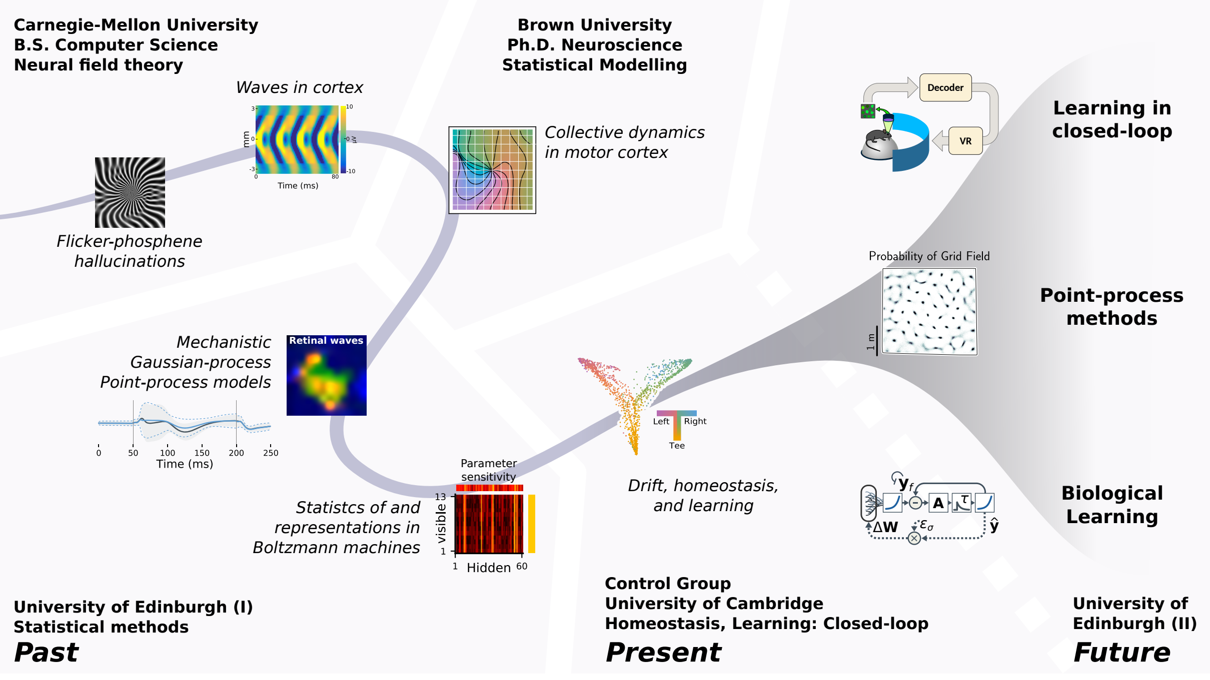
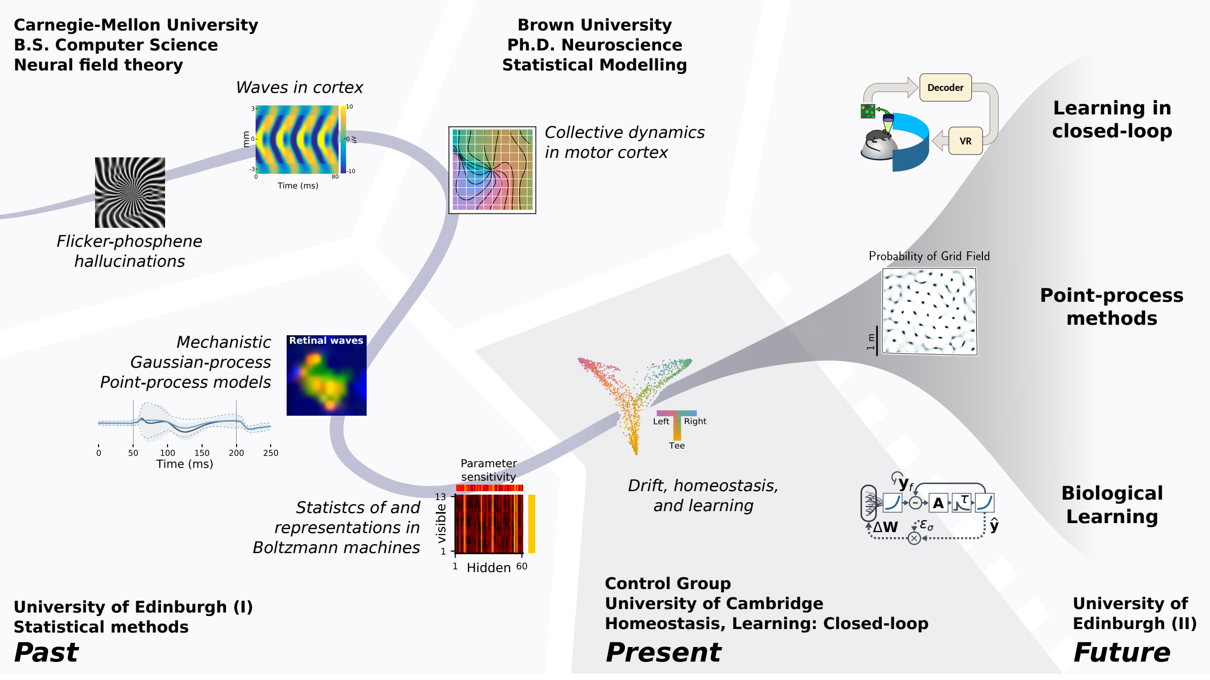
Self-Healing Neural Codes

Changing to stay the same

M. Rule

A job talk for lecturer in machine learning at the University of Edinburgh School of Informatics
20th June, 2022


Carnegie-Mellon University
B.S. Computer Science
Neural field theory
Waves in cortex
mm
Time (ms)
Flicker-phosphene hallucinations
Brown University
Ph.D. Neuroscience
Statistical Modelling
Collective dynamics in motor cortex
Decoder
VR
Learning in closed-loop
Probability of Grid Field
1 m
Point-process methods
Mechanistic Gaussian-process Point-process models
Retinal waves
0
50
100
150
200
250
Time (ms)
Left
Right
Tee
Drift, homeostasis, and learning
Parameter sensitivity
13
visible
1
1
Hidden
60
Statistcs of and representations in Boltzmann machines
ΔW
A
Biological Learning
University of Edinburgh (I)
Statistical methods
Past
Control Group
University of Cambridge
Homeostasis, Learning: Closed-loop
Present
University of Edinburgh (II)
Future


Carnegie-Mellon University
B.S. Computer Science
Neural field theory
Waves in cortex
mm
Time (ms)
Flicker-phosphene hallucinations
Brown University
Ph.D. Neuroscience
Statistical Modelling
Collective dynamics in motor cortex
Decoder
VR
Learning in closed-loop
Probability of Grid Field
1 m
Point-process methods
Mechanistic
Gaussian-process
Point-process models
Retinal waves
0
50
100
150
200
250
Time (ms)
Left
Right
Tee
Parameter sensitivity
13
visible
1
Hidden
60
Statistcs of and representations in Boltzmann machines
Drift, homeostasis, and learning
ΔW
A
Biological Learning
University of Edinburgh (I)
Statistical methods
Past
Control Group
University of Cambridge
Homeostasis, Learning: Closed-loop
Present
University of Edinburgh (II)
Future

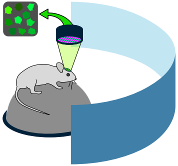
How do real neurons learn?

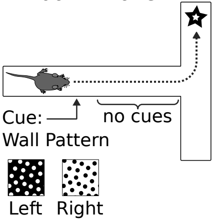
The brain is plastic.
The brain remembers.

How is the brain changing?

Driscoll & al. 2017 experiments:

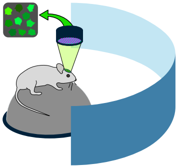
- Image neural population activity over time in a fixed virtual-reality task

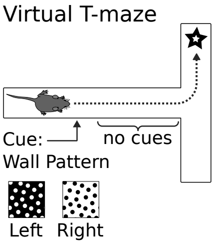




Driscoll & al. 2017 experiments:

- Image neural population activity over time in a fixed virtual-reality task
- Posterior Parietal Cortex (PPC): Association area required for task

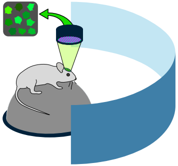


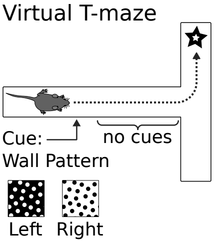


How is the brain changing?

Driscoll & al. 2017 experiments:

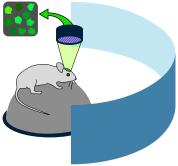
- Image neural population activity over time in a fixed virtual-reality task
- Posterior Parietal Cortex (PPC): Association area required for task
- Long-term recording at steady state: task performed at expert level

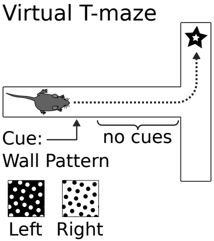




Driscoll & al. 2017 experiments:

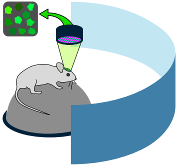
- Image neural population activity over time in a fixed virtual-reality task
- Posterior Parietal Cortex (PPC): Association area required for task
- Long-term recording at steady state: task performed at expert level
- No change in performance or behavior

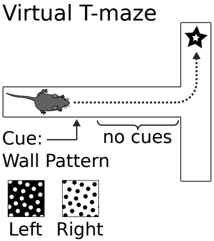


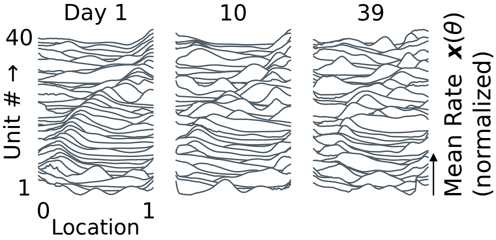


Driscoll & al. 2017 experiments:

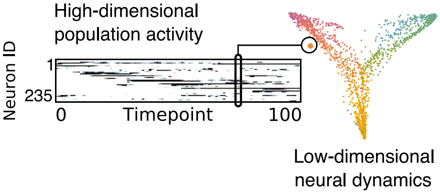
- Image neural population activity over time in a fixed virtual-reality task
- Posterior Parietal Cortex (PPC): Association area required for task
- Long-term recording at steady state: task performed at expert level
- No change in performance or behavior
- Neural population code “drifts”

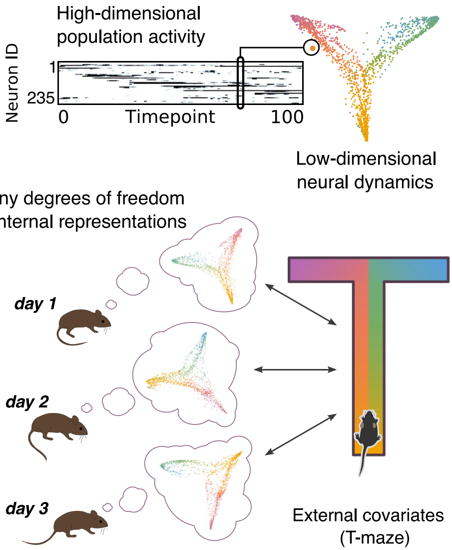




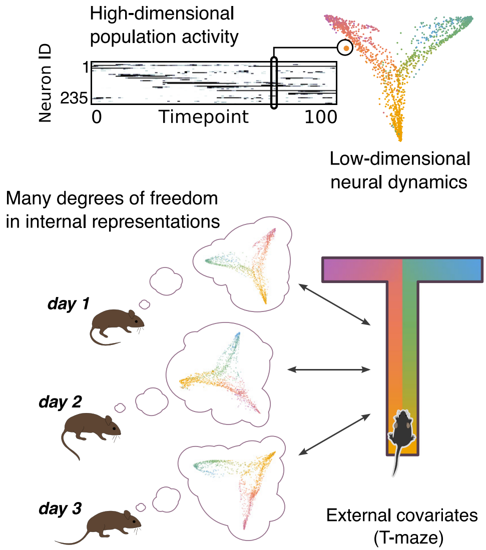


… Neurons’ roles change over time, even in representations supporting fixed, habitual tasks.


High-dimensional
population activity
Neuron ID
1
235
0
Timepoint
100
Low-dimensional
neural dynamics



Rule, M. E., O'Leary, T., and Harvey, C. D. (2019). Causes and consequences of representational drift. Current opinion in neurobiology, 58:141–147

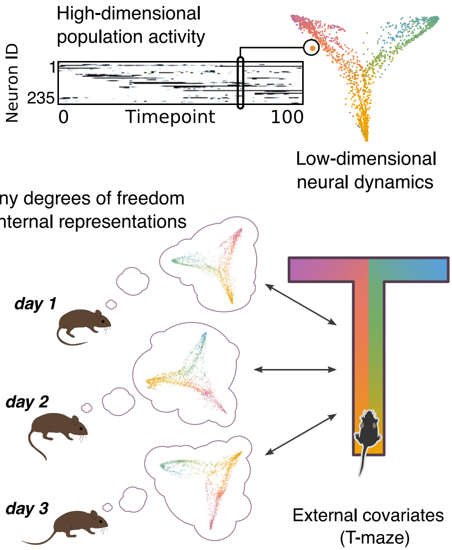


Ubiquitous, not uniform:

- **Hippocampus:**[1,2] Fast turnover; Episodic memory?
- **Olfactory:**[3] Experience-dependent: A code for novelty?
- **Visual**[4], **Prefrontal:**[5]: More stable near periphery?

[1]Ziv & al. (2013); [2]Levy & al. (2021); [3]Schoonover, Fink & al. (2020); [4]Deitch & al. (2020); [5]Singh & al. (2019);

Rule, M. E., O'Leary, T., and Harvey, C. D. (2019). Causes and consequences of representational drift. Current opinion in neurobiology, 58:141–147



Ubiquitous, not uniform:

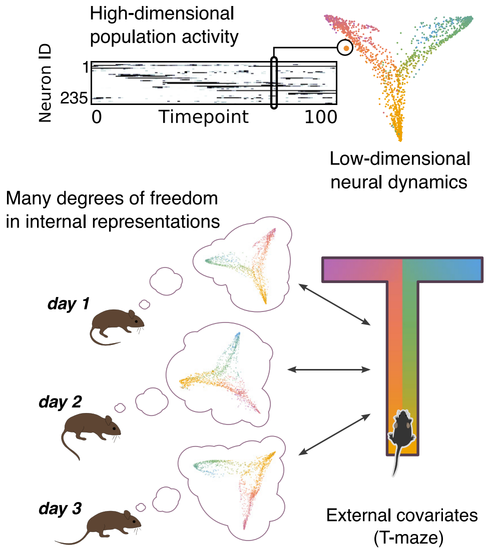
- **Hippocampus:**[1,2] Fast turnover; Episodic memory?
- **Olfactory:**[3] Experience-dependent: A code for novelty?
- **Visual**[4], **Prefrontal:**[5]: More stable near periphery?

[1]Ziv & al. (2013); [2]Levy & al. (2021); [3]Schoonover, Fink & al. (2020); [4]Deitch & al. (2020); [5]Singh & al. (2019);

What can drift tell us about learning?

- Learning in over-parameterized networks?
- Strategies for rapid learning?
- Representations in large generative models?
- Avoiding catastrophic forgetting in continual learning?

Rule, M. E., O'Leary, T., and Harvey, C. D. (2019). Causes and consequences of representational drift. Current opinion in neurobiology, 58:141–147



Ubiquitous, not uniform:

- **Hippocampus:**[1,2] Fast turnover; Episodic memory?
- **Olfactory:**[3] Experience-dependent: A code for novelty?
- **Visual**[4], **Prefrontal:**[5]: More stable near periphery?

[1]Ziv & al. (2013); [2]Levy & al. (2021); [3]Schoonover, Fink & al. (2020); [4]Deitch & al. (2020); [5]Singh & al. (2019);

What can drift tell us about learning?

- Learning in over-parameterized networks?
- Strategies for rapid learning?
- Representations in large generative models?
- Avoiding catastrophic forgetting in continual learning?

Rule, M. E., O'Leary, T., and Harvey, C. D. (2019). Causes and consequences of representational drift. Current opinion in neurobiology, 58:141–147

How does the brain achieve stable behavior despite internal change?

Could other neurons read a code like this?

How does the brain achieve stable behavior despite internal change?

Could other neurons read a code like this?

Proposition:

- Homeostasis stabilizes single neuron properties **and learned associations**
- Hebbian plasticity maintains learned representation **by re-enforcing existing correlation** structure

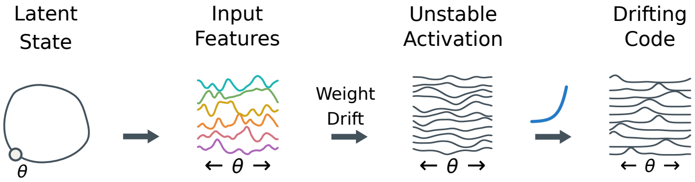
Supported by

LEVERHULME TRUST

early-career fellowship

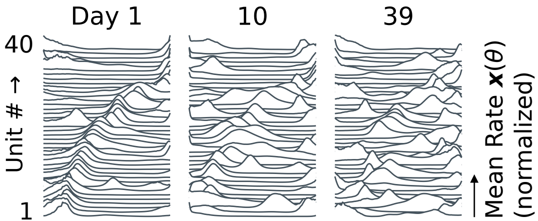
Unstable code, stable latent structure: A problem for long-term stability?

Model drift: Random features + drifting weights + homeostatic normalization

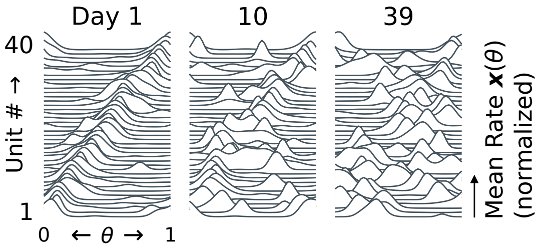


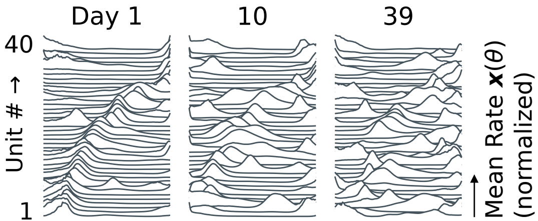
Sensitivity & threshold homeostasis preserve localized tuning†

† **Homeostasis:** Cannon & Miller (2017, 19),



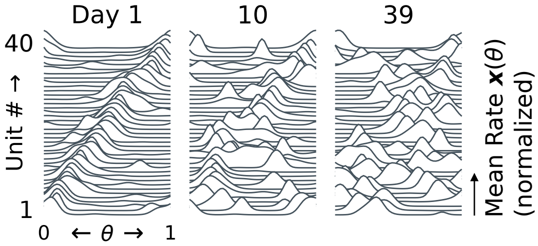
Simulated drift: which is data, which is model?

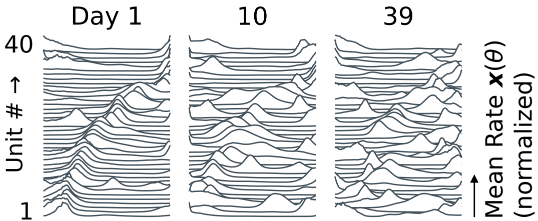




This one is data

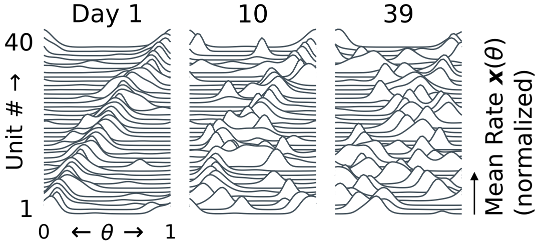
Simulated drift: which is data, which is model?





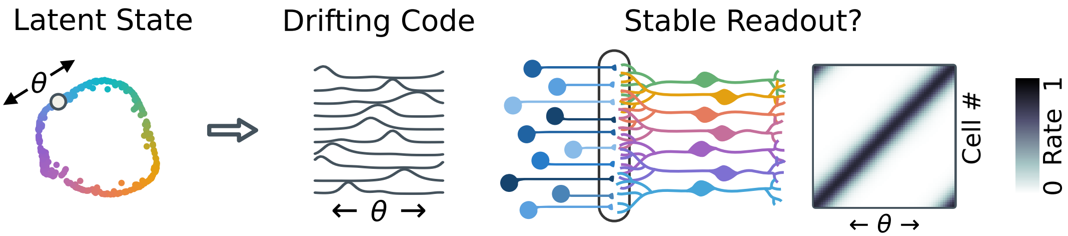
This one is data

Simulated drift: which is data, which is model?



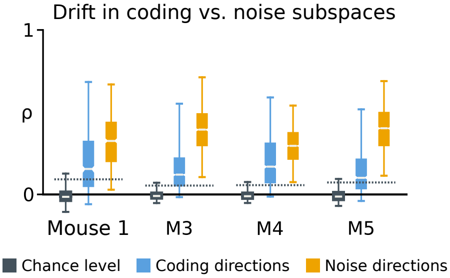
This one is model

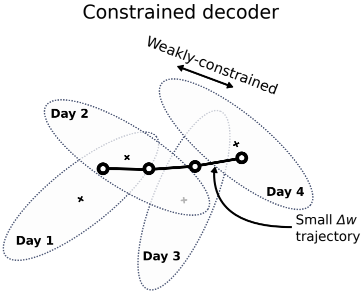
Is a stable readout possible?



A readout population learns to decode location from an encoding population that drifts.
Can it preserve this readout as the encoding changes?

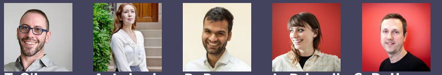
Yes, with some ongoing learning

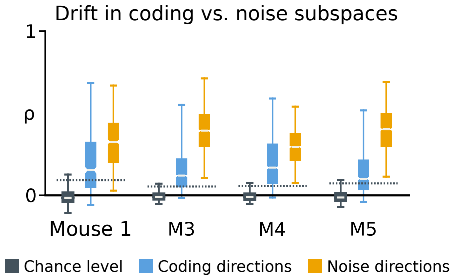


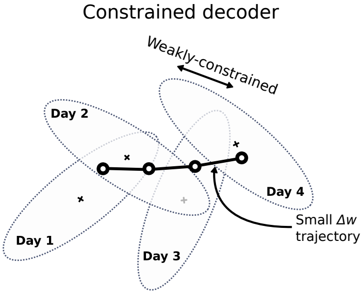


Rule, M. E., Loback, A. R., Raman, D. V., Driscoll, L. N., Harvey, C. D., and O'Leary, T. (2020). Stable task information from an unstable neural population. Elife, 9:e51121

Yes, with some ongoing learning



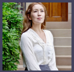




Rule, M. E., Loback, A. R., Raman, D. V., Driscoll, L. N., Harvey, C. D., and O'Leary, T. (2020). Stable task information from an unstable neural population. Elife, 9:e51121

- Most drift looks like trial-to-trial variability on slow time-scales

Yes, with some ongoing learning

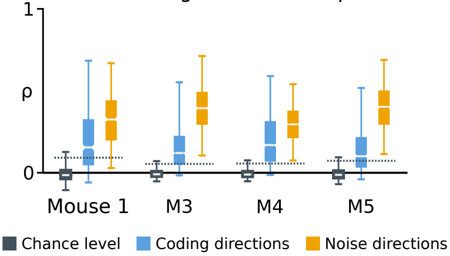

T. O'Leary

A. Loback

D. Raman

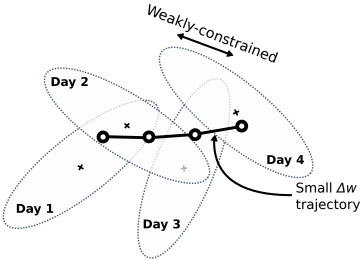
L. Driscoll

C. D. Harvey

Drift in coding vs. noise subspaces



Constrained decoder

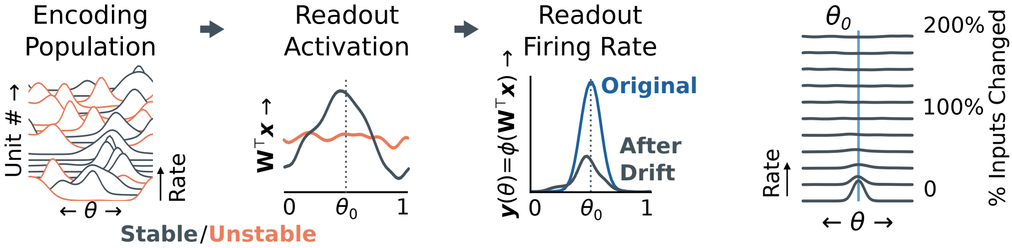


Rule, M. E., Loback, A. R., Raman, D. V., Driscoll, L. N., Harvey, C. D., and O'Leary, T. (2020). Stable task information from an unstable neural population. Elife, 9:e51121

- Most drift looks like trial-to-trial variability on slow time-scales
- Change in the underlying representation is gradual
  - Only small weight changes needed to stabilize decoding

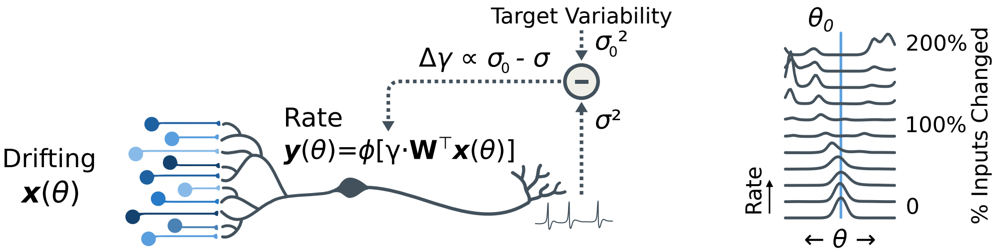
Can we track a drifting code for continuous variables
without an external reference?

Fixed weights: Drift attenuates excitatory drive



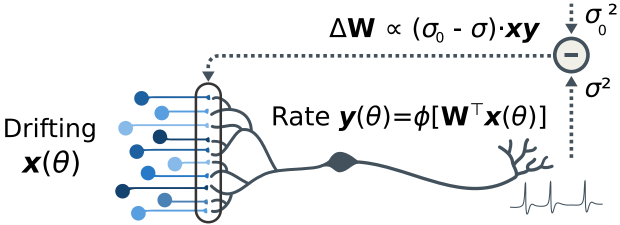
Readout neurons require a specific conjunction of inputs to fire.
Random drift destroys this excitatory drift, but doesn't change tuning.

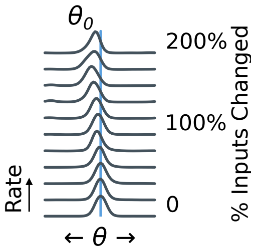
Sensitivity homeostasis: Tuning is robust to modest change



Homeostatically scaling up weights stabilizes activity against small amounts of drift, since the input is somewhat redundant.

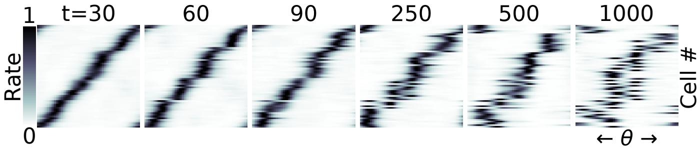
Hebbian homeostasis: Use the neuron's own output as a training signal



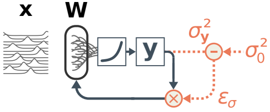


Using the neuron's own output to adjust weights stabilizes localized, bump-like tuning. This uses unsupervised Hebbian learning to homeostatically restore activity as drift destroys excitatory drive.

Hebbian Homeostasis



Schematic



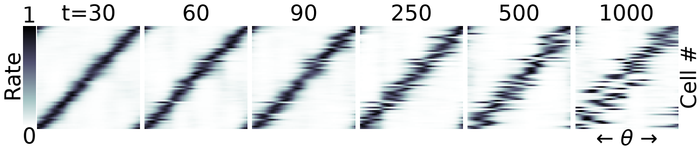
$$\Delta \mathbf{W} \propto \varepsilon_\sigma (\boldsymbol{x}\boldsymbol{y}^\top - c\mathbf{W})$$

Hebbian Homeostasis provides stability, but tuning decays toward high-variance components†

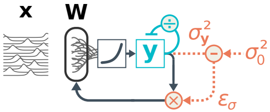
† c.f. Oja (1982)

Use recurrent predictions as a learning signal

Population Competition



Schematic



$\Delta \mathbf{W} \propto \varepsilon_\sigma (\mathbf{x}\mathbf{y}^\top - c\mathbf{W})$

Hebbian Homeostasis provides stability, but tuning decays toward high-variance components[†]

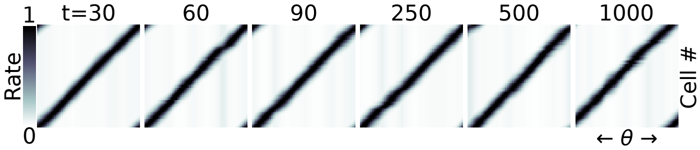
Competition between readout cells stabilizes further[‡]

[†] c.f. Oja (1982)

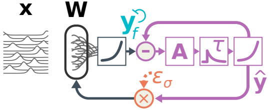
[‡] e.g. Földiák & Fdilr, Barlow & Földiák (1989); Pehlevan & al. (2015, 17, 19, 20)

Use recurrent predictions as a learning signal

Recurrent Dynamics



Schematic



$$\Delta\mathbf{W} \propto \varepsilon_\sigma \left(\boldsymbol{x}\boldsymbol{y}^\top - c\mathbf{W}\right)$$

Hebbian Homeostasis provides stability, but tuning decays toward high-variance components[†]

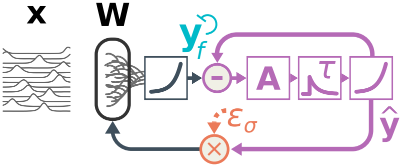
Competition between readout cells stabilizes further[‡]

Recurrent predictions improve error-correction ⟶ long-term stability

[†] c.f. Oja (1982)

[‡] e.g. Földiák & Fdilr, Barlow & Földiák (1989); Pehlevan & al. (2015, 17, 19, 20)

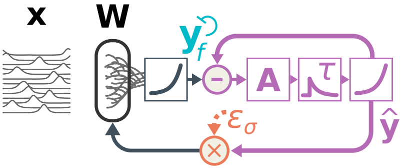
Repair readout using a predictive model stored in recurrent weights



Generative model of spikes $\boldsymbol{y} = \{y_1, .., y_n\}$

- Latent variables $\boldsymbol{z} = \{z_1, .., z_n\} \sim \mathrm{P}_{\boldsymbol{z}}$
- Observation model $\mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}}$
- $\mathrm{P}_{\boldsymbol{y}} = \int_{d\boldsymbol{z}} \mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}} \mathrm{P}_{\boldsymbol{z}}$

Repair readout using a predictive model stored in recurrent weights

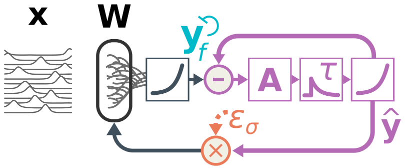


Generative model of spikes $\boldsymbol{y} = \{y_1, .., y_n\}$

- Latent variables $\boldsymbol{z} = \{z_1, .., z_n\} \sim \mathrm{P}_{\boldsymbol{z}}$
- Observation model $\mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}}$
- $\mathrm{P}_{\boldsymbol{y}} = \int_{d\boldsymbol{z}} \mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}} \mathrm{P}_{\boldsymbol{z}}$

Learn a prior on likely states in latent variables $\boldsymbol{z}$, and how to map $\boldsymbol{z} \leftrightarrow \boldsymbol{y}$. Save this prior, and use it to repair readouts in the face of drift.

Repair readout using a predictive model stored in recurrent weights



Specifically, a Log-Gaussian-Poisson model:

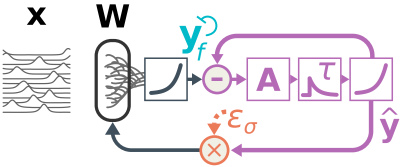
$$
\begin{aligned}
P_{\boldsymbol{z}} &= \mathcal{N}[0,\Sigma] \propto \exp(-\tfrac{1}{2}\boldsymbol{z}^\top\Sigma^{-1}\boldsymbol{z}) \\
P_{y_i|z_i} &= \text{Poisson}[\lambda_i = e^{z_i}] \propto \lambda_i^{y_i} e^{-\lambda_i} \\
\ln P_{\boldsymbol{z}|\boldsymbol{y}} &= \boldsymbol{y}^\top\boldsymbol{z} - \mathbb{1}^\top\boldsymbol{\lambda}(z) - \tfrac{1}{2}\boldsymbol{z}^\top\Sigma^{-1}\boldsymbol{z}
\end{aligned}
$$

Generative model of spikes $\boldsymbol{y} = \{y_1, .., y_n\}$

- Latent variables $\boldsymbol{z} = \{z_1, .., z_n\} \sim P_{\boldsymbol{z}}$
- Observation model $P_{\boldsymbol{y}|\boldsymbol{z}}$
- $P_{\boldsymbol{y}} = \int_{dz} P_{\boldsymbol{y}|\boldsymbol{z}} P_{\boldsymbol{z}}$

Learn a prior on likely states in latent variables $\boldsymbol{z}$, and how to map $\boldsymbol{z} \leftrightarrow \boldsymbol{y}$. Save this prior, and use it to repair readouts in the face of drift.

Repair readout using a predictive model stored in recurrent weights



Generative model of spikes $\boldsymbol{y} = \{y_1, .., y_n\}$

- Latent variables $\boldsymbol{z} = \{z_1, .., z_n\} \sim \mathrm{P}_{\boldsymbol{z}}$
- Observation model $\mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}}$
- $\mathrm{P}_{\boldsymbol{y}} = \int_{dz} \mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}} \mathrm{P}_{\boldsymbol{z}}$

Learn a prior on likely states in latent variables $\boldsymbol{z}$, and how to map $\boldsymbol{z} \leftrightarrow \boldsymbol{y}$. Save this prior, and use it to repair readouts in the face of drift.

Specifically, a Log-Gaussian-Poisson model:

$$\mathrm{P}_{\boldsymbol{z}} = \mathcal{N}[0, \Sigma] \propto \exp(-\tfrac{1}{2}\boldsymbol{z}^\top \Sigma^{-1} \boldsymbol{z})$$

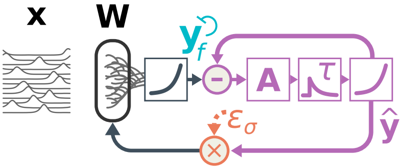
$$\mathrm{P}_{y_i|z_i} = \mathrm{Poisson}[\lambda_i = e^{z_i}] \propto \lambda_i^{y_i} e^{-\lambda_i}$$

$$\ln \mathrm{P}_{\boldsymbol{z}|\boldsymbol{y}} = \boldsymbol{y}^\top \boldsymbol{z} - \mathbb{1}^\top \boldsymbol{\lambda}(\boldsymbol{z}) - \tfrac{1}{2}\boldsymbol{z}^\top \Sigma^{-1} \boldsymbol{z}$$

Error correct: get $\boldsymbol{z}$ most likely to explain $\boldsymbol{y}$

$$\dot{\boldsymbol{z}} \propto \nabla_{\boldsymbol{z}} \ln \mathrm{P}_{\boldsymbol{z}|\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{\lambda}(\boldsymbol{z}) - \Sigma^{-1}\boldsymbol{z}$$

Repair readout using a predictive model stored in recurrent weights



Generative model of spikes $\boldsymbol{y} = \{y_1, .., y_n\}$

- Latent variables $\boldsymbol{z} = \{z_1, .., z_n\} \sim \mathrm{P}_{\boldsymbol{z}}$
- Observation model $\mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}}$
- $\mathrm{P}_{\boldsymbol{y}} = \int_{dz} \mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}} \mathrm{P}_{\boldsymbol{z}}$

Learn a prior on likely states in latent variables $\boldsymbol{z}$, and how to map $\boldsymbol{z} \leftrightarrow \boldsymbol{y}$. Save this prior, and use it to repair readouts in the face of drift.

Specifically, a Log-Gaussian-Poisson model:

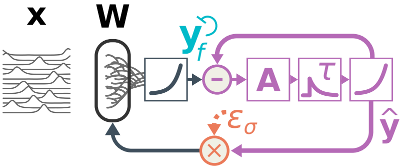
$$\mathrm{P}_{\boldsymbol{z}} = \mathcal{N}[0, \Sigma] \propto \exp(-\tfrac{1}{2}\boldsymbol{z}^\top \Sigma^{-1} \boldsymbol{z})$$
$$\mathrm{P}_{y_i|z_i} = \mathrm{Poisson}[\lambda_i = e^{z_i}] \propto \lambda_i^{y_i} e^{-\lambda_i}$$
$$\ln \mathrm{P}_{\boldsymbol{z}|\boldsymbol{y}} = \boldsymbol{y}^\top \boldsymbol{z} - \mathbb{1}^\top \boldsymbol{\lambda}(\boldsymbol{z}) - \tfrac{1}{2}\boldsymbol{z}^\top \Sigma^{-1} \boldsymbol{z}$$

Error correct: get $\boldsymbol{z}$ most likely to explain $\boldsymbol{y}$

$$\dot{\boldsymbol{z}} \propto \nabla_{\boldsymbol{z}} \ln \mathrm{P}_{\boldsymbol{z}|\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{\lambda}(\boldsymbol{z}) - \Sigma^{-1}\boldsymbol{z}$$

Equivalent: $\dot{\boldsymbol{z}} \propto -\boldsymbol{z} + \Sigma[\boldsymbol{y} - \boldsymbol{\lambda}(\boldsymbol{z})]$

Repair readout using a predictive model stored in recurrent weights



Generative model of spikes $\boldsymbol{y} = \{y_1, .., y_n\}$

- Latent variables $\boldsymbol{z} = \{z_1, .., z_n\} \sim \mathrm{P}_{\boldsymbol{z}}$
- Observation model $\mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}}$
- $\mathrm{P}_{\boldsymbol{y}} = \int_{dz} \mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}} \mathrm{P}_{\boldsymbol{z}}$

Learn a prior on likely states in latent variables $\boldsymbol{z}$, and how to map $\boldsymbol{z} \leftrightarrow \boldsymbol{y}$. Save this prior, and use it to repair readouts in the face of drift.

Specifically, a Log-Gaussian-Poisson model:

$$\mathrm{P}_{\boldsymbol{z}} = \mathcal{N}[0, \Sigma] \propto \exp(-\tfrac{1}{2}\boldsymbol{z}^\top \Sigma^{-1} \boldsymbol{z})$$

$$\mathrm{P}_{y_i|z_i} = \text{Poisson}[\lambda_i = e^{z_i}] \propto \lambda_i^{y_i} e^{-\lambda_i}$$

$$\ln \mathrm{P}_{\boldsymbol{z}|\boldsymbol{y}} = \boldsymbol{y}^\top \boldsymbol{z} - \mathbb{1}^\top \boldsymbol{\lambda}(\boldsymbol{z}) - \tfrac{1}{2}\boldsymbol{z}^\top \Sigma^{-1} \boldsymbol{z}$$

Error correct: get $\boldsymbol{z}$ most likely to explain $\boldsymbol{y}$

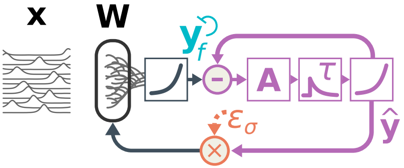
$$\dot{\boldsymbol{z}} \propto \nabla_{\boldsymbol{z}} \ln \mathrm{P}_{\boldsymbol{z}|\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{\lambda}(\boldsymbol{z}) - \Sigma^{-1}\boldsymbol{z}$$

Equivalent: $\dot{\boldsymbol{z}} \propto -\boldsymbol{z} + \Sigma[\boldsymbol{y} - \boldsymbol{\lambda}(\boldsymbol{z})]$

Predictions $\boldsymbol{\lambda}(\boldsymbol{z})$ cancel inputs $\boldsymbol{y}$ via inhibition, through recurrent weights Σ: A rate network, but not quite physiological[1].

[1] compare to Masset & al. (2022), which shares some assumptions;

Repair readout using a predictive model stored in recurrent weights



Generative model of spikes $\boldsymbol{y} = \{y_1, .., y_n\}$

- Latent variables $\boldsymbol{z} = \{z_1, .., z_n\} \sim \mathrm{P}_{\boldsymbol{z}}$
- Observation model $\mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}}$
- $\mathrm{P}_{\boldsymbol{y}} = \int_{dz} \mathrm{P}_{\boldsymbol{y}|\boldsymbol{z}} \mathrm{P}_{\boldsymbol{z}}$

Learn a prior on likely states in latent variables $\boldsymbol{z}$, and how to map $\boldsymbol{z} \leftrightarrow \boldsymbol{y}$. Save this prior, and use it to repair readouts in the face of drift.

Specifically, a Log-Gaussian-Poisson model:

$$\mathrm{P}_{\boldsymbol{z}} = \mathcal{N}[0, \Sigma] \propto \exp(-\tfrac{1}{2}\boldsymbol{z}^\top \Sigma^{-1} \boldsymbol{z})$$

$$\mathrm{P}_{y_i|z_i} = \mathrm{Poisson}[\lambda_i = e^{z_i}] \propto \lambda_i^{y_i} e^{-\lambda_i}$$

$$\ln \mathrm{P}_{\boldsymbol{z}|\boldsymbol{y}} = \boldsymbol{y}^\top \boldsymbol{z} - \mathbb{1}^\top \boldsymbol{\lambda}(\boldsymbol{z}) - \tfrac{1}{2}\boldsymbol{z}^\top \Sigma^{-1} \boldsymbol{z}$$

Error correct: get $\boldsymbol{z}$ most likely to explain $\boldsymbol{y}$

$$\dot{\boldsymbol{z}} \propto \nabla_{\boldsymbol{z}} \ln \mathrm{P}_{\boldsymbol{z}|\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{\lambda}(\boldsymbol{z}) - \Sigma^{-1}\boldsymbol{z}$$

Equivalent: $\dot{\boldsymbol{z}} \propto -\boldsymbol{z} + \Sigma[\boldsymbol{y} - \boldsymbol{\lambda}(\boldsymbol{z})]$

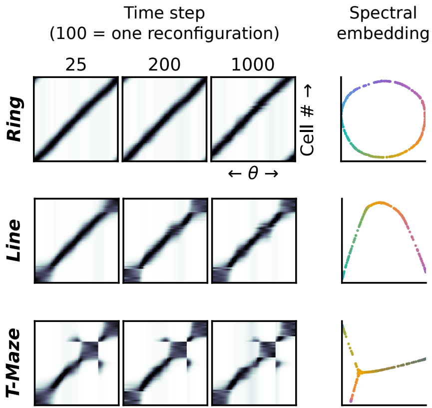
Predictions $\boldsymbol{\lambda}(\boldsymbol{z})$ cancel inputs $\boldsymbol{y}$ via inhibition, through recurrent weights Σ: A rate network, but not quite physiological[1].

These questions remain open: Future work.

[1] compare to Masset & al. (2022), which shares some assumptions;

Drift in low-dimensional representations is benign

Homeostasis and internal predictions confer long-term stability to drift



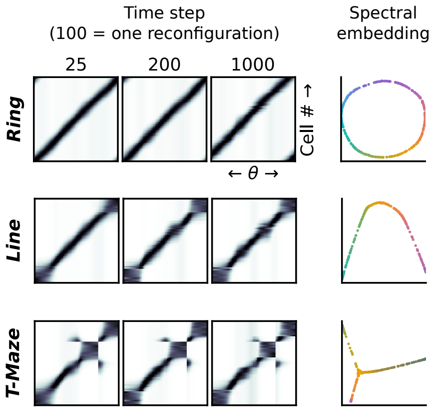
Rule, M. E. and O'Leary, T. (2022). Self-healing codes: How stable neural populations can track continually reconfiguring neural representations. Proceedings of the National Academy of Sciences, 119(7):e2106692119

Drift in low-dimensional representations is benign

Homeostasis and internal predictions confer long-term stability to drift

Drift: Unstable code, stable low-D structure

- Redundant codes have robust readout
- Monitor drift rate as deviation from homeostatic target



Rule, M. E. and O'Leary, T. (2022). Self-healing codes: How stable neural populations can track continually reconfiguring neural representations. Proceedings of the National Academy of Sciences, 119(7):e2106692119

Drift in low-dimensional representations is benign

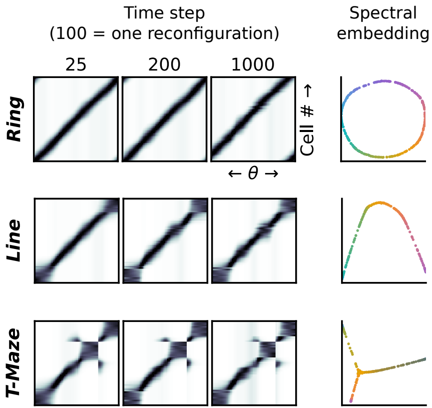
Homeostasis and internal predictions confer long-term stability to drift

Drift: Unstable code, stable low-D structure

- Redundant codes have robust readout
- Monitor drift rate as deviation from homeostatic target

Hebbian homeostasis:

- Correct errors using an internal training signal
- Re-learn tuning as inputs change



Rule, M. E. and O'Leary, T. (2022). Self-healing codes: How stable neural populations can track continually reconfiguring neural representations. Proceedings of the National Academy of Sciences, 119(7):e2106692119

Drift in low-dimensional representations is benign

Homeostasis and internal predictions confer long-term stability to drift

Drift: Unstable code, stable low-D structure

- Redundant codes have robust readout
- Monitor drift rate as deviation from homeostatic target

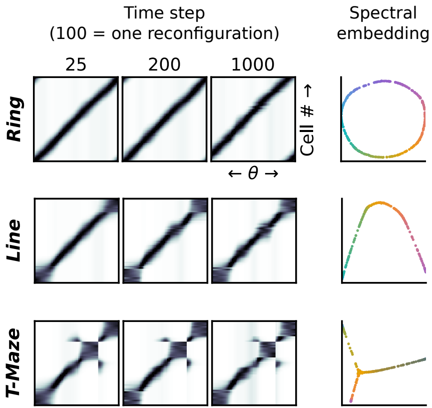
Hebbian homeostasis:

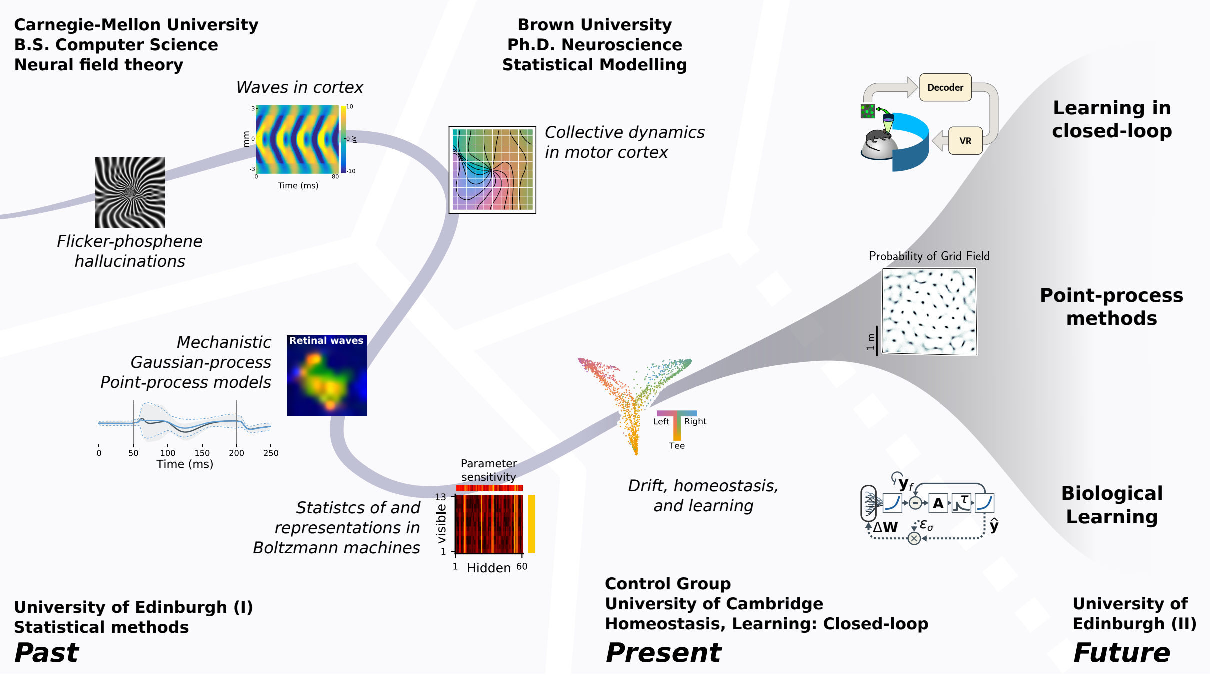
- Correct errors using an internal training signal
- Re-learn tuning as inputs change

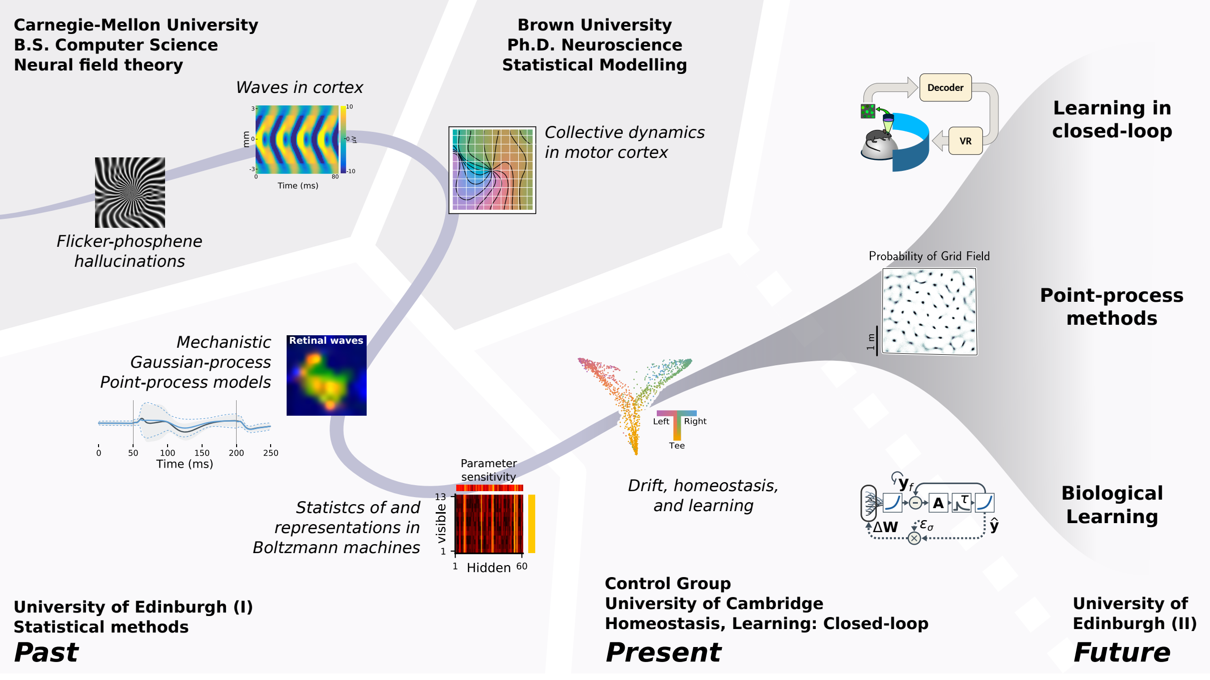
Recurrent population interactions

- Predictions correct errors, provide a stable readout

Rule, M. E. and O'Leary, T. (2022). Self-healing codes: How stable neural populations can track continually reconfiguring neural representations. Proceedings of the National Academy of Sciences, 119(7):e2106692119




Carnegie-Mellon University
B.S. Computer Science
Neural field theory
Waves in cortex
mm
Time (ms)
Flicker-phosphene hallucinations
Brown University
Ph.D. Neuroscience
Statistical Modelling
Collective dynamics in motor cortex
Decoder
VR
Learning in closed-loop
Probability of Grid Field
1 m
Point-process methods
Mechanistic Gaussian-process Point-process models
Retinal waves
0
50
100
150
200
250
Time (ms)
Left
Right
Tee
Drift, homeostasis, and learning
Biological Learning
Parameter sensitivity
Statistcs of and representations in Boltzmann machines
13
visible
1
1
Hidden
60
y_f
A
τ
ΔW
ϵ_σ
$\hat{y}$
University of Edinburgh (I)
Statistical methods
Past
Control Group
University of Cambridge
Homeostasis, Learning: Closed-loop
Present
University of Edinburgh (II)
Future


Carnegie-Mellon University
B.S. Computer Science
Neural field theory
Waves in cortex
mm
Time (ms)
Flicker-phosphene hallucinations
Brown University
Ph.D. Neuroscience
Statistical Modelling
Collective dynamics in motor cortex
Learning in closed-loop
Decoder
VR
Probability of Grid Field
1 m
Point-process methods
Mechanistic
Gaussian-process
Point-process models
Retinal waves
0 50 100 150 200 250
Time (ms)
Parameter sensitivity
visible
Hidden
Statistcs of and representations in Boltzmann machines
Left
Right
Tee
Drift, homeostasis, and learning
Biological Learning
University of Edinburgh (I)
Statistical methods
Past
Control Group
University of Cambridge
Homeostasis, Learning: Closed-loop
Present
University of Edinburgh (II)
Future

Background: *models of the brain, dynamical and statistical*


B. Ermentrout


W. Truccolo

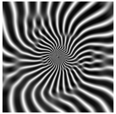
Undergraduate (*Bard Ermentrout*): *Neural-field theory*

(Rule et al., 2011) What do neural networks see?

- 5–40 Hz flickering light: “phosphene” hallucinations[1]
- To detects edges, visual cortex inhibits nearby inputs.
- Inverting input when inhibition arrives excites edges.
- Rate network with recurrent convolutions

(Heitmann, Rule et al. 2017) Optogenetic stimulation switches cortex from travelling waves to oscillations.

[1] Purkinje 1819

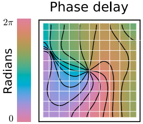


Neural-field theory simplifies collective activity for mathematical tractability.

Ph.D. (*Wilson Truccolo*): *Point-process models*

Collective Dynamics in Primate Motor Cortex

- **(Rule at al., 2017; 2018)** Before moving, rhythmic spiking encodes movement plans; Phase (re)alignments cause waves across cortex.
- **(Rule et al., 2015)** When moving, local electric fields relate to past/future movement; Past spiking predicts future variability.



Point-process models infer why neurons spike

Can we combine these? *Simplify point-process models, infer neural fields from spikes?*

Autoregressive point processes as state-space models

Autoregressive models predict the future from the past.

Coarse graining (“zoom out”):

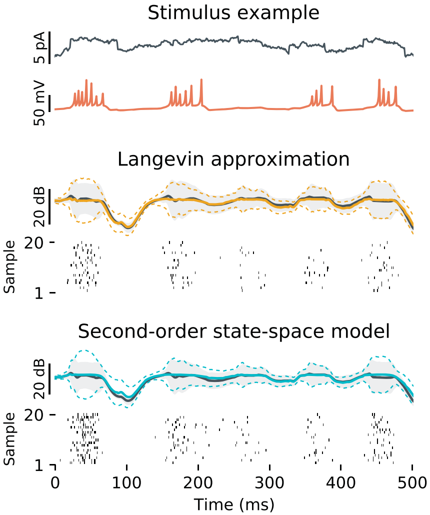
- Reduce the dimension of a large model
- Linear models with Gaussian noise are simpler to analyze

The probability of a neuron spiking can be predicted as a linear function of past variables, followed by a pointwise nonlinearity.

- Not Gaussian, but a Gaussian approximation is ok because predictions average over many spikes.

Process history ~ Gaussian process; Propagate in time using

- Moment-matching (“moment closure”), or
- Locally-quadratic approximation (improved stability)



Rule, M. and Sanguinetti, G. (2018). Autoregressive point processes as latent state-space models: A moment-closure approach to fluctuations and autocorrelations. Neural computation, 30(10):2757–2780

Neural field models for latent state inference

G. Sanguinetti D. Schnoerr M. H. Hennig E. Sernagor G. Hilgen

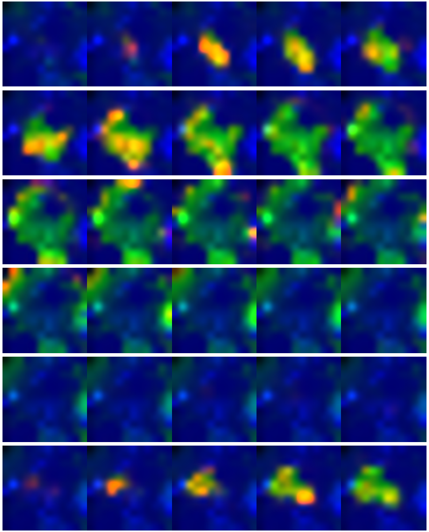
Neural field models are simple descriptions of spatiotemporal neural population dynamics.

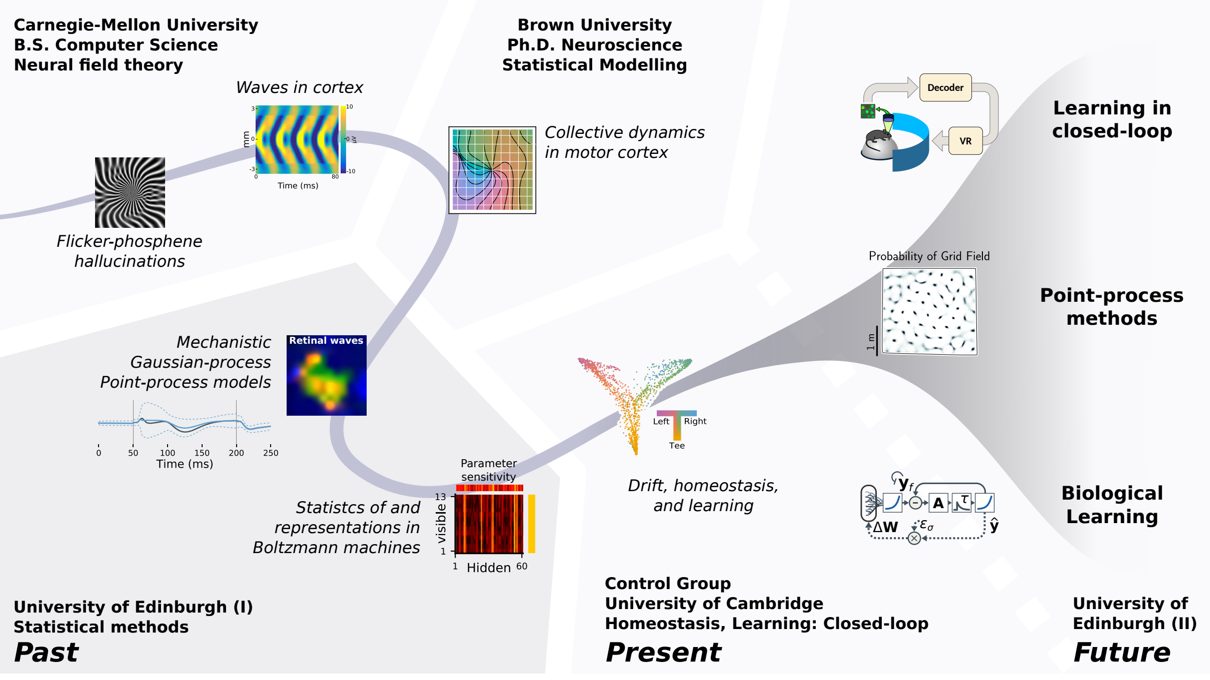
- Partial Differential Equations (PDEs) over space and time.
- # neurons often $\longrightarrow \infty$ so that noise averages away.

Model noise in finite populations $\longrightarrow$ *stochastic* PDE.

- Use as a latent-variable model of waves in the retina.
- 3 states: Quiescent, Active, Refractory
  (compare to Susceptible, Infected, Recovered)
- Approximate latent distribution as a Gaussian process
- Estimate from spiking observations via Bayesian filtering

Rule, M. E., Schnoerr, D., Hennig, M. H., and Sanguinetti, G. (2019). Neural field models for latent state inference: Application to large-scale neuronal recordings. PLoS computational biology, 15(11):e1007442




Carnegie-Mellon University
B.S. Computer Science
Neural field theory
Waves in cortex
mm
Time (ms)
Flicker-phosphene hallucinations
Brown University
Ph.D. Neuroscience
Statistical Modelling
Collective dynamics in motor cortex
Decoder
VR
Learning in closed-loop
Probability of Grid Field
1 m
Point-process methods
Mechanistic
Gaussian-process
Point-process models
Retinal waves
0
50
100
150
200
250
Time (ms)
Parameter sensitivity
13
visible
1
1
Hidden
60
Statistcs of and representations in Boltzmann machines
Left
Right
Tee
Drift, homeostasis, and learning
A
ΔW
ŷ
Biological Learning
University of Edinburgh (I)
Statistical methods
Past
Control Group
University of Cambridge
Homeostasis, Learning: Closed-loop
Present
University of Edinburgh (II)
Future

Are binary latent-variable models like the early sensory system?

M. Sorbaro

Restricted Boltzmann Machines (RBMs) are latent-variable models where the observed/latent variables are Bernoulli-distributed conditioned on each other.

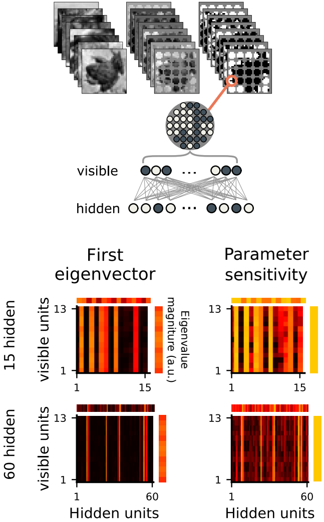
Martino Sorbaro was training RBMs to model population spiking activity, kept finding models close to "critical".

Rare stimuli suppressed variability

- Noise limits bandwidth; Rare stimuli need more bits to encode
- Incoming stimuli reduce neural variability[1]
- Suppression of firing can be informative[2]

∞-large models: 1/f statistics and rank $\approx$ 1 Fisher information matrix[3]

- Implicit prior on model's statistics
- Finite models also approximate this, given sufficient capacity.
- $\Longrightarrow$ Easy to measure a weight's importance from local activity



Rule, M. E., Sorbaro, M., and Hennig, M. H. (2020). Optimal encoding in stochastic latent-variable models. Entropy, 22(7):714

[1]Churchland & al. (2010), Echeveste & al. (2020), [2]Schneidman (2011), [3]Schwab (2014)

Can we understand developmental pruning via information theory?

C. Scholl

The brain removes neurons during maturation. How does it choose which ones?

Fisher Information Matrix (FIM): amount a model changes when we change its parameters ≈ Importance

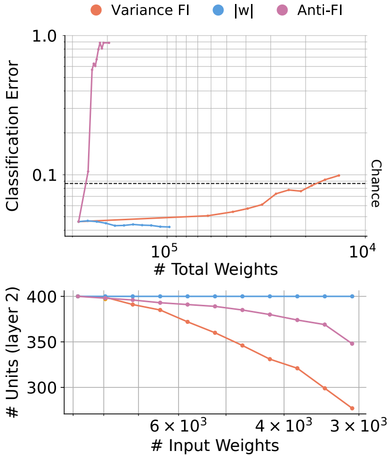
- Only the diagonal can be detected locally. But, if FIM ≈ rank-1, the diagonal is ok.

In Boltzmann machines, weight FI depends on pre/post activity

- More pre-post correlations ⟹ more important
- Prune small weights: **reduced** error, didn't shrink network
- FI pruning: good performance, shrinks network

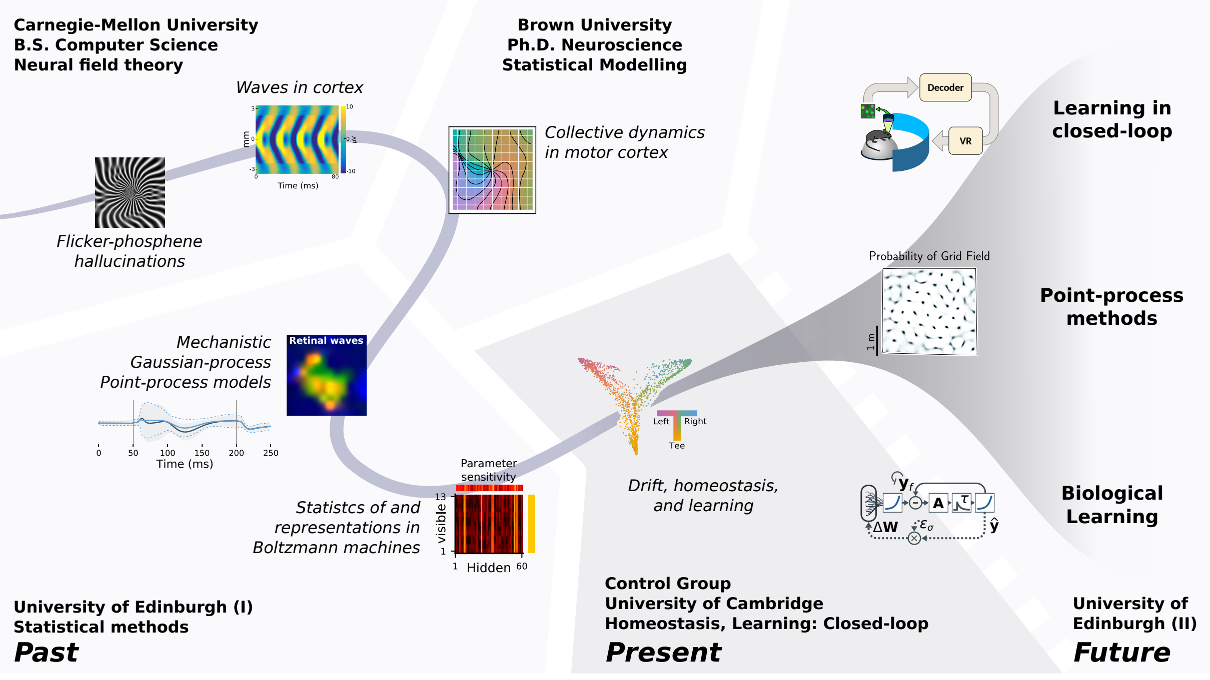
Must re-train post-pruning[1]

- *Use homeostasis for "internal" transfer learning during pruning?*

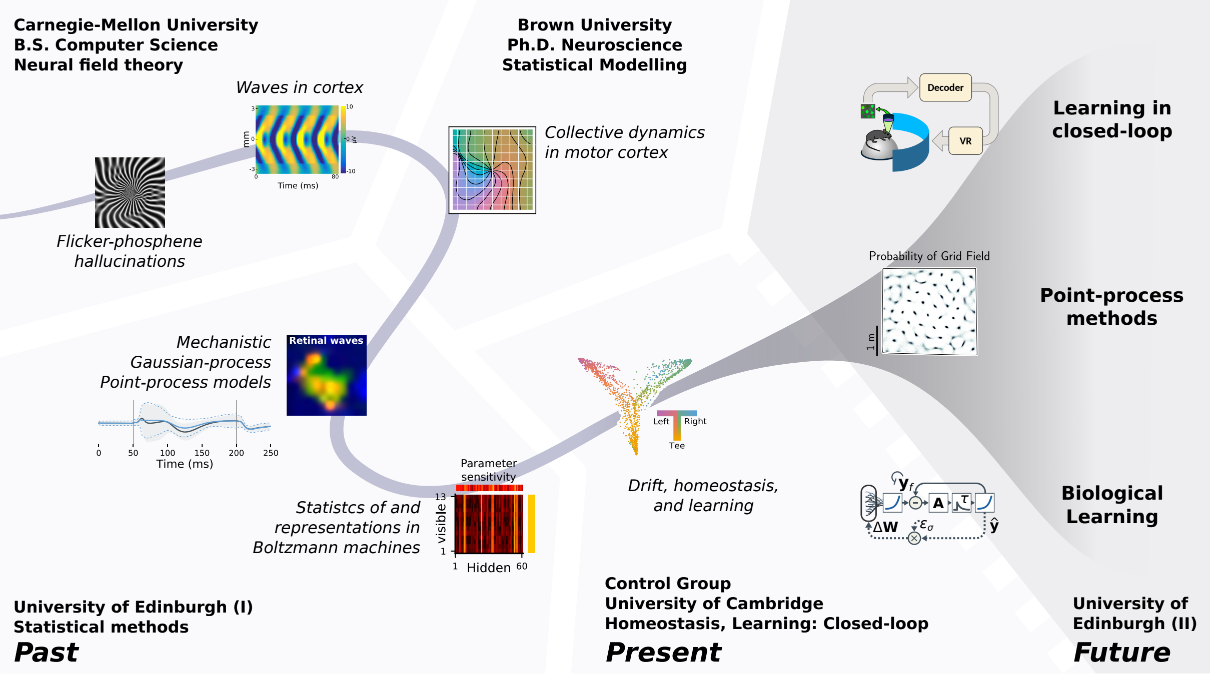


Scholl, C., Rule, M. E., and Hennig, M. H. (2021). The information theory of developmental pruning: Optimizing global network architectures using local synaptic rules. PLoS computational biology, 17(10):e1009458

[1]Not the best approach for industrial ML: Crowley & al. (2018, 2019)



Carnegie-Mellon University
B.S. Computer Science
Neural field theory
Waves in cortex
mm
Time (ms)
Flicker-phosphene hallucinations
Brown University
Ph.D. Neuroscience
Statistical Modelling
Collective dynamics in motor cortex
Decoder
VR
Learning in closed-loop
Probability of Grid Field
1 m
Point-process methods
Mechanistic
Gaussian-process
Point-process models
Retinal waves
Time (ms)
Parameter sensitivity
visible
Hidden
Statistcs of and representations in Boltzmann machines
Left
Right
Tee
Drift, homeostasis, and learning
ΔW
A
Biological Learning
University of Edinburgh (I)
Statistical methods
Past
Control Group
University of Cambridge
Homeostasis, Learning: Closed-loop
Present
University of Edinburgh (II)
Future


Carnegie-Mellon University
B.S. Computer Science
Neural field theory
Waves in cortex
mm
Time (ms)
Flicker-phosphene hallucinations
Brown University
Ph.D. Neuroscience
Statistical Modelling
Collective dynamics in motor cortex
Decoder
VR
Learning in closed-loop
Probability of Grid Field
1 m
Point-process methods
Mechanistic
Gaussian-process
Point-process models
Retinal waves
0
50
100
150
200
250
Time (ms)
Left
Right
Tee
Parameter sensitivity
13
visible
1
1
Hidden
60
Statistcs of and representations in Boltzmann machines
Drift, homeostasis, and learning
A
ΔW
Biological Learning
University of Edinburgh (I)
Statistical methods
Past
Control Group
University of Cambridge
Homeostasis, Learning: Closed-loop
Present
University of Edinburgh (II)
Future

Gaussian processes for grid cells


J. Krupic


P. C. Vayalambrone


T. O'Leary

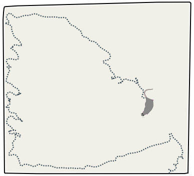
"Grid cells" in hippocampal formation fire as a periodic (hexagonal) function of animal's location "$\boldsymbol{x}$".

Can we model distribution of firing-rate maps $\mathrm{P}_{\mathrm{spike}}(\boldsymbol{x})$ from limited data?

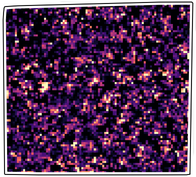
- (Latent) log-rate function $z(\boldsymbol{x})$ with Gaussian-process prior $\mathrm{P}_{z(\boldsymbol{x})} = \mathcal{GP}(\boldsymbol{\mu}_0, \Sigma)$
- Point process observations: $\mathrm{P}_{\boldsymbol{y}(\boldsymbol{x})|z(\boldsymbol{x})} = \mathrm{Poisson}[\boldsymbol{\lambda}(\boldsymbol{x}) = e^{z(\boldsymbol{x})}\, d\boldsymbol{x}]$

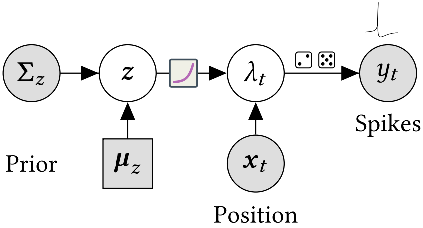
Covariance kernel for periodic grids

- Bessel function; 0^{th} order, 1^{st} kind
- Multiply by a local window



Spikes/Visit





M. Rule, P. Chaudhuri-Vayalambrone, J. Krupic, T. O'Leary (2023). Variational log-Gaussian point-process modelling: Application to hippocampal grid cells. In Preparation.

Gaussian processes for grid cells: Make it faster


J. Krupic


P. C. Vayalambrone


T. O'Leary

Pandemic lockdown: run everything on an old laptop, 2^{16} dimensions, large datasets, how?

Variational Bayes:

- $\mathrm{P}_{\boldsymbol{z}(\boldsymbol{x})|\boldsymbol{y}(\boldsymbol{x})} \approx \mathrm{Q}_{\boldsymbol{z}(\boldsymbol{x})} = \mathcal{GP}(\boldsymbol{\mu}, \Sigma)$
- Σ^{-1} = prior + diagonal update

Fit iteratively

- $\hat{\boldsymbol{\mu}} \leftarrow \underset{\boldsymbol{\mu}}{\operatorname{argmax}} \langle \text{log-prior} + \text{log-likelihood} \rangle_{\mathrm{Q}}{}^{*}$
- $\Sigma \leftarrow \Sigma_0 + \operatorname{diag}[\langle \boldsymbol{\lambda}(\boldsymbol{x}) \rangle]$

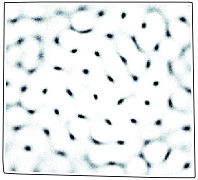
Evaluate on grid

- Bin data with interpolation to ‘pseudopoints’
- Σ_0 becomes a convolution

Low-rank spatial frequency subspace

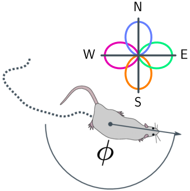
- Discard frequencies ≈ 0 in the prior
- (Fast) Hartley transform gives real-valued components compatible w. Krylov subspace solvers

Probability of Grid Field

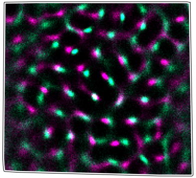


Heading-Dependent Weights

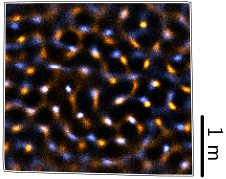
$w = \cos(\phi - \phi_0)^2$



Eastward (green) vs Westward (red)



Northward (blue) vs Southward (orange)



Calcium-imaging brain–machine interface

E. Sorrell T. O'Leary F. Forni C. D. Harvey D. Wilson

Closed-loop control in an adaptive system

- How does the brain adapt motor control to new circumstances?

Brain–Machine interface

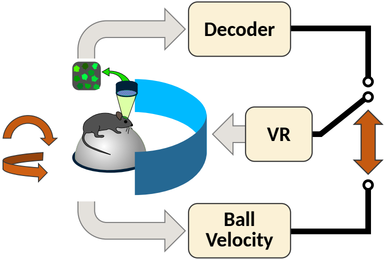
- Force parietal cortex to act as a motor output
- **We** control the motor response to neuronal firing

Decode directly from images

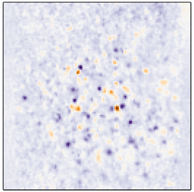
- Calibrate during behavior using stochastic gradient descent
- Test in closed loop

It works

- No substantial tuning drift over five days
- Mean-rates of neurons become sensitive to BMI/control context

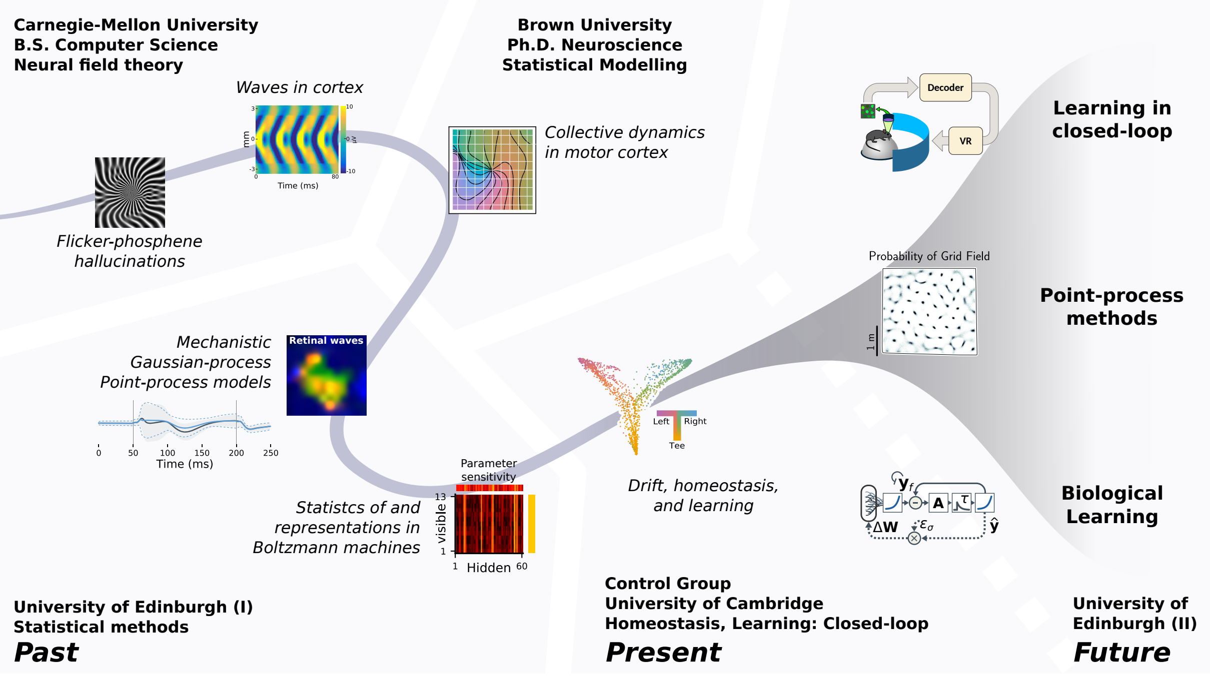


Normalized weights (forward velocity)



Sorrell, E., Rule, M. E., and O'Leary, T. (2021). Brain–machine interfaces: Closed-loop control in an adaptive system. Annual Review of Control, Robotics, and Autonomous Systems, 4

Sorrell, E, Wilson, D, Rule, M, Yang, H, Forni, F, Harvey, C, O'Leary, T. (2022) A Calcium Imaging Based Brain-Machine Interface for Virtual Navigation. In preparation.



Carnegie-Mellon University
B.S. Computer Science
Neural field theory
Waves in cortex
mm
Time (ms)
Flicker-phosphene hallucinations
Brown University
Ph.D. Neuroscience
Statistical Modelling
Collective dynamics in motor cortex
Decoder
VR
Learning in closed-loop
Probability of Grid Field
1 m
Point-process methods
Mechanistic
Gaussian-process
Point-process models
Retinal waves
0
50
100
150
200
250
Time (ms)
Left
Right
Tee
Drift, homeostasis, and learning
Parameter sensitivity
Statistcs of and representations in Boltzmann machines
visible
Hidden
Biological Learning
ΔW
Control Group
University of Cambridge
Homeostasis, Learning: Closed-loop
Present
University of Edinburgh (I)
Statistical methods
Past
University of Edinburgh (II)
Future