

Functional Connectivity in the Brains of Zebrafish Larvae

Ans IMRAN

Contents

1	Lab and Team Description	2
1.1	Laboratory	2
1.2	The Team	2
2	Scientific Context	2
2.1	Overview	2
2.2	Introduction	2
2.3	My project	3
3	Scientific Concepts and Techniques	4
3.1	Brain Activity	4
3.2	Voxels	4
3.3	Functional Connectivity	4
3.4	Structural Connectivity	4
3.5	compositional Restricted Boltzmann Machines - cRBMs	5
3.6	Coupling Matrix OR Functional Connectivity Matrix	5
3.7	Overfitting	6
3.8	Regularization	7
3.8.1	L2L1 Regularization	8
4	Description of Work	9
4.1	One Fish Multiple Trainings	9
4.1.1	Hyperparameter optimization - Optimizing L2L1 regularization parameter	9
4.1.2	Coupling matrices from good Vs bad cRBMs	14
4.1.3	Average nRMSE between a randomly choosen pair of coupling matrices	14
4.1.4	Comparing Coupling Matrices	19
5	Conclusion	22
6	Annex	23
6.1	good Vs bad cRBM visual comparison	23

1 Lab and Team Description

1.1 Laboratory

The Laboratoire Jean Perrin Laboratory (LJP) specializes in physics at the interface with biology and medicine, exploring responses of biological systems to external stimuli and developing bio-inspired mechanical and physico-chemical systems. They employ advanced experimental techniques, focusing on optics and microfabrication, to study complex biological systems at various scales.

1.2 The Team

I did my internship in the research activity entitled Zebrafish and Danionella Cerebrum behavior and calcium imagery.

The team focuses on leveraging advanced calcium imaging techniques, particularly Single-Plane Imaging Microscopy (SPIM), adapted for functional imaging of GCaMP transgenic zebrafish larvae. This innovation allows for simultaneous recordings of an unprecedented number of neurons at standard acquisition rates. By capturing such extensive neural activity, the team aims to obtain new insights into perceptual information processing. The resulting large datasets enable them to explore intricate statistical correlations between various brain regions, promising significant advancements in our understanding of how the brain processes sensory information.

Here are some recent publications from the team Rajan et al. [17] Ghosh et al. [6]; Hubert et al. [11] and Beiza-Canelo et al. [2] etc.)

2 Scientific Context

2.1 Overview

Recent advancements in large-scale neuronal recordings, which can capture data from tens of thousands of neurons, have significantly enhanced our ability to study brain activity. However, interpreting these vast amounts of data remains a substantial challenge. Here we use a method called compositional Restricted Boltzmann Machine - cRBM (Tubiana, Cocco, and Monasson [20]), designed to identify structures in this data, such as cell assemblies and functional connectivity, without requiring extensive computational resources. Remarkably, data from 40,000 neurons in zebrafish can be analyzed within a day (Plas et al. [16]), showcasing the efficiency of this approach. This advancement holds significant value for both data analysis and those interested in parsing neural data obtained by other large-scale recording techniques.

2.2 Introduction

The brain is a complex, multi-scale network, with activity patterns organized into neural assemblies (Harris [8]). These neural assemblies are thought to form elementary computational units and subserve essential cognitive functions such as short-term memory, sensorimotor computation or decision-making (Hebb [9]; Gerstein, Bedenbaugh, and Aertsen [5]; Harris [8]; Buzsáki [3]; Harris [7]; Palm et al. [14]; Eichenbaum [4]). Despite their

importance, extracting the assembly organization from entire brains and demonstrating that these assemblies explain overall neural dynamics remains a challenge. Advances in SPIM have enabled simultaneous recording of most neurons in the zebrafish brain at single-cell resolution, providing new opportunities to model neural activity comprehensively (Panier et al. [15]; Ahrens et al. [1]; Wolf et al. [24]; Wolf et al. [23]; Migault et al. [12]; Vanwallegem, Ahrens, and Scott [22]).

Previous attempts to identify neural assemblies through clustering (Panier et al. [15]), dimensionality reduction (Mu et al. [13]), or latent variable models (Triplett et al. [19]) have been limited to single brain regions and do not explicitly evaluate the assemblies' ability to reproduce observed brain activity statistics.

My hostlab has recently published a study that addresses these limitations by using a generative model constrained by a finite number of cell assemblies (Plas et al. [16]). The cRBM (further details in section 3.5), an extension of the classical RBM, infers neural assemblies solely from data statistics and can replicate empirical data through its generative nature.

The cRBM steers the assemblies description to the so-called compositional phase, where a small number of assemblies are active at any point in time, making the resulting model highly interpretable. The study successfully trained cRBMs on recordings from 41,000 zebrafish neurons, grouping neurons into 200 assemblies. These assemblies were anatomically localized, spanned the entire brain, and accurately replicated brain activity statistics. They demonstrated higher predictive power than fully connected models with many more parameters, validating that assemblies underpin collective neural dynamics.

Furthermore, the probabilistic nature of the cRBM model allowed the computation of a functional connectivity matrix (also called **Coupling Matrix**) by quantifying the effect of activity perturbations in silico. Details in section 3.6

This assembly-based functional connectivity was well-conserved across cRBMs trained on recordings from different zebrafish and consistent with anatomical connectivity at the mesoscale. This suggests that neural assemblies, as identified by the cRBM, play a crucial role in the brain's functional architecture.

In summary, the study presents an extensive assembly decomposition of the zebrafish brain, which accurately accounts for its activity statistics. The cRBM model provides a widely applicable tool for constructing low-dimensional representations of high-dimensional neural data, particularly valuable for very large-scale systems. Its generative capability allows the production of new (synthetic) activity patterns, making it suitable for direct in silico perturbation and ablation studies. This work significantly advances our understanding of neural assembly organization and offers a powerful method for analyzing large-scale neuronal data.

2.3 My project

Building on this work, the lab is currently investigating how comparable the brains of different zebrafish at the same developmental stage are in terms of stereotypicity of patterns of spontaneous brain activity, assembly descriptions, and functional connectivity.

One approach involves training a single model on data collected from multiple fish to determine how effectively such a model can replicate the spontaneous activity of individual animals. Given that individual fish slightly differ in neuron count and exact neuron locations, conducting studies at the scale of single neurons poses challenges. To manage this, the lab operates at a meta-scale by mapping each fish onto a spatial grid of voxels and defining the activity within each voxel as the average activity of the neurons that fall into it. The compositional Restricted Boltzmann Machine (cRBM) is then trained to reproduce the activity of these voxels. In comparison to the previously published model, the total number of parameters is reduced because the number of voxels is smaller than the number of neurons, and only 30 hidden units are used compared to 200 in the original model. To ensure that this model converges, the regularization parameters implemented in the training algorithm have to be reoptimized for the new model architecture. During my internship, I conducted an extensive sweep over the parameter space to identify the optimal regularization parameter to prevent overfitting while minimizing the probability of model convergence failure. For this study, I focused on optimizing the training of the model using neural activity recorded from a single fish. But my characterisation lays the groundwork for training the model on multiple fish in the future. Additionally, I investigated how model-inferred functional connectivity depends on model convergence quality and the choice of the regularization parameter.

3 Scientific Concepts and Techniques

3.1 Brain Activity

In the context of this internship, brain activity refers to spikes in neuronal activity. For example, during a specific time frame, if a neuron's activity spikes, we record a 1 for that neuron in that time frame; otherwise, we record a 0.

Also note, the length of each time frame is 0.4 seconds.

3.2 Voxels

We divide the whole volume of the fish brain into small cubes of equal sizes. We call each cube a voxel. The activity of a voxel is the average activity of all of the neurons that lie inside that voxel. Again, here, by neuronal activity, I mean neuronal spikes.

3.3 Functional Connectivity

If the activities of two voxels or neurons show a high degree of correlation over time, it means that there is a functional connection between these two voxels or neurons.

3.4 Structural Connectivity

Structural connectivity refers to the anatomical connections between different brain regions, specifically the physical pathways that connect neurons within the brain. Unlike functional connectivity, which measures the temporal correlation of neural or voxel activity, structural connectivity focuses on the physical wiring of the brain's neural network.

3.5 compositional Restricted Boltzmann Machines - cRBMs

A Restricted Boltzmann Machine (RBM) is a shallow, two-layered generative neural network capable of learning a probability distribution over its set of inputs. The first layer, known as the visible layer, represents the input data, while the second layer, called the hidden layer, captures dependencies between the inputs.

In the visible layer, each visible unit v_i corresponds to a voxel. In the hidden layer each Hidden Unit (HU) h_μ corresponds to the (weighted) activity of its voxel assembly and is chosen to be real-valued. The joint probability distribution $P(v, h)$ writes (Hinton and Salakhutdinov [10]; Tubiana and Monasson [21]):

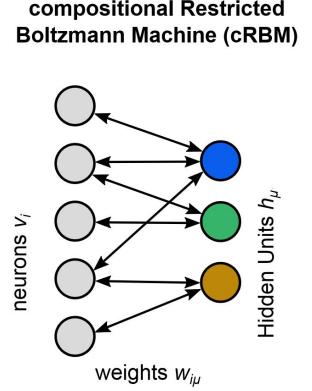


Figure 1: cRBM

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h})) = \frac{1}{Z} \exp \left(\sum_i g_i v_i - \sum_\mu U_\mu(h_\mu) + \sum_{i,\mu} w_{i,\mu} v_i h_\mu \right)$$

where E is the energy function and $Z = \sum_{\mathbf{v}} \int_{\mathbf{h}} dv dh \cdot \exp(-E(\mathbf{v}, \mathbf{h}))$ is the partition function. The weights g_i and potentials U_μ control the activity level of the visible units and the marginal distributions of the HUs respectively, and the weights $w_{i,\mu}$ couple the two layers. Note that while \mathbf{v} is directly observed from the neural recordings, \mathbf{h} is by definition unobserved (i.e. hidden) and is sampled from the observed \mathbf{v} values instead.

In this internship, I have utilized a modified form of Restricted Boltzmann Machine (RBM) known as the Compositional Restricted Boltzmann Machine (cRBM). One of the advantages of cRBM is that it has L2L1 regularization (section 3.8.1). Further details of a cRBM can be found here Plas et al. [16].

3.6 Coupling Matrix OR Functional Connectivity Matrix

The coupling matrix is a matrix that gives us the strength of coupling of each voxel with all of the other voxels in the fish brain. It is computed by using the formula:

$$J_{ij} = \sum_{\mu=1}^M w_{i\mu} w_{j\mu} \langle \text{Var}(h_\mu | \mathbf{v}) \rangle_{\text{data}}$$

where

J_{ij} : Strength of coupling between voxel- i and voxel- j

$w_{i\mu}$: The weight of the connection between the visible-unit- i and hidden-unit- μ

M : Total number of hidden units

h_μ : The output of the hidden unit μ

\mathbf{v} : Visible configuration of voxels for one time step

$(h_\mu | \mathbf{v})$: Output of hidden-unit- μ given \mathbf{v}

$\langle \text{Var}(h_\mu | \mathbf{v}) \rangle_{\text{data}}$: Avg of the variations in the output of hidden unit μ over all time steps

For more details see Plas et al. [16]

3.7 Overfitting

In machine learning projects, the goal is to train models that can make predictions on new, unseen data. Unseen data refers to data not included in the training set, so it wasn't used to adjust the model's parameters. Once our model can make accurate predictions on this unseen data, our task is accomplished.

During the training of machine learning models, the goal is to fit a model to the training data. This process involves adjusting the parameters of the model to best match the training data in a generalized way.

This can be understood easily in the context of a polynomial curve fitting. Let's say we use a polynomial of degree 'b' as our machine learning model. The coefficients of this polynomial can then be considered as the parameters or weights of the model.

For instance, in a polynomial equation like:

$$p(x) = a + bx + cx^2 + \dots + zx^b$$

the coefficients a, b, c, \dots, z can be considered to be the parameters of the our machine learning model. During training they have to be adjusted to fit the polynomial curve to the training data.

While fitting the parameters of the model to the data, we may encounter a phenomenon known as overfitting. Overfitting can occur when the machine learning model is too big/complex.

In the context of polynomial curve fitting, a too complex model refers to a polynomial of high degree, exceeding what is necessary to accurately represent the data. This complexity can lead to a fit focused too much on the noise in the data. Thus the polynomial curve will not be fitted in a 'generalized way' rather it'll be overfitted.

As a result of overfitting, the model performs well on the training data but poorly on unseen data.

For example, see Figure 2

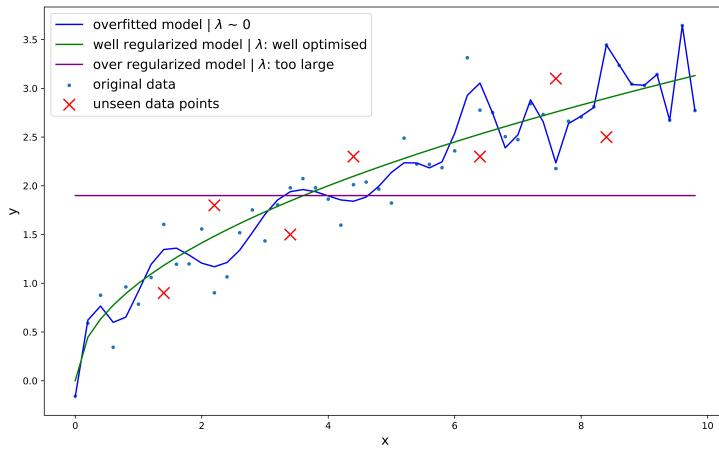


Figure 2: Overfitted curve Vs Regularized curve AND Generalization Error

The dotted blue points represent our training data. We've fitted a blue polynomial curve to this training data. Notice how the blue curve tries to pass through all the training data points. This is particularly very clear for points in the $x=[6,8]$ range. This

is indeed an illustration of overfitting in the context of polynomial curve fitting. The blue polynomial curve is overfitted to the training data.

Now, if we use the polynomial model corresponding to the blue curve to make predictions in the range $x=[6,8]$. Our predictions would precisely align with the blue curve, resulting in small errors between the predictions and the training data points in this region. However, if we were to make predictions on unseen data points, represented by red ‘X’ markers on the plot, the errors would be relatively large. This phenomenon is termed as the generalization error. Despite making accurate predictions on the training data, the model performs poorly on unseen data. This is the problem with overfitted models—they struggle to generalize well.

Now, observe the green curve. While it may not perfectly fit the training data, but predictions made by the model corresponding to this curve would exhibit significantly less generalization error (the error between predictions and unseen data points). This green curve represents the best generalized fit.

Now the question is how do we avoid overfitting and how do we make a model that generalizes well? Luckily, there are multiple ways to address overfitting! Regularization is one of the most important and effective method to combat overfitting. In the next section we delve into regularization.

3.8 Regularization

Machine learning models are generally trained using gradient descent algorithm see Ruder [18]. In gradient descent algorithm a loss function is used to update the parameters or weights of the model. Minimizing the loss function with respect to the parameters of the model gives us the optimal values for the parameters. However as we have seen in the last section, this can result in an overfitting of the parameters of the model to the training data. In order to avoid this problem, we can use regularization.

In regularization we create a modified loss function such that

$$\text{modified loss} = \text{original loss}(w_1, w_2, \dots, w_j) + \text{regularization term}(w_1, w_2, \dots, w_j)$$

This regularization term (for L2 regularization) looks like this

$$\text{regularization term} = A\lambda \sum_j w_j^2$$

where A is a scaling factor and λ is the regularization parameter.

The details of A are not important for our current discussion. The regularization parameter λ , however, is important for our current discussion.

Now, notice the full form of regularization term,

$$\text{regularization term} = A(\lambda w_1 + \lambda w_2 + \dots + \lambda w_j)$$

The regularization term contains products of the form

$$\lambda w_j$$

For all these products the value of λ is fixed and is given by us, so for the algorithm this value is fixed until unless we change it by hand. Therefore, the only parameters that the algorithm can change are w_j .

As we discussed earlier that the parameters of the model are learned by minimizing the loss function. Since now we have added another term in the original loss function, now, in order to minimize the modified loss function, this new regularization term must also be minimized. And since the the algorithm cannot change the value of λ , therefore, it decreases the value of the regularization term by decreasing the values of the different w_j in the products of the form λw_j (which are the constituents of the regularization term). And thus the result is that, that the presence of the regularization term forces the algorithm to have smaller values for w_j (the parameters of the model). Notice the bigger the value of the regularization hyper-parameter λ , the smaller will be the values of the parameters w_j , in order to keep the regularization term small.

Also as it can be seen in the figure 2, if λ is too large, we get underfitting. On the other hand, if the λ is too small we get overfitting. So, we have to find an optimal value for λ .

Unfortunately there is no single magical value for λ , the optimal value of λ varies from case to case. This is a value that we have to find ourselves. In fact finding the optimal value for λ was one of the main tasks that was assigned to me during this internship. See section 4.1.1 for a detailed discussion on hyperparameter optimization.

3.8.1 L2L1 Regularization

The regularization that we just discussed is called L2 regularization, however, in the cRBM that I have used in this internship, uses L2L1 regularization. L2L1 regularization is not a standard regularization (in a sense that it is not normally used in cRBMs in the Machine Learning community). The L2L1 regularization term is given by:

$$\text{L2L1 regularization} = \frac{\lambda^2}{2Nq} \sum_{\mu} \left(\sum_{i,v} |w_{i\mu}(v)| \right)^2$$

where

- $w_{i\mu}$: The weight of the connection between the visible-unit- i and hidden-unit- μ
- λ : The L2L1 regularization parameter
- v : Visible configuration of voxels for one time step

The reasons why we use L2L1 regularization and not L1 or L2 regularization are as follows:

- Using only L2 regularization does not result in sparse weights. For more details on why sparsity is important to us, refer to this study: Plas et al. [16].
- On the other hand, using only L1 regularization tends to make the weights too sparse. This can lead to hidden units having connection strengths with all visible units that are very close to zero or zero, making them effectively disconnected from any visible unit. We refer to these as orphan hidden units.

So, we want sparsity, but not too much sparsity. This is where L2L1 regularization comes into play. It provides the right balance of sparsity without creating orphan hidden units. For more details on L2L1 regularization, refer to Tubiana, Cocco, and Monasson [20].

4 Description of Work

4.1 One Fish Multiple Trainings

4.1.1 Hyperparameter optimization - Optimizing L2L1 regularization parameter

In addition to weights, machine learning models have other parameters as well, called hyperparameters. While the weights are generally learned through gradient descent, the hyperparameters typically need to be optimized by humans. Hyperparameters are optimized to address various issues in machine learning models and to enhance their performance and efficiency. One common issue in machine learning models is overfitting—see section 3.7. In our case we also had the issue of bad cRBMs see section 4.1.1. One of the ways to avoid bad cRBMs and to combat overfitting is to use regularization—see section 3.8.

So, in order to address these issues, I had to use regularization. Specifically, I needed to determine the optimal value of the L2L1 regularization parameter λ i.e. the λ value that, on average, generated the best brain activity statistics and had the lowest percentage of bad cRBMs.

To find the optimal value of λ , I went through the following procedure:

I split the data into train and test sets. The test set is necessary to check for overfitting, as discussed in Section 3.7.

I then trained cRBMs using data from one fish for uniformly increasing values of λ . At each value of λ in the training range, I trained 20 cRBMs.

I used 2 uniform ranges.

- The first range that I used was $[0.005, 0.1]$ with a step size of 0.005.
- The 2nd range that I used was $[0.5, 5.0]$ with a step size of 0.5.

I then assessed the quality of the trained cRBM model by quantifying how well the model reproduced the following five brain activity statistics:

Brain Activity Statistics

Brain Activity Statistics: In this report, when say Brain activity statistics, we mean the following statistics:

- $\langle v_i \rangle$: The mean activity of neuron-i or voxel-i (a collection of neurons).
- $\langle h_\mu \rangle$: The mean activity of hidden-unit- μ (an assembly of voxels).
- $\langle v_i h_\mu \rangle$: The interaction between voxel-i and hidden-unit- μ
- $\langle v_i v_j \rangle$: The pairwise interaction between voxel-i and voxel-j
- $\langle h_\mu h_\nu \rangle$: The pairwise interaction between hidden-unit- μ and hidden-unit- ν

As a metric, I computed the nRMSE values between the activity generated by the cRBMs and the activity from the test dataset for all five brain activity statistics.

normalized Root Mean Square Error - nRMSE

normalized Root Mean Square Error - nRMSE: Depending on the availability of an optimal RMSE, two different formulas were used to calculate nRMSE. The optimal RMSE is the RMSE calculated between the statistics derived from the training data and the validation data.

$$nRMSE(X, Y) = \frac{RMSE(X, Y)}{RMSE(X_{\text{shuffled}}, Y_{\text{shuffled}})}$$

without optimal

$$nRMSE(X, Y) = 1 - \frac{RMSE(X, Y) - RMSE(X_{\text{shuffled}}, Y_{\text{shuffled}})}{RMSE(X_{\text{optimal}}, Y_{\text{optimal}}) - RMSE(X_{\text{shuffled}}, Y_{\text{shuffled}})}$$

with optimal

The nRMSE computed with optimal gives us some extra information. E.g, if:

$nRMSE = [-\infty, 0]$:	The nRMSE is better than the optimal RMSE
$nRMSE = 0$:	The nRMSE is as good as optimal RMSE
$nRMSE = 1$:	The nRMSE is as bad as shuffled RMSE
$nRMSE = (1, +\infty]$:	The nRMSE is worse than shuffled RMSE

Using the nRMSE values (with optimal) just computed, I was able to identify all the bad cRBMs.

bad cRBMs

bad cRBMs “bad cRBMs” refers to those cRBMs for which the nRMSE value for any of the five brain activity statistics exceeded 1.0 or returned NaN. A visual comparison between good and bad cRBMs can be found in Annex figure 13.

Originally, I had 600 cRBMs, then I removed all the bad cRBMs. I was now left with 519 good cRBMs. I named this group of good cRBMs as the desired cRBMs.

Now, I wanted to find the value of λ that yielded cRBMs with following properties:

- Lower nRMSE values for brain activity statistics.
- A small enough percentage of bad cRBMs, so that this λ can be confidently used later to train cRBMs on Multiple Fish Data.

To measure these properties, I used following techniques and tools:

nRMSE Vectors:

First, I wanted to differentiate between a relatively good and a relatively bad cRBM based on how well they reproduced the brain activity statistics as compared to other cRBMs. To achieve this I created an nRMSE vector.

nRMSE Vector

I treated all the nRMSE values for the 5 brain activity statistics (except for $\langle v_i \rangle$) as elements of a 4-dimensional Euclidean vector. I then computed the L2 norms of these 4d euclidean vectors (nRMSE vectors) for all of the desired cRBMs.

I did not use the $\langle v_i \rangle_{nRMSE}$ for the following reasons.

The $\langle v_i \rangle$ statistic is the easiest to reproduce for all visible units. This is because I performed z-score normalization on my input data v (the voxel activity). Due to the z-score normalization, $\langle v_i \rangle \sim 0$. Since $\langle v_i \rangle \sim 0$, this statistic is very easy to reproduce for cRBMs.

Even poor cRBMs (those with high nRMSE values for the other four statistics) can reproduce it well. Therefore, it provides little information when trying to differentiate between relatively good and relatively bad cRBMs. So, I wanted to prevent it from unduly influencing the norm of the nRMSE vector.

Now, I had a method to differentiate between relatively good and bad cRBMs in the context of how well they reproduce the brain activity statistics. I simply had to examine the L2 norm of the nRMSE vector associated with each cRBM. If the L2 norm was higher (relative to other cRBMs), it would indicate that the underlying cRBM was not reproducing the brain activity statistics well (relative to other cRBMs). Conversely, if the L2 norm was lower (relative to other cRBMs), it would indicate that the underlying cRBM was reproducing the brain activity statistics well (relative to other cRBMs).

Note that for each λ value, I trained 20 cRBMs. I now calculated the mean of the L2 norms of the nRMSE vectors for each value of λ . By comparing these means, I aimed to identify which λ value produces the lowest norm nRMSE vectors. This would help me determine the optimal λ value. I also plotted these results; see Figure 3.

Percentage of bad cRBMs:

Percentage of bad cRBMs

Since at each λ value, 20 cRBMs were trained, I simply used the following formula:

$$(\text{Percentage of bad cRBMs})_{\text{at } \lambda=x} = \frac{20 - (\text{Number of Desired cRBMs})_{\text{at } \lambda=x}}{20} * 100$$

to get the percentage of bad cRBMs.

The results of the last two computations can be seen in figure 3.

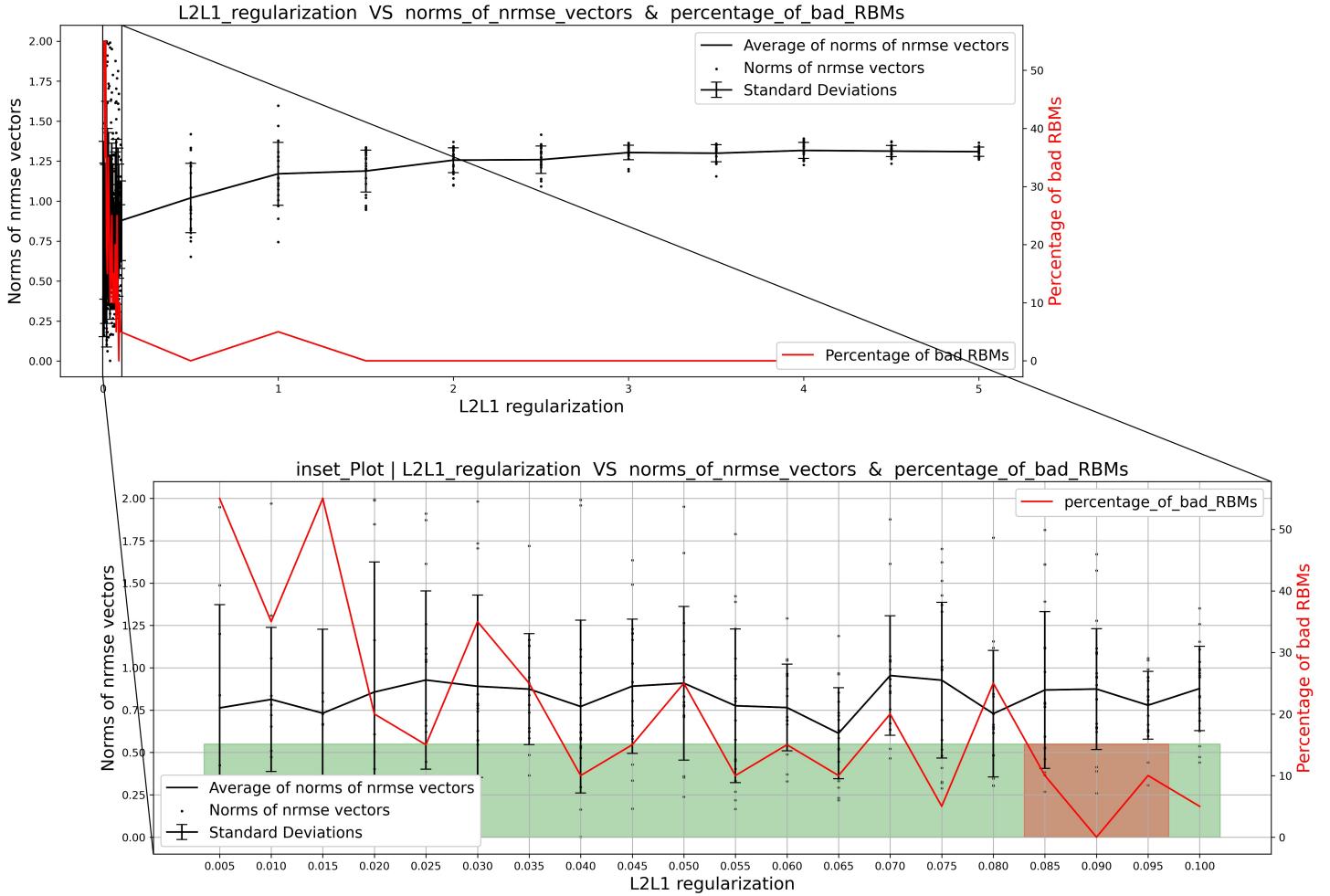


Figure 3: Choosing the best value of regularization

Note, in this figure we have two y-axes, sharing the same x-axis.

The first y-axis, labeled “Norms of nRMSE vectors,” measures how well a certain cRBM reproduces the brain activity of zebrafish larva.

The black dots in the figure are related to this 1st y-axis. They represent the norms of nRMSE vectors. In general, the lower the norm, the higher the quality of the underlying cRBM.

The black line in the figure is passing through the means of norms at each regularization value. The black vertical bars represent standard deviations around the mean values of norms.

The red line in the figure is associated with the 2nd y-axis, labeled “Percentage of bad cRBMs”.

inset plot: Notice that there is an inset plot in this figure. The inset plot is the same as the parent plot, with one exception. It only shows the λ values from the first range, i.e., $\lambda = [0.005, 0.1]$.

The green region in the inset plot contains the cRBMs that are relatively better than all of the other cRBMs.

On top of that, in the red region, the percentage of bad cRBMs also drops below 10%. This region is the best candidate for choosing a regularization value.

Note:

The 2nd range of λ values ($\lambda = [0.5, 5]$), in figure 3, won't be discussed here as there were some problems with the cRBMs trained in this range. For a discussion of cRBMs in this range kindly refer to section 4.1.2.

So, here we'll only talk about the λ values in the 1st range i.e. $\lambda = [0.005, 0.1]$, which is the inset plot in figure 3.

good cRBM:

If we examine the identity plot of any statistic - see figure 4, we notice that when the associated nRMSE is 0.55 or lower, the statistic, to some extent, follows the identity line. This indicates that this statistic of the generated activity, is, on average, similar to the corresponding statistic of the zebrafish brain activity.

Additionally, notice that the L2 norm has the property that if the L2 norm of a vector is ' t ', then none of the elements of the vector has a magnitude greater than ' t '. Therefore, if the L2 norm of a vector is 0.55, all elements of the vector will have magnitudes smaller than 0.55 i.e. none of the statistics of the generated activity will have an nRMSE value greater than 0.55.

So, we can use these two facts to set a cRBM selection threshold of 0.55. So, any cRBM whose generated activity has an nRMSE vector norm smaller than or equal to 0.55 will be considered as a good cRBM.

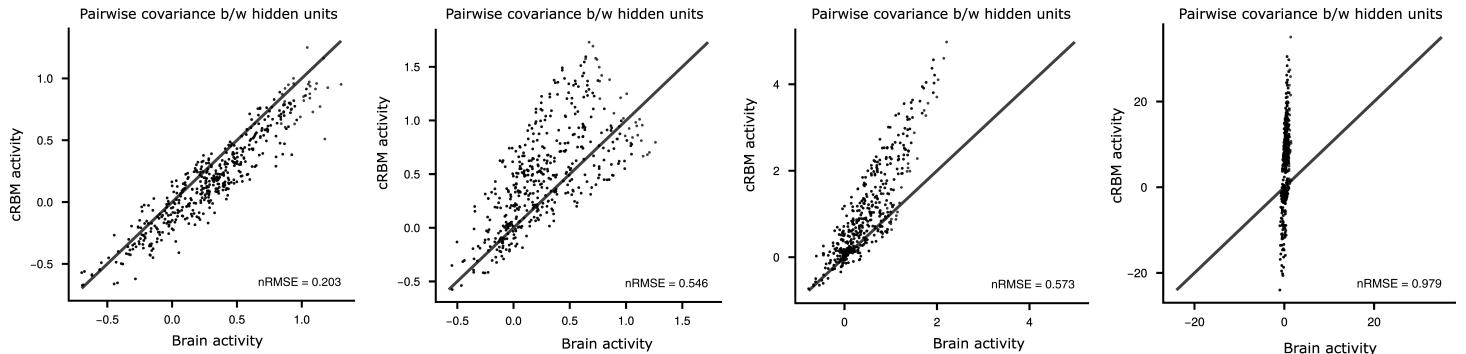


Figure 4: Validation of Normalized Root Mean Square Error (nRMSE) for Similarity Assessment.

Four examples of identity plots are shown for different nRMSE values, illustrating pairwise covariances between hidden units. The x-axis represents covariances derived from zebrafish brain activity, while the y-axis corresponds to covariances generated by compositional Restricted Boltzmann Machines (cRBMs). For nRMSE values below 0.55, the plots show covariances closely aligned with the identity line, indicating minimal deviation between the two data sources. This close alignment suggests that cRBMs with nRMSE values under 0.55 provide accurate simulations of zebrafish brain activity. Conversely, cRBMs with higher nRMSE values demonstrate greater deviations, signifying less accurate representations.

If we look at the green highlighted region in figure 3 (inset plot). All of the cRBMs in this region have an nRMSE vector norm of 0.55 or lower. Thus all of the cRBMs in this region can be considered to be relatively good.

Now, we see the trend that the percentage of bad cRBMs is decreasing with increasing strength of λ . Infact in the red-region in figure 3 (inset plot), the percentage of bad cRBMs is, on average less than 10 percent.

Furthermore, in this region we are also getting, on average, at least 3 cRBMs with norms less than 0.55

In summary:

- All of the cRBMs in the green region are relatively good ($nRMSE < 0.55$).
- On top of that, in the red region, the percentage of finding a bad cRBM ($nRMSE > 1$ OR $nRMSE = \text{Nan}$) is less than 10 percent.
- Furthermore, on average, at least 3 good cRBMs out of 20 cRBMs can always be found in the red region region.

Thus, based on these observations I choose the range of λ values, that fall within the red region in figure 3 (inset plot), to be the optimal values of λ .

Furthermore, since the middle of range is 0.09, therefore, the optimal λ value that I choose was 0.09.

4.1.2 Coupling matrices from good Vs bad cRBMs

In Figure 5, we see a representation of a coupling matrix. For details on the coupling matrix, please refer to the "Functional Connectivity Matrix" section 2.2. The figure demonstrates that regardless of the λ value, the large-scale structure of both coupling matrices (the one corresponding to a good cRBM and the one corresponding to a bad cRBM) remains the same. Furthermore, we also see that the coupling matrices at λ values of 0.5 and 5.0 are too much sparse i.e. they are indicating almost zero functional connectivity in the fish brain.

4.1.3 Average nRMSE between a randomly choosen pair of coupling matrices

Now, I wanted to understand how the coupling matrices obtained from two different cRBMs compare with each other. And what is the average nRMSE value between a randomly chosen pair of coupling matrices.

So, I calculated the nRMSE values between all pairs of coupling matrices. I got a distribution of nRMSE values. I plotted the distribution. See figure 6

I was expecting a single peak (a gaussian like distribution), but, to my surprise, I observed two peaks. I tried to understand what's really happening here. So, I made a hypothesis.

Hypothesis:

My hypothesis was that there are actually two different groups of coupling matrices.

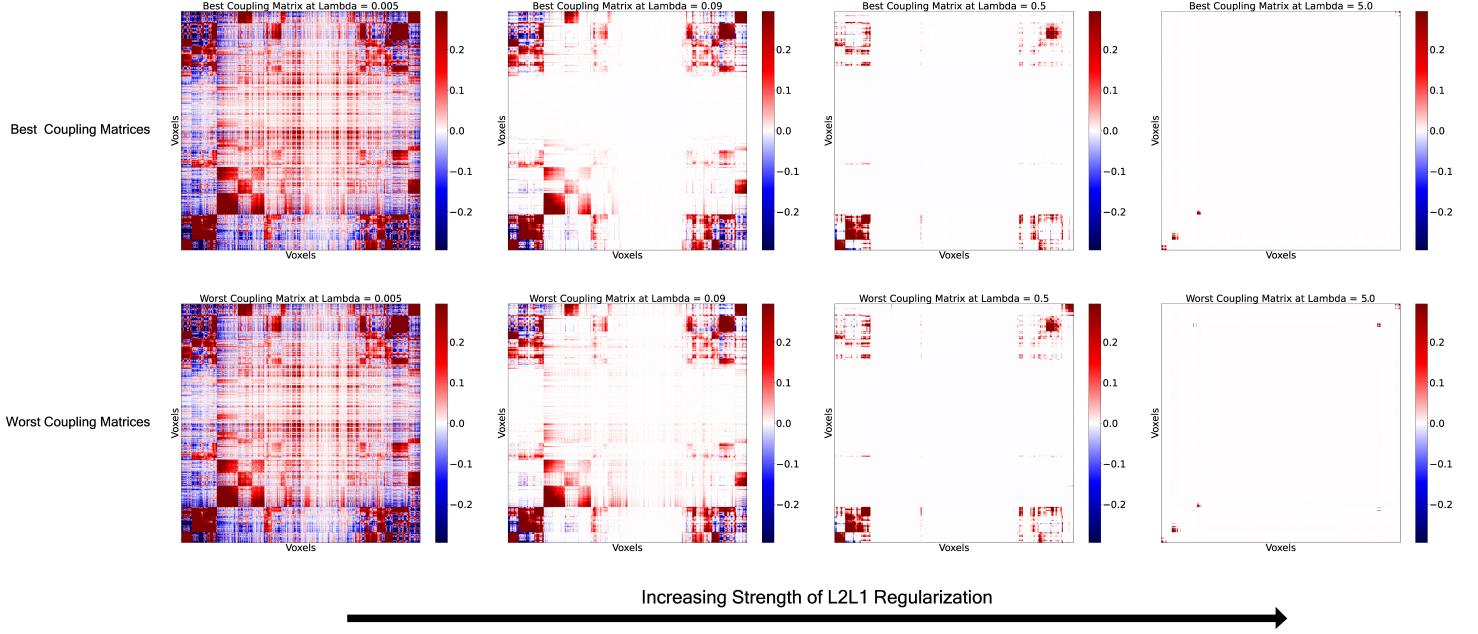


Figure 5: **Coupling matrices corresponding to good and bad cRBMs**

This figure shows the coupling matrices corresponding to good cRBMs as well as bad cRBMs at fixed λ values. We see that at a fixed λ , the general structure of both (the one corresponding to good cRBM as well as the one corresponding to a bad cRBM) coupling matrices is the same. The second thing that this figure shows is that the coupling matrices from cRBMs trained in the 2nd range of λ values i.e. [0.5, 5.0] are very very sparse. They are showing negligible functional connectivity among the voxels.

When an nRMSE value is calculated between two coupling matrices that lie in the same group, we get a low nRMSE value close to zero, hence the first peak near zero.

On the other hand, when an nRMSE value is calculated between two coupling matrices that belong to different groups, the nRMSE value is high, close to 1, hence the second peak near 1.0.

Therefore, the presence of two distinct groups of coupling matrices might be due to the fact that these matrices originate from cRBMs trained on two different ranges of λ values: [0.005, 0.1] and [0.5, 5.0]. Indeed, there was a significant gap between the λ values in the first range and those in the second range. In the second range, where λ values are higher, the weights of the cRBMs might have smaller magnitudes, as discussed in section 3.8. Conversely, in the first range with lower λ values, the weights of the cRBMs might have larger magnitudes.

Since a coupling matrix is constructed from the weights of a cRBM (see section 3.6), when the weights have larger magnitudes, the elements of the corresponding coupling matrix will have larger values. On the other hand, when the weights have smaller magnitudes, the elements of the corresponding coupling matrix will have smaller values.

Thus, when an nRMSE value is computed between two coupling matrices whose elements have small values, we will get a low nRMSE. Similarly, we will also get a low nRMSE value when it is computed between two coupling matrices whose elements have large values.

However, when one of the coupling matrices has lower valued elements while the other

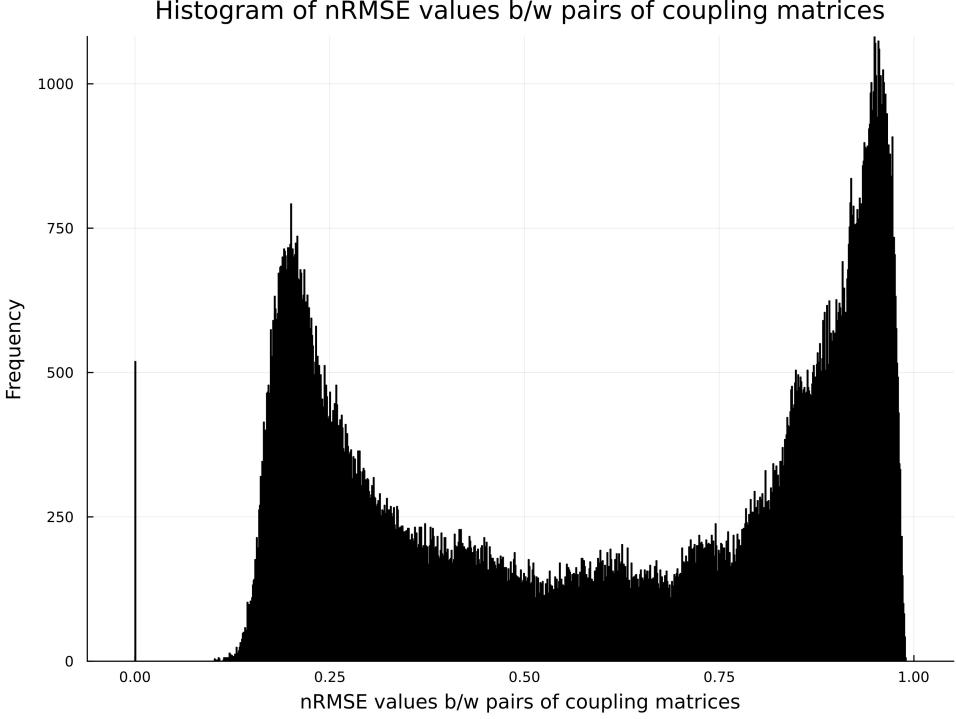


Figure 6: Two different groups of coupling matrices

The nRMSE values between all pairs of coupling matrices were computed. A histogram of these values can be seen in this figure. The histogram is showing two peaks. It suggests that there are two different groups of coupling matrices. When nRMSEs are calculated within the members of the same group, the nRMSEs are closer to zero. On the other hand when the nRMSEs are calculated between members from the opposite groups, the nRMSEs are closer to one. Thus we get two peaks, first one close to 0, the second one close to 1.

has large valued elements, or vice versa, we will get a high nRMSE value. This will explain the presence of the two peaks observed in the figure 6.

Testing the hypothesis - Principal Component Analysis and Clustering:

I first wanted to confirm whether there are indeed two different groups of coupling matrices or not. And, whether these two groups originate from the two different groups of cRBMs trained on two distinct ranges of λ values or not.

In order to do that I converted all of my coupling matrices into vectors. The coupling matrices were square matrices of size 1421×1421 . I converted them into $1421 \times 1421 = 2019241$ dimensional vectors. I basically treated each element of a matrix as a component of a 2019241 dimensional vector.

Now, I had 519 vectors, all of them living in a 2019241 -dimensional vector space. Next, I performed Principal Component Analysis (PCA) to project these high-dimensional vectors onto two dimensions. Figure 7 shows all of the 519 higher-dimensional vectors projected onto two dimensions, represented by colored dots.

I also performed K-means clustering on this data to obtain cluster centroids, as shown in Figure 7. Subsequently, I predicted the cluster associations of all the coupling matrices. All of the coupling matrices obtained from cRBMs trained in the 1st range of λ values were predicted to be the part of one cluster while all the coupling matrices obtained from

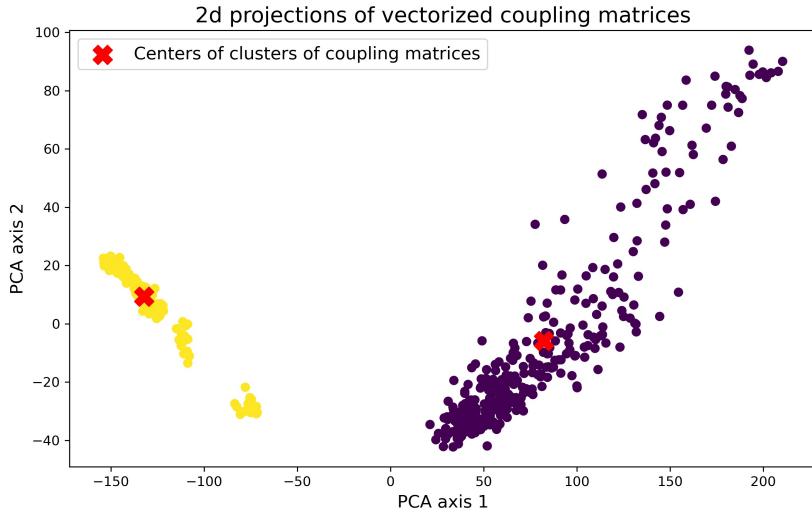


Figure 7: Two Separate clusters of coupling matrices

All of the 519 coupling matrices were converted into vectors and then projected onto 2 dimensions using PCA (principal component analysis). They are represented by colored dots in this figure. The projections suggest that our hypothesis was correct and that there are indeed two separate groups of coupling matrices.

Furthermore a clustering algorithm (Kmeans clustering) was also used to find the cluster associations of each coupling matrix.

All of the coupling matrices in the left hand side cluster were found to have underlying cRBMs, trained in the 2nd range of λ values i.e. $\lambda=[0.5,5]$

All of the coupling matrices in the right hand side cluster were found to have underlying cRBMs, trained in the 1st range of λ values i.e. $\lambda=[0.005,0.1]$

cRBMs trained in the second range of λ values were predicted to be a part of the second cluster, as shown in Figure 7.

This suggests that our hypothesis is true, with two distinct groups of coupling matrices corresponding to the two groups of cRBMs trained on the two different ranges of λ values.

However, it still remained to be shown that the reason behind all of this is the difference in the magnitudes of weights of cRBMs trained at two different ranges of λ values. In order to show that I used the following simple technique.

Testing the hypothesis - Sums of Coupling Matrices:

I simply summed all of the elements of each coupling matrix, for all of the 519 coupling matrices. I got a distribution of the sums. My idea was that if my hypothesis is correct then I should see two groups of peaks in the distribution of sums. Figure 8 shows the said histogram and it indeed contains two separate peaks.

The first group of peaks in the histogram shown in Figure 8 consists of exactly 199 sum values, which matches the number of cRBMs trained within the second range of λ values.

The second peak in the same histogram, Figure 8, contains exactly 320 sum values, which matches the number of cRBMs trained within the first range of λ values.

I further investigated these results by looping through the sums of coupling matrices

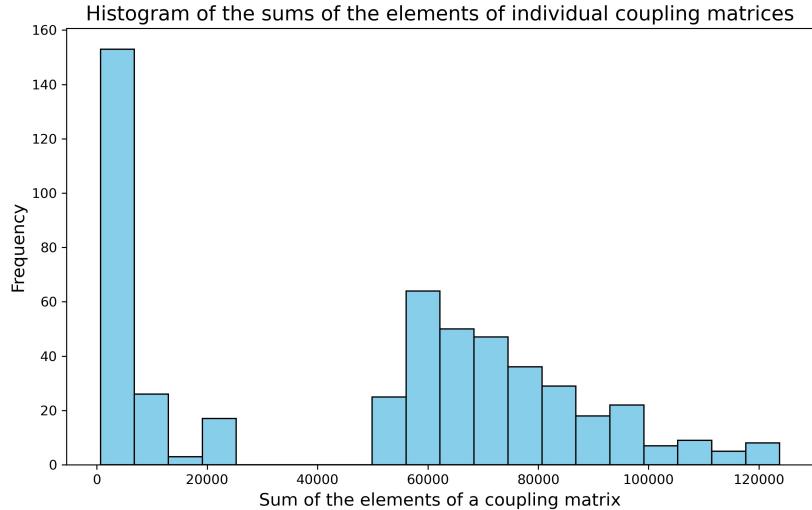


Figure 8: **Two Groups of Weights: High Magnitudes vs. Low Magnitudes**

Coupling matrices are built from the weights of underlying cRBMs. Consequently, coupling matrices whose underlying cRBMs had weights with higher magnitudes will have elements with higher values, and vice versa. The magnitudes of these weights are inversely proportional to the strength of regularization. Therefore, lower λ strengths will result in higher-value coupling matrices, while higher λ strengths will lead to lower-value coupling matrices.

In this figure:

All of the elements of individual coupling matrices were summed. The distribution of sums can be seen in this figure.

There are two groups of peaks.

The coupling matrices whose underlying cRBMs were trained in the second range i.e. $\lambda=[0.5,5]$ fall in the first group of peaks.

The coupling matrices whose underlying cRBMs were trained in the first range i.e. $\lambda=[0.005,0.1]$ fall in the second group of peaks.

obtained from the cRBMs trained in the first range of λ values. None of the sum values were smaller than 40,000. This means that they all belonged to the second group of peaks in Figure 8.

Similarly, I also looped through the sums of coupling matrices obtained from the cRBMs trained in the second range of λ values. None of the sum values were greater than 40,000. This means that they all belonged to the first group of peaks in Figure 8.

As depicted in Figure 8, the value of 40000 can be considered as the boundary separating the two groups of peaks.

Hypothesis Testing - Histogram of nRMSE values between pairs of coupling matrices for $\lambda=[0.005, 0.1]$

To further corroborate my initial hypothesis, I plotted the histogram of nRMSE values between pairs of coupling matrices for $\lambda = [0.005, 0.1]$. If my previous analyses are true, then when plotting a histogram of nRMSE values between pairs of coupling matrices for only one range of λ values, I should observe a single peak instead of two peaks. Indeed, a single peak was observed. See Figure 9.

Therefore, it is clear from all of this hypothesis testing that my original hypothesis was indeed correct.

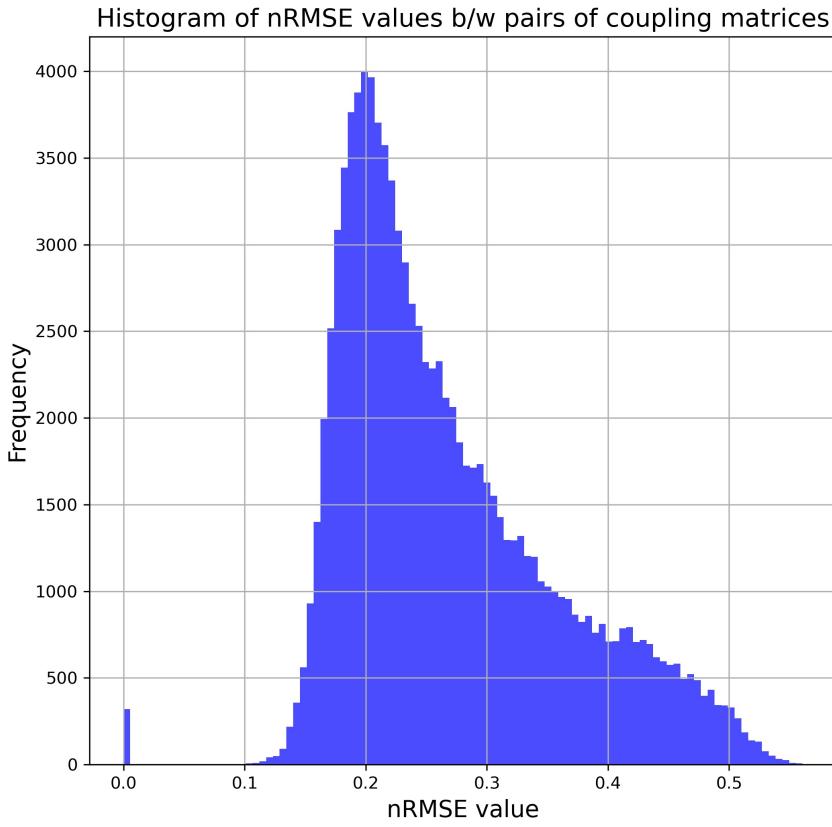


Figure 9: Coupling matrices are not very different from each other

The first result is that there is indeed a single peak for the first range of λ values!

The second result is that the average nRMSE between two coupling matrices is **0.27**, which is a small value. This means that, on average, the coupling matrices are not very different from each other.

The average nRMSE obtained from the distribution in Figure 9 is **0.27**. An nRMSE of 0.27 is not bad; it is a good result. This means that, on average, the coupling matrices produced by two different cRBMs are not very different from each other in the first range of λ values. This is a very important result that needs to be explored further. In the next section I explore it further.

4.1.4 Comparing Coupling Matrices

nRMSE as a function of λ

In the previous section, I wanted to determine the nRMSE between a pair of coupling matrices. In the process, I developed a deeper understanding of coupling matrices and their underlying cRBMs.

Now, based on this understanding, I predicted that the nRMSE between coupling matrices obtained from the same λ value would be smaller compared to the nRMSE between coupling matrices obtained from two different λ values. Additionally, the farther

apart two coupling matrices are on the λ axis, the greater the nRMSE between them would be.

To test this, I computed the nRMSE between the best coupling matrices. I fixed the first coupling matrix (B1) to be the best at $\lambda = 0.09$. The remaining coupling matrices were the best at different λ values: B2 at 0.06, B3 at 0.04, B4 at 0.02, and B5 at 0.005.

I then computed the following nRMSE values: nRMSE(B1,B2), nRMSE(B1,B3), nRMSE(B1,B4) and nRMSE(B1,B5). Where “nRMSE(B1,B2)” is the nRMSE between the coupling matrices B1 and B2.

The results are shown in figure 10

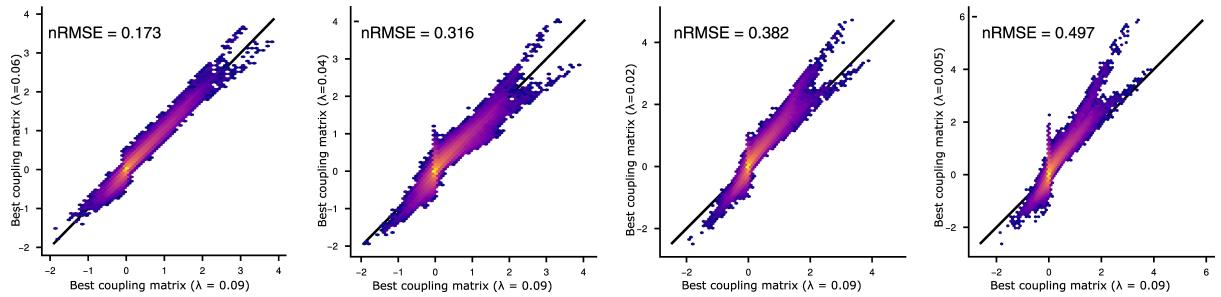


Figure 10: The greater the λ difference, the larger the nRMSE

The nRMSE between the best coupling matrix at $\lambda = 0.09$ and the best coupling matrices at $\lambda: 0.06, 0.04, 0.02$, and 0.005 was calculated.

The nRMSE increased as a function of the difference in λ values. Specifically, the nRMSE was larger for coupling matrices where the difference in their λ values was greater, and smaller for coupling matrices where the difference in λ values was smaller. In summary, the greater the λ difference, the larger the nRMSE.

The results are as follows. The nRMSE between coupling matrices is indeed directly proportional to the difference in their corresponding λ values. This relationship is straightforward to interpret. As discussed in section 4.1.3, the elements of coupling matrices from different λ values will have different magnitudes. The larger the λ difference, the greater the difference in the magnitudes of the elements, resulting in a larger nRMSE between the corresponding coupling matrices.

All of the 5 cRBMs underlying these coupling matrices, had an nRMSE vector with a norm smaller than 0.55. Thus the performance of all of these cRBMs is similar. Therefore the difference in the corresponding coupling matrices, indeed comes from the differences in the magnitudes of underlying weights and not due to the differences in the performance.

nRMSE at fixed λ values

Next, I wanted to understand what happens when comparing the best and worst coupling matrices at the same λ . Is the nRMSE then constant? i.e. Do we get similar nRMSE values if we perform these computations at different λ values?

To investigate this, I computed the nRMSEs between the best and worst coupling matrices at λ values of 0.005, 0.02, 0.04, 0.06, and 0.09. The results are shown in Figure 11.

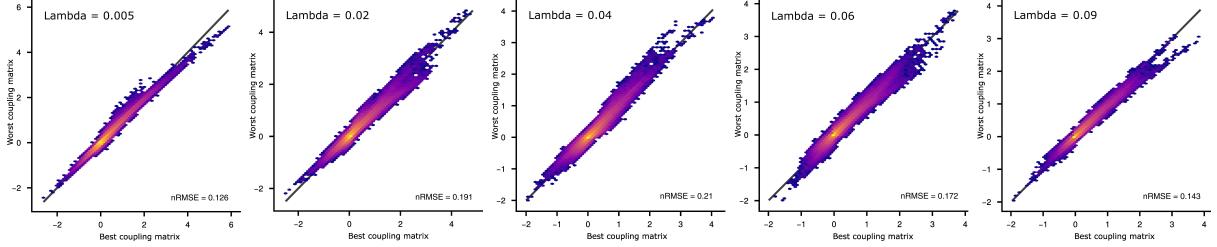


Figure 11: **nRMSE at fixed λ is almost constant**

The nRMSE between the best and the worst coupling matrices at a fixed λ was calculated. This process was then repeated for different λ values.

The nRMSE consistently turned out to be low and nearly the same in all cases.

This suggests that the main differences in the coupling matrices arise from variations in λ values rather than from how well or poorly they reproduce the brain activity statistics.

These results are surprising. Even when comparing the best and worst coupling matrices, the nRMSE at a fixed λ value is almost constant, ranging from 0.1 to 0.2. This suggests that all cRBMs, regardless of the λ value, may be producing the same coupling matrix.

To be more precise, voxel couplings may be consistent across all cRBMs: voxels negatively coupled in one cRBM are negatively coupled in all others, voxels positively coupled in one cRBM are positively coupled in all others, and voxels that are not coupled remain uncoupled across all cRBMs. The only difference may be the strength of the coupling, which varies due to differences in the magnitudes of the underlying weights from which the coupling matrices are derived.

Coupling matrices at different λ values

In order to confirm this hypothesis, I looked at the coupling matrices from different λ values, see figure 12

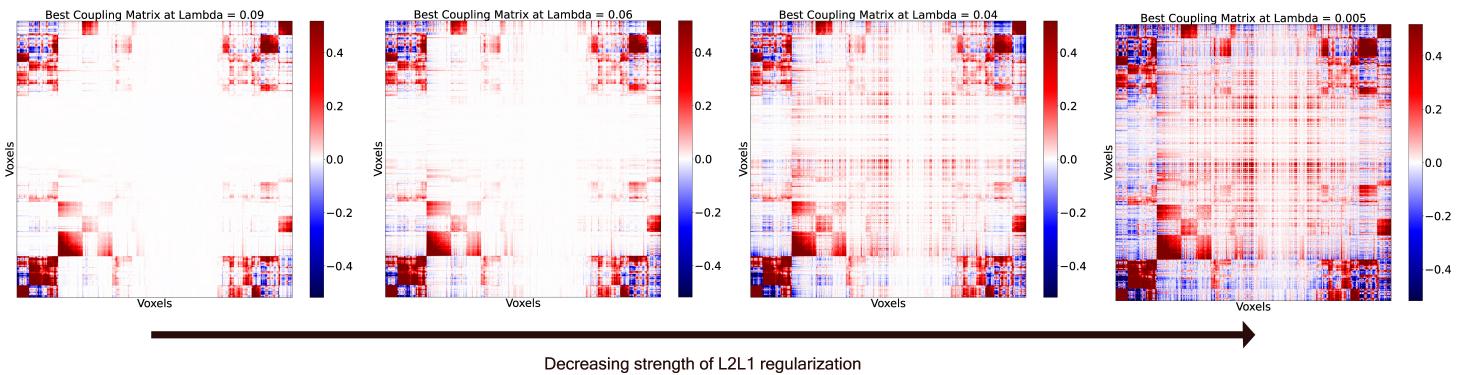


Figure 12: **Coupling matrix is conserved across cRBMs**

Coupling matrices from different λ values were plotted side by side. A commonly shared structure was found among all coupling matrices, suggesting that voxels are functionally grouped into voxel assemblies.

The only difference observed was in the sparsity of the coupling matrices. Those at higher λ values were more sparse, indicating that they were well regularized and not capturing the noise in the data.

The results are remarkable. We observe a consistent structure across all coupling matrices. Specifically:

- If a pair of voxels is positively coupled in one matrix, it tends to be positively coupled in all matrices.
- If a pair of voxels is negatively coupled in one matrix, it is generally negatively coupled in all matrices.
- If a pair of voxels is uncoupled in one matrix, it typically remains uncoupled in all matrices.

The primary distinction is that the coupling matrices at higher λ values are more sparse. This is good, as it indicates that the underlying cRBMs are well regularized and not overfitting to the noise in the data.

Let's try to understand the physical implications of these findings. It means that all of the cRBMs consistently reveal that certain voxels in the zebrafish brain are well connected with other voxels. Which suggests the presence of assemblies of voxels in the zebrafish brain that are functionally interconnected.

5 Conclusion

Results:

I trained multiple compositional Restricted Boltzmann Machines (cRBMs) on voxelized data from the brain of a zebrafish larva. The objective for the cRBMs was to learn the brain statistics of the larva. While some cRBMs accurately reproduced these statistics, others did not perform as well.

Nevertheless, even the cRBMs that were less successful in reproducing the brain statistics produced coupling matrices similar to those generated by the more successful models. This conservation of coupling matrices suggested functionally connected voxel assemblies within the zebrafish larva's brain.

Difficulties Encountered:

The first challenge was to determine the optimal value of the L2L1 regularization parameter λ . To address this, I created Figure 3.

Other significant difficulties were related to interpreting the results and identifying patterns. To address these challenges, I employed various machine learning and statistical tools, including identity plots, normalized root mean square errors (nRMSEs), histograms, principal component analysis, K-means clustering, and heat maps of coupling matrices.

What's Next?

While performing these analyses, I developed a deeper understanding of cRBMs and the brain of zebrafish larvae. Now equipped with this insight, I plan to investigate whether this functional connectivity is conserved across multiple zebrafish larvae. Specifically, I aim to determine if training cRBMs on data from different larvae reveals the same functional connections among voxels. If consistent functional connectivity is observed, the next step will be to explore the extent to which structural connectivity can be inferred from this functional connectivity.

6 Annex

6.1 good Vs bad cRBM visual comparison

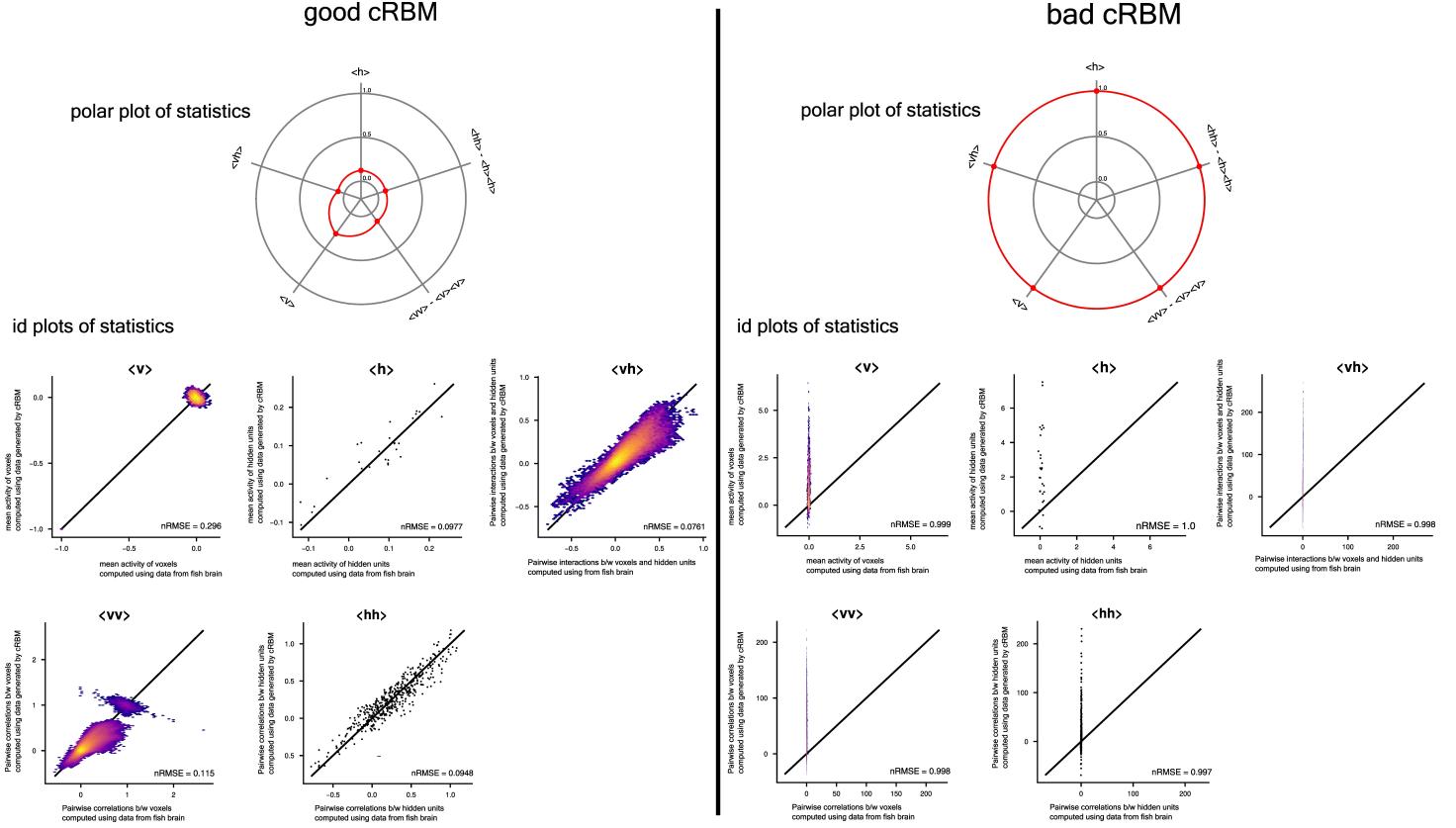


Figure 13: Comparison of a good cRBM with a bad cRBM

The polar plots show the nRMSE between statistics computed using data from the brain of fish vs the data generated by the cRBM. We see that for the bad cRBM the nRMSE for all of the 5 statistics is very close to one, except for the $\langle h \rangle$ statistic. For the $\langle h \rangle$ statistic is greater than 1. This is the reason why this cRBM was classified as a bad one.

The id plots of statistics show that the activity generated by the bad cRBM is not at all similar to the brain activity of the zebra fish.

References

- [1] Misha B Ahrens et al. “Whole-brain functional imaging at cellular resolution using light-sheet microscopy”. In: *Nature methods* 10.5 (2013), pp. 413–420.
- [2] Natalia Beiza-Canelo et al. “Magnetic actuation of otoliths allows behavioral and brain-wide neuronal exploration of vestibulo-motor processing in larval zebrafish”. In: *Current Biology* 33.12 (Sept. 2023), 2438–2448.e6.
- [3] György Buzsáki. “Neural syntax: cell assemblies, synapsembles, and readers”. In: *Neuron* 68.3 (2010), pp. 362–385.
- [4] Howard Eichenbaum. “Barlow versus Hebb: When is it time to abandon the notion of feature detectors and adopt the cell assembly as the unit of cognition?” In: *Neuroscience letters* 680 (2018), pp. 88–93.
- [5] George L Gerstein, Purvis Bedenbaugh, and Ad MHJ Aertsen. “Neuronal assemblies”. In: *IEEE Transactions on Biomedical Engineering* 36.1 (1989), pp. 4–14.
- [6] Marcus Ghosh et al. “Multimodal units fuse-then-accumulate evidence across channels”. In: *BioRxiv* (July 2023).
- [7] Kenneth D Harris. “Cell assemblies of the superficial cortex”. In: *Neuron* 76.2 (2012), pp. 263–265.
- [8] Kenneth D Harris. “Neural signatures of cell assembly organization”. In: *Nature reviews neuroscience* 6.5 (2005), pp. 399–407.
- [9] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.
- [10] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786 (2006), pp. 504–507.
- [11] Antoine Hubert et al. “Random-access two-photon holographic optogenetic stimulation combined with brain-wide functional light-sheet imaging in larval zebrafish”. In: *Advances in Microscopic Imaging IV* 12630 (Sept. 2023), pp. 11–14.
- [12] Geoffrey Migault et al. “Whole-brain calcium imaging during physiological vestibular stimulation in larval zebrafish”. In: *Current Biology* 28.23 (2018), pp. 3723–3735.
- [13] Yu Mu et al. “Glia accumulate evidence that actions are futile and suppress unsuccessful behavior”. In: *Cell* 178.1 (2019), pp. 27–43.
- [14] Günther Palm et al. “Cell assemblies in the cerebral cortex”. In: *Biological cybernetics* 108 (2014), pp. 559–572.
- [15] Thomas Panier et al. “Fast functional imaging of multiple brain regions in intact zebrafish larvae using selective plane illumination microscopy”. In: *Frontiers in neural circuits* 7 (2013), p. 65.
- [16] Thijs L van der Plas et al. “Neural assemblies uncovered by generative modeling explain whole-brain activity statistics and reflect structural connectivity”. In: *eLife* 12 (Jan. 2023). Ed. by Peter Latham and Laura L Colgin, e83139. ISSN: 2050-084X. DOI: 10.7554/eLife.83139. URL: <https://doi.org/10.7554/eLife.83139>.
- [17] Gokul Rajan et al. “An analysis pipeline to compare explorative locomotion across fish species”. In: *STAR Protoc.* 3.4 (Nov. 2022).

- [18] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).
- [19] Marcus A Triplett et al. “Model-based decoupling of evoked and spontaneous neural activity in calcium imaging data”. In: *PLoS computational biology* 16.11 (2020), e1008330.
- [20] Jérôme Tubiana, Simona Cocco, and Rémi Monasson. “Learning compositional representations of interacting systems with restricted boltzmann machines: Comparative study of lattice proteins”. In: *Neural computation* 31.8 (2019), pp. 1671–1717.
- [21] Jérôme Tubiana and Rémi Monasson. “Emergence of compositional representations in restricted Boltzmann machines”. In: *Physical review letters* 118.13 (2017), p. 138301.
- [22] Gilles C Vanwalleghem, Misha B Ahrens, and Ethan K Scott. “Integrative whole-brain neuroscience in larval zebrafish”. In: *Current opinion in neurobiology* 50 (2018), pp. 136–145.
- [23] Sébastien Wolf et al. “Sensorimotor computation underlying phototaxis in zebrafish”. In: *Nature communications* 8.1 (2017), p. 651.
- [24] Sébastien Wolf et al. “Whole-brain functional imaging with two-photon light-sheet microscopy”. In: *Nature methods* 12.5 (2015), pp. 379–380.