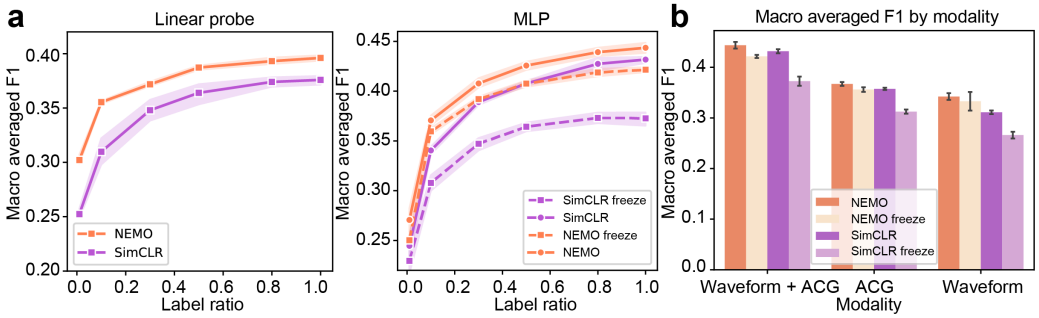
H INDEPENDENT LEARNING ABLATION FOR NEMO

We compared the cell-type classification performance between NEMO with independent and joint learning. The results are shown in Supplementary table 9.

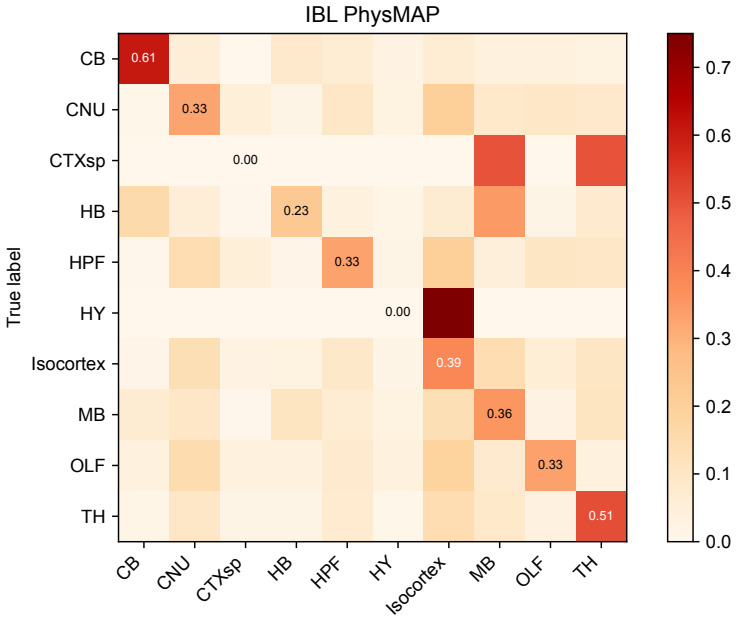
Supplementary Table 9: Cell-type classification for the NP-Ultra dataset for independent vs. joint learning NEMO.

Model	Linear		MLP		Finetuned MLP	
	Acc	F1	Acc	F1	Acc	F1
independent NEMO	**0.84 ± 0.00**	**0.83 ± 0.00**	0.83 ± 0.00	0.82 ± 0.01	0.87 ± 0.01	0.87 ± 0.01
joint NEMO	0.83 ± 0.01	**0.83 ± 0.01**	**0.84 ± 0.01**	**0.84 ± 0.01**	**0.88 ± 0.01**	**0.88 ± 0.01**

Supplementary Figure 6 shows the macro-averaged F1 scores for single neuron classification of brain region by label ratio with the IBL dataset, complementary to Figure 5.



Supplementary Figure 6: Left and middle, macro-averaged F1 scores for linear and MLP-based single-neuron classification of brain region by label ratio for independent learning ablation, as in Fig. 5(a), replacing accuracy with F1 score. Right, single-neuron MLP-classification balanced F1 scores for uni- and bimodal models, as in Fig. 5(b), replacing accuracy with F1 score.



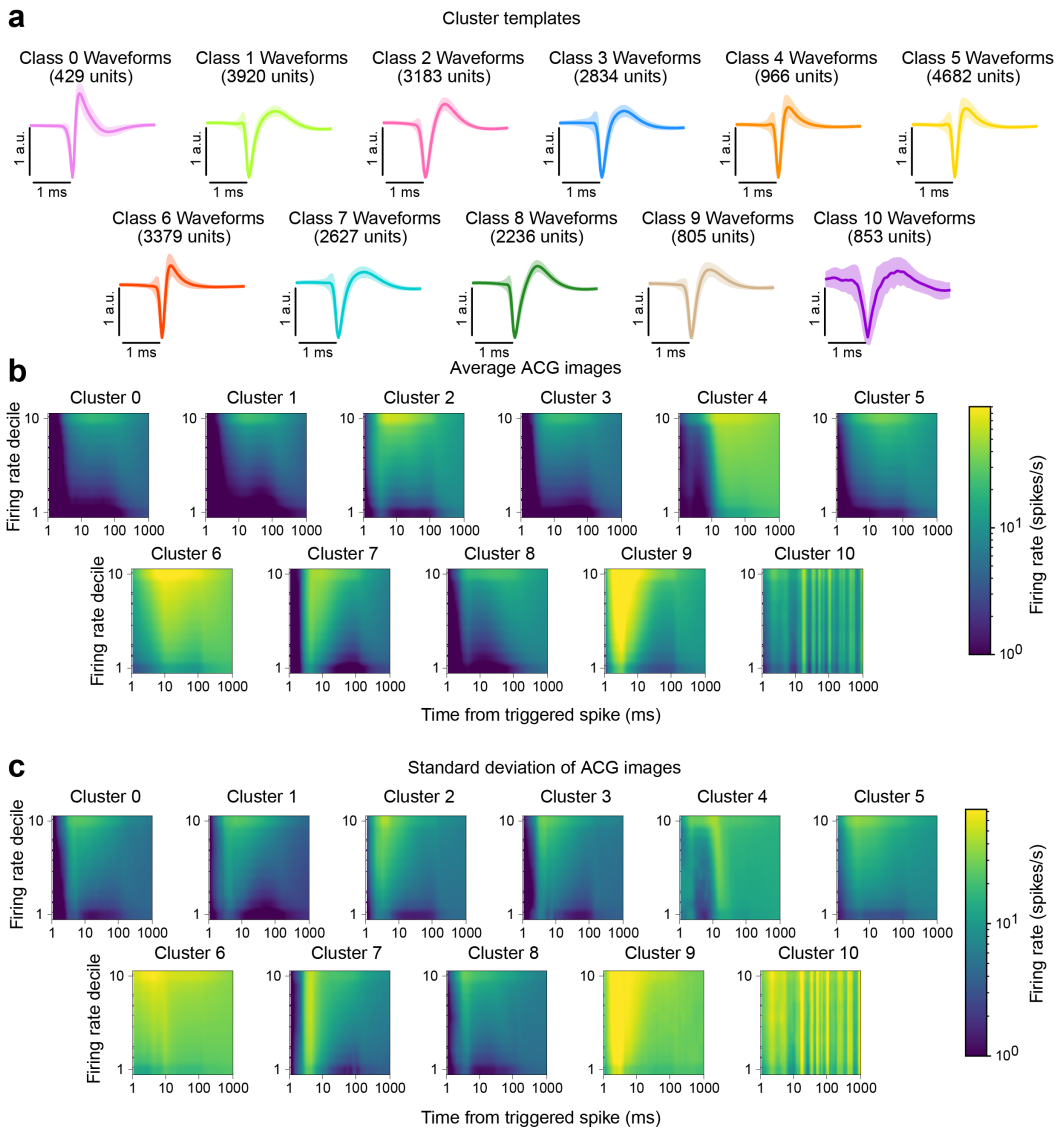
Supplementary Figure 7: Classification results of PhysMAP on the IBL brain region classification task. Due to label imbalance, PhysMAP is unable to predict CTXsp and HY, which leads to a low balanced accuracy and F1 score.

I IBL UNIT CLUSTERING RESULTS WITH NEMO

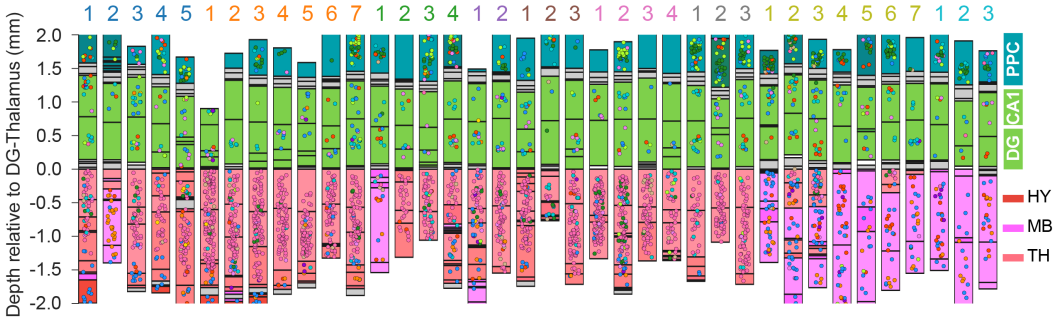
Supplementary Figure 8 shows averages and standard deviations for each cluster's template waveforms and ACG images. Supplementary Figure 9 shows the distribution of cluster labels over brain regions, separated by individual insertions and by the recording lab.

J IBL UNIT CLUSTERING RESULTS WITH RAW FEATURE INPUT

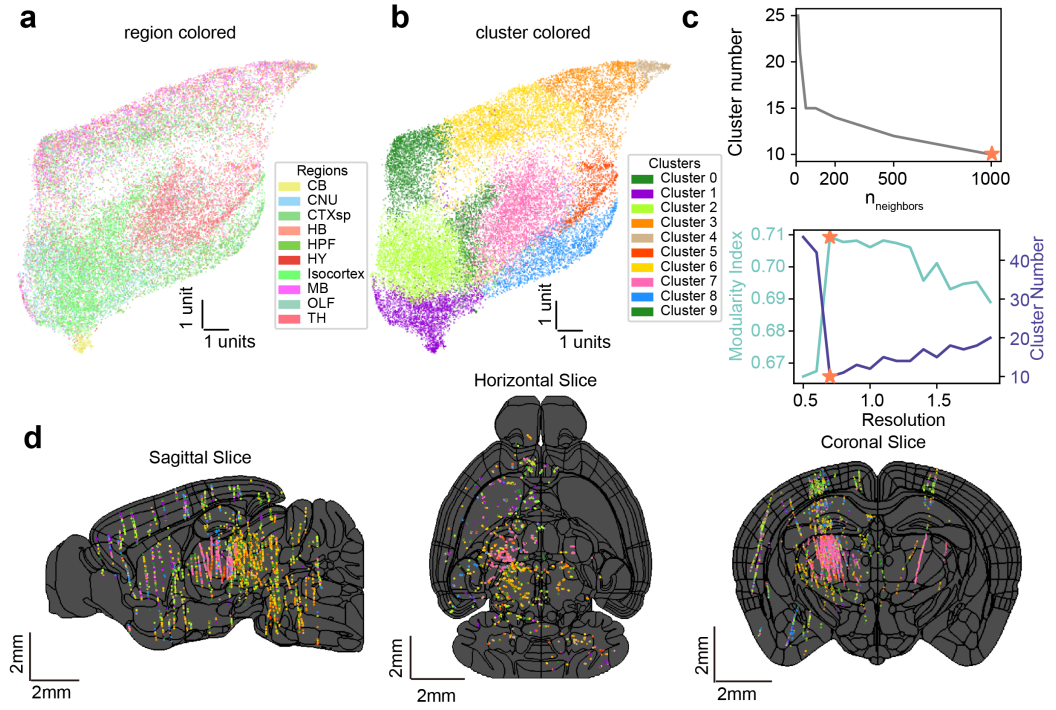
Supplementary Figure 10 shows clustering results for the IBL dataset. The hyper-parameters were selected with similar criteria as in Section 6. Since the number of clusters does not show a similar 'elbow,' but keeps decreasing as $n_{neighbor}$ increases, we picked $n_{neighbor} = 1000$ and used a resolution γ that maximizes the the modularity and minimizes the number of clusters. These clusters are less spatially organized compared to the clusters clustered using NEMO. As shown in Supplementary Figure 11, the clustering result based on NEMO overall shows lower entropy, which indicates higher region-selectivity.



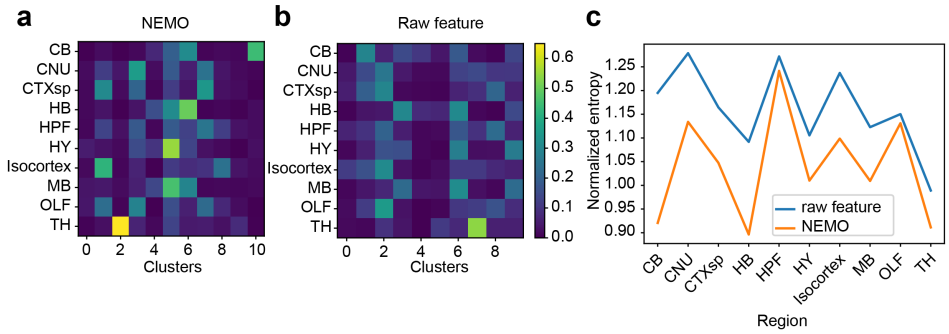
Supplementary Figure 8: Average and standard deviation of the template waveforms and ACGs for the units in each cluster, as shown in Figure 4. (a) The waveforms are consistent within clusters and distinct across clusters. (b) The ACG images are are also distinct across clusters. These results suggest that NEMO is able to find distinct clusterings of neurons across the whole-brain.



Supplementary Figure 9: Visualization of clustering results across repeated site in the IBL dataset (IBL et al., 2022). The Neuropixels probes target the same brain regions (including posterior parietal cortex, hippocampus, and thalamus) in these insertions. The color of the labels on top of each column indicates the lab ID of each insertion. Our results reveal that the clusters are highly distinguishable by region, with each region containing a distinct group of neurons. Moreover, the dominant cluster IDs for the same region remain consistent across different insertions.



Supplementary Figure 10: IBL neuron clustering using raw features. (a) A UMAP visualization of raw features on the training data colored by anatomical brain region. (b) The same UMAP as shown in (a) but instead colored by cluster labels using a graph-based approach (Louvain clustering). (c) We tuned the neighborhood size in UMAP and the resolution for the clustering. These parameters were selected by maximizing the modularity index which minimized the number of clusters. (d) 2D brain slices across three brain views with the location of individual neurons colored using the cluster IDs shown in (b). The black lines show the region boundaries of the Allen mouse atlas.



Supplementary Figure 11: Region distribution of clustering results using different NEMO features and raw features (normalized by region). For each region, we get a cluster distribution vector. We then computed the normalized entropy of that distribution. Cluster result based on NEMO overall shows lower entropy, which indicates higher region-selectivity.