Exploring Ensemble Coding in Retina

by

Christopher J. Warner II

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Biophysics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Friedrich Sommer, Co-chair
Professor Marla Feller, Co-chair
Professor Bruno Olshausen
Professor Frederic Theunissen

Summer 2019

Exploring Ensemble Coding in Retina

Abstract

Exploring Ensemble Coding in Retina

by

Christopher J. Warner II

Doctor of Philosophy in Biophysics

University of California, Berkeley

Professor Friedrich Sommer, Co-chair

Professor Marla Feller, Co-chair

Computational models founded on the prevailing paradigm of retinal processing, while able to replicate the coarse structure of responses to white noise stimulus, fail to replicate responses to natural stimuli. The textbook view of retina, which posits independent filters that decorrelate stimulus features, reduce representational redundancy [**barlow1961**] and encode local features in retinal ganglion cell (RGC) spike rates, leaves severe puzzles, unexplained about observed retinal anatomy and activity. We present, here, an addendum to the prevailing paradigm hypothesizing that *perhaps*, the retina reduces uninformative correlations in stimulus with outer layers, as claimed, in order to reintroduce informative correlations, observed in RGC responses to ethologically relevant stimuli, with the circuitry in the inner retinal network (bipolar, amacrine, ganglions). Rather than strict independent coding and redundancy reduction, a notion which Barlow himself amended [**barlow2001**], we explore ensemble coding in retina and what information beyond the traditional view might exist in the retinal code. This work is in two related, yet independent parts. First, we develop a proof-of-concept abstract computational model of image segmentation using phase coding in the retina, hypothesizing that fine-time correlations in spike trains are induced by phase interactions influenced by the visual stimulus and that these fine-time correlations, informative about segments in an image, are multiplexed into spike-trains along with rate-coded local stimulus features. Following, we present a latent variable statistical model that aims to detect cell assemblies, or groups of cells that fire are often co-active, possessing fine-time correlations irrespective of their source and apply it to retinal spike-trains responding to both white noise and natural movie stimulus. The work presented here is novel and controversial and therefore, worth a read.

hello world.

Contents

Contents **ii**

List of Figures **iii**

List of Tables **xi**

Introduction **1**

1 Image Segmentation in Retina **3**
1.1 Background 3
1.2 Methods 6
1.3 Results 12
1.4 Discussion 26

2 Probabilistic Cell Assembly Model **29**
2.1 Background 29
2.2 Cell Assembly Model 31
2.3 Model Training 34
2.4 Model Validation on Synthetic Data. 37
2.5 Retinal Data Exploration 48
2.6 Discussion 64

Conclusion **67**

A Image Segmentation Supplement **69**
A.1 Optimal Gaussian RF size 69
A.2 Motivating modularity 71

B Cell Assembly Model Supplement **78**

List of Figures

1.1 **Image Segmentation Model:** (a) Input image with superimposed retinal receptive fields (dashed cyan circles). (b) Network of retinal neurons. The neural firing rates r_i represent local contrast in the receptive fields. The phase interactions K_{ij} are displayed by the links between neurons. Line thickness represents the strength of the interaction which is set by the similarity of local features. Recurrent propagation in the network produces the phase structure ϕ_i of the periodic spike trains. (c) Resulting spike trains. Information about local features is represented in firing rates and segmentation is represented in phase structure. 5

1.2 **Performance and benchmarking:** Input image patch and associated human drawn ground truth boundaries (gT) provided by BSDS is displayed in the green box. The operations performed by model are displayed in the blue box. Other steps of model evaluation are illustrated in the remainder. (a) Filtering the raw image patch with a Gaussian kernel ($\sigma = 1$). (b) Phase relaxation in the network (Fig. 1.1b) produces a phase map. (c $_1$) Spatial gradient operation ($\delta/\delta r$) and normalization resulting in probabilistic boundary map ($pb \in [0, 1]$). (d) Thresholding pb map at several values yielded binary boundary (bb) maps. (e) Match set was computed for each bb-gT pair at different distance tolerances, d_t. (f) Precision, recall and F-measure were computed by ratios of boundary pixel sets. (c $_{2,3}$) To assess the performance of network models relative to baselines, we repeated steps (c) - (f) on Gaussian RF and image pixels independent sensors models, comparing F-measures by subtraction. 7

1.3 **Hyper-parameter optimization:** Network neighborhood graph structure r_M and coupling spring-constant scaling k_s are important meta parameters of the algorithm, discussed in Sections 1.2 and 1.2 respectively. We plot mean and standard deviation across 500 image patches of ΔF-measure relative to Gaussian RF independent sensors for the 2D topographic modularity network. Colors indicate pixel distance tolerances d_t (see Fig. 1.2 for explanation). *Left* panel shows performance at three k_s values, with r_M fixed at optimal. *Right* panel shows performance at four r_M values, with k_s fixed at optimal. Fig. 1.4 shows the effects of the different parameters on a single example image patch. 11

1.4 **Effect of hyper-parameters, single image patch example:** Probabilistic boundary maps shown for resulting phase distribution from TM 2D method for combinations of 3 k_s (rows) and 4 r_M (columns) hyper-parameters. 12

1.5 **Modularity null models & space**: In the null model of Newman's modularity **[newman2006]** *(panel a)* the average weight between nodes i and j is proportional to the product of their node degrees ($D_i \cdot D_j$). The topographic modularity's null model *(panels b & c)* additionally includes a distance-dependent factor, R_{ij}, which is the average edge weight between all node pairs in the graph separated by the same distance that separates nodes i and j. *Panel b* illustrates R_{ij} for a schematized 1D graph, shown with edges colored based on distance between the nodes they connect. Inset plot shows geometric factor in the topographic null model. Each term in $R_{ij}^{(1D)}$ is an off-diagonal sum in the adjacency matrix. *Panel c* shows the mask associated with a single geometric distance in a 2D image. Here $R_{ij}^{(2D)}$ at 1 pixel separation has a complex structure in the Adjacency matrix for even the simple binary image shown in the inset. 13

1.6 **Null model consistency**: *Top row* from left to right shows image patch, the adjacency (black) constructed from the patch with $r_{max} = 5$, and null models for modularity (blue), 1D topographic modularity (green) and 2D topographic modularity (red), with colorbar indicating edge weight. Models represented by line colors in plots as well. *Center plot* shows average node degree (row sums in each matrix) sorted by strength in adjacency. *Bottom plot* shows average edge weight as a function of distance in the image plane. 15

1.7 **Modularity performance comparison:** Each scatter point represents one image patch. Newman's modularity (M) in blue, 1-dimensional topographic modularity (TM 1D) in green and 2-dimensional topographic modularity (TM 2D) in red. Points above the unity line indicate image patches with improved image segmentation with network phase relaxation over-and-above Gaussian RF independent sensors. P-values quantify the difference between F-measure distribution across 500 image patches before and after network computation. 16

1.8 **Spectral methods vs. Kuramoto Net Examples:** Two example image patches (top two rows and bottom two rows) show probabilistic boundaries found by different network (TM, M, AA, GL) and baseline models (ImPix, GaussRF, ISO). Network models are segmented using eigen-methods (1st and 3rd row) and Kuramoto Net phase relaxation (2nd and 4th row). Qualitatively, boundaries found with spectral methods are less crisp and more localized than those found with Kuramoto Net phase relaxation. 17

1.9 **Spectral methods vs. Kuramoto Net Statistics:** F-measure computed across 500 image patches, mean and standard error errorbars. Colors indicating different network and baseline models are used consistently throughout this paper. Circles indicates that F-measure for each image patch taken for maximum matching GT and x's shows mean value across all GT's. Network models built with Gaussian RF features are segmented by the best combination of the top 3 eigenvectors on the x-axis and by the phase distribution after Kuramoto Net relaxation on the y-axis. The dashed unity line indicates equal performance and the independent sensors baseline models (magenta and cyan) do not deviate from it. 18

1.10 **Gaussian RFs improves segmentation**: Performances of 4 anisotropic diffusion and 3 baseline models are compared using raw image pixel features and Gaussian RF features, *center and right columns respectively*, lines representing average and bars standard error across 500 BSDS image patches. (*a*) Colors indicate different models and line styles indicate ground truth comparison as in Fig. 1.9. (*b*) Optimal spread, σ, of Gaussian RF's chosen by maximizing change in F-measure relative to the independent raw pixels baseline model, ΔF_i, averaged across all image patches. Recall d_t is the "distance tolerance" when computing the pixel match set, Fig. 1.2. Optimal performance for all d_t values obtained for Gaussian RF $\sigma = 1$. (*c*) F-measure and (*d*) ΔF_i when models receive raw image pixels as features. (*e*) F-measure and (*f*) ΔF_i when models receive Gauss RF activation as input features. 19

1.11 **ΔF-measure model comparison:** Violin plots show ΔF_G distribution with moments of ΔPrecision, ΔRecall and ΔF-measure distributions across 500 image patches in red, blue and green, respectively. ΔF relative to Gaussian RF Independent Sensors model. Optimal hyper-parameters(r_M,k_s), statistical significance, p-values and distribution moments indicated above each method. Performance of ISO, TM-1D and TM-2D models relative to Gauss RF are statistically significant, as determined by Mann-Whitney U (aka rank-sum) test. 21

1.12 **Precision & Recall model comparison:** (a) F-measure, (b) precision and (c) recall across 1000 image patches for Gaussian RF independent sensors baseline model and 4 network models with optimized parameters and $d_t = 2$. Distribution μ and σ denoted above. Note colors same as in Figs. 1.9&1.10. 22

1.13 **ΔPrecision and ΔRecall with TM-2D model:** In *panel a*, arrows show change in P & R for 467 image patches where network increased F-measure. Arrow tails indicate values before network relaxation and heads values after. Surrounding are distributions showing P,R,F before network (in cyan) and after (in blue). *Panel b* shows the same for 33 image patches where network decreased F-measure. Distributions before network in magenta. 22

1.14 **Representative image patches:** Each row shows one example image patch ordered by change in F-measure between Gaussian RF independent sensors baseline and TM models (indicated on left). Columns show image pixels, gT boundaries and pb maps obtained from raw pixels, Gaussian RF and 5 network models. Mean F-measure value across all gT's noted below each pane is red if $\Delta F_G > 0$. 24

1.15 **Examples of TM 2D model performance:** *Top* panel scatters F-measure in Gaussian RF independent sensors model vs. Δ-F after 2D topographic modularity network phase relaxation. Out of 500 total image patches, 467 show positive improvement. Best fit line to scatter points in magenta. Colored numbers indicate randomly sampled image patches (shown in bottom panel) where Δ-F performance is best (#1-4), average (#5-8) and worst (#9-12). *Bottom* panel shows image patches with best matching ground truth boundaries, in black. Yellow points indicate pixels found to be boundaries both by the Gaussian RF independent sensors model and the topographic modularity network model. Cyan number and points indicate F-measure under Gaussian RF model and boundaries found only by it. Red number and points indicate Δ-F after TM 2D network phase diffusion and boundaries found only by TM-2D. Note that image patches are shown at 1/2 contrast to highlight boundaries found. 25

1.16 **Two examples of cartoonization:** Original images on left and resulting phase of TM-2D network computation on right 27

2.1 **Schematic of Cell Assembly model**: Observed spikes within spike-words $\vec{y}$ arise from two sources. First, each cell has some probability of firing without any cell assembly activity, expressed by N-vector $\vec{P}_i$. The second cause of cell activity is cell assembly activity, expressed by the $\bar{P}_{ia}$ matrix and $\vec{z}$. Finally, the scalar Q parameter sets a binomial prior on the activity in the latent variable, $\vec{z}$. 32

2.2 **Fitting synthetic model to spike-word moments:** Comparison of spike-word moments from synthetic data fit to natural movie responses in red, retinal responses to white noise in green, and natural movie in blue. *Top row from left to right* shows probability density functions for spike-word length, $|\vec{y}|$, average single-cell activity, $\langle y_i \rangle$. and, pairwise cell coactivity, $\langle y_i \cdot y_j \rangle$. *Bottom row* shows quantile-quantile (QQ) plots - a pair of cumulative density function plotted against each other. See legend, left plot. *QQ* values measure average deviation from the unity line, larger values indicating differences in distributions. 42

2.3 **Cosine similarity between models:** for synthetic natural movie responses (a) and real retinal responses to white noise(b). Within each panel, left column shows matrix of *cs* values between all cell assembly pairs across a pair models. Top, with arbitrary order due to learning algorithm stochasticity. Bottom, with cell assemblies matched across models based on *cs*. Right of each panel shows the pair of $\bar{P}_{ia}$ matrices with columns, cell assemblies, aligned to maximize *cs* across all matched pairs. *cs* for each match is shown in blue bars in panel bottom right. Panel b illustrates the necessity of a null model for the *cs* quality metric. Although both panels illustrate similar *cs* values after CA matching (0.75 vs 0.72), the improvement from null model before matching (0.13 vs 0.58) is quite different. 44

2.4 **Modelling synthetic responses to natural movie vs. white noise stimulus:** Cosine similarity for 6 learned models and ground truth (GT) for synthetic natural movie responses (a) and synthetic white noise responses (b). Value and color in each box indicate improvement above null model after CA matching. Details for box at intersection of Mod2 and Mod1B (0.61) in panel a are illustrated in Fig 2.3a. Each model's overlap with the GT is roughly correlated with the model's overlap with other models. This correlation is further unpacked for Mod2 and Mod1B in panel a and for Mod2 and Mod1 in panel b in Fig 2.5a and 2.5b respectively. 45

2.5 **CA structure in multiple models matches GT**. In panel a, for synthetic natural movie responses and in panel b, for synthetic white noise responses. Within each panel, left plot shows *cs* between each matched CA in a model pair on the x-axis and between each model's CA and its matching GT CA on the y-axis. One model is in blue, the other in green, with red 'o' indicating smaller *cs* between model and ground truth. Black diamond indicate where two models agree on same ground truth CA. Larger blue and green '+' show error bars mean and std of each model's CA population. Right plot in panel shows distribution of points from the unity line. Mass near zero indicates CA triplets that share similar *cs* values, that is where structure in the GT is robustly found by multiple models and structure found by only one model is not in the GT. The CA match is stronger between model pair as well as between ground truth and each individual model with natural movie vs white noise model. Pearson correlation coefficients for three scatter groups in left plots are shown in legend on the right. 46

2.6 **Comparing Models learned on synthetic data fit to different stim responses:** natural-movie-like (a) and white-noise-like (b) responses. Within each panel, *top boxes* show $\bar{P}_{ia}$ matrix in GT and learned model, columns indicating CAs and y-axis, cells. *Right bottom* shows the signed error between GT and learned $\bar{P}_{ia}$'s. Note sigmoid colorbar to accentuate small differences. Unfilled boxes in learned $\bar{P}_{ia}$ and error show active cells in GT CAs. *Left bottom* scatter plot shows Q, in green, and $\vec{P}_i$, in blue, parameters in learned model (y-axis) vs. GT (x-axis). Points near $y = x$ indicate correctly learned parameters. Parameter initialization are shown in gray. Cyan points in *left middle* show the number of times each CA was inferred across all data after model learning. In the white noise model, fewer cell assemblies are learned as indicated by more blue and red in the signed error. The model also relies on noisier $\vec{P}_i$ parameters (blue o's further from 1 and below $y = x$ line) to account for increased spike-word variability. This model is more difficult to learn because cell assembly participation is weaker, lighter gray squares in GT $\bar{P}_{ia}$. 47

2.7 **PSTHs and RFs of Off-Brisk Transient RGC:** Color indicates #Trials in which an offBT neuron (y-axis) spiked during a 1ms interval of stimulus presentation (x-axis). *Top*, responses to white noise. *Bottom*, responses to natural movie. Geometric RF relationships in visual space maintained in cell ordering, *panel b*. 48

2.8 **Models trained on responses to different stimuli:** P_{ia} matrices for models trained on retinal responses to white noise stimulus*(a)*, natural movie stimulus *(b)* and GLM simulated responses to same natural movie stimulus *(c)*. Cell ID on y-axis, CA ID on x-axis. Yellow indicates cell membership in CA, $p(y_i = 1|z_a = 1) \sim 1$. 49

2.9 **Repeatable structure in responses to different stimuli:** Similarity of CA membership structure across model pairs for six models trained on each of three spike-word data sets. Note relationship to Fig. 2.4. Models trained on white noise retinal in responses *(a)*, natural movie responses in *(b)* and GLM simulated responses to same natural movie stimulus in *(c)*. *Within each panel*, matrix off-diagonal elements shows average Δcs (relative to null without CA matching) between all matched CA pairs within a model pair. Here diagonal value indicates average between a model and all other models, i.e. the average across a row. Numbers on left show average conditional probability computed on hold out set of half of all spike-words, i.e., cross-validation. Vector on right shows average change from initialization for all CAs in model, defined as $1 - cs$. 50

2.10 **Membership Crispness C_M examples:** CAs ranging from diffuse *(a)* to crisp *(c)*. In each panel, numbered ovals represent cell RFs with redness indicating strength of CA membership. Legend above. Corresponding column in P_{ia} shown on right. 52

2.11 **Cross-validation Robustness R_X example:** *(a)*. Raster plot time vs. trial. CA activity in red and member cell spikes in other colors. *(b)*. PSTH from shown CA on top line and PSTHs from matched CAs in other models below, (model, CA id) indicated on the left. PSTHs normalized with total number of activations in white on right. *(f)*. Columns of P_{ia} matrices for shown CA, on left, and its counterparts in other models. *(c)*. Membership vs. temporal cosine similarity for each CA in model with matching CAs in 5 other models, averaged across 5 matches. For clarity, $\langle cs_M \rangle_X$ on x-axis computed from panel f and $\langle cs_\tau \rangle_X$ on y-axis computed from panel b. *(d)*. Cross-validation Robustness R_X vs. Membership Crispness C_M metrics for all CAs in one model. CA shown here highlighted with red "50" in panels c & d. *(e)*. Cell RFs, redness indicates CA membership strength, P_{ia} value, and outline colors match raster colors in panel a. 53

2.12 **Cell-type Heterogeneity H example:** Two cell assemblies with high heterogeneity. *(a)* Single CA comprised of offBT *(red, left)* and onBT *(blue, right)* cell types. Ovals indicate cell RFs and color intensity indicates membership strength, i.e., P_{ia} value. Legend above. 54

2.13 **Difference from Null $\Delta P(y)_{null}$ example:** PSTH of z_a in red. In green, $p(\vec{y})$ under GLM null model for all $\vec{y}$ observed when $z_a = 1$. Binning curves at [1,10,50,100] ms yields ΔPy = [.79,.78,.61,.51]. In blue, KL-divergence between N-dimensional multivariate Bernoulli distributions of $p(y_i)_{null}$ and $p(y_i|z_a = 1, z_{a'} = 0)$, discussed further in supplemental section B. 55

2.14 **Statistics of CA metrics in typical model** trained on natural movie responses from 94 [offBT,onBT] RGCs. *(a).* Sorted CA sizes, colors consistent throughout. *(b).* P_{ia} matrix, white line indicates break between population offBT below and onBT above. *(c).* Membership Crispness C_M vs. Cross-validation Robustness R_X, each point a CA. Cross shows μ & σ across all CAs. Note that vertical color gradient indicates correlation between C_M and CA size. *(d).* Difference from Null ΔPy with 1ms binning vs. with 100ms binning. *(e).* Heterogeneity H metric histogram. 56
2.15 **CAs Membership Crispness examples:** From 3 separate models trained on [offBT] natural movie responses. In each panel, colored ovals RFs of cell members from 3 separate CAs with similar C_M values. Small inset scattters R_X on x-axis vs. C_M on y for all CAs in each model with shown CAs highlighted in matching color. 57
2.16 **Two elongated, yet crisp sample CAs:** RFs from CAs z_{16} and z_{27} in panels *(a) & (b)* are same as green and blue ellipses in Fig. 2.15b. In *(c)*, crispness and robustness scattered for all CAs in model with two shown highlighted in red. In *(d)*, ΔPy at 1ms vs 100ms scattered in *(a)*. Temporal response traces of CA PSTH *in red* and GLM $\langle p(\vec{y}) \rangle$ predictions *in green* shown in *(e) & (f)*. 58
2.17 **Six robust CAs with high ΔPy in one model:** Each surrounding panel shows RFs of [offBT] cells with redness reflecting strength of membership in CA. CA id in colored number in bottom left of each panel. Scatter plots in bolded box show metric values for each CA in model with shown CAs highlighted in colored numbers matching id. 59
2.18 **Heterogeneity of CAs across cell-types:** *(a).* Model trained on natural movie responses from 55 offBT and 43 offBS RGCs. *(b).* Model trained on responses from 55 offBT and 39 onBT RGCs. In each panel, P_{ia} matrices shown on top with columns indicating CAs. Dashed white line shows boundary between offBT cells below and other type above. Bottom shows histogram of H metric values for all CAs in model. Black arrow indicates consistent difference across multiple trained models. 60
2.19 **Heterogeneous [offBT,onBT] CAs:** Four strongly heterogeneous CAs learned within a single model are shown on bottom. Each boxed plot represents one CA, red ovals showing RFs and participation of offBT cells and blue ovals, onBT cells. Scatter plots above show R_x vs. C_M on right and ΔPy at 1ms and 100ms time resolutions on left. Shown CAs are highlighted in color. Green labeled CAs are significantly different from GLM predictions and red labeled CAs are not. . . . 61
2.20 **CA z68 and stimulus:** *Not* significantly different from GLM null model. Within each panel, *Center top* shows cell RFs of offBT (red) and onBT (cyan) member cells. Box matches image dimensions approximately. *Bottom* shows PSTH of CA activation. *Right top* shows stimulus at time of CA activation, blue peak in PSTH. *Left top* shows stimulus $\sim$ 333ms prior to CA activation. 62
2.21 **CA z67 and stimulus:** *Not* significantly different from GLM. Explanation in Fig.2.20. 63

2.22 **CA z5 and stimulus:** Significant difference from GLM. Explanation in Fig.2.20. 63
2.23 **CA z19 and stimulus:** Significant difference from GLM. Explanation in Fig.2.20. 64
2.24 **Measured temporal response profiles by cell type:** Units of x-axis are stimulus frames or $\sim$ 16ms. 64

A.1 **Primate center-surround RFs:** modeled as difference-of-Gaussians. Note: R_c and R_s in image pixels. Values are given for magnocellular projecting (P) and parvocellular projecting (M) cells averaged across all eccentricities (avg) and at the visual periphery ($-40°$) Image of measured retinal RF size from Croner 1995 [**croner1995**] 70
A.2 **Visual angle calculation schematic** 71
A.3 **Grid-Distance Dependence:** Distance mask in **A** matrix: Elements within the adjacency matrix that are separated by distance $d = |r_i - r_j|$ in an 11x11 network arranged on a 2D lattice. 75
A.4 **Adjacency edge weight vs distance**: Average edge weight between node pairs in the adjacency matrix separated by distance r as a function of distance in image. Colored lines denote individual image patches and black line with grey error bars indicates μ and σ across 1500 image patches that are 50x50pixels. 77

B.1 **Coactive CAs can be synergystic or redundant:** *Bottom* PSTH traces show high temporal overlap between activations of 3 CAs. *Top left* shows high RF overlap for those CAs as well. 79
B.2 **CA individual inference and coactivity statistics:** Two panels show statistics for inference on all spike-words in data corpus after model is learned and fixed. Cell-type listed in panel caption. In each panel, CAs on x-axis. Blue points show number of time each CA was inferred across all spike-words. Red points show total number of times it was inferred with a partner. Top plot shows pairwise inference coactivity with 5 largest values circled in red. Coactivity among CAs is pretty insignificant. 79

List of Tables

2.1 Hyper-parameters used for model synthesis and data generation. 38
2.2 Hyper-parameters fit to offBriskTransient RGC responses to white noise and natural movie stimuli. Key differences highlighted in red. 43

Acknowledgments

This work has taken 8 years. That is, 90° from ∞. I want to give acknowledgement those who have helped keep me sane, whether they know it or not. To God; for serendipity, for all and more, for good and bad. Turns out, it's all (for the) good. To family, close and distant; for moral support, prayers and general rooting. To the Bridge; for a community, and the challenge and a home-away-from-home. To friends at the Missouri Lounge and elsewhere in the Berk scene; for the woodshed and my weekly dose of weird. To band and jam mates in Outta Thin Air and all its past (and future) incarnations; for support, inspiration, container, friendship. To those fellow-travellers inhabiting various rooms; for showing up for the real shit, for co-struggling, for all the presence. To Carolyn; for what I can't quite put into words. To all who have participated in the process of spiraling up. To fellow teachers and students and lifelong learners; Turns out Nobody else Knows either. To the Redwood Center; for being that magical and humble place that you are. Part of me still doesn't want to leave. Finally to Self, the common variable; Because you deserve it too. <3.

What's next?

Introduction

The complexity of the brain is staggering, even at its outermost sensory periphery. The retina is a favorite model system because of its relative simplicity; there is no direct top-down feedback from the rest of the brain. It is often modeled, taught and thought of as an oversimplified caricature in which a bank of simple independent, linear filters decorrelate stimulus features in space and time, reducing redundancy and [**barlow1961**] encoding local contrast in the spike rates of retinal ganglion cells (RGC) [**kuffler1953**]. However the complexity and heterogeneity of the retinal tissue has been heavily documented [**masland2011**], [**masland2012a**], [**werblin2011**], [**gollisch2010**]. Simple linear spatio-temporal filtering requires only a handful of cell types in the outer retina, leaving > 50 and nearly the entire inner retina extraneous. Further, textbooks fail to account for complex phenomena such as precise spiking of RGCs relative to the phase of network oscillations in the gamma range (50-80Hz) [**neuenschwander1996**] [**koepsell2009**]. Finally, the most advanced computational models founded on the textbook view fail to predict retinal responses natural and ethologically relevant stimuli [**chichilnisky2016**]. The hypothesis that ties both parts this work together is the following: *Perhaps, the retina reduces uninformative correlations /***pitkow2012***/ in stimulus with outer layers (photoreceptor, bipolar and horizontal cellls) in order to reintroduce informative correlations in precise spike timing with inner layers (bipolar, amacrine and ganglion cells).*

The first half of the statement forms the basis for the textbook retinal model. The second is deeply controversial. Herein, we present two projects that address this hypothesis from different directions. While connected to and inspired by the retina, each chapter stands on its own apart from the retina as well. Additionally, while related to one another by retina and more general ideas of phase coding well established in other neural systems [**fries2007**] [**singer2009**] [**buzsaki2013**], the two parts are also distinct from one another. In chapter 1, we explore image segmentation using phase coding in the retina, hypothesizing that fine-time correlations in spike trains are induced by phase interactions influenced by the visual stimulus and that these fine-time correlations, informative about segments in an image, are multiplexed into spike-trains along with rate-coded local stimulus features. In chapter 2, we explore a statistical model that aims to find cell assemblies, or groups of cells that fire are often co-active, possessing fine-time correlations irrespective of their source.

In chapter 1, we present an abstract proof-of-concept computational model of image segmentation in retina. Following and expanding on previous work, we cast image segmentation

as a graph clustering problem [**shi2000**], constructing a network from feature similarity in an image, and grouping together nodes based on interactions in a networks of phase-coupled oscillators [**arenas2006**]. Broadly, similarity between pairs of local features in the image determines the strength of phase interaction between the periodic structure in the spike trains. Phase diffusion through the network does not change firing rates but produces sets of neurons with similar spike times on a fine time scale. These sets of synchronous neurons represent spatially extended image features, image segments. The resulting spike trains multiplex two types of information, local contrast in individual spike rates, and image segments in sets of neurons that fire nearly synchronously [**koepsell2009**]. We make design choices to connect the model to retina, for example Gaussian receptive field image feature inputs determine coupling strength and phase initialization. However, this algorithm provides a general, distributed mechanism by which any objects with similar features can be grouped together in phase, and simultaneously separated from dissimilar ones.

Then in chapter 2, we introduce a novel probabilistic latent variable model, based on the Noisy-OR concept [**heckerman1990**], to detect "Cell Assemblies" (CAs), noisy repeats of groups of nearly synchronous cells, in observations of binary spiking neural data. Given a corpus of observed "spike-words", the task is to infer the sparse activity of a set of binary latent variables, CAs. If there are noisy repeats of the same firing configurations in the observation dataset, this repeating pattern is represented by a latent component. We apply our model to spiking responses recorded in retinal ganglion cells during stimulation with a movie. Again though applied in the current work to retinal data, the latent cell assembly model is completely general and can be leveraged to find ensembles of nearly synchronous neurons in other brain regions.

In fact it can be applied to any binary data.

We must stress that these two projects are separate in some key ways. While it may be tempting to connect cell assemblies in chapter 2 to segments in chapter 1, the cell assembly model is purely statistical in nature. It has no access to the stimulus and is cause agnostic, reporting the existence of noisy repeats but saying nothing about whether they are induced by stimulus or the anatomical retinal network or something else entirely. Further, there is no notion of phase or even time beyond spike-word binning. The model is trained by randomly sampling from the spike-word data corpus. The two projects are unified in that they challenge the textbook model of retina as a bank of independent, linear spatio-temporal filters that preprocesses images to reduce redundancy and remove uninformative correlations in the neural signal. Chapter 1 proffers a possible computation which reintroduces informative correlations back into the neural signal. And chapter 2 develops an algorithm to detect correlations, regardless of information content or source.

Chapter 1

Image Segmentation in Retina

1.1 Background

For decades the commonly accepted view of retinal processing has been that it provides a bank of independent, linear filters that decorrelate stimulus features in space and time, reducing the redundancy in the retina's representation [**barlow1961**]. Linear spatio-temporal filters factorized into center-surround spatial and biphasic temporal components followed by pointwise non-linearities encode local stimulus features in the spike rates of retinal ganglion cells (RGC) [**kuffler1953**]. There remain, however, severe puzzles, unexplained by the textbook view of retina.

First, for retinal ganglion cells it would be inefficient to use spikes exclusively in a rate code with rather long integration window. This assumption is in conflict not only with theoretical principles, such as the efficient coding hypothesis [**atick1992**], but with experimental observations. For example, it has been shown that time to first spike in RGCs can be very reliable, containing nearly as much information about the stimulus as spike rates [**gollisch2008**].

Second, the circuitry in the anatomical retinal network is exquisitely complex, consisting of >60 distinct neuron types stratified into at least 12 parallel and interconnected circuits providing roughly 20 diverse representations of the visual world, discussed at length in [**masland2011**], [**masland2012a**], [**werblin2011**], [**gollisch2010**]. Simple linear spatio-temporal filtering requires only a handful of cell types in the outer retina, leaving the rest of the network unexplained. By "occam's razor", the simple textbook view must be at least incomplete.

Third, the textbook model of retina fails to account for complex phenomena such as precise spiking of RGCs relative to the phase of network oscillations in the gamma range (50-80Hz) [**neuenschwander1996**] [**koepsell2009**]. Although the function of retinal oscillations is yet unknown in mammals, they have been observed in mouse [**menzler2011**], cat [**neuenschwander1999**] and primate [**ogden1973**]. Further, gamma-band retinal oscillations have been causally connected to the perception of spatially extended stimuli in the frog

[**ishikane2005**] Specifically, it has been observed that neurons in the cat lateral geniculate nucleus (LGN) often receive periodic retinal spike trains in the gamma band. Estimates of information rate in LGN spike trains suggest that in cells with periodic inputs, the spike train could mulitplex two different types of information. While rate modulation in a courser time window encodes local stimulus contrast, a significant fraction of the total information is encoded by spike timing at a fine time scale, conveying the phase of the gamma frequency in the neurons input [**koepsell2009**].

Fourth, computational models reflecting the text book view, such as the linear nonlinear Poisson (LNP) model and independent generalized linear model (GLM), predict RGC responses to a simple white noise stimulus [**schwartz2006**] with reasonable accuracy. However, looking more closely, one observes pairwise correlations in retinal activity, even in the absence of stimulus (correlations) [**schneidman2006**]. Taking into account these pairwise activity correlations improves decoding of retinal responses to white noise [**pillow2008**] – but does not explain why the retina introduces such correlations to begin with. The situation with ecologically relevant natural movie stimuli, in which pixels possess dependencies across space and time, is even more puzzling. The model prediction by independent encoding models becomes rather poor [**schwartz2006**], and even encoding models that include second-order correlations fail to replicate responses to natural movie stimuli [**chichilnisky2016**]. We suggest to take these mismatches between retina and its current computational models as an encouragement to design and investigate novel computational models of retina.

Here we approach the challenge to design better retina models from a computational perspective and ask: "What type of image analysis could be computed in an array simple sensors with access to (center surround) image features, like found in retina, above and beyond independent sensors proposed in the textbook model?" Specifically, we follow the lead suggested in the discussion of experimental work [**ishikane2005**] and investigate whether, in addition to encoding local image features, the retinal network can also extract spatially extended visual features and multiplex the extracted information into the retinal output using phase synchrony in periodic spike trains [**koepsell2009**].

To concretely design a sensor network model with this function we build on contributions provided in various streams of earlier work, the insight that image segmentation (IS) can be cast as a graph clustering problem [**shi2000**], and the insight that, in addition to spectral methods, graph clustering can be efficiently solved in networks of phase-coupled oscillators [**arenas2006**]. To evaluate the performance of the model, the Berkeley Image Segmentation Dataset (BSDS) was essential. While the motivation for this work is highly retinal, it should be noted that the network model we propose is still quite abstract. The model aims to serve as a proof of principle that the network computation could be efficiently performed by biological retinas, and not intended as a neurobiolocically detailed circuit model.

A coarse overview of the model is given in Fig. 1.1. The firing rate r_i in a coarse time window represents the local image contrast in the classical receptive field of neuron i. The similarity between pairs of local features in the image determines the strength of phase interaction between the periodic structure in the spike trains. Phase diffusion through the phase couplings does not change firing rates but produces sets of neurons with similar spike

times on a fine time scale. These sets of synchronous neurons represent spatially extended image features, image segments. The resulting spike trains multiplex two types of information, local contrast in individual spike rates, and image segments in sets of neurons that fire nearly synchronously [**koepsell2009**]. In our example, two image segments are represented by groups of neurons with different phases. Note that in this study, we only consider models of the phase dynamics, omitting aspects of spikes and spike rates.

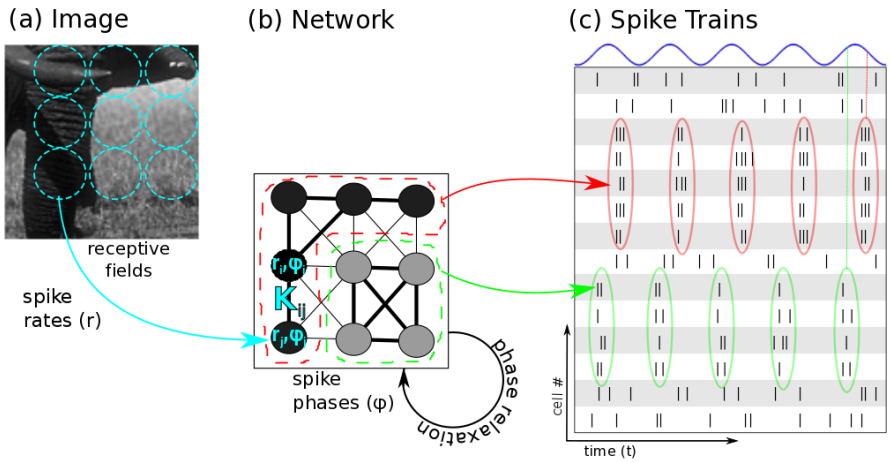


Figure 1.1: **Image Segmentation Model:** (a) Input image with superimposed retinal receptive fields (dashed cyan circles). (b) Network of retinal neurons. The neural firing rates r_i represent local contrast in the receptive fields. The phase interactions K_{ij} are displayed by the links between neurons. Line thickness represents the strength of the interaction which is set by the similarity of local features. Recurrent propagation in the network produces the phase structure ϕ_i of the periodic spike trains. (c) Resulting spike trains. Information about local features is represented in firing rates and segmentation is represented in phase structure.

The remainder of this paper is structured as follows. The Methods section describes prerequisites for our study from the literature. Section 1.2 concisely defines the putative computation of our retina model, image segmentation (IS) using simple image features available in retina, local contrast values or local center surround image features. The evaluation pipeline proposed in the BSDS image segmentation database [**martin2001**] is explained, which is essential to quantitatively compare the performances of different models. Following [**shi2000**], section 1.2 describes how image segmentation can by cast as a graph clustering problem, and how an adjacency graph is constructed for a particular image. Section 1.2 describes three common graph clustering methods from the literature, average association, graph Laplacian and modularity, that we will compare in our image segmentation experiments. Section 1.2 describes how, as an alternative to the spectrum of a graph edge matrix, relaxation of phase-coupled oscillators can be used to solve graph clustering problems. This step is critical in mapping the computation of image segmentation to the network model in Fig. 1.1b.

The Results section contains original contributions of our study. Section 1.3 describes topographic modularity, a novel graph-clustering method based on modularity [**newman2006**] that we propose for clustering multigraphs. For image segmentation the clustering of multigraphs is important because the graph representing local features of an image is a multigraph with two types of edges, one type representing feature similarity and the other geometric vicinity of the features in the image plane. Section 1.3 compares the performance of image segmentation of commonly used eigenvector-based "spectral methods" [**chung1997**] for graph clustering to the method of phase relaxation [**arenas2006**]. We find that phase relaxation generally outperforms spectral methods, independent of the choice of a particular image graph or receptive field structure. Thus, our further experiments focus on phase relaxation, the method that also has the advantage of being easily implementable as an oscillation-based computation [**koepsell2010**]. The central experimental results of our study are described in section 1.3. We compare segmentation performances of different network models to a baseline segmentation algorithm based on thresholding image feature histograms, a computation which does not require a network. While the standard graph clustering methods are not able to significantly outperform histogram thresholding, one model stands out significantly, the network implementing topographic modularity. Section 1.3 describes experiments to elucidate why the network with topographic modularity outperforms the competitor models. We find that phase diffusion through a network defined by topographic modularity quantifiably improves image segmentation, increasing edge precision at the expense of recall.

In the Discussion section we describe the various implications of the presented results. We describe the predictions our model makes for future neuroscience experiments and its potential for applications of image processing with coupled sensors.

1.2 Methods

Berkeley Segmentation Data Set

Image segmentation is a challenging and important problem in computer vision and the Berkeley Segmentation Data Set (BSDS) is a standard benchmarking data set for many computer vision image segmentation algorithms [**martin2001**, **arbelaez2011**]. It consists of 500 large ($\sim 400x300$ pixels) color images each with multiple (~ 5) human drawn boundary contours (green box in Fig. 1.2), as well as code provided for standard benchmarking and comparison of algorithms. Since image segmentation is closely related to boundary detection and quantification of boundary detection performance is more straightforward than that of image segmentation, segments in images are often recast as boundaries for benchmarking. Binary boundary pixel locations are compared to human drawn boundaries using the precision, recall, f-measure framework. In this context, "Precision" is the proportion of image pixels hypothesized by a method to belong to segment boundaries that agree with the ground truth. "Recall" is the percentage of ground truth boundary pixels that are found by a method. F-measure is the harmonic mean of Precision and Recall.

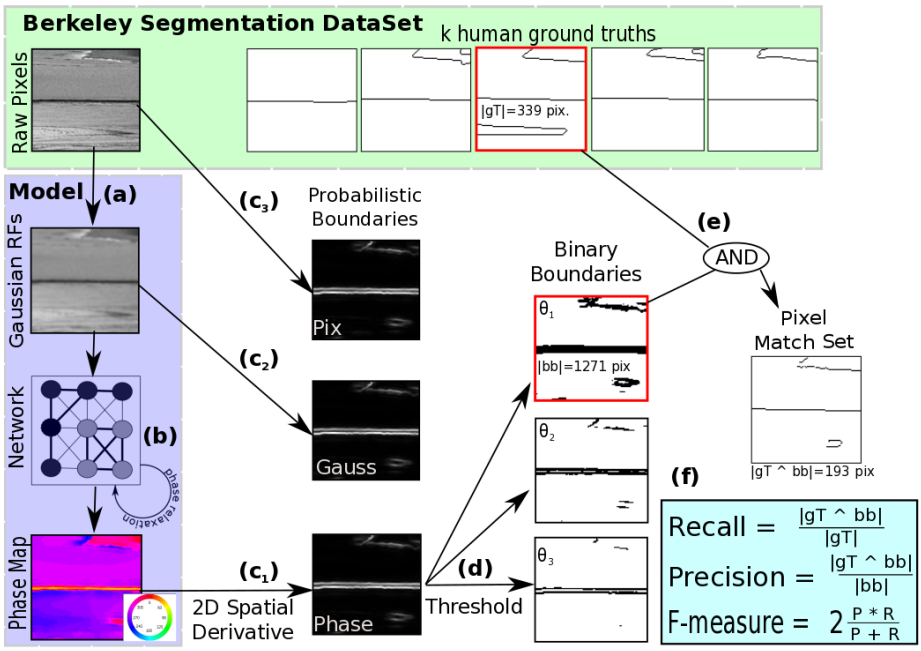


Figure 1.2: **Performance and benchmarking:** Input image patch and associated human drawn ground truth boundaries (gT) provided by BSDS is displayed in the green box. The operations performed by model are displayed in the blue box. Other steps of model evaluation are illustrated in the remainder. (a) Filtering the raw image patch with a Gaussian kernel ($\sigma = 1$). (b) Phase relaxation in the network (Fig. 1.1b) produces a phase map. (c $_1$) Spatial gradient operation ($\delta/\delta r$) and normalization resulting in probabilistic boundary map ($pb \in [0, 1]$). (d) Thresholding pb map at several values yielded binary boundary (bb) maps. (e) Match set was computed for each bb-gT pair at different distance tolerances, d_t. (f) Precision, recall and F-measure were computed by ratios of boundary pixel sets. (c $_{2,3}$) To assess the performance of network models relative to baselines, we repeated steps (c) - (f) on Gaussian RF and image pixels independent sensors models, comparing F-measures by subtraction.

In order to leverage the BSDS resource, we must first convert the output of a segmentation model - a phase, spectral or feature activation map (blue box in Fig. 1.2) - into binary boundaries. Intuitively, a good segmentation of an image has been achieved if the model output map has very similar values within segments and large discontinuities at boundaries. We compute spatial derivatives ($\delta/\delta r$) in the output map and normalize the values between 0 and 1, allowing us to interpret resulting probabilistic boundary (pb) as the algorithm's confidence that there is a boundary between segments at a particular image location. We can threshold pb's at multiple values and compare each resulting binary boundary map (bb) to each human drawn ground-truth boundary map (gT), generating a pixel match set by a logical AND operation. Because human drawn boundaries are not precise down to the pixel, we allow small misalignment between gT and bb pixel including a pair in the match set if

they are within d_t pixels of one another.

We compared the ability of different phase coupled oscillator models to segment images from the Berkeley Segmentation Dataset (BSDS) [**martin2001**]. The models differed in the phase couplings. One baseline model contained isotropic couplings, while the couplings in the other models AA, GL, M and TM were the transformations of the adjacency of features described above. We set the parameters σ_f, σ_d and σ_ω to adequate common values and performed for each method a parameter grid search in neighborhood connectivity R_M and scale K_s to maximize the average ΔF_b across 500 image patches. The oscillator frequency was 60 Hz (typical for retinal gamma oscillations) and we gave the networks 300ms to relax the phases, corresponding to an interval between saccades.

Image segmentation as a problem of graph clustering

Within a stage of visual processing, in which a set of local visual features is extracted, image segmentation can be viewed as a graph clustering problem [**shi2000**]. Consider an image and its corresponding neural representation in retina or LGN, in which the activity in individual cells represent the strength of local center-surround features. Image segmentation consists of clustering sets of local image features that share properties and thus likely correspond to larger objects in the image. However, much more efficient than clustering pixel values, is to apply clustering on more sophisticated local image features, e.g., center-surround, edges, multi-scale textures, as done in state-of-the-art segmentation methods [**shi2000**, **sarkar1998**].

The problem of graph clustering, that is finding "cliques" of strongly connected nodes in a graph, is obviously related to finding pixel sets with similar local features. To recast image segmentation in terms of graph clustering, one first uses kernels to construct an adjacency matrix, in which an element is large whenever two features have similar values and lie nearby each other in the image plane [**shi2000**, **sarkar1998**]. The segmentation of the image corresponds to finding the communities (subsets of nodes) that are strongly interconnected within the community, and well separated from nodes outside. The goal then is to find non-trivial subsets of nodes that can be separated from one another by cutting through the minimum weight of edges, know as the "mincut" problem. Though a brute force, optimal solution to this problem would be combinatorically intractable, approximate solutions can be found efficiently by leveraging the machinery of "spectral graph theory" [**chung1997**].

Following [**shi2000**], we define a graph for segmenting an image by the *adjacency* matrix:

$$\mathbf{A}_{ij} = e^{-\frac{(f_i - f_j)^2}{2\sigma_f^2}} \cdot e^{-\frac{(r_i - r_j)^2}{2\sigma_r^2}} \cdot \left(1 - H(\sqrt{(r_i - r_j)^2} - R_M)\right) \tag{1.1}$$

with $H(x)$ the Heaviside step function. The first factor reflects the dissimilarity of the local features f_i and f_j, in our case local contrasts. It was found experimentally that $\sigma_f = 0.2$ provides reasonable dynamic range in adjacency weight distribution. The second and third terms reflect the distance between the local features in the image plane. Since we are interested how well segmentation can be performed in networks with local neighborhood

connectivity and for simplicity, we we null out the second term by setting $\sigma_r = \infty$ and add the third term, a binary rectangular Heaviside function 1 - $H(\sqrt{(r_i - r_j)^2} - R_M)$ that is 1 within a maximum radius, R_M, and 0 outside. We explored R_M values of 1,3,5 and 10.

Three common graph clustering methods

The simplest strategy of graph clustering, referred to as *average association* (AA), is to analyze the adjacency matrix directly [**sarkar1998**]. Eigenvalues of the adjacency quantify the amount of correlated structure and the associated eigenvectors characterize the location of the correlated structure in the image. Other methods of graph clustering utilize transformations of the adjacency matrix, often incorporating the node "degree", $d_i = \sum_j A_{ij}$, which captures the total weight of connections to each node from all other nodes in the network. One such transformation we considered is the normalized *graph Laplacian* (GL) or Kirchhoff matrix: $L = D^{-1/2}(D - A)D^{-1/2}$ with diagonal matrix $D_{ij} = \delta_{ij} \sum_k A_{kj}$, δ_{ij} the Kronecker symbol. This strategy, combined with more sophisticated image features, forms the basis of a very successful image segmentation algorithm, the "Normalized Cut" [**shi2000**]. The eigenvectors and associated smallest eigenvalues of the Laplacian matrix find divisions in the input characterized by large feature differences.

A second transformation we considered is *modularity* (M) [**newman2006**], which has successfully discovered community structure in social and information networks, outperforming the graph Laplacian in these tasks. The modularity matrix can be written as

$$Q = A - N \quad \text{with} \quad N_{ij} = D_i D_j \quad \text{and} \quad D_i := \frac{d_i}{\sqrt{2m}} \tag{1.2}$$

where D_i and D_j denote the "degree " of nodes i and j respectively, normalized by the total weight of edges in the graph, $2m = \sum_k d_k$. Importantly, the null model matrix, N, contains the expectation of the weight value between each node pair N_{ij} based on the strength of connectivity of both nodes. In this way, an expected graph is constructed by assuming an otherwise random graph with node degrees constrained (an Erdos-Renyi random graph). Comparing the observed adjacency graph to the null model by subtraction reveals graph structure beyond what could expectedly be introduced by heterogeneous node degrees. In section 1.5, we discuss modularity further and introduce an extension, called *topographic modularity* (TM), with null model adapted for graphs embedded in space.

Once an associated matrix representing a graph is constructed, spectral methods have been predominantly used within the graph clustering community to find clusters within because eigenvalues and eigenvectors efficiently find an approximate solution to the combinatorially intractable "mincut" problem. It has been observed on simple networks that the eigenvalue spectrum of an associated matrix resembles the temporal progression of clusters discernible from phases of nodes in a Kuramoto network [**arenas2007**], this time evolution of clusters forming a hierarchical clustering of a network. Given this observation, we compute the time evolution of a phase coupled oscillator network dynamical system as an alternative to eigenvector-based graph clustering methods.

Kuramoto Phase Relaxation Model

The described graph clustering methods in 1.2 compute the eigenvectors of the associated matrices [**chung1997**] which, in essence, is assessing anisotropic diffusion in these networks. This process has also been related to the path a random walker would take through the graph where edge weights represent transition probabilities and the distribution of electrical potentials on nodes in a resistor network where an edge weight represents the conductance of a particular resistor [**grady2006**]. A further parallel has been between eigenvector based methods for graph clustering and the "fundamental mode(s) of a spring-mass system" [**shi2000**]. To rigorously investigate this last claim, we simulate phase relaxation in a network of Kuramoto coupled oscillators [**kuramoto1984**] with networks defined by methods described in 1.2.

Here we followed [**arenas2007**] and assessed diffusion properties by relaxing a network of phase-coupled oscillators :

$$\Delta\phi_i = \omega_i + \sum_j K_{ij} sin(\phi_i - \phi_j), \qquad K_{ij} = k_s M_{ij} \tag{1.3}$$

with each node's natural frequency $\omega_i = 60Hz$ and where M_{ij} is one of the graph matrices mentioned above. For intuition, Eq. 1.3 loosely simulates a lattice of oscillating masses connected by different size and signed springs. The lattice is shaken at initialization and through the relaxation dynamics, masses connected by strong positive springs are attracted in phase while strong negative springs repel one another. In the original Kumamoto model [**kuramoto1984**], couplings K were set to be uniform, supporting isotropic diffusion. As a baseline, we also investigated the effects of isotropic diffusion (ISO) for image segmentation. Unlike the uniform network, a network with heterogeneous weights relax to stable states containing multiple distinct clusters of phase aligned oscillators.

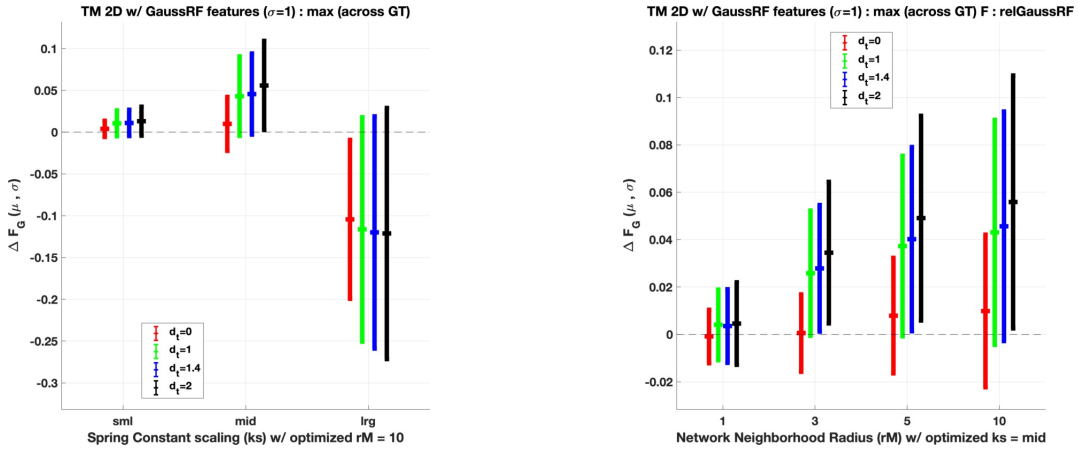
In the implementation of the model, the overall positive scaling factor k_s is critical. If coupling weights are too large, phasers will spin wildly in response to even small phase differences. Conversely, if too small, oscillators will adjust their phase too slowly and the relaxation will not converge in time. Importantly, the phase relaxation was limited to 300ms or 20 periods of the 60Hz signal, which is the average duration of fixation before a saccade brings the eye's gaze to a new point, refreshing the input and beginning the computation once again. The value for the k_s parameter was set for each graph individually based on mathematical considerations in equation 1.3. A middle value k_s^{mid} was chosen so that the phase change of the node with largest degree $D_{k_{max}}$ is limited to $\pi/2$ radians in one full period of the 60Hz signal when all its neighbors are aligned $\pi/2$ radians away and exerting maximal pull.

$$k_s^{mid} = 60Hz \cdot \frac{2\pi}{\pi/2} \cdot D_{k_{max}} \tag{1.4}$$

We then bracketed that value above and below by an order of magnitude.

The final result of the phase relaxation simulation is a phase map with a phase value, $\phi_i \in [0, 2\pi]$, associated with every node, i, in the network and corresponding location i in the image. Spectral methods also yield a value associate with each location, i, in the image with $v_i \in [-\infty, \infty]$. In order to compare our results to other algorithms using the BSDS resources, we convert these maps to probabilistic boundaries and recast the image segmentation problem as a boundary detection one as discussed in Section 1.2 and illustrated in Fig. 1.2.

In practice, two meta parameters, r_M defining the neighborhood structure of the Adjacency graph and k_s defining an overall scaling on the strength of phase interactions in the network, impacted image segmentation performance. They were optimized for each method and results shown are with optimized parameters, shown in Fig. **??**. To optimize parameters for each method, we performed segmentation of 500 image patches with four r_M values ranging from 1 to 10 and bracketing k_s as discussed above and chose the parameter settings with best average performance across all images and across d_t. Fig. 1.3 illustrates the procedure for one particular method. It shows average performance across k_s values for optimal r_M on the left and performance across r_M values for optimal k_s on the right. Fig. 1.4 shows the effect of different parameter settings on one example image patch.



(a) (b)

Figure 1.3: **Hyper-parameter optimization:** Network neighborhood graph structure r_M and coupling spring-constant scaling k_s are important meta parameters of the algorithm, discussed in Sections 1.2 and 1.2 respectively. We plot mean and standard deviation across 500 image patches of ΔF-measure relative to Gaussian RF independent sensors for the 2D topographic modularity network. Colors indicate pixel distance tolerances d_t (see Fig. 1.2 for explanation). *Left* panel shows performance at three k_s values, with r_M fixed at optimal. *Right* panel shows performance at four r_M values, with k_s fixed at optimal. Fig. 1.4 shows the effects of the different parameters on a single example image patch.

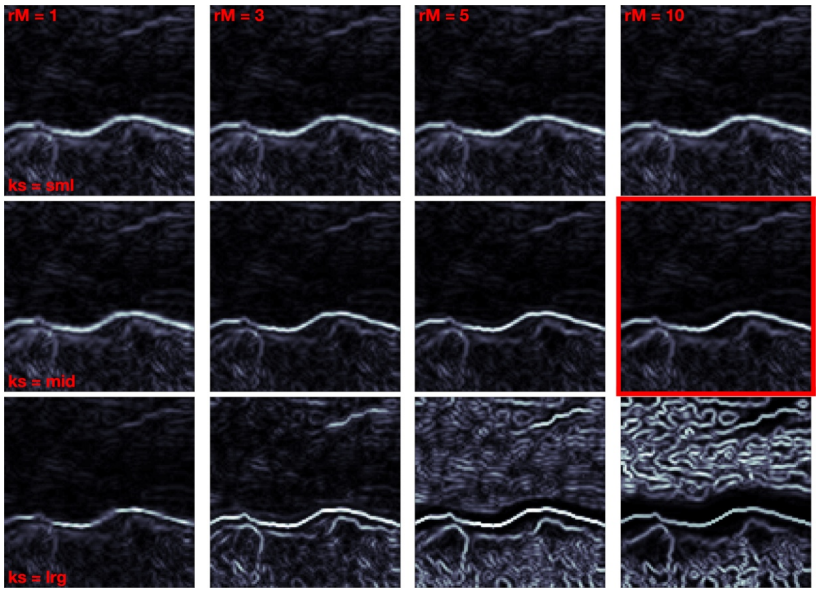


Figure 1.4: **Effect of hyper-parameters, single image patch example:** Probabilistic boundary maps shown for resulting phase distribution from TM 2D method for combinations of 3 k_s (rows) and 4 r_M (columns) hyper-parameters.

1.3 Results

Modularity null models for images

An image can be described by a multi-graph, in which pixels or local image features are represented by nodes and each pair of pixels has two different types of edges connecting them. One edge type represents geometric distance in the image plane and the other edge type represents feature differences. The two types of edges are given by adjacency matrices, resulting from the two types of distances and corresponding kernel functions (like a Gaussian kernel), as in Eq. 1.1. Shi and Malik [**shi2000**] proposed a way to collapse this multi-graph of an image to an ordinary graph by forming the Hadamard product of the two adjacency matrices. An entry in the resulting single adjacency matrix A represents the two distinct similarities between pixels, geometric proximity and feature similarity by a single number. Specifically, an entry in A can only be large, if both, distance and feature differences are small in the corresponding pair of pixels. In order to find image segments, researchers then used "spectral" graph clustering methods on the matrix A [**sarkar1998**, **shi2000**].

For some graph clustering methods, such as modularity [**newman2006**], the collapsing of the multi-graph into an ordinary graph destroys information, which is critical for segmenting images. The modularity matrix consists of the difference of the adjacency matrix and a null model. The null model represents an average adjacency value. In the standard modularity

method, Eq. 1.2, the average is computed from the degrees of the two nodes involved, the row and column sum of the collapsed graph. However, in natural images, the average feature similarity of a pair of pixels is a function of geometric distance [**ruderman1994**], see also Fig. A.4 in supplemental section A.2. Thus, an appropriate null model for images should also depend on the geometric adjacency matrix.

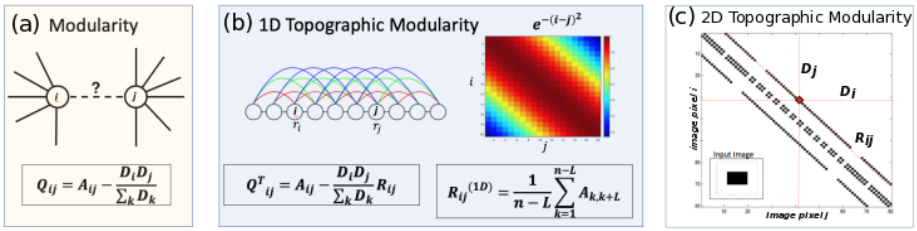


Figure 1.5: **Modularity null models & space**: In the null model of Newman's modularity [**newman2006**] *(panel a)* the average weight between nodes i and j is proportional to the product of their node degrees ($D_i \cdot D_j$). The topographic modularity's null model *(panels b & c)* additionally includes a distance-dependent factor, R_{ij}, which is the average edge weight between all node pairs in the graph separated by the same distance that separates nodes i and j. *Panel b* illustrates R_{ij} for a schematized 1D graph, shown with edges colored based on distance between the nodes they connect. Inset plot shows geometric factor in the topographic null model. Each term in $R_{ij}^{(1D)}$ is an off-diagonal sum in the adjacency matrix. *Panel c* shows the mask associated with a single geometric distance in a 2D image. Here $R_{ij}^{(2D)}$ at 1 pixel separation has a complex structure in the Adjacency matrix for even the simple binary image shown in the inset.

To address this issue, we devised a novel graph clustering method called *topographic modularity* (TM) in which the null model takes topographic distance in the image plane into account. Like the standard modularity [**newman2006**], see Fig. 1.5a and Eq. 1.2, an entry of the topographic modularity matrix, Q^T, is the difference between the entry A_{ij} and the expected connectivity, captured by the null model, N_{ij}. Here, the topographic null model N_T accounts for distance dependent factors in feature similarity with the R_{ij} term in addition to node degree heterogeneity.

$$Q_{ij}^T = A_{ij} - N_{ij}^T \quad \text{where} \quad N_{ij}^T = D_i \cdot D_j \cdot R_{ij} \tag{1.5}$$

The R_{ij} factor represents the average connectivity between all node pairs that are separated by the same geometric distance as the nodes i and j. For a network in space along a 1D line, Fig. 1.5b, the distance dependent contribution to the null model can be written mathematically as

$$R_{ij}^{(1D)} = (\frac{1}{n-L}) \sum_{k=1}^{n-L} A_{k,k+L} \tag{1.6}$$

where L is the distance separating nodes i and j (i.e., $L = r_i - r_j$) and n is the total number of nodes (or pixels or features in the image). In 1D, the average connectivity of all nodes separated by a distance L is equal to the mean along the L'th diagonal.

For networks with 2D grid-like geometry, like Adjacency graphs constructed from images, the computation of $R_{ij}^{(2D)}$ is more involved, yet the interpretation is the same. Reshaping a 2D image into a 1D vector so that similarity relationships can be represented in a 2D matrix introduces discontinuities in spatial relationships between entries in the matrix. Weights between nodes separated by a particular distance can be labeled by a mask specific to the dimensions of a particular image. Fig. 1.5c shows the weights between all neighboring nodes ($L = 1$) in the network derived from the 11 x 11 binary image in the inset. For completeness, we show $R_{ij}^{(2D)}$ masks for other pixel separations in supplement section A.2 Fig. A.3.

Before comparing the different null models in an image segmentation we compare how well they capture the structure of an adjacency matrix of an image. The null model in Newman's modularity, by construction, is a "consistent" estimator of node degrees [**chang2012**], ensuring that $\sum_j N_{ij} = \sum_j A_{ij}$ (blue line in Fig. 1.6 middle). However, it is clearly the wrong null model for natural image Adjacency graphs for two reasons. First, the null model incorrectly contains positive diagonal weights in proportion to D_i^2, although the diagonal elements of the adjacency matrix are zero. Second, it does not capture the distance dependence of the adjacencies, thereby underestimating average adjacency between proximal nodes and overestimating it for distant node pairs. Both problems manifest in the difference between the blue and the dashed lines in Fig. 1.6 bottom.

While the TM-1D and TM-2D null models are not strictly consistent in node degree or distance dependence, they are nearly so (green and red lines respectively in Fig. 1.6). Introducing distance-dependent statistics into the TM-1D null model corrects for the spatial "inconsistency", vastly improving estimates of edge-weight over M. TM-2D offers improvement over TM-1D due to further refinement of its null model, see Eq. 1.6 and surrounding text. Further discussion of null model consistency and bias in the supplemental section **??**.

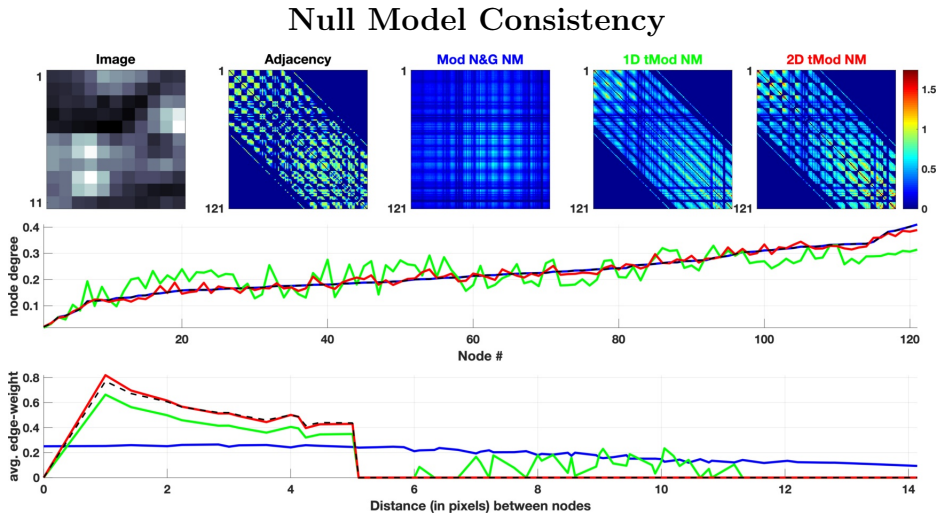


Figure 1.6: **Null model consistency**: *Top row* from left to right shows image patch, the adjacency (black) constructed from the patch with $r_{max} = 5$, and null models for modularity (blue), 1D topographic modularity (green) and 2D topographic modularity (red), with colorbar indicating edge weight. Models represented by line colors in plots as well. *Center plot* shows average node degree (row sums in each matrix) sorted by strength in adjacency. *Bottom plot* shows average edge weight as a function of distance in the image plane.

Importantly, the difference between an adjacency value and its average in the modularity can become negative. In a Kuramoto net relaxation, these negative weights mediate phase repulsion and introduce targeted phase desynchronization, see Sec. 1.2, at boundaries in an image where gross image statistics change. In contrast, if the modularity value between a node pair is positive, it contributes to phase synchronization. Fig. 1.7 illustrates image segmentation performance before and after phase relaxation through connections defined by M (in blue), TM-1D (in green) and TM-2D (in red). While M does not significantly change image segmentation performance over Gaussian RF independent sensors, TM-1D does so (p-value ~ 0.004) and TM-2D does so even more (p-value $\sim 4 \cdot 10^{-7}$). With TM-2D, we see improvement for $^{\sim 460}/_{500}$ image patches.

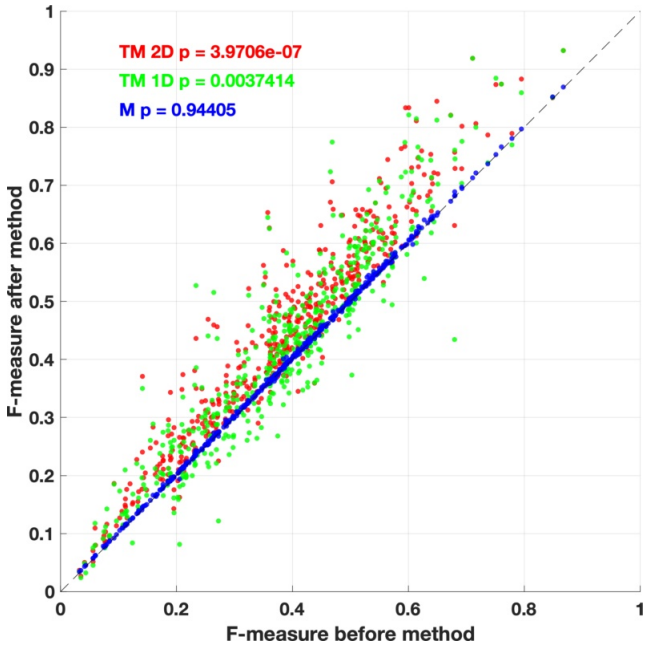


Figure 1.7: **Modularity performance comparison:** Each scatter point represents one image patch. Newman's modularity (M) in blue, 1-dimensional topographic modularity (TM 1D) in green and 2-dimensional topographic modularity (TM 2D) in red. Points above the unity line indicate image patches with improved image segmentation with network phase relaxation over-and-above Gaussian RF independent sensors. P-values quantify the difference between F-measure distribution across 500 image patches before and after network computation.

Broad comparison of models on image segmentation

Following [**koepsell2010**], we investigate the idea whether a phase-coupled network of simple sensors of local image features, similar to those in the retina, could at the same time represent local and contextual image features in its output. Specifically, phase interactions mediated through heterogeneous network edges which are influenced by local features similarities can segment an image, grouping regions within a segment into the same relative phase and introducing phase breaks at segment boundaries. In a biological system, the contextual image information encoded by phase can be represented by the timing of spikes and be multiplexed into spike trains, whose rates represent the local features Fig. 1.1.

This idea is tested on images provided in the Berkeley Segmentation Data set (Sec. 1.2). For an image patch, we construct a graph based on local features in the image (Sec. 1.2) and segment the image by either computing eigenvectors or by performing anisotropic phase diffusion in a Kuramoto net. Computing a spatial derivative on either eigenvectors or the final phase distributions and normalizing values between 0 and 1 converts the output into probabilistic boundaries, which can be quantitatively compared to assess relative performance of different image segmentation methods.

We ask whether a phase-coupled sensor array can add to an image segmentation that can be done based on the independent sensor measurements alone. Thus, the network computation must outperform two baseline methods. The first method computes normalized spatial gradients on the raw image pixels (magenta, RawPix). In the second method the image pixels are first convolved with Gaussian receptive fields, roughly similar to those measured in retina (cyan, GaussRF). As a third baseline method, we include isotropic diffusion in a network with homogeneous phase couplings between nearest-neighbor nodes (black, ISO).

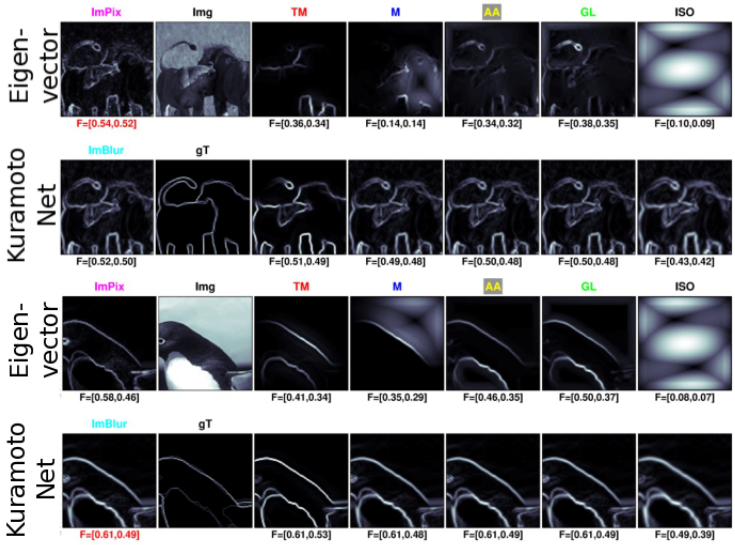


Figure 1.8: **Spectral methods vs. Kuramoto Net Examples:** Two example image patches (top two rows and bottom two rows) show probabilistic boundaries found by different network (TM, M, AA, GL) and baseline models (ImPix, GaussRF, ISO). Network models are segmented using eigen-methods (1st and 3rd row) and Kuramoto Net phase relaxation (2nd and 4th row). Qualitatively, boundaries found with spectral methods are less crisp and more localized than those found with Kuramoto Net phase relaxation.

Probabilistic boundaries (pb) can be interpreted as the algorithm's confidence that a boundary exists between two segments at a particular location in the image. Fig. 1.8 shows pb's resulting from the segmentation of different networks constructed from the same image patch, either by computing eigenvectors and by performing Kuramoto net relaxation. Qualitatively, we observe that eigenvectors seem to focus a spotlight on a region of the image patch while information propagated through the Kuramoto Net covers all parts of the image patch. Regardless of the network method used, boundaries found with the Kuramoto net are crisper and extend further across the image patch than do those found by computing eigenvectors.

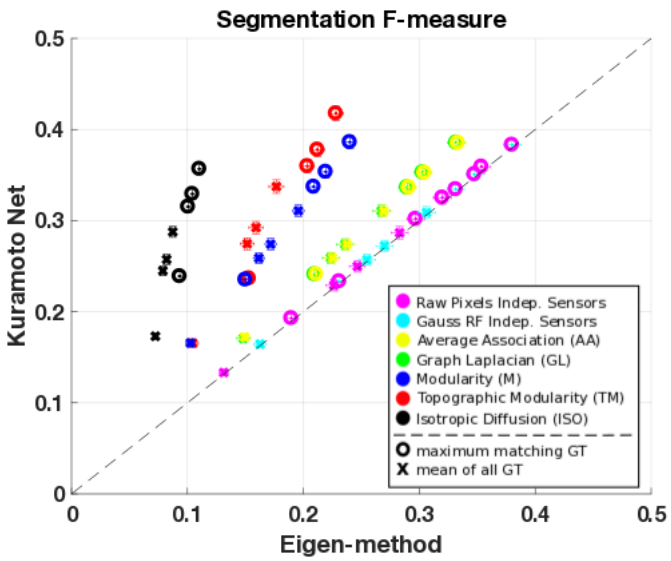


Figure 1.9: **Spectral methods vs. Kuramoto Net Statistics:** F-measure computed across 500 image patches, mean and standard error errorbars. Colors indicating different network and baseline models are used consistently throughout this paper. Circles indicates that F-measure for each image patch taken for maximum matching GT and x's shows mean value across all GT's. Network models built with Gaussian RF features are segmented by the best combination of the top 3 eigenvectors on the x-axis and by the phase distribution after Kuramoto Net relaxation on the y-axis. The dashed unity line indicates equal performance and the independent sensors baseline models (magenta and cyan) do not deviate from it.

To assess whether this trend in image segmentation performance is statistically significant, we calculate Precision, Recall and F-measure across 500 image patches, shown in Fig. 1.9. Plotting F-measure statistics for network and baseline models segmented by Kuramoto-net and Eigen-methods, we find that segmentation without network computation (magenta and cyan) outperforms the results from the best combination of the top 3 eigenvectors, regardless of the model. We also find that all scatter points lie above the unity line, indicating superior image segmentation performance of anisotropic phase diffusion in a Kuramoto net verses the spectral clustering methods. As a consequence of this observation, we focus in the reminder on the superior methods based on Kuramoto Nets.

Influence of receptive fields choice

We further observe that the features from which networks are constructed influence segmentation performance achieved. This comes as no surprise since state-of-the-art image segmentation algorithms rely on a combination of sophisticated spatially-extended features.

We constrain our investigation to the relatively simple and local stimulus features that retina is supposed to have access to. Specifically, we investigate the difference in segmentation caused by switching between raw pixels and Gaussian receptive fields with different radii. Again, we compare the segmentation performance of networks with phase relaxation to baseline models representing independent sensors, and a model with isotropic diffusion through a homogeneous neigbor connections. We find that Gaussian receptive field features provide better segmentation than raw image pixels both when used as independent sensors and to construct phase interaction networks. Fig. 1.10 shows the segmentation performance (F-measure and the change in F-measure relative to the independent sensors image pixels baseline model) as a function of pixel match distance tolerance (d_t).

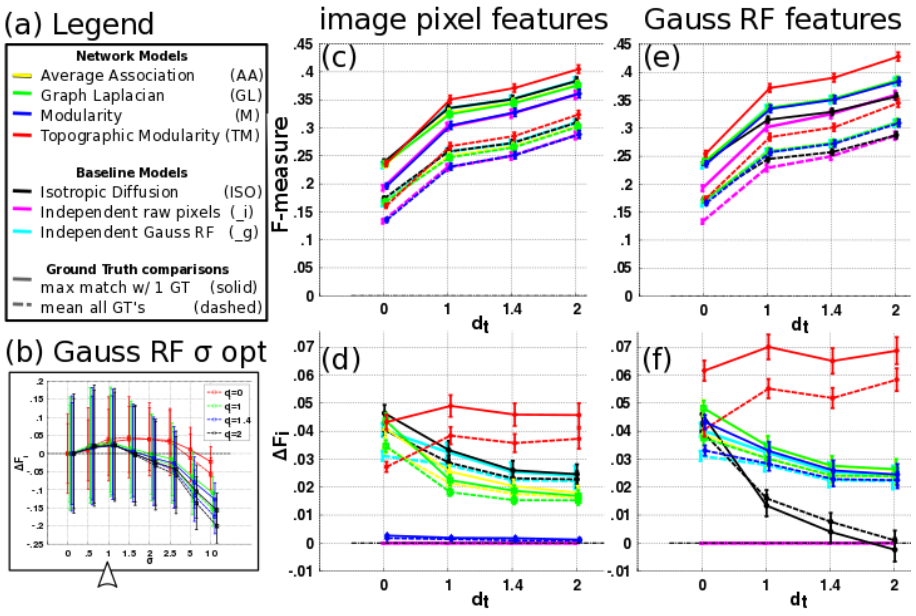


Figure 1.10: **Gaussian RFs improves segmentation**: Performances of 4 anisotropic diffusion and 3 baseline models are compared using raw image pixel features and Gaussian RF features, *center and right columns respectively*, lines representing average and bars standard error across 500 BSDS image patches. (a) Colors indicate different models and line styles indicate ground truth comparison as in Fig. 1.9. (b) Optimal spread, σ, of Gaussian RF's chosen by maximizing change in F-measure relative to the independent raw pixels baseline model, ΔF_i, averaged across all image patches. Recall d_t is the "distance tolerance" when computing the pixel match set, Fig. 1.2. Optimal performance for all d_t values obtained for Gaussian RF $\sigma = 1$. (c) F-measure and (d) ΔF_i when models receive raw image pixels as features. (e) F-measure and (f) ΔF_i when models receive Gauss RF activation as input features.

For small tolerances d_t in the F-measure (see section 2.2) the simpler isotropic phase diffusion model was a surprisingly strong competitor, even beating some of the anisotropic networks (black lines in Fig. 1.10c and d). Isotropic diffusion with optimized parameters

provides mild smoothing of image structure, which operates indiscriminately within and across segments. To introduce the effect of smoothing in other models, we introduced Gaussian RF features. The filters corresponded to optical blur and the extended (centers of) receptive fields in retinal ganglion cells. Fig. 1.10b shows segmentation performance as a function of the width of the Gaussian filter, σ, and tolerance parameter, d_t. We find that Gaussian RF features with $\sigma = 1$ were beneficial and near optimal across different tolerance values. Interestingly, the size of the optimal Gaussian coincides with the size of retinal ganglion cell receptive fields measured in primate retina [**croner1995**]. See supporting information A.1 for further discussion.

Fig. 1.10e and 1.10f show the improvement in segmentation performance using Gaussian features above using image pixel features. In particular the method TM displayed a significant increase in ΔF_i which became more prominent for larger pixel match distance tolerances d_t. Among all methods TM was able to improve segmentation performance the most, compared to that achievable with the Gaussian RF independent sensors model.

Detailed model comparison between most promising models

To assess the overall performance of different models on the diverse input images, each model was run with optimized parameters. Fig. 1.11 shows image segmentation performance improvement from Gaussian RF independent sensors. Here the models TM-1D, TM-2D and ISO were significantly different from the three other methods that stayed near baseline $\Delta F = 0$. ISO stayed below baseline because the input kernels provide near optimal blur and therefore additional isotropic blurring deteriorated the segmentation performance. TM-1D performs well too, but not as well as TM-2D. This is because the null models are increasingly accurate, section 1.3. Shown results are with best matching ground truth. Results hold with average across all human drawn ground truths, though less pronounced.

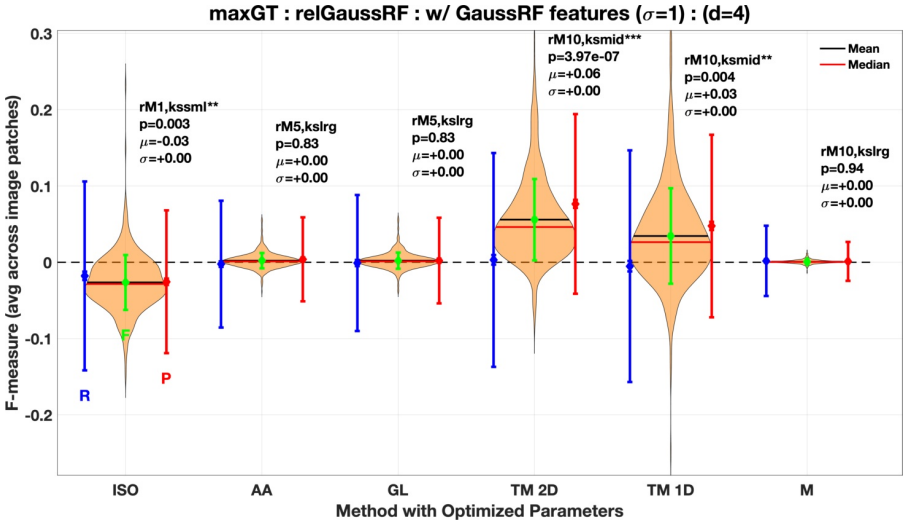


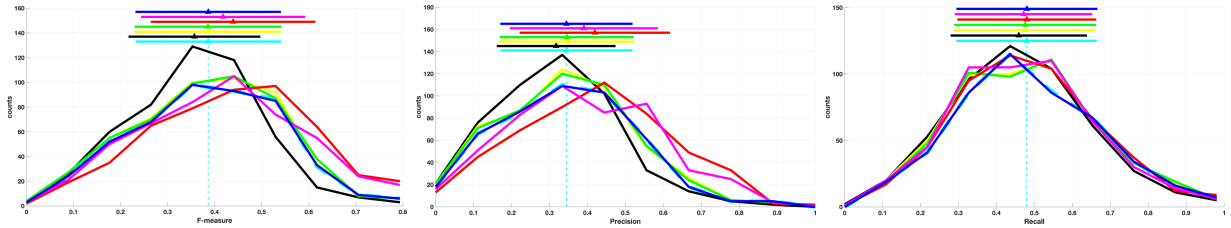
Figure 1.11: **ΔF-measure model comparison:** Violin plots show ΔF_G distribution with moments of ΔPrecision, ΔRecall and ΔF-measure distributions across 500 image patches in red, blue and green, respectively. ΔF relative to Gaussian RF Independent Sensors model. Optimal hyper-parameters(r_M,k_s), statistical significance, p-values and distribution moments indicated above each method. Performance of ISO, TM-1D and TM-2D models relative to Gauss RF are statistically significant, as determined by Mann-Whitney U (aka rank-sum) test.

Segmentation performance via anisotropic phase diffusion in a Kuramoto net depends critically on the structure of the phase couplings. Kuramoto nets using the graph Laplacian, average association or Newman's modularity as the phase couplings do not improve segmentation performance significantly over the independent sensors Gaussian RF baseline model. Only the Kuramoto model with the topographic modularity as phase couplings increases segmentation information over baseline independent sensors, homogeneous network and competitor heterogeneous network models, as quantified by the F-measure.

Why is the Kuramoto model with topographic modularity superior?

The F-measure combines the performance measures Precision and Recall, each with intuitive interpretations described in section 1.2. To analyze the differences between our different models, we separately plot the precision and recall distributions in Fig. 1.12. Note the position of curves for each network method relative to the independent sensors Gaussian RF baseline model (cyan dashed curve). Focusing first on the F-measure, in panel a, three of the network models (AA in yellow, GL in green, M in blue) did not show significant differences. The ISO model (black) degraded segmentation performance while the TM models (red & magenta) improved relative to the Gaussian RF baseline. In panels b and c, the precision distribution of both TM models shifts significantly to higher values while the recall distribution shifts

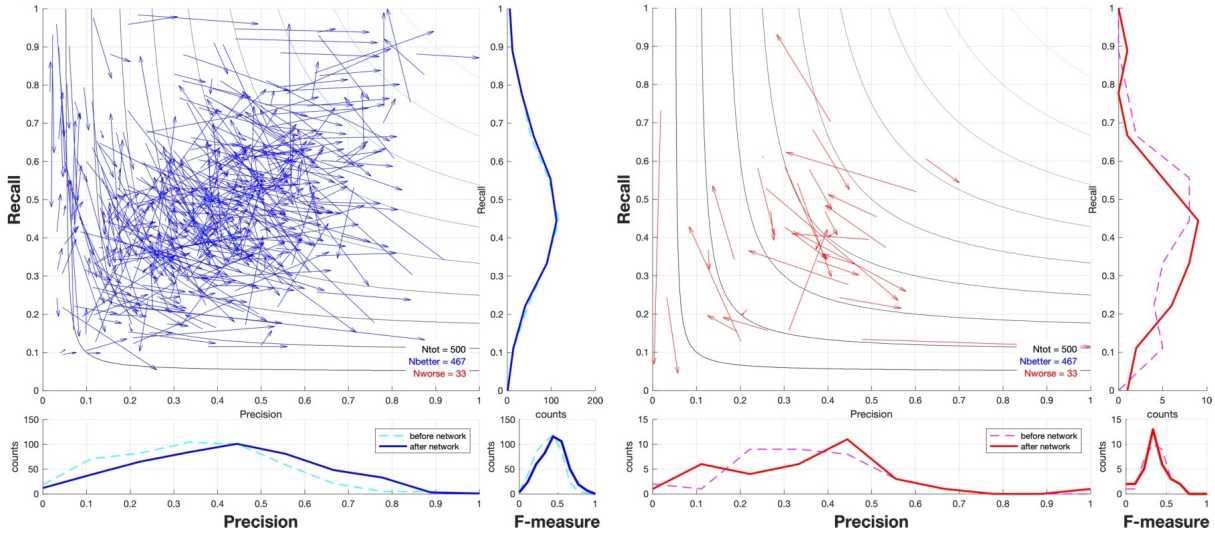
only slightly to lower values. Thus, the performance improvement of the TM model is mainly caused by increased precision, reflecting superior ability to suppress spurious boundaries, texture or "noise" in the probabilistic boundary maps.



(a) F-measure (b) Precision (c) Recall

Figure 1.12: **Precision & Recall model comparison:** (a) F-measure, (b) precision and (c) recall across 1000 image patches for Gaussian RF independent sensors baseline model and 4 network models with optimized parameters and $d_t = 2$. Distribution μ and σ denoted above. Note colors same as in Figs. 1.9&1.10.

To better understand the computation in the TM-2D model, we visualize changes to Precision and Recall together for individual image patches in Fig. 1.13.



(a) Improved segmentation: $\Delta F > 0$ (b) Degraded segmentation: $\Delta F < 0$

Figure 1.13: **ΔPrecision and ΔRecall with TM-2D model:** In *panel a*, arrows show change in P & R for 467 image patches where network increased F-measure. Arrow tails indicate values before network relaxation and heads values after. Surrounding are distributions showing P,R,F before network (in cyan) and after (in blue). *Panel b* shows the same for 33 image patches where network decreased F-measure. Distributions before network in magenta.

The TM-2D network relaxation improved segmentation for $\sim 93\%$ of all image patches, in blue, panel a. Clear positive shifts in the precision and F-measure distributions can be observed from the independent sensors Gaussian RF model (dashed cyan) to the phase output from the TM-2D network relaxation (solid blue). No clear trend emerges for the recall distribution with improved images. No clear trend exists for images where the network relaxation decreased performance. For some precision increased, and recall decreased. For others, vice versa.

Visual assessment of model performances

Finally, to provide some intuition what a ΔF value means for individual images, some examples are shown in Fig. 1.14. Compared to the results from other methods, the TM model produces probabilistic boundaries (pb's) that are often thinner and cleaner.

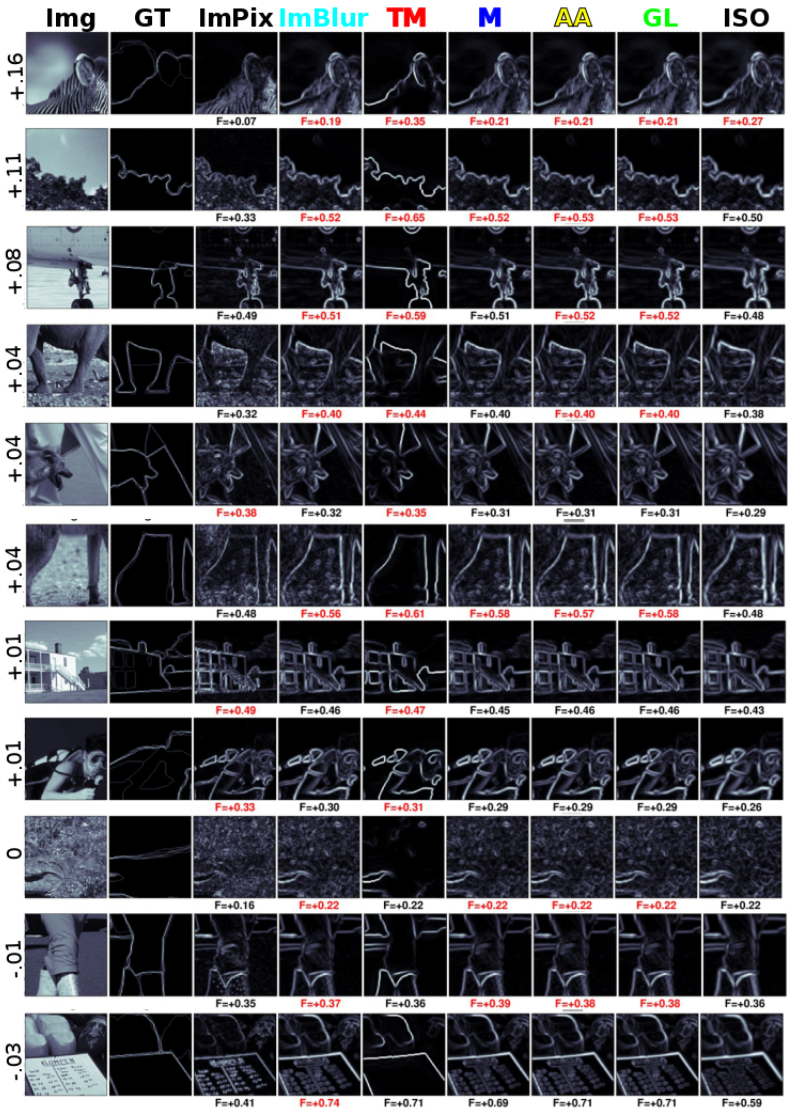


Figure 1.14: **Representative image patches:** Each row shows one example image patch ordered by change in F-measure between Gaussian RF independent sensors baseline and TM models (indicated on left). Columns show image pixels, gT boundaries and pb maps obtained from raw pixels, Gaussian RF and 5 network models. Mean F-measure value across all gT's noted below each pane is red if $\Delta F_G > 0$.

Further, in Fig. 1.15, we show samples of image patches with varying image segmentation performance relative to the Gaussian RF independent sensors model.

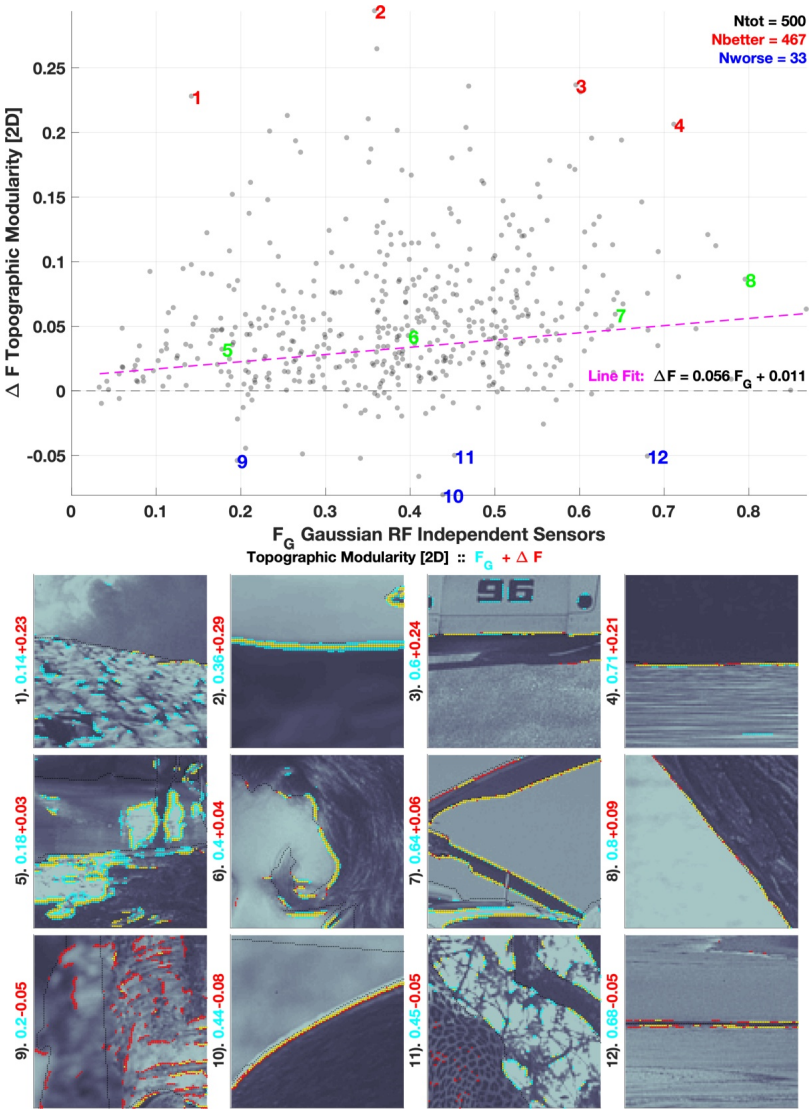


Figure 1.15: **Examples of TM 2D model performance:** *Top* panel scatters F-measure in Gaussian RF independent sensors model vs. Δ-F after 2D topographic modularity network phase relaxation. Out of 500 total image patches, 467 show positive improvement. Best fit line to scatter points in magenta. Colored numbers indicate randomly sampled image patches (shown in bottom panel) where Δ-F performance is best (#1-4), average (#5-8) and worst (#9-12). *Bottom* panel shows image patches with best matching ground truth boundaries, in black. Yellow points indicate pixels found to be boundaries both by the Gaussian RF independent sensors model and the topographic modularity network model. Cyan number and points indicate F-measure under Gaussian RF model and boundaries found only by it. Red number and points indicate Δ-F after TM 2D network phase diffusion and boundaries found only by TM-2D. Note that image patches are shown at $^1/_2$ contrast to highlight boundaries found.

1.4 Discussion

In this work, we have shown that phase relaxation in coupled oscillators receiving inputs from simple image sensors (with unoriented Gaussian receptive fields) can provide image segmentation performance above and beyond the baseline, the segmentation performance that can be achieved by just using local contrast measurements. First, we have demonstrated that the type of graph clustering matters, the common spectral methods do not perform as well as relaxation in a Kuramoto model (Arenas). Second, we have demonstrated that the graph derived from the image structure matters. Specifically, we introduced topographic modularity, a modularity matrix that can capture the distance dependence in the statistics of image features. We find that a Kuramoto model using the topographic modularity matrix as phase couplings was the only network model that significantly outperformed the baseline.

A critical element of the successful model are its negative phase coupling weights, which introduce phase desynchronization at segment boundaries. Interestingly, we saw the best segmentation results with Gaussian receptive fields sizes similar to those measured in retina [**croner1995**]. In essence, the successful segmentation model provides a "cartoonization" [**yin2005**] of images - smoothing texture and variation within segments while maintaining crisp segment boundaries. Examples of phase relaxation results on two sample images are shown in Fig 1.16. Note the halos at the base of the lizard tail and surrounding the elk, where low contrast segment boundaries have been accentuated.

Figure 1.16: **Two examples of cartoonization:** Original images on left and resulting phase of TM-2D network computation on right

We quantify performance on the BSDS and show that anisotropic phase diffusion through the TM-2D improves F-measure significantly, in general, by increasing Precision while slightly deteriorating Recall. However, there are caveats with BSDS. First, BSDS is designed for state-of-the art image segmentation methods that requires combinations of sophisticated image filters, etc. In contrast, context extraction in the retina can only use the simple image feature extraction in retinal cells. Second, human segmenters that provide the ground truth in the BSDS databas can take advantage of the full image in color, while our model has only access to a $100x100$ pixel image patch in greyscale. Third, human segmenters use consciously and unconsciously high-level semantic information to draw boundaries while our algorithm just uses information from the image patch.

The model presented in this work is abstract and does not aim to directly capture biological features of retina. However, some evidence supports the plausibility of such a computation in retina. Ganglion cell spike trains have been observed to be periodic in the Gamma frequency [**neuenschwander1996**] and the phase of that frequency is transmitted with high precision through the thalamus to cortex along with precise spike times [**koepsell2009**]. The time to first spike in ganglion cells is quite precise [**gollisch2008**] and provides a possible mechanism for phase initialization following global suppression during eye saccades [**roska2003**]. Phase coupling without amplitude coupling could result simply from weak

retinal interactions, that slightly advance or delay spikes without adding or removing them. Both, phase synchronization and desynchronization through positive and negative weights in the model can be mapped onto excitation, inhibition and inhibition-of-inhibition circuits in retina. A retinal mechanism for fast adaptation of phase couplings to a particular stimulus image remains unclear. The spatial null model's distance dependent term, R_{ij} term in Eq. 1.2, which requires global knowledge in the model could be implemented in retina via sampling through long distance inhibitory interactions from polyaxonal amacrine cells [**olveczky2003**] or through eye movements implementing a temporal null model based on comparing feature similarity at a current stimulus location to feature similarity at a previous fixation.

Given the results of the model on BSDS and observations cited in the previous paragraph, we hypothesize that a coarse image segmentation or grouping/clustering of image features could be computed at the first stage of visual processing, in retina. While individual cell spike rates encode local stimulus contrast features through Gaussian-like receptive fields of ganglion cells, fine-time spike relationships across the cell population encode extra-classical receptive field features, such as extended segments. Fine-time correlations are multiplexed into ganglion cell spike-trains alongside with the rate-coded local stimulus features. Perhaps, the retina reduces uninformative correlations [**pitkow2012**] in stimulus with outer layers (photoreceptors, bipolars and horizontals) in order to reintroduce informative correlations in precise spike timing with inner layers (bipolar, amacrine, ganglions).

Chapter 2

Probabilistic Cell Assembly Model

2.1 Background

In this work, we introduce a novel, probabilistic latent variable model to detect "Cell Assemblies" (CAs) in spiking neural data. Given a corpus of observed sparse binary variables, "spike-words", the task is to infer the sparse activity of a set of binary latent variables, CAs. CAs are noisy repeats of groups of nearly synchronous cells, commonly coactive within a small time window, in observations. Our model is based on the "Noisy-OR model" [**heckerman1990**], used previously for disease and topic modelling. Analogous to binary soft-clustering, we wish to assign probabilistic binary observations to binary latent states. Each component in the latent state, if set to one, reduces the probabilities of certain populations of neurons to be silent. Thus, if there are noisy repeats of the same firing configurations in the observation data set, this repeating pattern is represented by a latent component. The conditional probability kernels of different latent components must be learned from the data in an expectation maximization scheme, involving inference of latent states and parameter adjustments. We apply our model to spiking responses recorded in retinal ganglion cells (RGCs) during stimulation with a movie.

State-of-the-art models of retina describe the function of (RGC) retinal ganglion cells, the output neurons in retina, as a bank of linear filters and pointwise nonlinearities that decorrelate stimulus features in space and time, reducing the redundancy in the retina's representation [**barlow1961**]. The ganglion cells encode local stimulus features in the spike rates [**kuffler1953**]. Computational models reflecting the text book view, such as the linear nonlinear Poisson (LNP) model and independent generalized linear model (GLM), predict RGC responses to a simple white noise stimulus [**schwartz2006**] with reasonable accuracy. Considering nearest-neighbor, pairwise activity correlations improves decoding of retinal responses to white noise [**pillow2008**]. However, model prediction by independent encoding models becomes rather poor [**schwartz2006**], and even encoding models that include second-order correlations fail to replicate responses to ecologically relevant natural movie stimuli [**chichilnisky2016**].

In addition to the explanatory gap between theory and observed activity, anatomical observations beg explanation. Circuitry in the retinal network is exquisitely complex, consisting of >60 distinct neuron types stratified into at least 12 parallel and interconnected circuits providing roughly 20 diverse representations of the visual world [**masland2011**], [**masland2012a**], [**werblin2011**], [**gollisch2010**]. Simple linear spatio-temporal filtering requires only a handful of cell types in the outer retina, leaving the bulk of the network unexplained. By "occam's razor", the simple textbook view must be at least incomplete.

We approach the topic from a statistical stand point, seeking hallmarks in retinal neural activity that cannot be explained by standard retinal modeling. We apply unsupervised learning to find repeating patterns of co-activity in neural activity. These found patterns will then be visualized and further analyzed in terms of underlying mechanisms and computational function. The application to retinal data is presented as an example, our unsupervised learning model can be applied to any spiking neural recording data, or any similar clustering problems involving binary data.

In this work, we present the model, and validate its performance on synthetic data, building up a performance assessment framework that can be applied to real data, which lacks correct answers. We then use the model to extract latent structure in retinal spiking responses to different types of stimuli. We show biological results which, while early and incomplete, point to activity in retinal spike-trains beyond local contrast representations independent encoded in RGC spike rates. The remainder of this paper is structured as follows. In section 2.2, we motivate the model and derive the math which underlies it, clearly stating assumptions and design choices. Section 2.3 we explain how the model parameters are learned from spike-word observations using an iterative expectation maximization algorithm. We describe a greedy approximate inference procedure for the latent variables and derive learning rules for model parameters. In section 2.4, we provide a procedure for validating the model's performance on synthetic data. We provide details for model synthesis and data generation procedures, including a discussion of parameters and pseudocode. We synthesize two models to match moments in distributions if the spike-words they generate to those observed in spike-trains recorded from population of 55 offBriskTransient ganglion cells from *in vitro* rat retina responding to white noise and natural movie stimuli. Interestingly, we find that parameters for best fit models to the two stimuli differ in interpretable ways, with responses to natural movie containing stronger cell assembly activity. Further, we show that a model trained on synthetic data matched to natural movie responses correctly learns CA structure and does so robustly across multiple models trained on the same data corpus. This last fact allows us some measure of confidence in the structure discovered in real neural data.

In section 2.5, we apply the algorithm to real spike-trains collected from a diverse population of retinal ganglion cell-types responding to both white noise and natural movie stimuli. We compare models trained on responses to different stimuli as well simulated data from a GLM responding to the natural movie stimulus. We develop a battery of intuitive and informative metrics along the way, with which we can quantify and characterize CAs discovered. Using these metrics to guide our exploration, we find a number of crisp CAs with temporally precise responses to natural movie stimulus that are reliable from trial-to-trial.

We also find some CAs with elongated structure, revealing structure in retinal activity beyond pairwise nearest-neighbor correlations previously proposed. We provide a few tantalizing examples of CAs that cross the ON-OFF cell-type boundary and align with prominent large scale stimulus features shortly before activation. Interesting CAs are learned robustly across multiple trained models (and even redundantly within individual models). Spike-words observed during CA activations have low probability under an independent GLM null model.

Finally, in the Discussion section, we address factors which limit our analysis in the current data set, collected before the advent of the model. We consider other neural data to which this model might be successfully applied. We speculate what additional questions could be asked with more complete data, and suggest additional experiments to further explore the model and the correlational structure of retinal activity.

2.2 Cell Assembly Model

A standard approach in analyzing spike rasters is to bin the data in time, with a bin width small enough so that the resulting data is binary, i.e., for every neuron a time bin has either one or zero spikes. Thus the observation data are sequences of binary vectors $\vec{y}(1), \vec{y}(2), ..., \vec{y}(T)$, with T the number of observations.

Here we design a probabilistic latent variable model to analyze the structure in observation vectors. The latent variables in the model are also binary. Each component of the latent vector $\vec{\mathbf{z}}$ is an indicator that in a time bin a cell assembly is active, which probabilistically causes a certain subset of cells to fire. We assume that observations in all time bins can be modeled with a fixed set of cell assemblies. In other words, we assume that cell assemblies can be switched on and off over time but their individual structure is stationary across the observations. Figure 2.1 shows a schematic of the model.

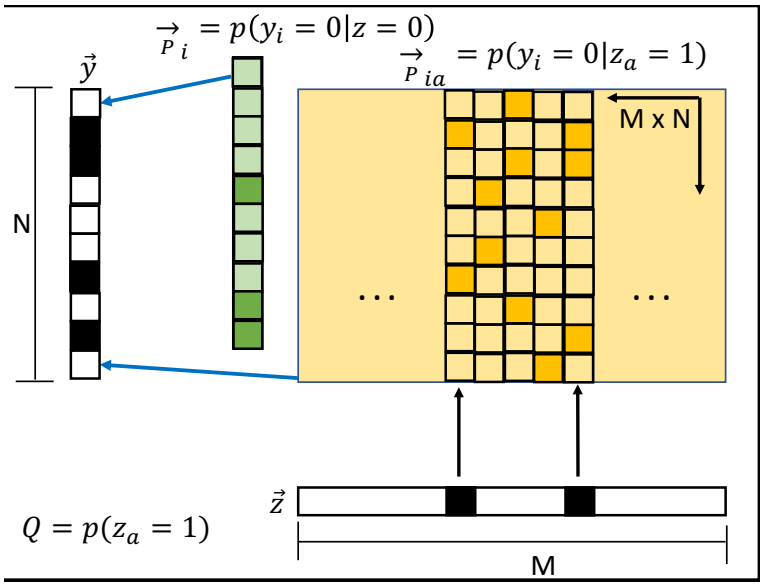


Figure 2.1: **Schematic of Cell Assembly model**: Observed spikes within spike-words $\vec{y}$ arise from two sources. First, each cell has some probability of firing without any cell assembly activity, expressed by N-vector $\vec{P}_i$. The second cause of cell activity is cell assembly activity, expressed by the $\bar{P}_{ia}$ matrix and $\vec{z}$. Finally, the scalar Q parameter sets a binomial prior on the activity in the latent variable, $\vec{z}$.

The model assumes that different cell assemblies which are active simultaneously, increase the firing probability of a cell, according to a noisy-OR combination. Specifically, the generative model for an observation vector is given by the product

$$p(y_i = 0|\vec{z}) = \prod_{a=1}^{M} p(y_i = 0|z_a = 1)^{z_a} \cdot p(y_i = 0|z_a = 0)^{1-z_a} \tag{2.1}$$

Note that (2.1) is a noisy version of the OR function $y_i = f(\mathbf{z}) = 1 - \prod_a z_a$. A similar model was proposed for analyzing relationships between diseases and symptoms by Heckerman [**heckerman1990**].

Second, the latent causes of observations are assumed to be sparse, that is, each observation vector is explained by a few active cell assemblies. This means that the majority of cell assemblies are inactive in any particular observation and it allows us to reduce the number of free parameters in the model by applying a mean field approximation. We assume that if cell assemblies are inactive, they all have the same (average) influence on the generation of a data vector, there are no individual differences between inactive assemblies. Rather than modeling the influence on each cell y_i by a vector of conditional probabilities, it can be modeled by a

single parameter for each component of the data vector:

$$P_i := p(y_i = 0|\vec{z} = \vec{0}) = \prod_{a=1}^{M} p(y_i = 0|z_a = 0) \tag{2.2}$$

and with this definition, equation (2.1) can be approximated as:

$$p(y_i = 0|\vec{z}) = P_i^{\left(1-\frac{|\vec{z}|}{M}\right)} \prod_{a-1}^{M} (P_{ia})^{z_a} \tag{2.3}$$

where $\vec{P_i} \in [0,1]^M$ is the vector of free parameters describing probabilities that cells are silent given no cell assembly is active. Further, $\bar{P}_{ia} \in [0,1]^{NxM}$ is the matrix of free model parameters describing the conditional probabilities of cells to be part of a cell assembly. That is, $P_{ia} = p(y_i = 0|z_a = 1)$.

A third model assumption is conditional independence of an observation vector $\vec{y}$, given a vector of latent variables $\vec{z}$. With this, the conditional probability of an arbitrary observation vector can be written:

$$p(\vec{y}|\vec{z}) = \prod_{i=1}^{N} \left[P_i^{\left(1-\frac{|\vec{z}|}{M}\right)} \prod_{a=1}^{M} (P_{ia})^{z_a} \right]^{(1-y_i)} \left[1 - P_i^{\left(1-\frac{|\vec{z}|}{M}\right)} \prod_{a=1}^{M} (P_{ia})^{z_a} \right]^{y_i} \tag{2.4}$$

A fourth model assumption is that the activation of different cell assemblies is uniform and independent. Thus the prior on $\vec{z}$ is given by a binomial distribution:

$$p(\vec{z}) = \binom{M}{|\vec{z}|} \cdot Q^{|\vec{z}|} \cdot (1-Q)^{\left(M-|\vec{z}|\right)} \tag{2.5}$$

with scalar parameter $Q = p(z_a = 1) << 1 \in [0,1]$ the probability that any individual cell assembly z_a is active. Combining prior (2.5) and likelihood (2.4), yields the joint probability $p(\vec{y}, \vec{z})$ which, for fixed data probability, is proportional to the posterior probability $p(\vec{z}|\vec{y})$:

$$p(\vec{z}|\vec{y}) \propto p(\vec{y}, \vec{z}) = p(\vec{y}|\vec{z})p(\vec{z}) \tag{2.6}$$

The joint probability for a single cell's activity y_i and a latent vector $\vec{z}$ is given by:

$$p(y_i, \vec{z}) = \binom{M}{|\vec{z}|} \cdot Q^{|\vec{z}|} \cdot (1-Q)^{\left(M-|\vec{z}|\right)} \left[P_i^{\left(1-\frac{|\vec{z}|}{M}\right)} \prod_{a=1}^{M} (P_{ia})^{z_a} \right]^{(1-y_i)} \left[1 - P_i^{\left(1-\frac{|\vec{z}|}{M}\right)} \prod_{a=1}^{M} (P_{ia})^{z_a} \right]^{y_i} \tag{2.7}$$

Taking the natural logarithms, we get:

$$
\begin{aligned}
log\, p(y_i, \vec{z}) = log \binom{M}{|\vec{z}|} + |\vec{z}\,| log\, Q + \Big(M - |\vec{z}|\Big) log\, [1-Q] \qquad (2.8a)\\
+ (1-y_i)\Big(1 - \frac{|\vec{z}|}{M}\Big) log\, P_i \\
+ (1-y_i) \sum_{a=1}^{M} z_a log\, P_{ia} \\
+ y_i log \left(1 - P_i^{\left(1-\frac{|\vec{z}|}{M}\right)} \prod_{a=1}^{M} P_{ia}^{z_a}\right)
\end{aligned}
$$

Observing that models often did not use all available cell assemblies, we explored an alternative prior on cell assembly activation that allows individual cell assemblies to have a different activation probability. The "Homeostatic Egalitarian" prior **[perrinet2010]** encourages all elements of $\vec{z}$ to be active an approximately equal number of times by decreasing each $p(z_a = 1)$ proportional to its previous activation. We define $\vec{Q} \in R^M$ where an element $Q_a = p(z_a = 1) << 1 \in [0,1]^1$ is the probability that cell assembly a is active. This yields:

$$
p(\vec{z}) = \prod_{a=1}^{M} p(z_a) = \prod_{a=1}^{M} Q_a^{z_a}\, (1-Q_a)^{(1-z_a)} \quad \text{with} \quad Q_a = Q\, \frac{\frac{1}{M}\sum_{a'=1}^{M} r_{a'}(t)}{r_a(t)} \qquad (2.9)
$$

where Q_a is a weighted version the still scalar Q parameter and $r_a(t)$ is the activation rate of cell assembly a at time t, i.e. the number of times it has been inferred active during the EM learning algorithm. With the new, slightly more general activity-dependent prior, the activation probability cell assembly a is scaled by the prior activation history of that cell assembly. Taking the natural logarithm of Eq. 2.9, changes the first line in the log joint in Eq. 2.8 to

$$
log\, p(\vec{z}) = \sum_{a=1}^{M} z_a \left\{ logQ + log \frac{\frac{1}{M}\sum_{a'=1}^{M} r_{a'}(t)}{r_a(t)} \right\} + (1-z_a) log \left(1 - Q \cdot \frac{\frac{1}{M}\sum_{a'=1}^{M} r_{a'}(t)}{r_a(t)}\right) \qquad (2.10)
$$

2.3 Model Training

The model is trained using Expectation Maximization to perform iterative gradient ascent procedure on the log joint. Model parameter values are initialized to nearly silent with some small Gaussian random variability. For each observed spike-word $\vec{y}$, learning proceeds in two steps. First, the latent variables ($\vec{z}$) are inferred with the current fixed model parameters, Sec. 2.3). Then, model parameters are adjusted to maximize the derivative of the log joint Eq. 2.8 with respect to each parameter, Sec. 2.3.

Inference of latent variables

Given a fixed model and a single observed $\vec{y}$, we run the model in reverse to infer the most likely latent state $\vec{z}$ that generated the observed state. Generally, finding the optimal binary latent vector $\vec{z}$ for a given binary observation $\vec{y}$ is a combinatorial problem that can only be solved by an exhaustive search over all possible latent states, which quickly becomes computationally prohibitive as the length of $\vec{z}$ grows. For tractability, we solve a greedy relaxation of this problem, which finds the a small number of the best cardinality-1 $\vec{z}$ solutions and chooses the combination of z_a's which maximizes the joint in that smaller subset using combinatorial search.

Specifically, the greedy inference algorithm proceeds as follows: We compute the joint probability in Eq. 2.8 of all M 1-hot $\vec{z}$'s as well as the $\vec{z} = \vec{0}$ solution. Sorting the M+1 values in descending order, we form combinations of cell assemblies that individually yield higher joint probability than the $\vec{z} = \vec{0}$ solution. Two parameters of the inference procedure allow us to adjust the number of 1-hot solutions to include when trying combinations of z_a's. The first, I_0, allows a number of z_a's with joint probability lower than $\vec{z} = \vec{0}$ into the combination step. The second parameter, I_{max}, sets a maximum on the number of z_a's to include in the combination step. With this smaller set of cell assemblies, we can tractably compute the joint probability of pairs, triplets and higher-order combinations of those z_a's that form solutions with $|\vec{z}| > 1$. The resulting inferred $\vec{z}$ is the one which maximizes Eq. 2.8 out of all combinations checked.

We choose parameters $I_0 = 9$ and $I_{max} = 10$, which uses the top 10 1-hot cell assemblies in the combination step. While the inference procedure would be faster with smaller values, it is more likely to infer a sub-optimal $\vec{z}$. This procedure acts as an interpolation between the full combinatorial search of all possible $\vec{z}$'s and the efficient but greedy approach of taking the best 1-hot / 0-hot $\vec{z}$. Note that full combinatorial search results from choosing $I_0, I_{max} > M$. The best 1-hot / 0-hot $\vec{z}$ solution is obtained by setting $I_{max} = 1$. If one chooses $I_0 = 0$ and $I_{max} = M$, the the inference procedure only considers 1-hot $\vec{z}$'s that have higher joint probability than the $\vec{z} = \vec{0}$ solution. While this is a heuristic procedure, it works well in practice - correctly inferring ground truth $\vec{z}$'s and learning ground truth model parameters in synthetic data as well as inferring non-trivial $\vec{z}$'s and learning interesting cell assembly structure in real retinal data.

Learning model parameters

Given an observed $\vec{y}$ and an inferred $\vec{z}$ at each iteration of the EM algorithm, we adjust each parameter in the model to increase $p(\vec{y}, \vec{z})$ via gradient ascent. We compute derivatives of Eq. 2.8 w.r.t. each individual model parameter in Eqs. 2.13,2.14,2.15 and 2.16. In order to perform *unconstrained* gradient ascent, we use a logistic parameterization for all variables that describe probability values

$$P = \sigma(\rho) = \frac{1}{1 + e^{-\rho}} \tag{2.11}$$

where capital P's $\in [0,1]$ indicate probability values and lower-case ρ's can take any real value. With the logistic parametrization and the binomial $\vec{z}$ prior, Eq. 2.8 can be rewritten as:

$$\begin{aligned} log\, p(y_i, \vec{z}) = log \binom{M}{|\vec{z}|} + |\vec{z}\,| log\, \sigma(q) + \Big(M - |\vec{z}|\Big) log\, [1 - \sigma(q)] \qquad (2.12a) \\ + (1 - y_i)\Big(1 - \frac{|\vec{z}|}{M}\Big) log\, \sigma(\rho_i) \\ + (1 - y_i) \sum_{a=1}^{M} z_a log\, \sigma(\rho_{ia}) \\ + y_i log\, \left(1 - \sigma(\rho_i)^{\left(1 - \frac{|\vec{z}|}{M}\right)} \prod_{a=1}^{M} \sigma(\rho_{ia})^{z_a}\right) \end{aligned}$$

Derivatives with respect to each model parameter are shown below. We leave the calculation of derivatives to the reader.

$$\frac{\partial\, log\, p(y_i, \vec{z})}{\partial q} = |\vec{z}| \left(1 - \sigma(q)\right) - \left(M - |\vec{z}|\right) \sigma(q) \qquad (2.13)$$

$$\frac{\partial\, log\, p(y_i, \vec{z})}{\partial \rho_i} = \left(1 - \frac{|\vec{z}|}{M}\right) \left(1 - \sigma(\rho_i)\right) \left[(1 - y_i) - \frac{y_i\, C_i}{(1 - C_i)}\right] \qquad (2.14)$$

$$\frac{\partial\, log\, p(y_i, \vec{z})}{\partial \rho_{ia}} = z_a \left(1 - \sigma(\rho_{ia})\right) \left[(1 - y_i) - \frac{y_i\, C_i}{(1 - C_i)}\right] \qquad (2.15)$$

$$\text{where} \quad C_i \coloneqq \sigma(\rho_i)^{\left(1 - \frac{|\vec{z}|}{M}\right)} \prod_{a=1}^{M} \sigma(\rho_{ia})^{z_a}$$

The learning rule for q with the alternative "Egalitarian Homeostatic" prior in Eq. 2.10 is

$$\frac{\partial\, log\, p(y_i, \vec{z})}{\partial q} = (1 - \sigma(q)) \sum_{a=1}^{M} \left[z_a - \frac{(1 - z_a)\, Q_a}{1 - Q_a}\right] \qquad (2.16)$$

where Q_a is defined in Eq. 2.9.

Inspection of the learning rules reveals some nice symmetry and intuitive interpretations. C_i is just $p(y_i = 0|\vec{z})$, the conditional probability that cell y_i is silent given its participation in all active cell assemblies (P_{ia} and $\vec{z}$) as well as its own inherent chattery-ness (P_i). The C-ratio ($\frac{C}{1-C}$) in Equations 2.14 and 2.15 can be seen as the relative probability (or "log odds") that a cell is silent rather than active. These ratios contribute to the derivative with a

negative sign to decrease a parameter value, either ρ_i or ρ_{ia} when the cell is active ($y_i = 1$) with a strength proportional to the models misprediction that the cell should be silent. The derivative with respect to ρ_{ia} in Eq. 2.15 is gated by whether a cell assembly is active (i.e., $z_a = 1$) and the derivative with respect to ρ_i in Eq. 2.14 is weighted by the probability that cell assemblies are inactive ($1 - |\vec{z}|/M$). That is, parameters in column a of ρ_{ia} are only changed when z_a has been active. Further, since the ρ_i parameters are modeling cell activity in the absence of cell assembly activity, the changes to ρ_i are weighted by how many cell assemblies are inactive.

With the learning rule for the "Homeostatic Egalitarian" prior on $\vec{z}$ in Eq. 2.16, $Q_a = p(z_a = 1)$ and is directly analogous to C_i. Its ratio is the relative probability that cell assembly a is active rather than inactive and it contributes with a negative sign, decreasing the scalar q parameter, when z_a is inactive. For the binomial prior on $\vec{z}$, the derivative with respect to q in Eq. 2.13 contains two terms with opposite signs that represent discrepancies between model and observations. The first term, which is proportional to the number of active cell assemblies $|\vec{z}|$ and to the model's prediction that cell assemblies are inactive ($1 - \sigma(q)$), tends to increase the value of q to predict more active cell assemblies. The second term decreases q when the model overpredicts the number of active cell assemblies.

2.4 Model Validation on Synthetic Data.

As a key initial step, we validate how well the model is able to learn known structure embedded into synthetic data because we can not be certain that neural data structure discovered by the model reflects true causal structure since neural data does not provide ground truth. We construct a realistic synthetic ground truth (GT) model by setting parameters to fit moments of generated spike-words to those observed in retinal spike trains. With the data corpus of generated $(\vec{y}, \vec{z})$ pairs, we learn model parameters and infer $\vec{z}$ activity, as described in section 2.3. Post-learning, we assess model performance by comparing learned model parameters and inferred $\vec{z}$ activity to the ground truth values used to generate the data. We demonstrate that structure learned by multiple models trained on different partitions and random samplings of the same data reflects structure that also exists in the ground truth with high probability. For reference, we provide a table of model synthesis and data generation parameters with a short description of their effect on model parameters here in table 2.1. They are discussed in greater detail within.

Hyper-parameter	Effect on synthetic model parameters ($\bar{P}_{ia}$, $\vec{P}_i$, and Q) and data generation
N	number of cells or elements in $\vec{y}$
M	number of cell assemblies or elements in $\vec{z}$
K	number of active cell assemblies. Truncated binomial probability
K_{min}	minimum number of active cell assemblies in data generation $\lvert\vec{z}\rvert \sim [\text{Bin}(K)]_{K_{min}}^{K_{max}}$
K_{max}	maximum number of active cell assemblies in data generation
C	number of cells per cell assembly. Truncated binomial probability
C_{min}	minimum number of cells per cell assembly. ${}^{cells}/_{CA} \sim [\text{Bin}(C)]_{C_{min}}^{C_{max}}$
C_{max}	maximum number of cells per cell assembly.
$\mu_{P_{ia}}$	mean difference from deterministic values in P_{ia}. Truncated normal distribution.
$\sigma_{P_{ia}}$	std in difference from deterministic values in P_{ia}. i.e. $[\mathcal{N}(\mu_{P_{ia}}, \sigma_{P_{ia}})]_0^1$
μ_{P_i}	mean difference from deterministic values in P_i. Truncated normal distribution.
σ_{P_i}	std in difference from deterministic values in P_i. i.e. $[\mathcal{N}(\mu_{P_{ia}}, \sigma_{P_{ia}})]_0^1$
σ_Q	std of truncated Gaussian on Q with mean $= {}^{K}/_{M}$

Table 2.1: Hyper-parameters used for model synthesis and data generation.

Model Synthesis

A trained model can only discover structure that is embedded in the data. Therefore, to avoid embedding pathological structure into the training data due to variance on binomial and normal distributions, the procedure for ground truth model synthesis and data generation is somewhat involved. We carefully balance randomness by setting reasonable bounds on the resulting model and generated data statistics and resampling from distributions when bounds are exceeded. With the additional bounding and resampling steps, we ensure structure similar to what we expect to see in real retinal data, avoiding cell assemblies which are too large or too small as well as probability values greater than 1 or less than 0.

A brief overview of the model synthesis, data generation procedure is provided here and unpacked below. Dimensions of the system are determined with N, number of cells and M, the number of cell assemblies. Each of N neurons is driven by its own internal activity/processes (represented in the $\vec{P}_i$ parameter vector $\in [0,1]^N$) and its participation in M cell assemblies (represented in the $\bar{P}_{ia}$ parameter matrix $\in [0,1]^{NxM}$). First, deterministic cell assembly structure is built into the binary $\bar{P}_{ia}$ matrix, with columns resampled to ensure reasonably sized cell assemblies with minimal overlap. Then, stochasticity is added by varying

binary values based on a normal distribution truncated at 0 and 1. Next, single cell noisiness is built into the $\vec{P}_i$ vector by similarly drawing values from a truncated normal distribution. Finally, the independent Bernoulli probability that any single cell assembly will be active (represented in the scalar Q parameter $\in [0, 1]$) defines the cardinality in the latent variable vector, that is $|\vec{z}|$.

Construction of $\bar{P}_{ia}$, defining which cells participate in which cell assemblies, is a multistep process. In addition to a description here, we provide pseudo-code detailing how $\bar{P}_{ia}$ is constructed from model hyper-parameters $\{C, C_{min}, C_{max}, \mu_{P_{ia}}$ and $\sigma_{P_{ia}}\}$. First, C defines a Bernoulli probability of drawing a 1 for each element in the binary cell assembly membership matrix, $p(y_i = 1|z_a = 1) \in \{0, 1\}$, which is equivalent to $1 - \bar{P}_{ia}$. Because a Bernoulli random process has a variance of $C(1 - C)$, it is possible to generate data with pathological structure - like cell assemblies containing ≤ 1 active cells. The C_{min} and C_{max} parameters determine lower and upper bounds on the number of cells in any given assembly. When the sum of a column in $1 - \bar{P}_{ia}$ exceed these bounds, the N values within that column are resampled. After ensuring that CAs are of reasonable size, we implement an iterative procedure that shuffles cells within cell assemblies to minimize the overlap between different cell assemblies and which encourages all cells to participate in approximately the same number of cell assemblies. On each iteration, we activate a cell which participates in few cell assemblies in a random CA and inactivate one of the cells in that CA. We compare the average overlap of all cell assemblies and keep the change if that value was decreased (see pseudo-code for further details). Given deterministic cell assemblies of reasonable size with minimal redundancy, values in the deterministic $\bar{P}_{ia}$ matrix are replaced by values drawn from a truncated normal distribution where $\bar{P}_{ia}^{prob} = [\bar{P}_{ia}^{det} \pm \mathcal{N}(\mu_{P_{ia}}, \sigma_{P_{ia}})]_0^1$ and samples which exceed the truncation values are resampled. The result of this procedure is a probabilistic ground truth model and noisy data generated stochastically from that model on which to train.

Pseudo-code for ground truth model $\bar{P}_{ia}$ matrix construction

1. Generate deterministic, binary $\bar{P}_{ia}$:

   (a). construct sparse binary matrix, ensuring reasonable #Cells per CA

   – draw elements in $\bar{P}_{ia}$ from binomial with probability, $p(1) = C$

   – redraw column while not $C_{min} < \sum_a P_{ia} < C_{max}$

   (b). change cells in CAs to minimize CA overlap and equalize cells' participation across CAs

   compute overlap matrix, OVL_a, for all CA pairs

   while count $<$ maxCount:

   randomly draw CA j from M

   compute $CperA$, N-vector of number of CAs per cell

   find cell i that participates in few CAs. Sort $CperA$

if cell i not active in CA j:
  activate cell i
  randomly inactivate an already active cell in CA j
compute OVL_b
if mean(OVL_b) < mean(OVL_a):
  keep the change
  $OVL_a = OVL_b$

2. Add stochasticity into $\bar{P}_{ia}$.

for each element, p, in $\bar{P}_{ia}$:
  q = -1 # to enter while loop
  while not $0 \leq q \leq 1$:
    draw q from normal distribution $\mathcal{N}(p \pm \mu_{P_{ia}}, \sigma_{P_{ia}})$
  replace p with q; i.e., $P_{ia} = q$

3. Invert $\bar{P}_{ia}$ so that $\bar{P}_{ia} \coloneqq p(y_i = 0 | z_a = 0)$.
  $\bar{P}_{ia} = 1 - \bar{P}_{ia}$

The $\vec{P}_i$ vector defines the probability that each cell i will be active without being caused by cell assembly activity, $p(y_i = 1 | z = 0) = 1 - \vec{P}_i$. Values in $\vec{P}_i$ are drawn from a truncated normal distribution similar to values in $\bar{P}_{ia}$. That is, $\vec{P}_i \sim [1 - \mathcal{N}(\mu_{P_i}, \sigma_{P_i})]_0^1$. Finally, the scalar Q parameter represents the probability that any single cell assembly is active, $p(z_a = 1)$, assumed independent of the activity of other cell assemblies. Q is drawn from a truncated normal distribution with mean K/M and variance σ_Q, i.e., $Q \sim \mathcal{N}(K/M, \sigma_Q)$. The K hyper-parameter determines the sparseness of the latent $\vec{z}$ or how many cell assemblies are active in any single observation.

Data Generation

Once the GT model is synthesized (Q, $\vec{P}_i$, $\bar{P}_{ia}$ parameters fixed as described in 2.4), we construct training and test data $\vec{y}, \vec{z}$ pairs to learn a model and validate its performance. A sparse binary $\vec{z}$ is generated by independently sampling each z_a from a Bernoulli distribution with $p(z_a = 1) = Q$. Recalling that Q is itself drawn from a Gaussian distribution with mean K/M, $\vec{z}$ will be K-hot on average, that is having K nonzero values. Due to variance in binomial distributions, which determines the sparsity of $\vec{z}$, we set reasonable bounds on the number of nonzero entries or active cell assemblies in $\vec{z}$ with the K_{min} and K_{max} parameters. If a sampled $\vec{z}$ falls outside these bounds, it is discarded and resampled.

$$z_a \sim Bern(Q) \quad \text{subject to} \quad K_{min} \leq |\vec{z}| \leq K_{max}$$

From the sampled $\vec{z}$, the GT model is run in the generative direction to produce a vector of probabilities, $p(y_i|\vec{z})$. From these probabilities, binary spike-word $\vec{y}$ is then constructed by independently sampling the state of each cell i from a Bernoulli distribution with parameter, $p(y_i = 1|\vec{z})$. See Fig. 2.1 for an illustration of the generative process.

$$y_i \sim Bern(\ p(y_i = 1|\vec{z})\ ) \quad \text{where} \quad p(y_i = 1|\vec{z}) = 1 - P_i \prod_{a-1}^{M} (P_{ia})^{z_a}$$

Fitting Model parameters to spike-word statistics

Early experiments showed that models were more easily learned when their parameters were nearly deterministic, values close to 0 and 1, and when single cells were not noisy, $\vec{P_i} \approx 1$. With increased stochasticity and single-cell noise, models learned fewer cell assemblies and required more data to fully sample the probability distributions from observed binary spike words.

To test the performance of these models on data similar to neural data, we set parameters for the synthetic model in order to match some key moments of spike word distributions measured from *in vitro* rate RGC responses to white noise and natural movie stimuli. Data obtained from the G. Field lab at Duke University consists of spike trains 55 off brisk transient cells responding to white noise and natural movie stimuli. The data are described fully in section 2.5. Spike trains are binned at 5ms to generate spike-words, generally sparse binary vectors representing cells which are coactive. We measure distributions of single cell average activity, average pairwise coactivity and spike-word length.

Performing a grid search over model parameters K, K_{min}, K_{max}, C, C_{min}, C_{max} $\mu_{P_{ia}}$, $\sigma_{P_{ia}}$, μ_{P_i}, σ_{P_i}, we select values that minimize a linear combination of QQ values for the three distribution moments. QQ values capture the average difference between a pair of cumulative density functions, with near-zero values indicating very similar distributions. Figure 2.2 shows distributions for spike-word length (left), single cell activity (center) and pairwise coactivity (right) for measured and the data generated by the synthetic model which best fits retinal responses to natural movie stimulus, blue. Note closer matches between red and blue curves as compared to red and green in top plots, that blue curves are closer to unity line in bottom plots and that blue QQ values are smaller than green. While spiking statistics from synthesized model are not a perfect match to observed data (indicated by differences in blue), these synthetic model parameters provide both a challenging, realistic data set with which to test our algorithm and ground truth with which to validate its performance.

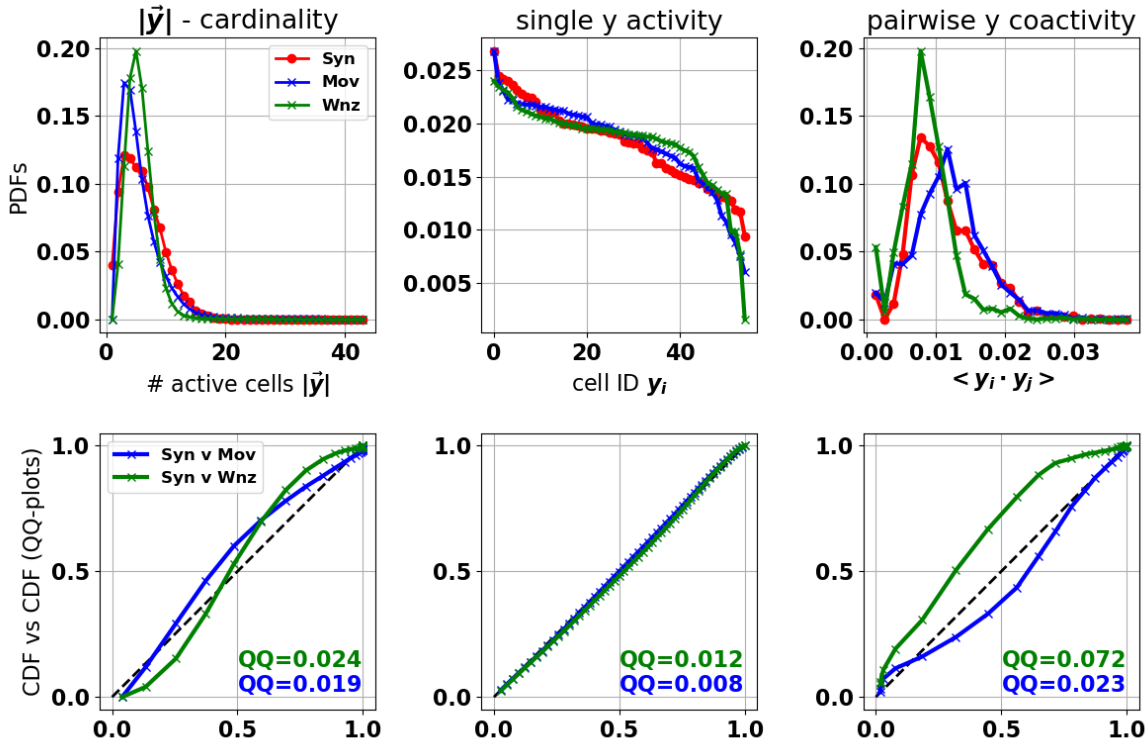


Figure 2.2: **Fitting synthetic model to spike-word moments:** Comparison of spike-word moments from synthetic data fit to natural movie responses in red, retinal responses to white noise in green, and natural movie in blue. *Top row from left to right* shows probability density functions for spike-word length, $|\vec{y}|$, average single-cell activity, $\langle y_i \rangle$. and, pairwise cell coactivity, $\langle y_i \cdot y_j \rangle$. *Bottom row* shows quantile-quantile (QQ) plots - a pair of cumulative density function plotted against each other. See legend, left plot. *QQ* values measure average deviation from the unity line, larger values indicating differences in distributions.

Additionally, an interesting and interpretable result emerges from the difference in the synthetic model hyper-parameters fit to activity of the same RGC population in response to these two types of stimuli. The best-fit parameters, shown in table 2.2, reveal that, under the assumptions of the model, responses to natural movie stimulus contain fewer active cell assemblies (smaller K) in any observed spike-word, while each individual cell assembliy contains more cells (larger C) with stronger membership or participation (smaller $\mu_{P_{ia}}$) in that assembly, when compared to responses of the same cell population responding to white noise stimulus.

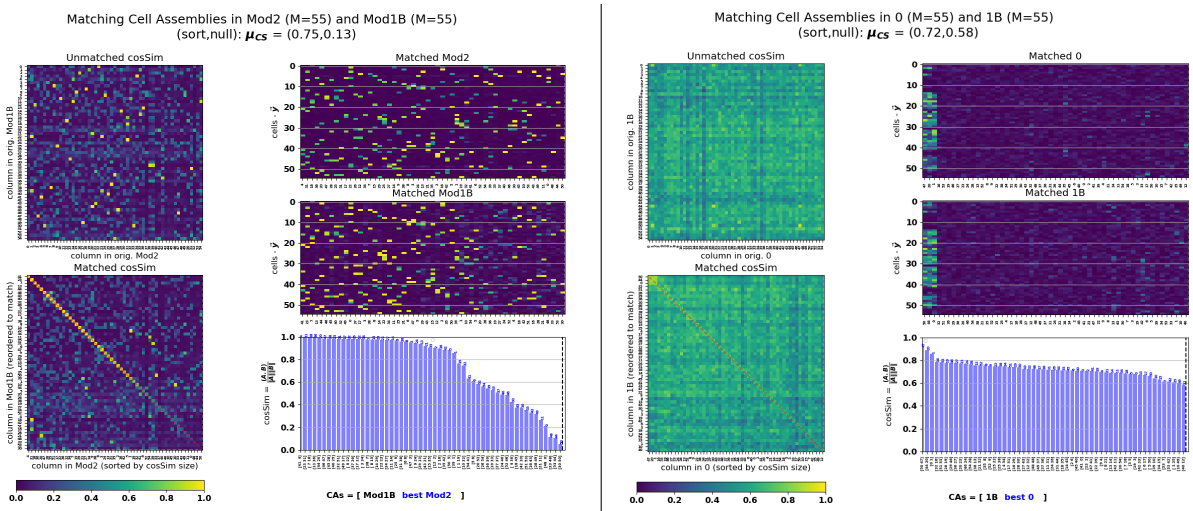
Stimulus	K	K_{min}	K_{max}	C	C_{min}	C_{max}	$\mu_{P_{ia}}$	$\sigma_{P_{ia}}$	μ_{P_i}	σ_{P_i}
Natural Movie	1	0	4	6	2	6	0.3	0.1	0.04	0.02
White Noise	2	0	4	2	2	6	0.55	0.05	0.04	0.02

Table 2.2: Hyper-parameters fit to offBriskTransient RGC responses to white noise and natural movie stimuli. Key differences highlighted in red.

Performance assessment and results on synthetic data

With each synthetic data set, we independently train multiple models on different splits and samplings of the data and compare models both to one another and to the ground truth model in order to assess the validity of structure that the model discovers. The cosine similarity (cs) measure between a pair of cell assemblies gives a normalized measure of the angle between them. In the special case of probabilities, since cell assemblies are vectors in $\mathbb{R}^N \in [0,1]^N$, cosine similarity is non-negative, being 0 if vectors are orthogonal and 1 if parallel. Computing cosine similarity for all cell assembly pairs in two equal-sized models yields an MxM matrix. Since the specific order of cell assemblies in a model is arbitrary due to initialization and sampling stocasticity, it is necessary to uniquely match cell assembly pairs. For this, we leverage the Hungarian algorithm **[Kuhn1955]**, which permutes matrix columns to minimize the trace of $1-cs$. The mean value along the diagonal of the matched cs matrix yields a single number to quantify the match between models. Figure 2.3a illustrates the process of matching up cell assemblies and quantifying a the match between a learned model and ground truth.

$$cs(P_{ia}, P_{ib}) = \frac{P_{ia} \cdot P_{ib}^T}{\|P_{ia}\|\|P_{ib}\|} \tag{2.19}$$



(a) Synthetic natural movie model

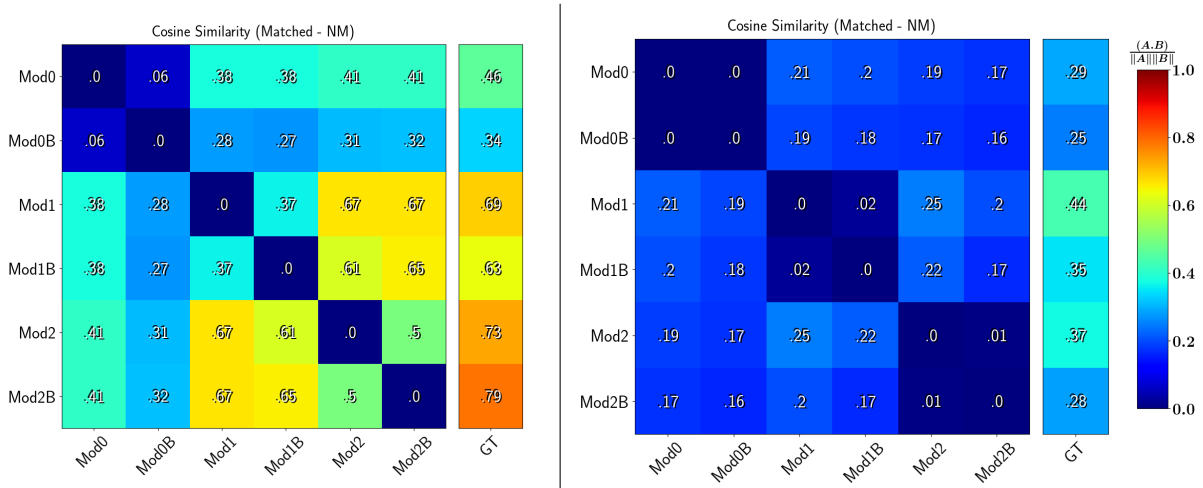
(b) Real white noise model

Figure 2.3: **Cosine similarity between models:** for synthetic natural movie responses (a) and real retinal responses to white noise(b). Within each panel, left column shows matrix of *cs* values between all cell assembly pairs across a pair models. Top, with arbitrary order due to learning algorithm stochasticity. Bottom, with cell assemblies matched across models based on *cs*. Right of each panel shows the pair of $\bar{P}_{ia}$ matrices with columns, cell assemblies, aligned to maximize *cs* across all matched pairs. *cs* for each match is shown in blue bars in panel bottom right. Panel b illustrates the necessity of a null model for the *cs* quality metric. Although both panels illustrate similar *cs* values after CA matching (0.75 vs 0.72), the improvement from null model before matching (0.13 vs 0.58) is quite different.

Raw values of this average cosine similarity can be biased by the statistics of vectors compared. Specifically, the measure can take systematically larger values for populations of random vectors with near-zero elements because it includes division by vector norms. This effect is particularly pronounced in models trained on retinal responses to white noise stimulus because $\bar{P}_{ia}$ is initialized with small random noise and cell assemblies do not change from that initialization when they are rarely inferred due to lack of structure in training data, see figure 2.3b. Thus, it is important to compare the cosine similarity measure averaged across all matched cell assemblies to a null model, which characterizes the expected average value for a random matching of the same group of vectors. We use the diagonal mean in the cosine similarity matrix prior to cell assembly matching (compare right top panel of figure 2.3b to same in figure 2.3a) for a null model.

In real retinal data, ground truth cell assembly structure and activity does not exist. Regardless, we can build confidence in the cell assembly structure discovered by the algorithm by a cross-validation protocol. Here, with synthetic data, we show that structure found reliably and robustly by multiple models trained independently on different splits and random

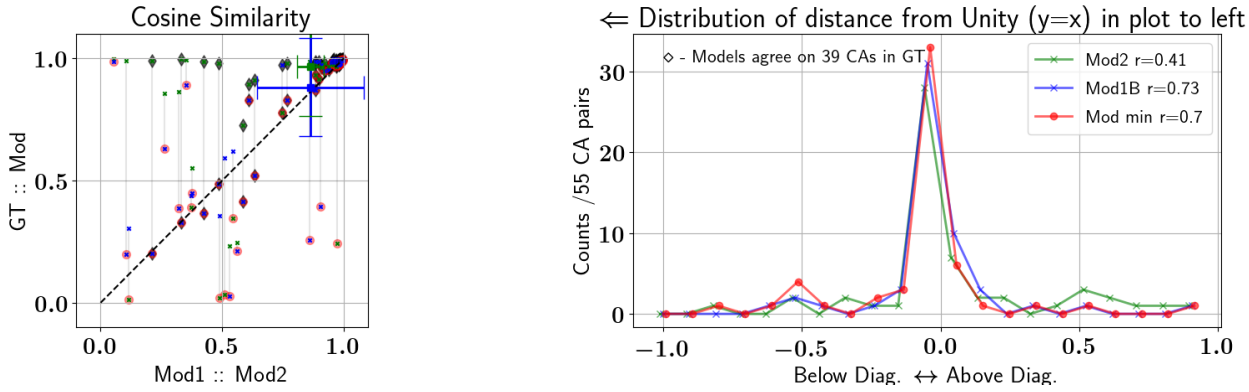
samplings of data also exists in the ground truth model with high likelihood. Fig. 2.4 shows the average CA cosine similarity between pairs of models as well as between each model and the ground truth. When multiple models have significant overlap, they also overlap with the ground truth to a similar degree in both the natural movie and white noise cases. Fig. 2.5 shows that this is also true for individual cell assemblies.



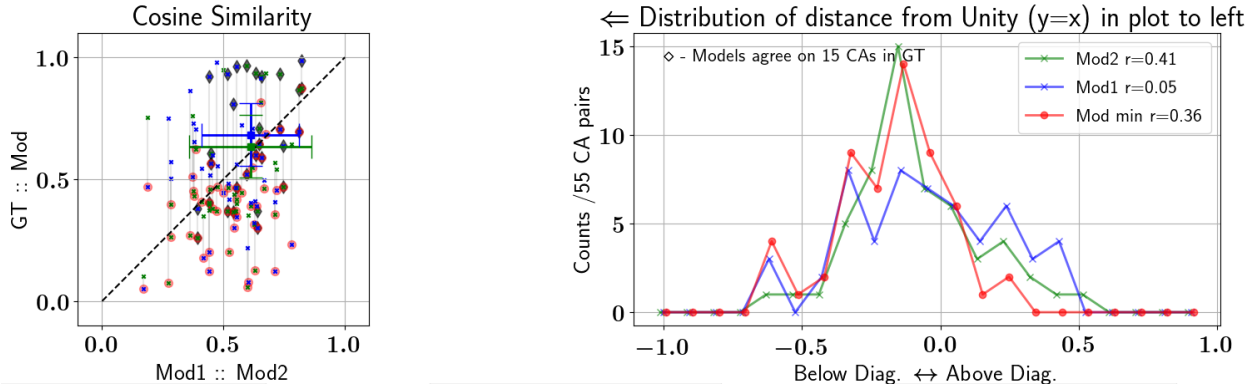
(a) Synthetic natural movie model

(b) Synthetic white noise model

Figure 2.4: **Modelling synthetic responses to natural movie vs. white noise stimulus:** Cosine similarity for 6 learned models and ground truth (GT) for synthetic natural movie responses (a) and synthetic white noise responses (b). Value and color in each box indicate improvement above null model after CA matching. Details for box at intersection of Mod2 and Mod1B (0.61) in panel a are illustrated in Fig 2.3a. Each model's overlap with the GT is roughly correlated with the model's overlap with other models. This correlation is further unpacked for Mod2 and Mod1B in panel a and for Mod2 and Mod1 in panel b in Fig 2.5a and 2.5b respectively.



(a) Synthetic natural movie model

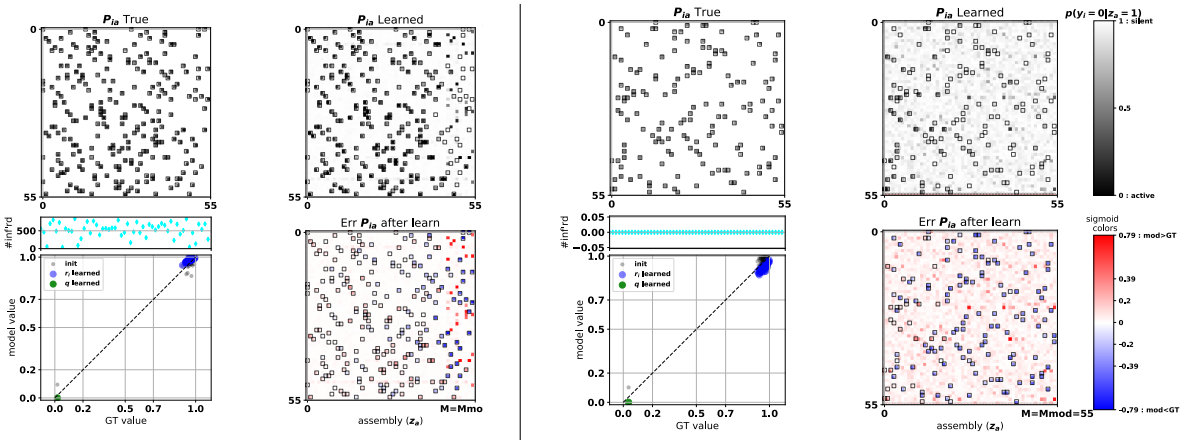


(b) Synthetic white noise model

Figure 2.5: **CA structure in multiple models matches GT**. In panel a, for synthetic natural movie responses and in panel b, for synthetic white noise responses. Within each panel, left plot shows *cs* between each matched CA in a model pair on the x-axis and between each model's CA and its matching GT CA on the y-axis. One model is in blue, the other in green, with red 'o' indicating smaller *cs* between model and ground truth. Black diamond indicate where two models agree on same ground truth CA. Larger blue and green '+' show error bars mean and std of each model's CA population. Right plot in panel shows distribution of points from the unity line. Mass near zero indicates CA triplets that share similar *cs* values, that is where structure in the GT is robustly found by multiple models and structure found by only one model is not in the GT. The CA match is stronger between model pair as well as between ground truth and each individual model with natural movie vs white noise model. Pearson correlation coefficients for three scatter groups in left plots are shown in legend on the right.

Directly comparing models fit to retinal response statistics to natural movie and white noise stimuli, we find interpretable differences in parameters of those fit models (end of section 2.4). Under the assumptions of the model, natural movie stimuli activate fewer cell assemblies

which contain more cells that more consistently participate in the assembly. By contrast, responses to white noise stimuli are captured in a model with noisier, smaller cell assemblies, more of which must be active on average to explain observed spike-words. Moreover, training models on data generated from these two synthetic models, we find difference in the model's ability to learn the structure in these data. Data containing larger and stronger cell assemblies fewer of which are active at any given time is understandably easier to learn. In figures **??** and **??**, we compare single learned models trained on data generated from synthetic natural-movie-response-like and white-noise-response-like models, respectively. We observe that structure is more easily found in synthetic responses to natural movies. Finally, we also observe that ground truth structure is more robustly learned across multiple trained models for synthetic natural-movie-like responses than for synthetic white-noise-like responses, compare figures 2.4a and 2.4b.



(a) Synthetic natural movie model

(b) Synthetic white noise model

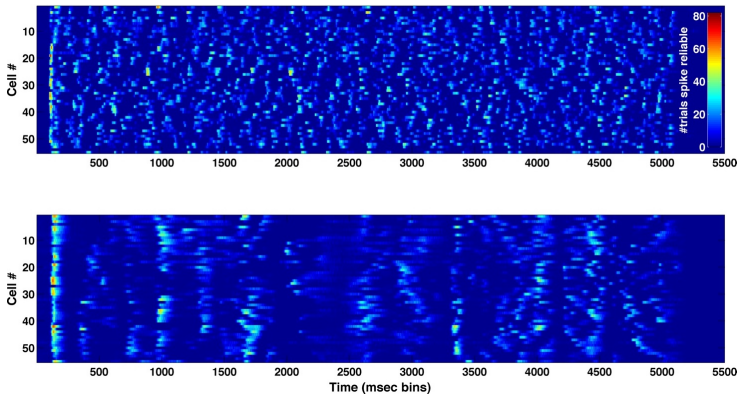
Figure 2.6: **Comparing Models learned on synthetic data fit to different stim responses:** natural-movie-like (a) and white-noise-like (b) responses. Within each panel, *top boxes* show $\bar{P}_{ia}$ matrix in GT and learned model, columns indicating CAs and y-axis, cells. *Right bottom* shows the signed error between GT and learned $\bar{P}_{ia}$'s. Note sigmoid colorbar to accentuate small differences. Unfilled boxes in learned $\bar{P}_{ia}$ and error show active cells in GT CAs. *Left bottom* scatter plot shows Q, in green, and $\vec{P}_i$, in blue, parameters in learned model (y-axis) vs. GT (x-axis). Points near $y = x$ indicate correctly learned parameters. Parameter initialization are shown in gray. Cyan points in *left middle* show the number of times each CA was inferred across all data after model learning. In the white noise model, fewer cell assemblies are learned as indicated by more blue and red in the signed error. The model also relies on noisier $\vec{P}_i$ parameters (blue o's further from 1 and below $y = x$ line) to account for increased spike-word variability. This model is more difficult to learn because cell assembly participation is weaker, lighter gray squares in GT $\bar{P}_{ia}$.

2.5 Retinal Data Exploration

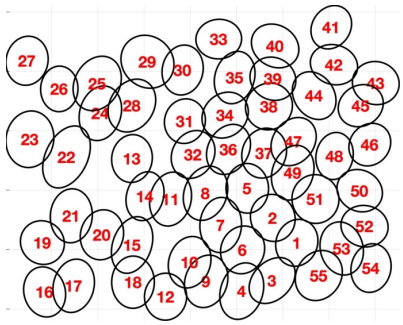
Outline

We apply the model to spike trains collected from *in vitro* rat retinal ganglion cells (RGCs). Activity from 329 cells of 11 different cell types was recorded using a multielectrode array by the lab of Greg Field. We used data from 55 Off-Brisk Transient (offBT), 39 Off-Brisk Sustained (offBS) and 43 On-Brisk Transient (onBT) cells. The remaining 8 cell-types did not have data from a sufficient numbers of cells for our analysis. Cell receptive fields (RFs) were fit using responses to 1 hour presentation of white noise stimulus. The data we analyze consists of RGC spike-train responses to 200 trial repeats of 5 second clips of white noise and natural movie stimulus. Natural movie stimulus from the "Cat Cam" data set [**betsch2004**].

Responses from offBT cells to white noise and natural movie, shown in Fig. 2.7, are clearly different visually. Loosely, responses to natural movie are more smeared out in time and more structured spatially across the cell population. Importantly, the geometric organization of cell RFs in 2D visual space is maintained in the cell ordering shown. That is, nearby RFs are adjacent on y-axis. Though individual cells seem less reliable in time under natural movie stimulation, our analysis aims to uncover whether trial-to-trial variability is shared across the population. In other words, we search for groups of cells within temporal smears in the bottom of Fig. 2.7, varying together so that population spike-words remain within single trials even if the precise time of spike-words relative to the stimulus changes. We call these groups, "cell assemblies" (CAs).



(a) responses



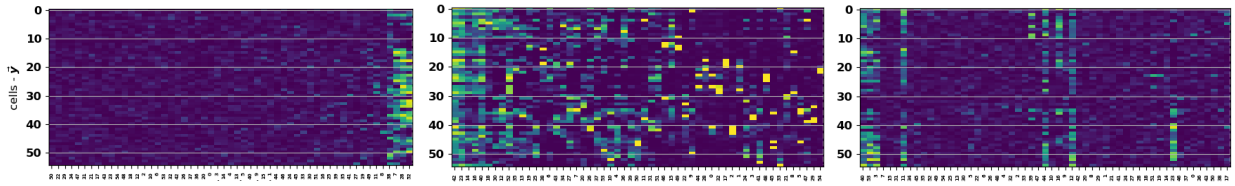
(b) receptive fields

Figure 2.7: **PSTHs and RFs of Off-Brisk Transient RGC:** Color indicates #Trials in which an offBT neuron (y-axis) spiked during a 1ms interval of stimulus presentation (x-axis). *Top*, responses to white noise. *Bottom*, responses to natural movie. Geometric RF relationships in visual space maintained in cell ordering, *panel b*.

Raw spike trains are binned at 1, 3 and 5ms to form spike-words, wider bin widths allowing

for detection of near-synchronous activity, to construct a data corpus of between $500k$ and $1M$ spike-words. Sample distributions of spike-word statistics are shown in Fig. 2.2 for offBT cells show both white noise and natural movies, in green and blue respectively. After binning, spike-words are randomly sampled, ignoring activation time and stimulus, to train a model using EM algorithm, as described in section 2.3. We choose the latent dimension ($\vec{z}$) to be the same size as the observed dimension ($\vec{y}$) and explore both binomial and "Egalitarian Homeostatic" priors.

First, looking only at models trained on each data set, we inquire: "Is CA structure different in white noise responses vs. natural movie responses vs. GLM simulated responses?" We focus our discussion here on models fit to 55 offBT RGC responses to both stimuli and to simulated responses to natural movie from a GLM model fit to these cells. Similar results were found for models trained on responses from [offBT,offBS] and [offBT,onBT] cell-type combinations as well. Cell assembly structure discovered in natural movie responses was qualitatively different from both models trained on white noise responses and GLM simulated natural movie responses, while the two others resembled one another. Fig. 2.8 shows three typical P_{ia} matrices trained on each data set. They clearly indicate that the model trained on natural movie responses in *(b)* has learned more CAs with varied structure than model trained on white noise responses from the same cells in *(a)* or to model trained on GLM simulated responses to the natural movie in *(c)*.

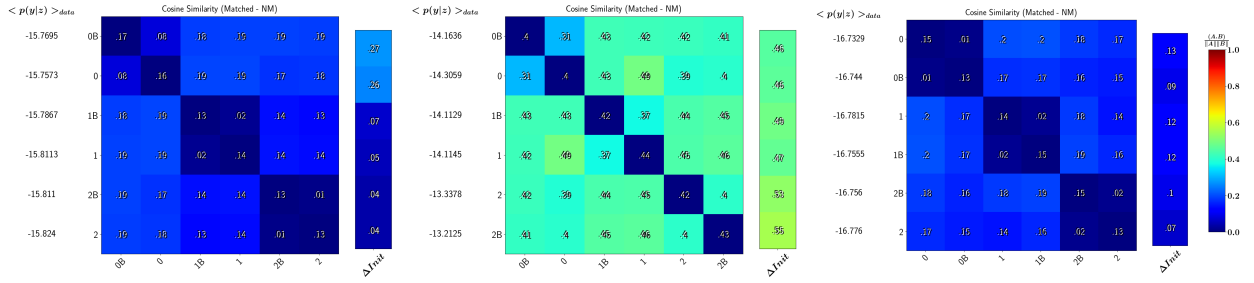


(a) white noise responses (b) natural movie responses (c) GLM simulated responses

Figure 2.8: **Models trained on responses to different stimuli:** P_{ia} matrices for models trained on retinal responses to white noise stimulus*(a)*, natural movie stimulus *(b)* and GLM simulated responses to same natural movie stimulus *(c)*. Cell ID on y-axis, CA ID on x-axis. Yellow indicates cell membership in CA, $p(y_i = 1|z_a = 1) \sim 1$.

Following the cross-validation framework laid out in section 2.4, we split spike-words in half and train one model on each split, using the other half for cross-validation. We repeat this process 3 times, training a total of six models for each data set. In synthetic data, we observed that structure found by multiple models matched embedded ground truth with high probability, see Figs. 2.4 & 2.5. We propose that CA structure discovered reliably across multiple models reflects valid structure in real spike-trains as well. Comparing Fig. 2.9 *(b)* to panels *(a)* & *(c)* indicates that models trained on natural movie responses have learned valid higher order structure, while the other two have not. First, models trained on natural movie responses are changed more from their initialization, vector of ΔInit in *(b)*. Latent z_a's

are activated more often and change to represent commonly occuring structure in natural movie responses. Second, models independently trained on natural movie responses share more structure, evinced by higher values in matrix off diagonals in *(b)*. Finally, the average conditional probability of data in the cross-validation set, $p(\vec{y}|\vec{z})$, is highest for models trained on natural movie responses. These provide three clear indications that CA structure is discovered in natural movie responses, that multiple models robustly learn that structure and that each model generalizes to data that was not used to train it.



(a) white noise responses (b) natural movie responses (c) GLM simulated responses

Figure 2.9: **Repeatable structure in responses to different stimuli:** Similarity of CA membership structure across model pairs for six models trained on each of three spike-word data sets. Note relationship to Fig. 2.4. Models trained on white noise retinal in responses *(a)*, natural movie responses in *(b)* and GLM simulated responses to same natural movie stimulus in *(c)*. *Within each panel*, matrix off-diagonal elements shows average Δcs (relative to null without CA matching) between all matched CA pairs within a model pair. Here diagonal value indicates average between a model and all other models, i.e. the average across a row. Numbers on left show average conditional probability computed on hold out set of half of all spike-words, i.e., cross-validation. Vector on right shows average change from initialization for all CAs in model, defined as $1 - cs$.

Given these findings, we focus our analysis to models trained on natural movie responses and models trained on GLM simulated responses to natural movie for significance assessment.

With real data, each CA can be placed in space because cell RFs have been fit using reverse correlation on white noise responses, see Fig. 2.7b. Further, after a model is trained, CA activations can be placed in time by inferring latent activity with fixed model for each observed spike-word in the entire data corpus. Because CAs can be placed in space and time with real data, we can further probe the spatial membership structure and temporal activity of CAs in retinal responses to natural movie stimulus, asking questions such as:

1. How big are CAs? Are their membership boundaries crisp?
2. Are individual CAs found robustly across models, with similar spatial membership structure and similar temporal response profiles?

3. Are spike-words observed during CA activity significantly different from what the textbook retinal model would predict for the same stimulus?

4. Qualitatively, what shapes do CAs take in the image plane?

5. In models trained on multiple cell-types, do CAs cross cell-type boundaries?

6. Do CAs seem to activate in response to certain stimulus features?

The remainder of this section is organized as follows. We first introduce four metrics that will be leveraged to address some of the questions listed above. We present the model's findings on retinal responses, relying on statistics and metrics developed. We conclude by showing some intriguing example CAs with structure more complex than believed to exist in retina. We remind the reader that this is the first time this method has been applied to real neural data and the results, while inconclusive, are encouraging. Finally in the discussion section, we place the method and results in a broader context, highlighting remaining model development, future experiments and data analysis.

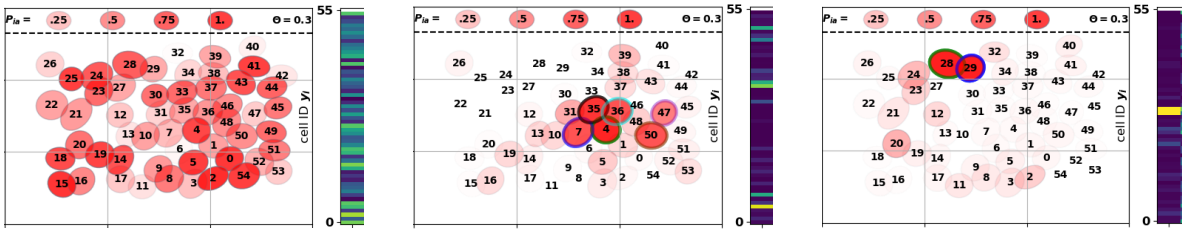
Metrics for cell assembly model assessment

Here, we motivate and construct 4 metrics establishing a framework to assess the types, quality and significance of cell assemblies discovered by various models. Note that each metric is computed for single cell assemblies, not full models as done above. Along with the formulae, we provide a brief description of the structure each captures along with illustrative examples from CAs discovered in retinal data.

(M1). Membership Crispness (C_M) provides a measure of how well defined CA "boundaries" are. Based on a d' metric, from Signal Detection Theory, C_M quantifies how separable a signal distribution (cells determined to be "in" a CA) is from a noise distribution (cells "out" of CA), both assumed normal. Three illustrative examples of CAs with varying C_M values are shown in Fig. 2.10. C_M is computed as

$$C_M = \frac{\mu_{in} - \mu_{out}}{\sqrt{\sigma_{in}^2 + \sigma_{out}^2}} \tag{2.20}$$

with μ_{in} and σ_{in} the mean and standard deviation of P_{ia} values of cells determined to be "in" the CA, the remained of cells being labeled as "out". The method we use for defining cells that are "in" a CA is based on ordering membership probabilities, P_{ia} column values, computing derivatives in sorted values, ΔP and choosing elements that pass $\mu + \sigma$ of both P_{ia} and ΔP. Thus, based on P_{ia} values, we can determine which cells are members of a CA and quantify how sharp are its' boundaries.



(a) $C_M = 0.2$ (b) $C_M = 0.6$ (c) $C_M = 0.9$

Figure 2.10: **Membership Crispness C_M examples:** CAs ranging from diffuse *(a)* to crisp *(c)*. In each panel, numbered ovals represent cell RFs with redness indicating strength of CA membership. Legend above. Corresponding column in P_{ia} shown on right.

(M2). Cross-validation Robustness (R_X) quantifies how reliable or repeatable cell assembly membership structure and temporal activations are across multiple models trained on the same data. For a single CA, average membership cosine similarity (cs_M) with matching CAs in other models can be computed as discussed in section 2.4. Additionally, for real data where CA activations occur in time, the analogous temporal quantity (cs_τ) can be computed by binning CA rasters and computing cosine similarity between the PSTHs of matched CAs. Observing that membership and temporal $\langle cs \rangle$ across models are highly correlated, see Fig. 2.11c, we combine them into a single Robustness measure,

$$R_X = \sqrt{\langle cs_\tau \rangle_X \cdot \langle cs_M \rangle_X} \tag{2.21}$$

where $\langle cs \rangle_X$ indicates average similarity across matching CAs in other models trained on the same data. R_X is bounded between 0 and 1, obtaining large values only when temporal and membership similarity across matching CAs in multiple models are both high.

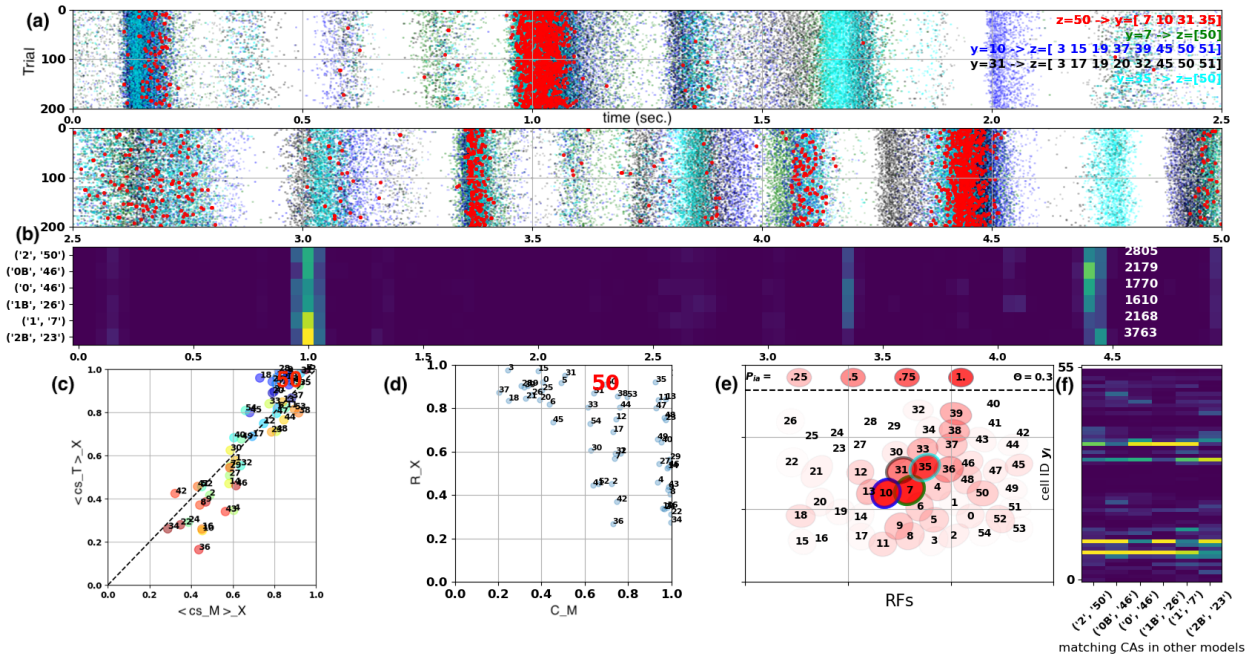


Figure 2.11: **Cross-validation Robustness R_X example:** *(a).* Raster plot time vs. trial. CA activity in red and member cell spikes in other colors. *(b).* PSTH from shown CA on top line and PSTHs from matched CAs in other models below, (model, CA id) indicated on the left. PSTHs normalized with total number of activations in white on right. *(f).* Columns of P_{ia} matrices for shown CA, on left, and its counterparts in other models. *(c).* Membership vs. temporal cosine similarity for each CA in model with matching CAs in 5 other models, averaged across 5 matches. For clarity, $\langle cs_M \rangle_X$ on x-axis computed from panel f and $\langle cs_\tau \rangle_X$ on y-axis computed from panel b. *(d).* Cross-validation Robustness R_X vs. Membership Crispness C_M metrics for all CAs in one model. CA shown here highlighted with red "50" in panels c & d. *(e).* Cell RFs, redness indicates CA membership strength, P_{ia} value, and outline colors match raster colors in panel a.

(M3). Cell-type Heterogeneity (H) is a measure of how mixed the membership of a cell assembly is across a pair of cell-types. We define heterogeneity as

$$H = \frac{min(\#_{ct1}, \#_{ct2})}{avg(\#_{ct1}, \#_{ct2})} \tag{2.22}$$

where $\#_{cti}$ is the number of cells of type i participating in the CA. Method for defining CA members discussed in section on membership crispness metric. H is bounded between 0 and 1, requiring mixed CA participation to be nonzero, and taking a maximum value of 1 when each cell type contributes half of the cells to the CA. A sample of a cell assembly involving offBT and offBS cells with high heterogeneity value is shown in Fig. 2.12.

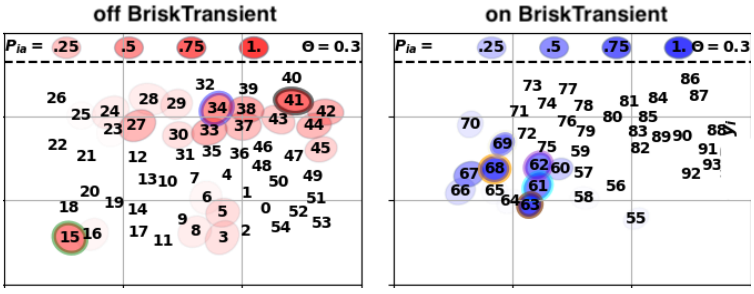


Figure 2.12: **Cell-type Heterogeneity H example:** Two cell assemblies with high heterogeneity. *(a)* Single CA comprised of offBT *(red, left)* and onBT *(blue, right)* cell types. Ovals indicate cell RFs and color intensity indicates membership strength, i.e., P_{ia} value. Legend above.

(M4). Difference from Null (ΔPy) is a measure of the significance of a CA. It captures the degree to which observed cell firing associated with a CA differs from predictions of the textbook independent, rate-coding retinal model. We employ an independent GLM null model where each cell has access to the stimulus within its RF and its own spike-history. For each observed spike-word $\vec{y}_s$ during each z_a activation, we compute $p(\vec{y}_s)_{null}$ at every point in time based on GLM simulated spike-rates and $\vec{y}_s$. Averaging across all spike-words observed while z_a is active yields $\langle p(\vec{y}_s)_{null} \rangle_{\forall \vec{y}_s | z_a=1}$, the green curve in Fig 2.13. PSTH for z_a is shown in red. Differences highlight spike-train structure captured by z_a in the latent variable model which is not explained by rate-coded stimulus correlations. We quantify the difference at a particular time resolution by binning PSTH and $p(\vec{y}_s)_{null}$ and computing the cosine similarity between their traces. Specifically,

$$\Delta Py = 1 - cs_\tau(PSTH(z_a), \langle p(\vec{y}_s)_{null} \rangle_{\forall \vec{y}_s | z_a=1}) \tag{2.23}$$

where cs_τ is the temporal cosine similarity (see Eq. 2.19) between the PSTH of z_a and the probability of spike-words observed during z_a activation under the GLM null model. Binning at different time resolutions reveals temporal dependencies between synchronous activity and spike-rates, not nearby large peaks in Fig 2.13 and decreasing ΔPy for coarser binning. There are some caveats for this metric, the discussion of which we save for supplemental material.

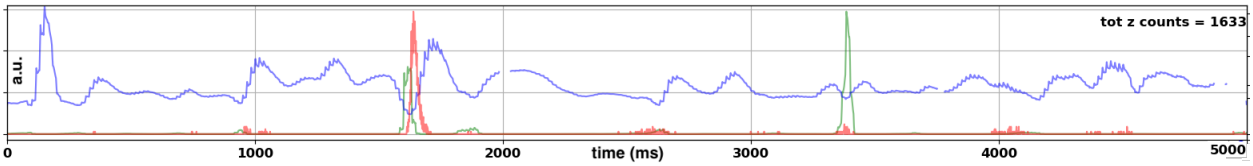


Figure 2.13: **Difference from Null $\Delta P(y)_{null}$ example:** PSTH of z_a in red. In green, $p(\vec{y})$ under GLM null model for all $\vec{y}$ observed when $z_a = 1$. Binning curves at [1,10,50,100] ms yields ΔPy = [.79,.78,.61,.51]. In blue, KL-divergence between N-dimensional multivariate Bernoulli distributions of $p(y_i)_{null}$ and $p(y_i|z_a = 1, z_{a'} = 0)$, discussed further in supplemental section B.

Results: Exploratory Data Analysis

Although we can show clearly that structure is discovered in retinal spike-train responses to natural movie stimulus, it is unclear what that structure represents and what induces it. It would be premature at this stage to draw strong conclusions or make general statements from the model results because CAs found are quite varied and the stimulus presented was limited. Rather here, armed with the metrics developed above, we attempt to catalog cell assemblies discovered in retinal responses to natural movie and showcase some intriguing examples of CAs demonstating spatial membership structure and temporal activity beyond what is believed to exist in retina.

Statistics of these metrics computed for all CAs in one typical model trained on [offBT, onBT] responses to natural movie are shown in Fig. 2.14. Statistics shown look very similar for other models trained on [offBT, onBT] responses, and qualitatively similar for models trained even on offBT responses and [offBT,offBS] responses to natural movie. Here, CA sizes range from 2-20 cells (out of 94 cells total). In models trained on [offBT] responses alone, with 55 cells, the upper end was around a dozen cells, distributions of CA sizes resembling *(a)*, with proportionally more small and crisp CAs. Sorting and coloring CAs by size reveals that C_M is correlated with CA size, evinced by the vertical color gradient in *(c)* and qualitative difference from left to right in the sorted P_{ia} matrix in *(b)*. Many CAs are robustly learned across models *(c)* and significantly different from GLM predictions on fine and coarse time-scales *(d)*. Finally, although the majority of CAs do not cross cell-type boundaries, a number of heterogeneous CAs are found as well *(e)*.

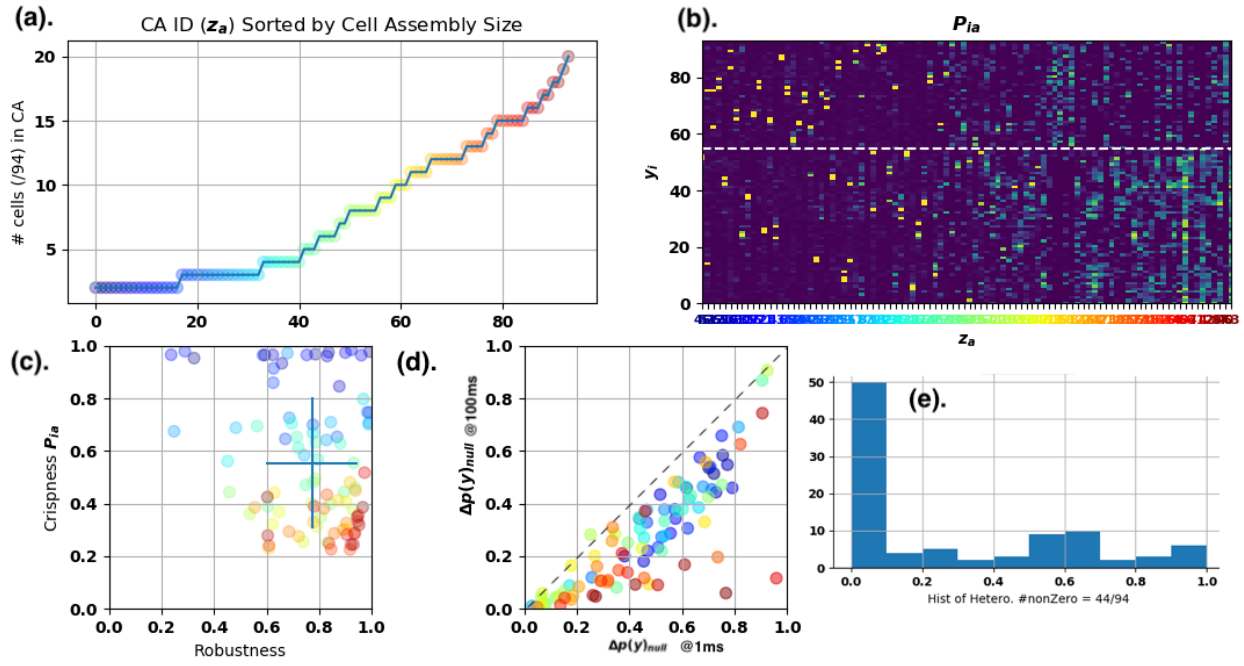
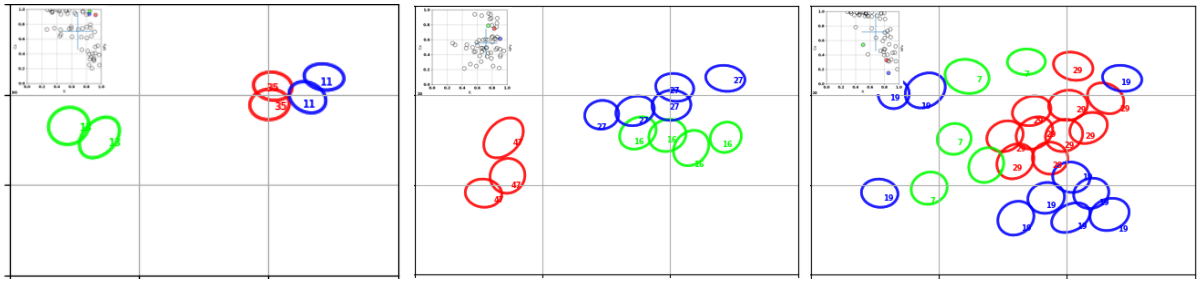


Figure 2.14: **Statistics of CA metrics in typical model** trained on natural movie responses from 94 [offBT,onBT] RGCs. *(a).* Sorted CA sizes, colors consistent throughout. *(b).* P_{ia} matrix, white line indicates break between population offBT below and onBT above. *(c).* Membership Crispness C_M vs. Cross-validation Robustness R_X, each point a CA. Cross shows μ & σ across all CAs. Note that vertical color gradient indicates correlation between C_M and CA size. *(d).* Difference from Null ΔPy with 1ms binning vs. with 100ms binning. *(e).* Heterogeneity H metric histogram.

Here, we share the exploratory data analysis process, showing examples, discussing of trends observed and noting unresolved issues. We resist the urge to present the results as crystalline because the process and bird's-eye view will be most useful if the work is to be extended in the future. First we sorted CAs based on one or a few of these metrics. Then we characterized their performance in the other metrics searching for correlations. We visualized the spatial and membership structure of participating cells as well as temporal responses of CAs themselves, searching for covariates in stimulus and GLM-predicted firing rates. Though posed separately, we address the questions at the beginning of this section in parallel because much of the insight to be gained about structure discovered emerges in their interactions.

We begin by addressing issues of CA crispness, cross-validation robustness and statistical significance relative GLM null model predictions within a single cell-type population. In responses from the [offBT] population, we find a variety of CAs from crisp nearest neighbor pairs *(a)* to large diffuse groups *(c)*. Fig.2.15 shows 9 CAs discovered in 3 different models. Although we find robust CAs that are significantly different from null predictions at all crispness values, reflecting perhaps limitations in the independent GLM null model and in the ΔPy metric (see supplement section 7), we focus on CAs with crispness values between

0.5 and 0.9, expecting interesting and interpretable CAs to contain a few well defined cell members.



(a) Crisp ($C_M > 0.9$) (b) ($0.8 > C_M > 0.6$) (c) Diffuse ($C_M < 0.5$)

Figure 2.15: **CAs Membership Crispness examples:** From 3 separate models trained on [offBT] natural movie responses. In each panel, colored ovals RFs of cell members from 3 separate CAs with similar C_M values. Small inset scattters R_X on x-axis vs. C_M on y for all CAs in each model with shown CAs highlighted in matching color.

Looking further into z_{16} and z_{27} the green and blue colored CAs in Fig. 2.15b respectively, Fig. 2.16 shows that they represent elongated, horizontally oriented correlated fine-time firing among groups of 4 & 5 neighboring cells. Both CAs have crisp, well-defined membership and are robust under cross-validation paradigm *(c)*. ΔPy computed at 1ms vs 100ms time resolutions scattered in *(d)* indicates that both CAs are involved in activity significantly different from GLM predictions. Significant differences persist at coarser time-scales. Temporal response traces in *(e)* & *(f)* reveal that even though the CAs are close in proximity and share horizontal orientation, CA activations *in red* and GLM predictions *in green* are different.

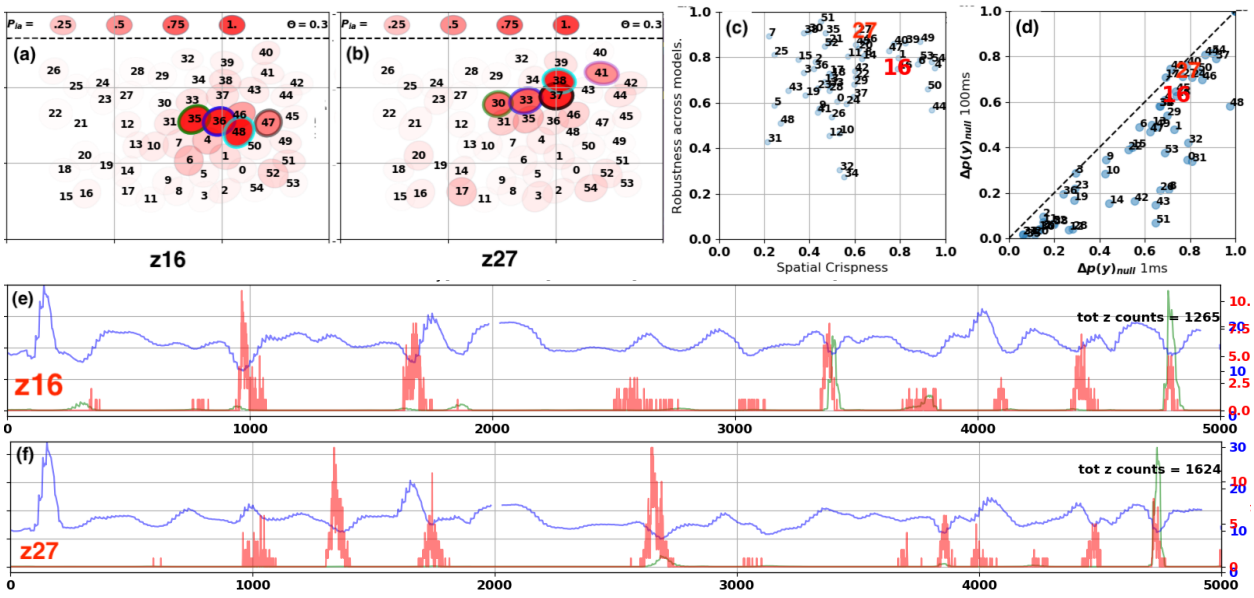


Figure 2.16: **Two elongated, yet crisp sample CAs:** RFs from CAs z_{16} and z_{27} in panels *(a)* & *(b)* are same as green and blue ellipses in Fig. 2.15b. In *(c)*, crispness and robustness scattered for all CAs in model with two shown highlighted in red. In *(d)*, ΔPy at 1ms vs 100ms scattered in *(a)*. Temporal response traces of CA PSTH *in red* and GLM $\langle p(\vec{y}) \rangle$ predictions *in green* shown in *(e)* & *(f)*.

Fig. 2.17 showcases six out of more than a dozen visually interesting CAs found in a single model with high cross-validation robustness and associated with activity significantly different from GLM-predictions. These were discovered by sorting CAs by R_X and ΔPy_{100ms} and examining high ranking results. Metric values are scattered in bolded box upper center with shown CAs highlighted by color. While all CAs featured take only modest crispness values ($0.4 < C_x < 0.6$), visual inspection reveals reasonably discernible boundaries. Moreover, several CAs resemble elongated shapes, edges or curves (specifically z_{34}, z_{32}, z_{35}, z_{11}), possibly encoding for extended edges in the natural movie stimulus. Recall that activity is more synchronous that predicted by a rate-code model and therefore can not be explained simply by stimulus correlations alone.

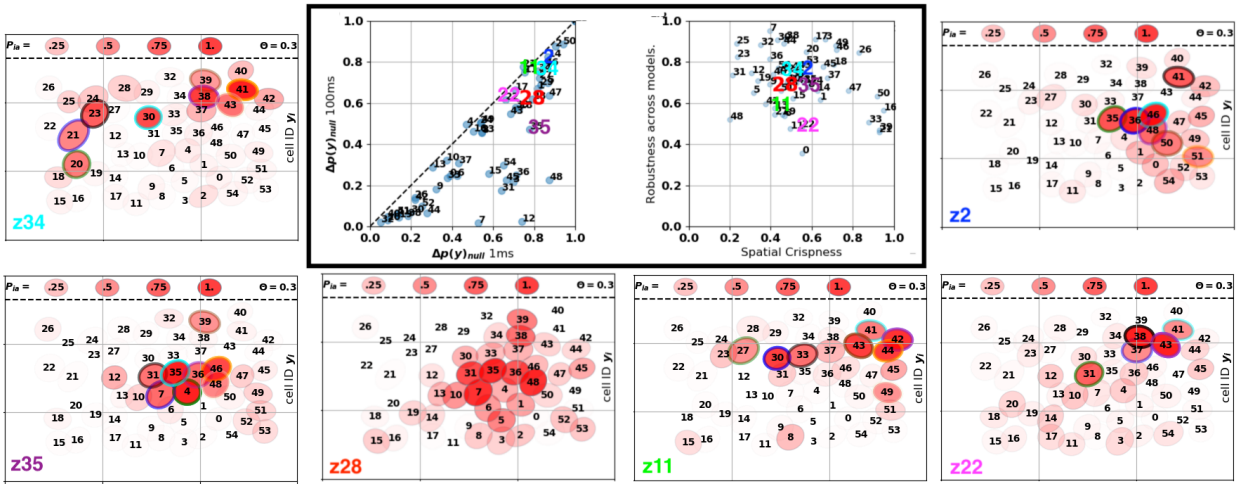
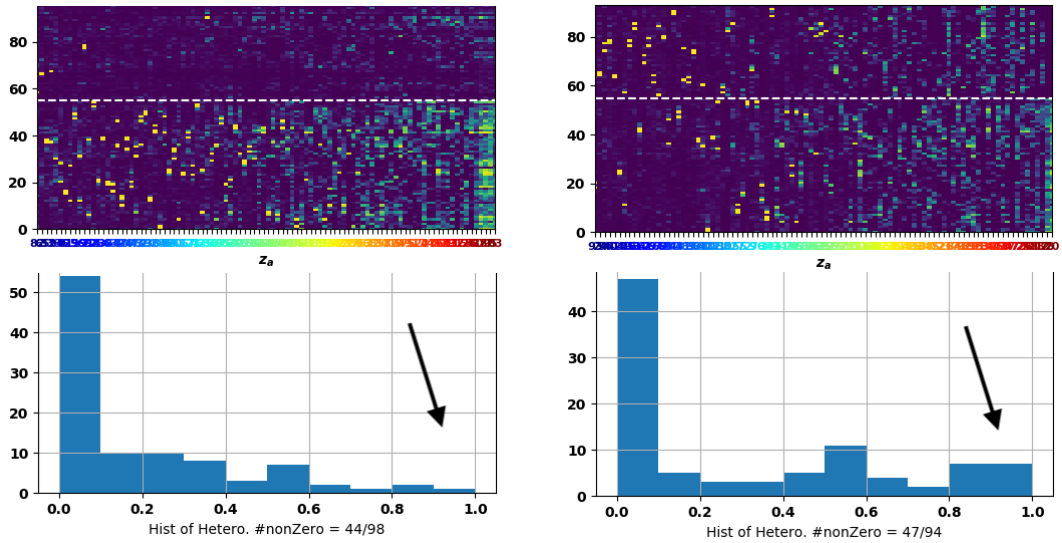


Figure 2.17: **Six robust CAs with high ΔPy in one model:** Each surrounding panel shows RFs of [offBT] cells with redness reflecting strength of membership in CA. CA id in colored number in bottom left of each panel. Scatter plots in bolded box show metric values for each CA in model with shown CAs highlighted in colored numbers matching id.

To our knowledge the question has not been asked yet whether fine-time correlated spiking activity exists across mixed populations of retinal cell-types responding to natural movie stimulus. We now shift focus to models trained on multiple cell types to investigate whether learned cell assemblies cross cell-type boundaries. Fig. 2.18 contrasts two typical models trained on mixed cell-types. For each data set, the other 5 models trained resemble those shown. In both data sets, the majority of CAs learned segregate within one or the other cell type. In responses of [offBT, onBT] cells, we find more heterogeneous indicated by the slight shift in the distribution to higher heterogeneity values in panel *(b)* relative to *(a)*. Though subtle, this trend towards more heterogeneous CAs crossing [offBT,onBT] cell-types is consistent across others models learned.



(a) [offBT,offBS] (b) [offBT,onBT]

Figure 2.18: **Heterogeneity of CAs across cell-types:** *(a).* Model trained on natural movie responses from 55 offBT and 43 offBS RGCs. *(b).* Model trained on responses from 55 offBT and 39 onBT RGCs. In each panel, P_{ia} matrices shown on top with columns indicating CAs. Dashed white line shows boundary between offBT cells below and other type above. Bottom shows histogram of H metric values for all CAs in model. Black arrow indicates consistent difference across multiple trained models.

We find a number of interesting heterogeneous CA within the [offBT,onBT] populations. They tend to form extended regions on and off cells that border one another, perhaps performing some sort of push-pull computation or edge enhancement. Some of these CAs are largely explained by spike-rates and some are not. It is unclear the difference between them other than the ΔPy metric. Fig. 2.19 shows four heterogeneous CAs learned in a single model. All are learned robustly across models and moderately crisp *right top*. All four have similar structure with clusters of on cells bordering clusters of off cells. Two however are significantly different from null model predictions and two are not. At present, it is unclear why.

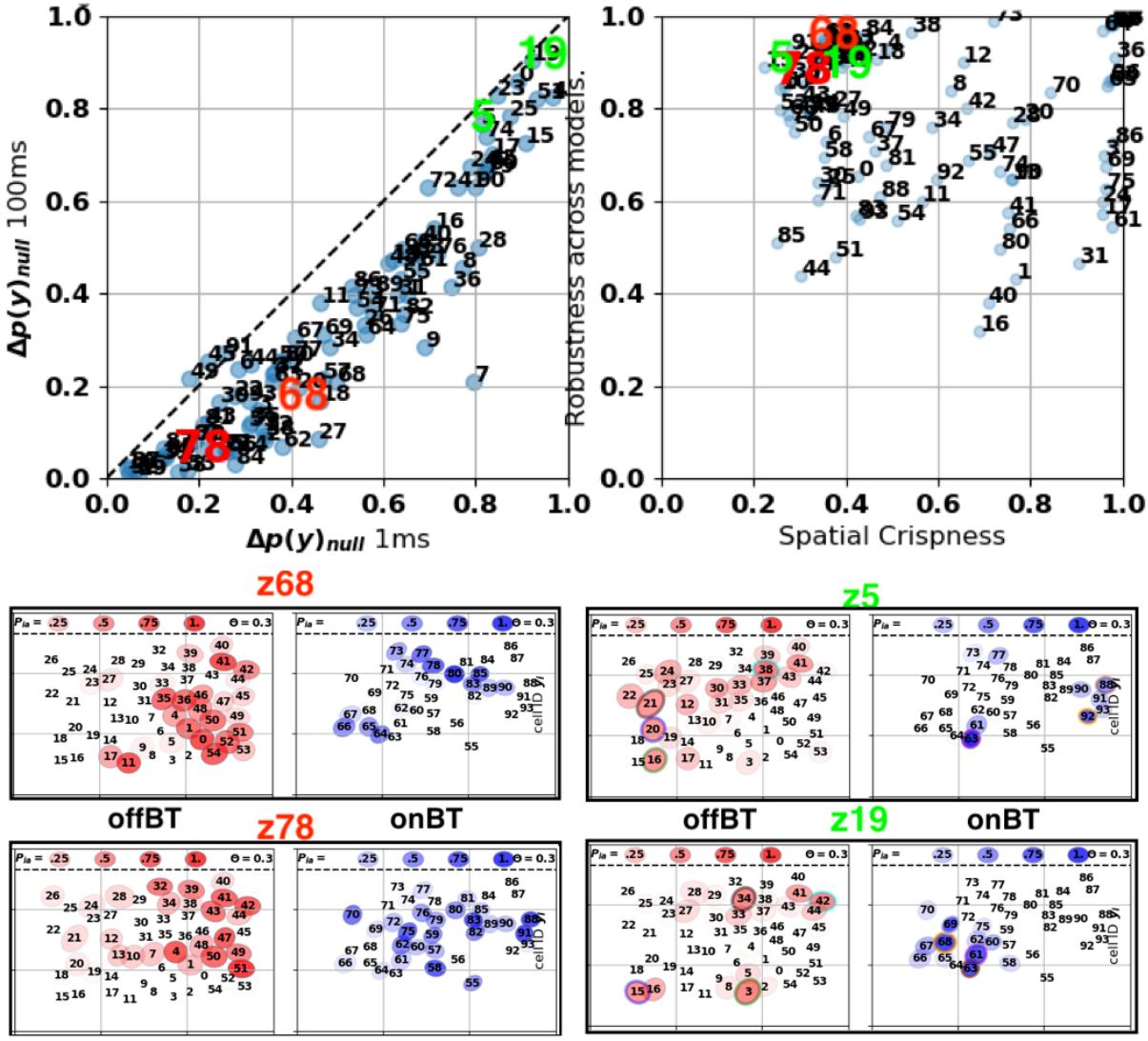


Figure 2.19: **Heterogeneous [offBT,onBT] CAs:** Four strongly heterogeneous CAs learned within a single model are shown on bottom. Each boxed plot represents one CA, red ovals showing RFs and participation of offBT cells and blue ovals, onBT cells. Scatter plots above show R_x vs. C_M on right and ΔPy at 1ms and 100ms time resolutions on left. Shown CAs are highlighted in color. Green labeled CAs are significantly different from GLM predictions and red labeled CAs are not.

Below, we present the four heterogeneous CAs displayed in Fig. 2.19 along with their temporal responses and stimulus frames during and $\sim$ 333 ms prior to their activation. Figs. 2.20, 2.21, 2.22, 2.23 show precise and repeatable activation across trials in bottom

PSTH trace and the spatial layout of offBT (red) and onBT (cyan) RFs in top center. Combined RFs of member cells appear to form extended shapes within cell-type which are interleaved across cell-types. Moreover, they appear to be loosely oriented with prominent features, changes and/or movement directions in the stimulus shortly before activation. Stimulus frames during and ~ 333 ms prior to activation are shown on top right and left of each figure respectively. The effect appears more pronounced in the two CAs with low ΔPy values, Figs. 2.20, 2.21, it is also there in the two CAs with high ΔPy values. We do not understand this, and leave it to future investigation to sort out.

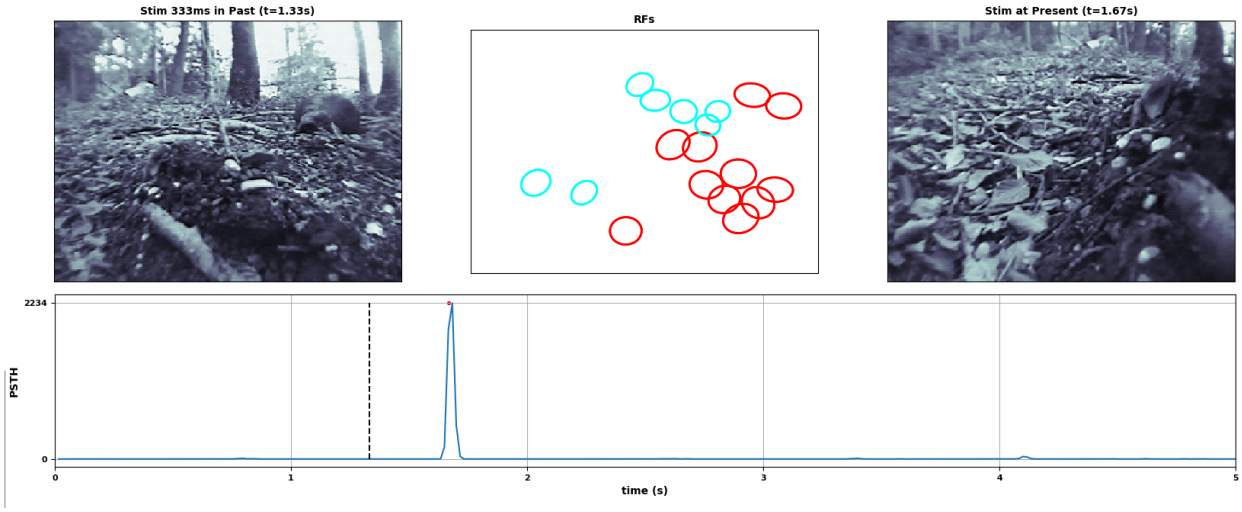


Figure 2.20: **CA z68 and stimulus:** *Not* significantly different from GLM null model. Within each panel, *Center top* shows cell RFs of offBT (red) and onBT (cyan) member cells. Box matches image dimensions approximately. *Bottom* shows PSTH of CA activation. *Right top* shows stimulus at time of CA activation, blue peak in PSTH. *Left top* shows stimulus ~ 333ms prior to CA activation.

Additionally, the time-lag found here is interesting considering that the temporal responses of individual cells are ~ 150ms. See Fig. 2.24. This time difference circumstantially supports the hypothesis presented in chapter 1 that spike alignment through phase relaxation is encoding extended stimulus feature grouping, i.e., a coarse image segmentation. The relaxation has only $\sim 1/3$ second to occur since that is the average fixation time between eye saccades. The computation introduces large spatial and temporal correlations into spike trains, which are then strongly suppressed during saccades to allow for a repeat computation at the next fixation point. To explore this computation, a useful stimulus for future work would be one which includes simulated eye saccades in natural movie/image stimulus. Though the evidence presented here is qualitative and anecdotal, it is intriguing and encourages further investigation. It points to the possibility that synchrony introduced into spike-trains of these

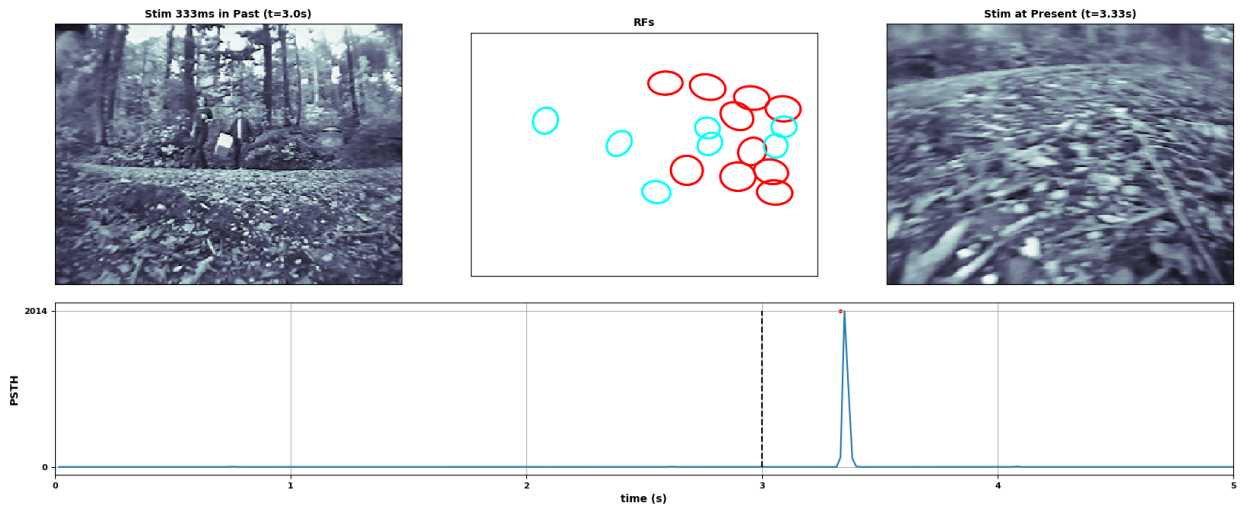


Figure 2.21: **CA z67 and stimulus:** *Not* significantly different from GLM. Explanation in Fig.2.20.

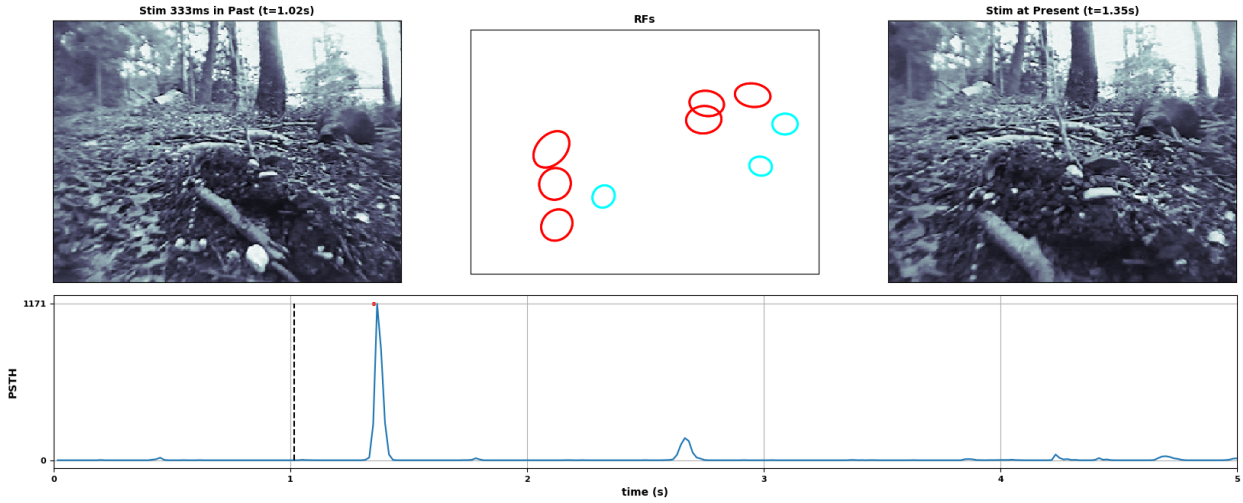


Figure 2.22: **CA z5 and stimulus:** Significant difference from GLM. Explanation in Fig.2.20.

cells might facilitate some sort of push-pull computation or edge enhancement, encoding a non-local gist representation of the stimulus.

Some final notes. Heterogeneous CAs were also found in [offBT,offBS] responses (*not shown*) but the interpretation of observed CAs spanning two off cell-types is murkier because it is unclear what they are expected to look like. Their analysis remains for future work. Also, analysis to connect [offBT] CA activity to stimulus has not been carried out rigorously either and it will likely yield some similarly tantalizing, yet inconclusive results. Further EDA

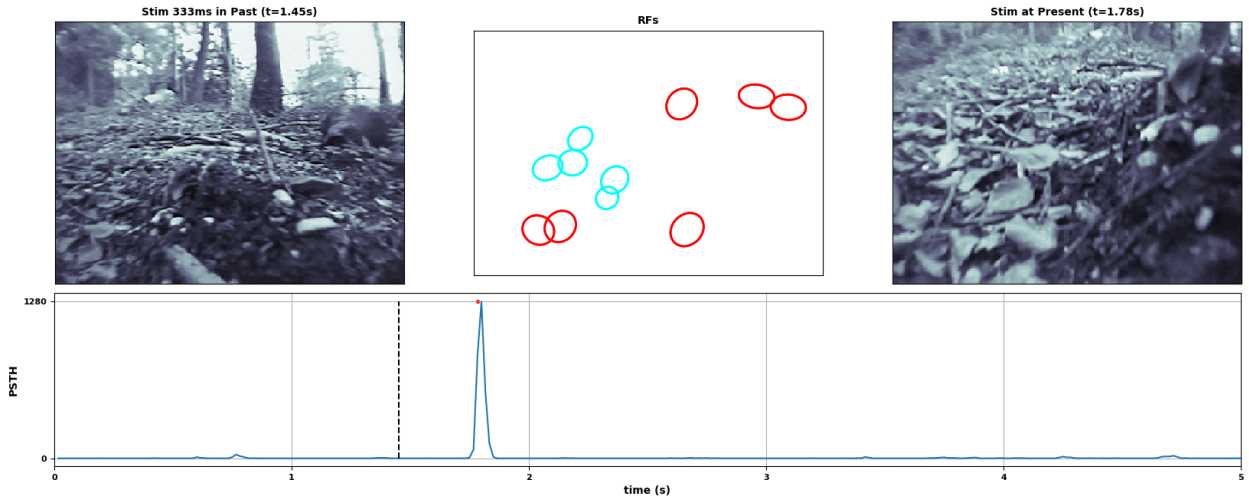


Figure 2.23: **CA z19 and stimulus:** Significant difference from GLM. Explanation in Fig.2.20.

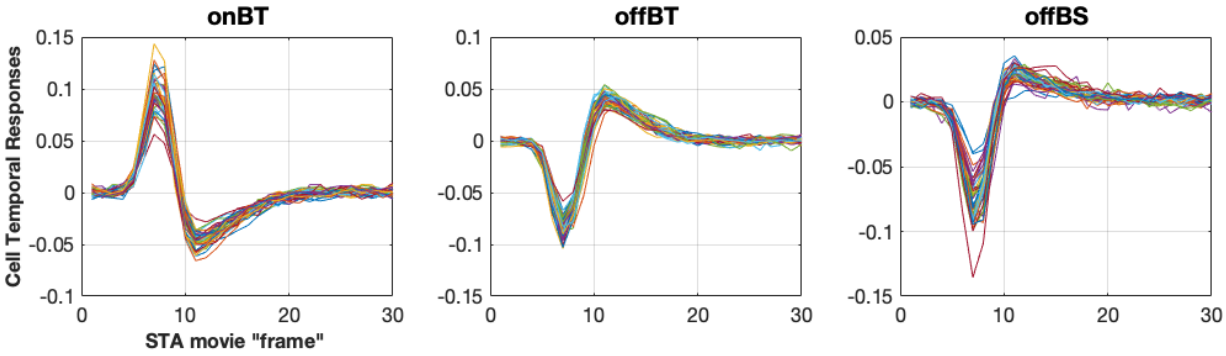


Figure 2.24: **Measured temporal response profiles by cell type:** Units of x-axis are stimulus frames or $\sim$ 16ms.

investigations should be performed on this data and more rigorous analysis on future data collections.

2.6 Discussion

In this work, we introduced a novel probabilistic latent variable model to detect cell assemblies in spiking neural data. We extended the "Noisy-OR model" [**heckerman1990**] to allow individual variability in observation vectors, adding P_i parameters. We also extend the Bernoulli prior on latent activation to a "Homeostatic Egalitarian" prior and find improvement

in some cases. The current model is related to binary soft-clustering, where binary data points in high dimensional space are assigned to cluster centers. It is also related to non-linear sparse coding [**olshausen1996**], with the difference, that in our method both observed and latent variables are binary, not real-valued. Other approaches have previously used latent variable models to analyze spiking data, for example, restricted Boltzmann machines (RBMs) [**koster2014**]. In contrast to an RBM, our model is a directed graphical model, a causal model of the data where $\vec{z}$'s can be interpreted as causes of spike-words. We are not aware of earlier approaches using directed graphical models to analyze neural data.

We have developed several variants of our model, differing in the priors for the latent representations, and vetted them on synthetic spike data, whose statistics was matched to neural responses to different types of stimuli, white noise and natural movies. We observed that our method discovered larger and more crisp cell assemblies, each with lower probability of being active at any one time, in synthetic data matched to the responses of natural movies, as compared to synthetic data matched to the responses to Gaussian noise. We validated how consistently the method found cell assemblies in the synthetic data fit to real spike-trains, finding that structure in the model fit to natural movie responses was more easily learned. Finding that ground truth CA structure embedded into the data was robustly learned across multiple models, we developed some assessment tools which we could then apply to models trained on real data.

We then applied our method to retinal spike-trains recorded from cells responding to white noise and natural movie stimulus. We showed biological results which, while early and incomplete, reveal cell assembly structure in retinal spike trains, inconsistent with the traditional model of retinal encoding. The CA structure we found was strongly dependent on the type of stimulation. While little CA structure was found in responses to white noise, our method discovered large numbers of robust cell assemblies of various sizes and shapes in retinal responses to natural movies. The crispest CAs often included a few cells which were nearest neighbors in the receptive field mosaic. The RFs of other cell assemblies formed elongated edges and curves in the mosaic, the least crisp CAs formed diffuse large clusters.

Using our method to analyze responses from different types of retinal ganglion cells in parallel revealed interesting results. A large fraction of CAs were entirely homogeneous, exclusively including one cell type. However, we also found CAs that were heterogeneous. For example, a few CAs included both offBT and onBT ganglion cells and aligned intriguingly with structure in the movie stimulus shortly before activation. The temporal response properties of CAs seemed to correlate with their size and complexity, with smaller pairwise PSTHs looking closer to single cell PSTHs and larger more complex CAs being activated precisely at one time in the stimulus. Importantly, spike-words observed during many of the CA activations had extremely low probability under an independent GLM model learned on the same data.

The extent of the analysis which could be performed on the retinal data provided was limited however due to several properties of the experimental data – which were collected before our method was available. Because both stimuli essentially consisted of only 150 image frames. Even performing simple reverse correlation of activity onto stimulus was infeasible

in such a data limited regime. In future experiments, diverse natural movie or naturalistic stimulus would be shown for longer duration without trial repeats.

What are the potential questions that can be addressed, using our method in combination with optimally designed experiments? Inspired by the work of [**deny2017**] and [**koepsell2009**] and building on the work discussed in chapter 1, a question that, for example, could be addressed is how complex natural scenes, correlated in both space and time, are encoded by retina. The experiment to address this question would collect responses to a variety of natural or simple naturalistic movies. Scenes should contain objects that move both laterally and in depth, move relative to one another, and occasionally occlude one another. Specifically, the stimulus should include frames when nearby cell RFs process a common segment and frames when the same set of nearby cells is separated by an image segment boundary. Contrasts and textures should be varied through out the data set to provide a rich and challenging assortment of complex scenes to parse. Including large and abrupt shifts in the visual scene which mimic eye movements would provide insight into how the retina uses or ignores large bursts of activity at stimulus onset or just after a fixation. The experimental data and the results of the CA analysis could then be compared with predictions of our image segmentation model of retina described in chapter 1.

Our cell assembly detection method is different from others, for example Unitary Events Analysis [**grun2010**], because we learn a real-valued probabilistic representation of cell assemblies and allow an observed spike-word to be represented by a combination of latent variables. This allowing us to detect noisy repeats of commonly occuring patterns. UEA detects only exact repeats of binary patterns and requires many trial repeats to elicit repeat responses. We do not require stimulus repeats and in fact suggest for future data collection to do away with stimulus repeats in order to more fully sample the space of natural images and drive the retinal cell population in a wider variety of ways. UEA assesses the significance of found patterns by comparing number of observations of an exact pattern relative to the number predicted by a null model. This key difference prohibits direct comparison to UEA.

Finally, we wish to reiterate that while much of the discussion in this work has focused on retina, this method is applicable to any neural data where near synchronization of spike activity is suspected to be meaningful. Of course, our method is agnostic of mechanistic cause, whether the synchrony is caused by common input, recurrent excitation or other causes. While much remains to be done in this arena, the work presented here lays a firm foundation to investigate fine-time ensemble coding in spiking neural data.

Conclusion

Motivated by the gulf between observed anatomical structure and response complexity in the retina on one hand and the parsimony in models and our understanding of that system, we endeavor in this body of work to explore what else this under-appreciated system might be doing. Modelling RGCs as a bank of independent filters that encode local image features in the spike rates flies in the face of "occam's razor" and begs the question what is the function of the rest, the majority, of the intricate retinal circuitry. That is, why connect neurons together if they are coding independently? Or perhaps the better question, why hypothesize a model of independent coding when retinal neurons interact through a complex anatomical network and demonstrate rich activity unexplained by that model during ethologically relevant stimulation?

In this work, we approach the topic of ensemble coding in retina in two ways. First we ask, what visual information exists in the retinal ensemble above and beyond the sum of independent rate-coded representations of individual ganglion cells? We explore image segmentation using phase coding in the retina, hypothesizing that fine-time correlations in spike trains are induced by phase interactions influenced by the visual stimulus and that fine-time correlations, informative about segments in an image, are multiplexed into spike-trains along with rate-coded local stimulus features. In the second effort we ask, how would one find evidence of these fine-time correlations exist in retinal spike-trains if each latent cause activates observations stochastically, observations from multiple latent causes can be mixed together and and if these observations are overlaid on top of other noisy signals? In pursuit, we explore a statistical model that aims to find cell assemblies, or groups of cells that fire are often coactive, possessing fine-time correlations.

While inspired by and applied to retina, each project stands apart from this system as well. Grouping related objects through phase interaction has been posited throughout the brain and is equally relevant in reasoning and cognition, an example being "the binding problem" as it is in segmenting images. Synchrony and fine time relationships between neural activity has been studied extensively throughout the brain. Moreover, while they are related to one another, the two halves of this work stand alone and contribute to the ongoing scientific conversation about retinal and, more generally, neural coding.

Both efforts lead to interesting results that call into question the textbook model of retina. In more ways than one, the retina is a window into the brain. It is a relatively simple, model system isolated from the rest of brain that more readily allows investigation than cortex

or deeper brains structures. We have shown proof-of-concept in both efforts that warrant further study both in retina and elsewhere. We hope that the reader agrees and will perhaps carry on the work.

Appendix A

Image Segmentation Supplement

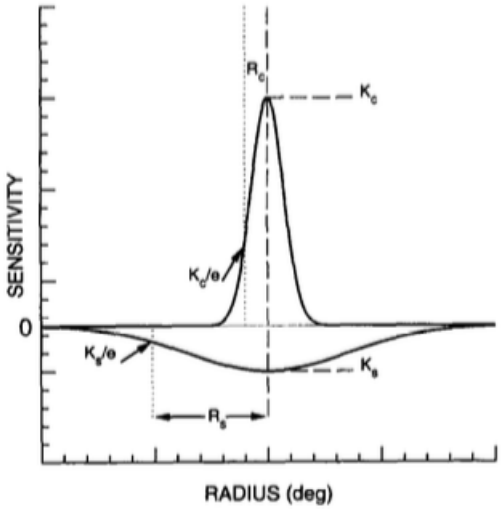
A.1 Optimal Gaussian RF size

There are multiple independent sensor models to which we could compare the network models. We constrain our sensors to have access to relatively simple image features similar to those which the retina would encode. For comparison, we compute image segmentation using two independent sensor null models. The first uses raw image pixels and the second passes image pixels through Gaussian filters that mimic retinal ganglion cell (RGC) receptive fields (RFs). Center-surround RGC RFs are modelled by a difference-of-Gaussian filter with an excitatory center and inhibitory surround. Gaussian filters fit to the centers and surrounds of primate midget and parasol ganglion cells were observed to be strongly center dominant [**croner1995**]. Thus the receptive field of an RGC can reasonably be modelled by a single excitatory Gaussian center to first approximation and the optimal Gaussian RF size reasonably matches average RGC RF sizes measured in primate retina.

In our simulations, the phase initialization of each individual oscillator as well as the connectivity strength between oscillators are both determined by the cell's activation - that is, how closely incoming stimulus matches the filter that is defined as a cell's receptive field. We began with the simplest receptive field model, each cell responding to the greyscale pixel intensity value at its location. Then, motivated by the biological fact that retinal receptive fields are spatially extended, we extended the receptive field model for each oscillating cell to be a localized Gaussian RF kernel. To determine the best Gaussian RF size (σ), we numerically explored a range of spread values and kept the one that provided best average segmentation performance across 500 image patches in the Berkeley Segmentation Dataset (BSDS) [**martin2001**]. Segmentation performance was determined by F-measure calculated on the match between spatial gradients in phase maps output by network models and ground truth boundaries drawn by human subjects. Interestingly, we determined that a Gaussian RF kernel with $\sigma = 1$ pixel performed best empirically, improving the F-measure value by a modest but statistically significant 0.04 points over raw image pixels.

Motivated further by the excitatory and inhibitory center-surround nature of biological

receptive fields in retina, we employ difference of gaussian (DoG) filters with parameters based on retinal physiology [**croner1995**]. The Croner paper provides parameters fit to DoG receptive fields for M and P cells in primate retina for eccentricites ranging from $0 - 40°$ in its Table 1. In contrast with LGN center-surround cells [**martinez2014**], retinal receptive fields have very weak surrounds ($\sim {}^{1}/_{100}{}^{th}$) compared with the strength of the center portion. From the many receptive field parameters fit to different cell types at different eccentricities in the primate retina, we distilled out 4 clusters that were different enough to test via simulations. In our simulations using DoG filters with P-avg and M-avg parameter values, we did not see image segmentation improvement over simple Gaussian filter with $\sigma = 1$.



	R_c	R_s	K_s/K_c
P-avg	1	8	0.01
P-40°	3	13	0.06
M-avg	3	14.5	0.01
M-40°	5	12.5	0.025

Figure A.1: **Primate center-surround RFs:** modeled as difference-of-Gaussians. Note: R_c and R_s in image pixels. Values are given for magnocellular projecting (P) and parvocellular projecting (M) cells averaged across all eccentricities (avg) and at the visual periphery ($-40°$) Image of measured retinal RF size from Croner 1995 [**croner1995**]

.

Using a simple back-of-the-envelope visual angle calculation, illustrated in Fig. A.2, and a few reasonable assumptions we approximate the size of retinal receptive field centers and surrounds in terms of image pixels for our models. The calculation goes as follows: Full images in the BSDS are 321 x 481 pixels and we assume that the displayed image size is 8.5" x 11". Given these assumptions, an image pixel is approximately 0.02" on a side. Next, we assume that the projection screen is placed 24" away from the eye. Then, the angle that a single pixel subtends on the retina is approximately 0.05°. Using this relation, we convert numbers provided in the Croner paper for retinal receptive field sizes into pixels and provide them in Fig. **??**.

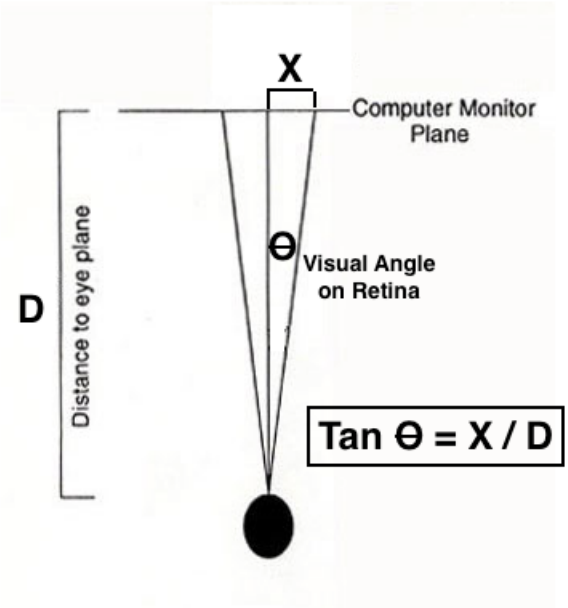


Figure A.2: **Visual angle calculation schematic**

A.2 Motivating modularity

Homogenization and Null Model as Expected Value of Weight

Most generally, an entry in the *modularity* matrix (Q_{ij}) is defined as the difference in weight between a pair of nodes in the actual network, characterized in the *adjacency matrix* (A_{ij}), and the expected value of that weight ($\mathbf{E}[A_{ij}]$) in a "homogenized network", with connections between nodes made to reflect gross statistics of the network's connectivity.

$$Q_{ij} = A_{ij} - \mathbf{E}[A_{ij}] \qquad \text{with} \qquad \mathbf{E}[A_{ij}] = \int A_{ij} p(A_{ij}) dA_{ij} \tag{A.1}$$

The expected value of weights is parameterized in the *null model* (N_{ij}) which is chosen to reflect the modeller's knowledge of network structure and connectivity.

$$Q_{ij} = A_{ij} - N_{ij} \qquad \text{where} \qquad N_{ij} = \mathbf{E}[A_{ij}] \tag{A.2}$$

The null model is constrained only by two considerations. First, because the networks considered have undirected edges, both adjacency and null model matrices are symmetric, with $N_{ij} = N_{ji}$ and $A_{ij} = A_{ji}$. Second, it is axiomatically required that the total weight of edges in the null model are equal to the total weight of edges in the actual network because $Q = 0$ when all the vertices are placed in the same partition. This leads to a normalizing constraint on the null model matrix,

$$\mathbf{\Sigma} = \sum_{ij} A_{ij} = \sum_{ij} N_{ij} \tag{A.3}$$

where $\mathbf{\Sigma}$ is twice the total weight of edges in the network to account for double counting in the double sum over vertices (Note: $\sum_{ij} := \sum i \sum j$). Beyond these basic requirements, we are free to choose from many possible null models, each one containing a different number of parameters, requiring a different number of computations and capturing the expectation of edge weights at different levels of homogeneity by calculating different statistics on the adjacency matrix.

I.I.D. or Homogeneous Random Graph

The simplest null model, based on a Bernoulli or Erdos-Renyi random graph with weights allowed to take real values (i.e. are not constrained to be binary), assigns a single uniform expectation weight to all edges in the network, $\bar{A} = \frac{\mathbf{\Sigma}}{n^2-n}$, which is the average edge weight in the actual network. Note that n is the number of nodes in the network and $\binom{n}{2} = \frac{n^2-n}{2}$ is the number of possible undirected edges that connect them with all-to-all connectivity, barring self-loops.

$$\mathbf{E}[A_{ij}|\frac{\mathbf{\Sigma}}{n^2-n}] = \int A_{ij} \cdot p(A_{ij}|\frac{\mathbf{\Sigma}}{n^2-n}) dA_{ij} = \int A_{ij}\delta(A_{ij} - c\frac{\mathbf{\Sigma}}{n^2-n}) dA_{ij} \tag{A.4}$$

$$N_{ij} = \mathbf{E}[A_{ij}|\frac{\mathbf{\Sigma}}{n^2-n}] = c\frac{\mathbf{\Sigma}}{n^2-n} \tag{A.5}$$

Solving for c by equation A.3, we find

$$c = \frac{n-1}{n}. \tag{A.6}$$

Combining the I.I.D. edge weight assumption with the constraint on total weight strength, we derive that the null model which assumes Bernoulli random graph connectivity patterns expects each weight in the network to take the following value.

$$N_{ij} = \frac{\mathbf{\Sigma}}{n^2} \tag{A.7}$$

This is a very simple representation of the network which requires only a single number - the average edge weight across the entire network ($\bar{A}$), however it is inadequate to capture the structure in all but the simplest networks.

Independent-Vertex or Inhomogeneous Random Graph (N&G Modularity)

Relaxing the "identical" assumption of the I.I.D. graph null model, the "Independent-Vertex" model allows the expected value of each weight in the null model network to be different (inhomogeneous). The expected value of a weight between two nodes is the product of the degree of each of those nodes. This null model capture the expectation that two strongly connected nodes are more likely to be connected to one another and two nodes which are generally weakly connected are unlikely to be connected to one another. Specifically,

$$\mathbf{E}[A_{ij}|\frac{d_i}{n},\frac{d_j}{n}] = \int A_{ij}p(A_{ij}|\frac{d_i}{n},\frac{d_j}{n})dA_{ij} = \int A_{ij}\delta(A_{ij} - c\frac{d_i}{n}\frac{d_j}{n})dA_{ij} \tag{A.8}$$

$$N_{ij} = \mathbf{E}[A_{ij}|\frac{d_i}{n},\frac{d_j}{n}] = c\frac{d_i}{n}\frac{d_j}{n} \qquad \text{where} \qquad d_i = \sum_{i=1}^{n} A_{ij} \tag{A.9}$$

where n is the number of vertices and d_i is the "degree" of node i or strength of connectivity from node i to all other nodes in the network, defined as the row (or equivalently column) sums of the adjacency matrix. Solving for c by equation A.3, we find

$$c = \frac{n^2}{\mathbf{\Sigma}} \tag{A.10}$$

making the full null model

$$N_{ij} = \frac{d_i d_j}{\mathbf{\Sigma}}. \tag{A.11}$$

This requires n numbers or statistics calculated from the network to characterize the null model, namely the degree of each node. This is the model used by Newman [**newman2006**] and works well finding community structure in networks with no inherrent spatial layout or topography.

Line-Distance Dependent, Independent-Vertex Random Graph in 1D (Mod SKH Adj)

In networks with 1D spatial relationships, where each vertex is more likely or more strongly connected to nearby vertices than to distant vertices, the independent-vertex null model which just considers vertex degrees fails to capture this spatial structure and the modularity's ability to find communities in such topographical networks suffers. The simplest spatial arrangement of nodes in a network is along a line in one dimension. Here, we can expand the vertex-independent null model to include a line-distance dependent ($b_{|i-j|}$) term which characterizes the expectation of a weight between nodes separated by a distance ($|i-j|$).

$$\mathbf{E}[A_{ij}|\frac{d_i}{n},\frac{d_j}{n},\frac{b_{|i-j|}}{n-|i-j|}] = \int A_{ij}p(A_{ij}|\frac{d_i}{n},\frac{d_j}{n},\frac{b_{|i-j|}}{n-|i-j|})dA_{ij} = \int A_{ij}\delta(A_{ij}-c\frac{d_i}{n}\frac{d_j}{n}\frac{b_{|i-j|}}{n-|i-j|})dA_{ij} \quad \text{(A.12)}$$

$$N_{ij} = \mathbf{E}[A_{ij}|\frac{d_i}{n},\frac{d_j}{n},\frac{b_{|i-j|}}{n-|i-j|}] = c\frac{d_i}{n}\frac{d_j}{n}\frac{b_{|i-j|}}{n-|i-j|} \quad \text{(A.13)}$$

where

$$d_i = \sum_{i=1}^{n} A_{ij} \qquad \text{and} \qquad b_{|i-j|} = \sum_{k=1}^{n-|i-j|} A_{k,k+|i-j|} \quad \text{(A.14)}$$

Solving for c by equation A.3 yeilds

$$c = \frac{n^2\mathbf{\Sigma}}{\sum_{ij}(d_i d_j \frac{b_{|i-j|}}{n-|i-j|})} \quad \text{(A.15)}$$

and the full null model is

$$N_{ij} = \frac{d_i d_j \frac{b_{|i-j|}}{n-|i-j|}\mathbf{\Sigma}}{\sum_{ij}(d_i d_j \frac{b_{|i-j|}}{n-|i-j|})} \quad \text{(A.16)}$$

where $\frac{d_i}{n}$ is the average weight from node i to other nodes in the network, and $\frac{b_{|i-j|}}{n-|i-j|}$ is the average weight between a pair of nodes separated by the distance $|i-j|$. Since nodes are arranged along a line, their separation distance in 1 dimensional space directly translates into distance from the diagonal in the adjacency matrix. Namely, the first off-diagonal contains weights between nodes separated by one distance unit, the second off diagonal by two units, and so on. This method requires $2n$ values computed from A to characterize the null model, the n normalized row (or column) sums and the n normalized diagonal sums. Although it is not entirely correct for networks arranged on a 2D grid, it can be used and yeilds better performance than the Independent-Vertex null model.

Grid-Distance Dependent, Independent-Vertex Random Graph in 2D (Mod SKH Euc)

A more correct null model for networks constructed from images admits the arrangement of nodes in a 2D lattice. The setup follows very closely the construction discussed above in the

Line-Distance Dependent case with independent contributions from node degrees and from the connectivity-distance relationship across the entire network. When nodes are arranged in a two dimensional grid, however, the relationship between distance in the network and location in the adjacency matrix is no longer simple to express mathematically, as in diagonal sums of **A** in the 1D case. Fig. A.3 below shows entries in the adjacency matrix representing the collection of edges separating pairs of nodes by the distance indicated in each pane in an 11x11 image patch.

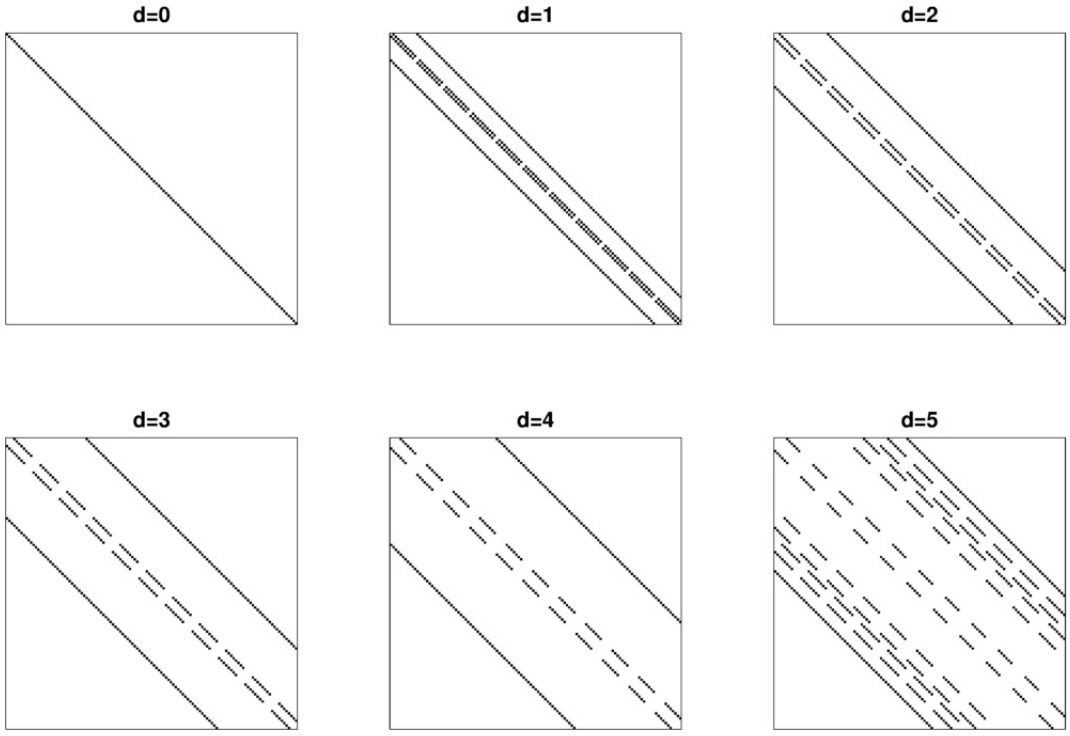


Figure A.3: **Grid-Distance Dependence:** Distance mask in **A** matrix: Elements within the adjacency matrix that are separated by distance $d = |r_i - r_j|$ in an 11x11 network arranged on a 2D lattice.

For all but $|r_i - r_j| = 0$, distances in the image plane translate into patterns in the adjacency matrix that are more complex than just off-diagonals. Note that each pattern includes some of the $|r_i - r_j|^{th}$ off-diagonal, with additional entries resulting from the way which the nxn image is rasterized to make to form the n^2xn^2 adjacency matrix. In our implementation, we do not attempt to express the $b_{|r_i - r_j|}$ term analytically, rather we algorithmically compute distances in the image plane and construct an adjacency matrix mask for each distance that we use to compute the distance-dependent average connectivity. Aside from difference in implementation, the motivation behind this model is identical to the 1D case. Here specifically,

$$\mathbf{E}[A_{ij}|\frac{d_i}{n},\frac{d_j}{n},\frac{b_{|r_i-r_j|}}{\#b_{|r_i-r_j|}}] = \int A_{ij}p(A_{ij}|\frac{d_i}{n},\frac{d_j}{n},\frac{b_{|r_i-r_j|}}{\#b_{|r_i-r_j|}})dA_{ij} = \int A_{ij}\delta(A_{ij}-c\frac{d_i}{n}\frac{d_j}{n}\frac{b_{|r_i-r_j|}}{\#b_{|r_i-r_j|}})dA_{ij} \quad \text{(A.17)}$$

$$N_{ij} = \mathbf{E}[A_{ij}|\frac{d_i}{n},\frac{d_j}{n},\frac{b_{|r_i-r_j|}}{\#b_{|r_i-r_j|}}] = c\frac{d_i}{n}\frac{d_j}{n}\frac{b_{|r_i-r_j|}}{\#b_{|r_i-r_j|}} \quad \text{(A.18)}$$

where

$$d_i = \sum_{i=1}^{n} A_{ij} \quad \text{(A.19)}$$

and $b_{|r_i-r_j|}$ is implemented by masks illustrated in Fig. A.3. Here, the $\#b_{|r_i-r_j|}$ term refers to the number of non-zero entries in the mask for the given distance. Since edges are undirected and $\mathbf{A}$ is symmetric, the distance mask could also be implemented using the upper or lower triangular version of the adjacency matrix.

Solving for c by equation A.3 yields

$$c = \frac{n^2\mathbf{\Sigma}}{\sum_{ij}(d_i d_j \frac{b_{|r_i-r_j|}}{\#b_{|r_i-r_j|}})} \quad \text{(A.20)}$$

and the full null model with the normalization constant is

$$N_{ij} = \frac{d_i d_j \frac{b_{|r_i-r_j|}}{\#b_{|r_i-r_j|}}\mathbf{\Sigma}}{\sum_{ij}(d_i d_j \frac{b_{|r_i-r_j|}}{\#b_{|r_i-r_j|}})}. \quad \text{(A.21)}$$

Temporal Modularity Null Model

While topographic modularity models are powerful tools for image segmentation, it is difficult to interpret how they could be implemented in retinal circuitry. The distance-dependent term $b_{|r_i-r_j|}$ requires that each edge in the network have access to global knowledge, namely the average edge weight across the entire network of all edges that span the same physical distance for the current input stimulus. However, the null model can constructed with only local information if each neuron pair samples and stores the average edge weight between them over an ensemble of past stimuli. Hebbian plasticity in the ganglion-amacrine cell anatomical connectivity network could nicely account for such a computation.

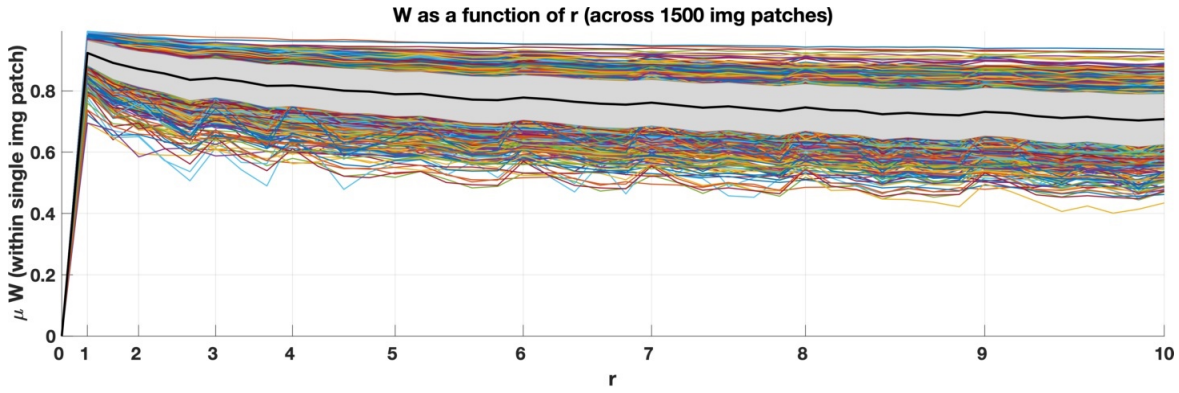


Figure A.4: **Adjacency edge weight vs distance**: Average edge weight between node pairs in the adjacency matrix separated by distance r as a function of distance in image. Colored lines denote individual image patches and black line with grey error bars indicates μ and σ across 1500 image patches that are 50x50pixels.

Within a single scene or image, this spatial statistic can be converted to a local, temporal statistic via eye movements in a persistent scene if the timescale of plasticity is shorter than the scene duration [**zenke2017**]. For longer Hebbian timescales, the argument holds across an ensemble of natural scenes in so far as the distance-dependent feature similarity in single images is captured by an average across the ensemble. Pixel values in images of natural scenes have been shown to be much more highly correlated for nearby pairs of pixels than for distant pairs [**atick1992**].Fig. A.4 shows the average weight in the Adjacency matrix across all node pairs i and j separated by a distance $r = |r_i - r_j|$ as a function of r, within single image patches as colored lines and the mean and standard deviation across an ensemble in black and grey.

A further advantage of a temporally sampled null model, beyond node degree and distance-dependence, is that *all* parameters describing the relationship between cells (such as cell types and direction) are trivially captured the cell pair itself is used to compute the null model. Thus the null model effectively controls for all influences to network connectivity other than image content, which is marginalized out over many samples across time. The temporal null model has not been explored in this work and is left for future development.

Appendix B

Cell Assembly Model Supplement

We reserve this supplemental text to catalog and give brief discussion on details of the Cell Assembly Model. This includes known issues and limitations, work thought about or begun but not followed through with, pointers to ideas to try in the future for anyone continuing with this, and other things not mentioned in the body of the chapter 2.

1. We focused our analysis on spike-words with 5ms binning because they yielded the most robust structure, although some structure was found in 3ms binned spike-words as well.

2. Run pairwise GLM simulation for groups of cells ($\sim 10 - 20$) that participate in particularly interesting looking CAs.

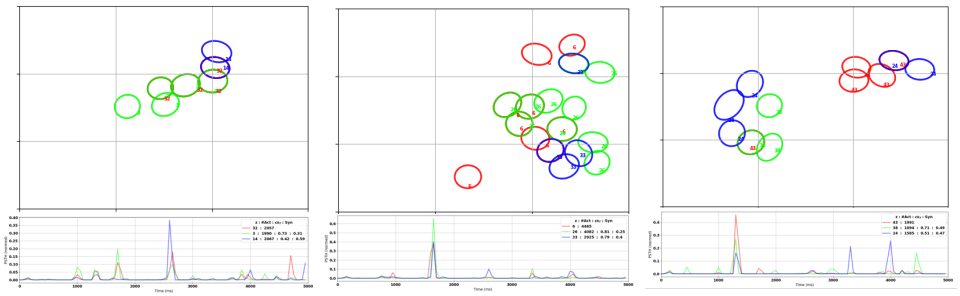
3. The size of the latent dimension, M, is a hyper-parameter of the model. We did some initial investigation into over- and under-complete models with synthetic data, with $M > N$ and $M < N$ respectively. We hypothesize that model completeness will effect the type of structure discovered by CAs, especially when using the "Homeostatic Egalitarian" prior.

4. When CAs in the same model are often co-active, do they provide redundant or synergistic information? Defined by:

$$S(a, b) = 1 - \sqrt{cs_\tau(a, b) \cdot cs_M(a, b)} \tag{B.1}$$

   where $cs_\tau(a, b)$ and $cs_M(a, b)$ are the temporal and membership cosine similarity between CAs a and b within the same model. For a single CA, we are most interested in the minimum S value across the rest of the population. Visualizing individual CAs does not reveal the full picture because observed spike words can be explained by simultaneous activity of multiple CAs. We look for CAs within a model that are commonly coactive, determined by cosine similarity of their PSTHs to determine if they have form interesting larger shapes. Fig. B.2 shows CA (in red) along with two additional CAs with which

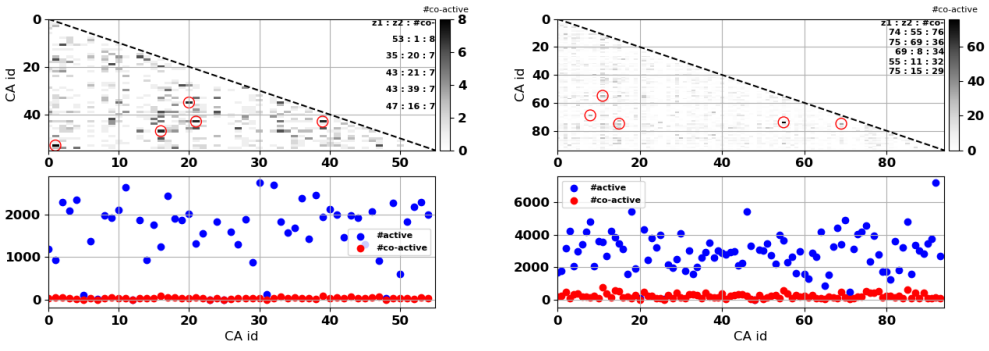
it shares a large temporal overlap. Red and green CAs have more overlapping spatial and temporal representation in panel a and more synergistic representation in panel c. Panel b shows $z0$ example from above in red. Even this does not reveal the full picture. These CAs are determined to be temporally coactive by cosine similarity of PSTHs binned at 50ms. First, this does not indicate that they are necessarily coactive in the same trials. Second, if coactive in the same trials, they could at different times within the 50ms bins.



(a) Redundant (b) $z6$ example above (c) Synergistic

Figure B.1: **Coactive CAs can be synergystic or redundant:** *Bottom* PSTH traces show high temporal overlap between activations of 3 CAs. *Top left* shows high RF overlap for those CAs as well.

Empirically on the time-scale of individual spike-words, CAs are not often co-active relative to the number of times they are active individually. Fig. **??** shows a few typical, randomly sampled examples.



(a) [offBT] (b) [offBT,onBT]

Figure B.2: **CA individual inference and coactivity statistics:** Two panels show statistics for inference on all spike-words in data corpus after model is learned and fixed. Cell-type listed in panel caption. In each panel, CAs on x-axis. Blue points show number of time each CA was inferred across all spike-words. Red points show total number of times it was inferred with a partner. Top plot shows pairwise inference coactivity with 5 largest values circled in red. Coactivity among CAs is pretty insignificant.

5. We construct spike-words with a bin-size of 5ms but a step-size of 1ms. So individual spikes are used in multiple spike-words. This process introduces noisy repeats into spike words used introducing a couple of confounds. First, spike-words are used for learning and introduced noisy repeat structure can be confounded with structure in with actual noisy repeats from cell assemblies. Second, these spike-words are also used inference, to construct CA activity rasters. We attempted to do an ISI / Fano factor analysis to uncover periodic structure in CA activity, but had many 1ms and 2ms ISIs introduced by how the data set was constructed.

6. ΔPy is an approximate, imperfect measure that introduces some confounds into the comparison with the GLM p(y). It correctly obtains high values when a CA is active and GLM rates predict low synchrony. However, large ΔPy values (ie. small cosine similarity values) may also result from GLM predicted activity not observed in z_a's PSTH. This may not reflect real significance because multiple CAs can learn overlapping cell membership and predicted activity at one moment can be partially or fully subsumed in the activity of another CA, leaving the first, z_a, inactive. However, ΔPy only considers the PSTH of z_a, and the high activity at one time in $\langle p(\vec{y}_{null})\rangle$ which is unmatched in the PSTH dramatically changes the angle between vectors in high dimensional space because it lowers the height or significance of other events in $\langle p(\vec{y}_{null})\rangle$ relative to that max. This results in a low cosine similarity and a high significance in cases where similar CAs in the same model work together to to represent the activity of cells at different times. A more complete significance measure would consider the membership overlap of CAs in a model and allow for additional CAs with similar membership to absorb some of the GLM model prediction. It is probably not too hard to extend it to allow the significance comparison to use PSTHs from a couple z_a's with very similar spatial/membership similarity.

7. An additional measure of the difference of spiking activity given $z_a = 1$ from null model predictions is the KL-divergence between N-dimensional multivariate Bernoulli distributions of $p(y_i)_{null}$ and $p(y_i|z_a = 1, z_{a'} = 0)$, shown and mentioned in Fig. 2.13. It more explicitly makes the same confounding and faulty assumption that all other $z_{a'}$'s are inactive. It may provide a more straight-forward path to generalize the ΔPy metric.