Nengo Summerschool 2019

Biologically Detailed Networks and Neuron Models

Peter Duggins, Andreas Stöckel

UNIVERSITY OF WATERLOO

CNRG computational neuroscience research group
compneuro.uwaterloo.ca

June 18, 2018

OUTLINE

I *Motivation & Background*
Why care about biological detail?

OUTLINE

I *Motivation & Background*
Why care about biological detail?

II *Conductance-based n-LIF neurons* $\leftarrow$ Andreas
How to computationally exploit nonlinear dendritic interaction

OUTLINE

I *Motivation & Background*
Why care about biological detail?

II *Conductance-based n-LIF neurons* ← Andreas
How to computationally exploit nonlinear dendritic interaction

III *Detailed NEURON Models and the NEF* ← Peter
How to integrate detailed neuron models into the NEF

OUTLINE

I *Motivation & Background*
Why care about biological detail?

II *Conductance-based n-LIF neurons* ← Andreas
How to computationally exploit nonlinear dendritic interaction

III *Detailed NEURON Models and the NEF* ← Peter
How to integrate detailed neuron models into the NEF

IV `nengo-bio` *hands-on* ← Andreas
Build networks adhering to Dale's principle and exploit dendritic computation

PART I

Motivation & Background

Motivation — *Levels of Analysis*

mechanistic functional

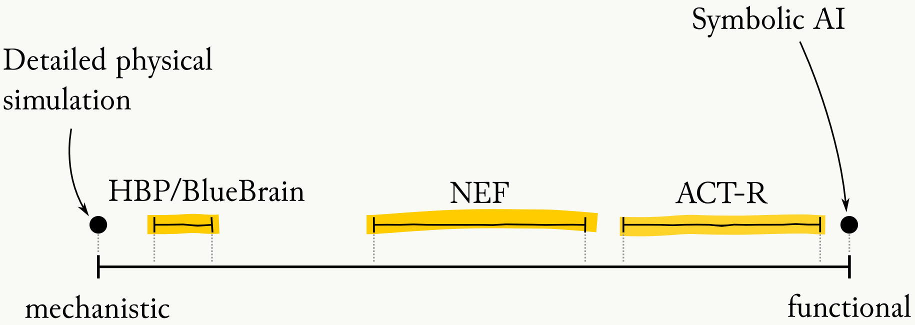
Motivation — *Levels of Analysis*

Motivation — *Levels of Analysis*

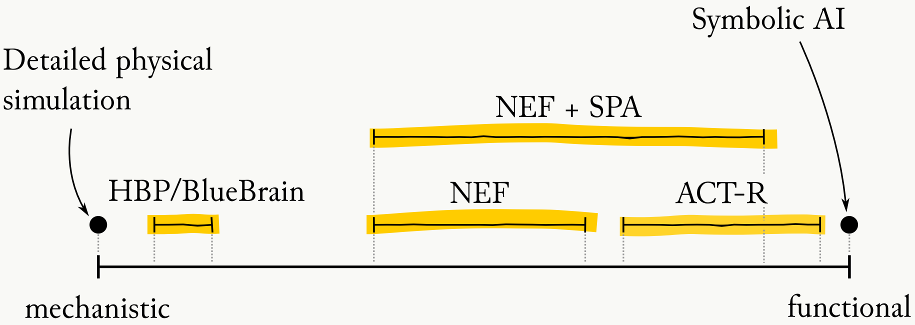
Motivation — *Levels of Analysis*

Motivation — *Levels of Analysis*

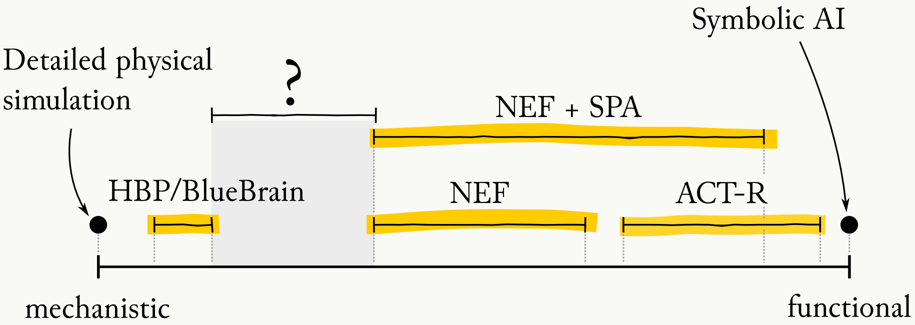
Motivation — *Levels of Analysis*



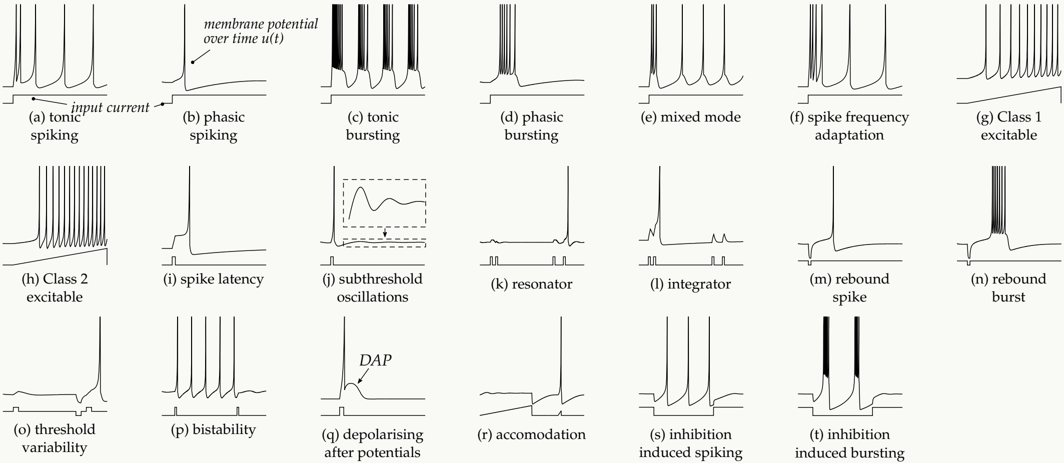
Motivation — *Levels of Analysis*



Motivation — *Levels of Analysis*

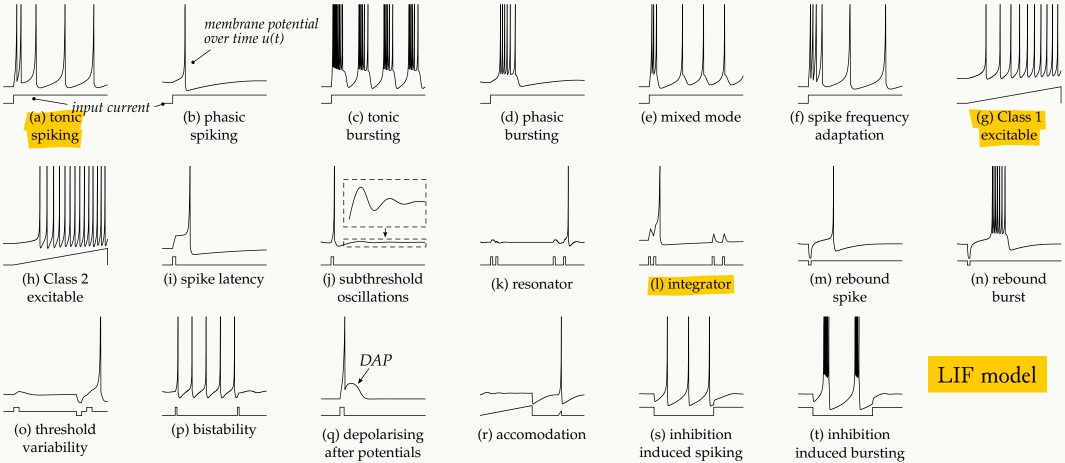


Motivation — *Computational Power of Biological Neurons*



Adapted from Izhikevich (2004), *Which Model to Use for Cortical Spiking Neurons?*

Motivation — *Computational Power of Biological Neurons*



Adapted from Izhikevich (2004), *Which Model to Use for Cortical Spiking Neurons?*

Motivation — *Computational Power of Biological Neurons*

- LIF model accounts for a fraction of the behaviour found in biological neurons

Motivation — *Computational Power of Biological Neurons*

- LIF model accounts for a fraction of the behaviour found in biological neurons
- **Questions:**
  - In how far is the additional detail functionally *relevant*?

Motivation — *Computational Power of Biological Neurons*

- LIF model accounts for a fraction of the behaviour found in biological neurons
- **Questions:**
  - In how far is the additional detail functionally *relevant*?
  - What is its role in terms of *computation*?

Motivation — *Computational Power of Biological Neurons*

- LIF model accounts for a fraction of the behaviour found in biological neurons
- **Questions:**
  - In how far is the additional detail functionally *relevant*?
  - What is its role in terms of *computation*?
  - Can we *harness* these details to perform computation?

Motivation — *Computational Power of Biological Neurons*

- LIF model accounts for a fraction of the behaviour found in biological neurons
- **Questions:**
  - In how far is the additional detail functionally *relevant*?
  - What is its role in terms of *computation*?
  - Can we *harness* these details to perform computation?
- **Example:**
  - Excitatory and inhibitory channels interact nonlinearly.

Motivation — *Computational Power of Biological Neurons*

- LIF model accounts for a fraction of the behaviour found in biological neurons
- **Questions:**
  - In how far is the additional detail functionally *relevant*?
  - What is its role in terms of *computation*?
  - Can we *harness* these details to perform computation?
- **Example:**
  - Excitatory and inhibitory channels interact nonlinearly.

  ⇝ Can we exploit this nonlinearity systematically?

Motivation — *Recreating Idiosyncrasies of the Brain*

① Study Perturbations

- Drugs
- Neurological/mental disorders

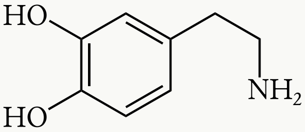
Motivation — *Recreating Idiosyncrasies of the Brain*

❶ Study Perturbations

- Drugs
- Neurological/mental disorders

❷ Neuromodulation

- Model additional pathways for information processing/learning



DOPAMINE

Motivation — *Recreating Idiosyncrasies of the Brain*

❶ Study Perturbations

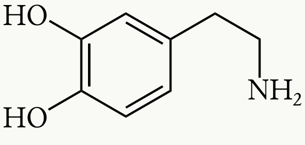
- Drugs
- Neurological/mental disorders

❷ Neuromodulation

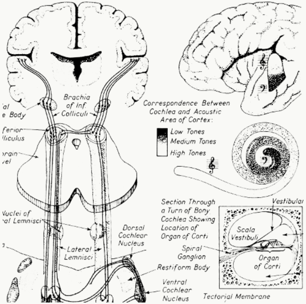
- Model additional pathways for information processing/learning

❸ Constrain Models

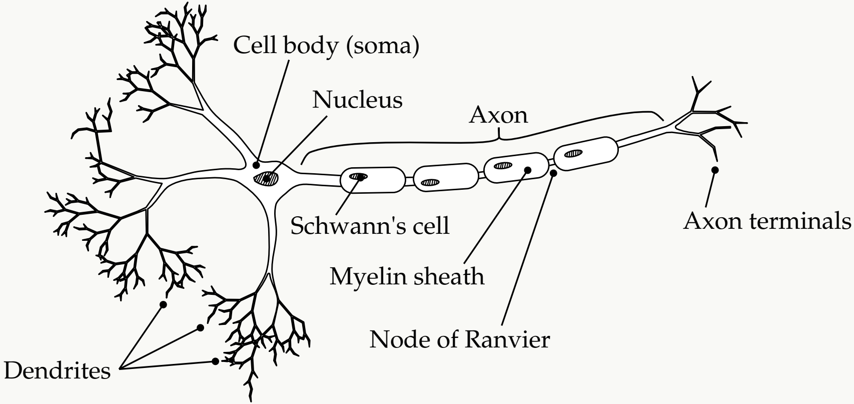
- Constrain models to available biophysical data



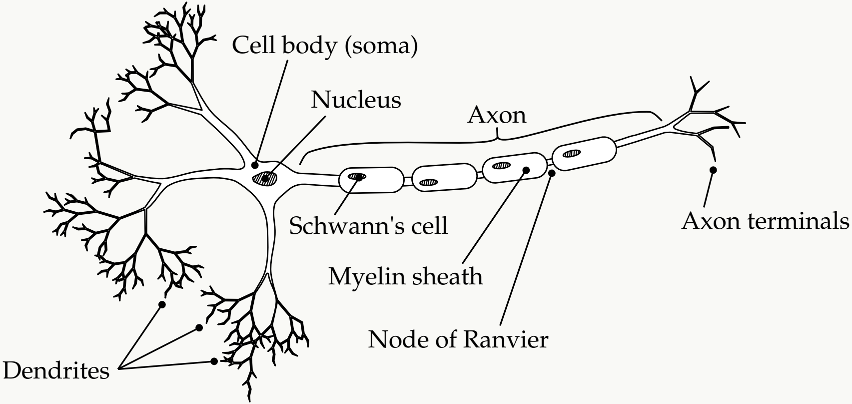
DOPAMINE



Neurobiology — *Idealized "Textbook" Neuron*

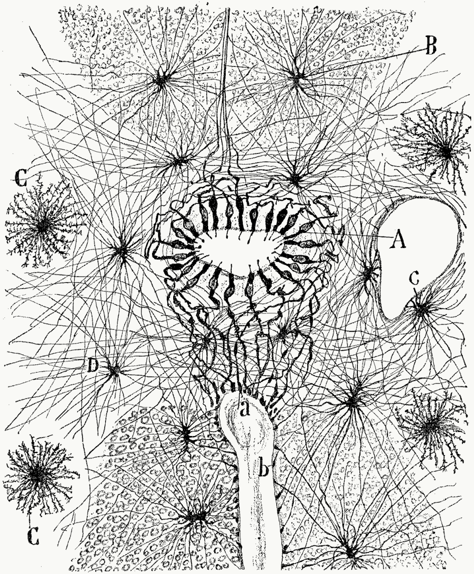


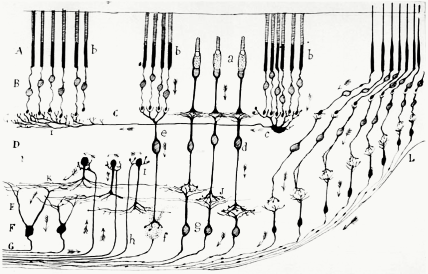
Neurobiology — *Idealized "Textbook" Neuron*

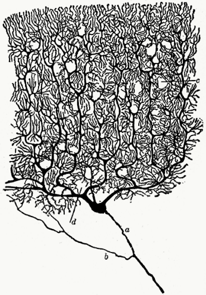


❶ Dendrites collect input ⟶ ❷ Integrated in soma ⟶ ❸ Output spikes travel along axon

Neurobiology — *Neural Heterogeneity*

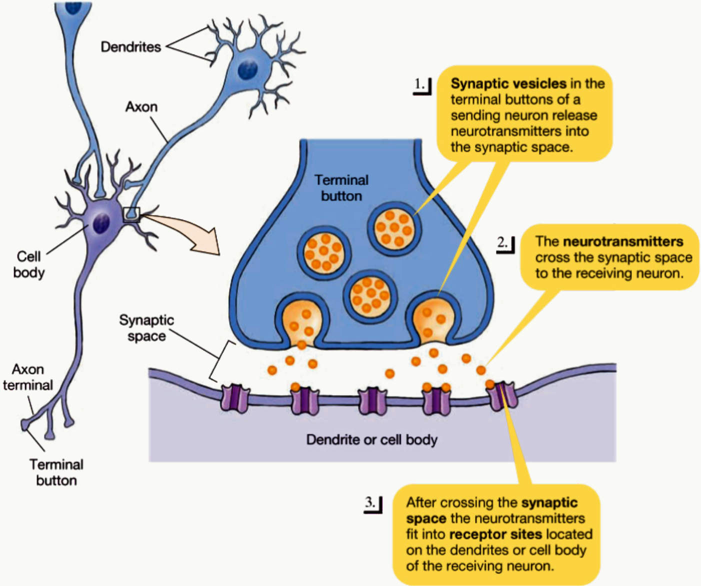


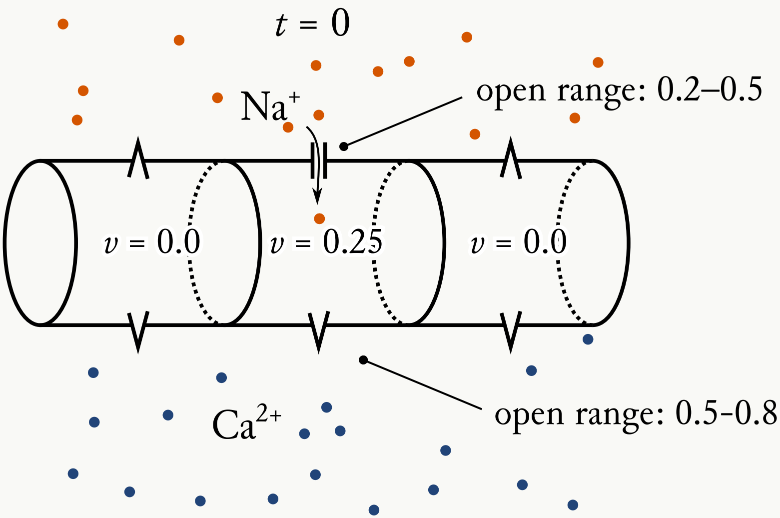


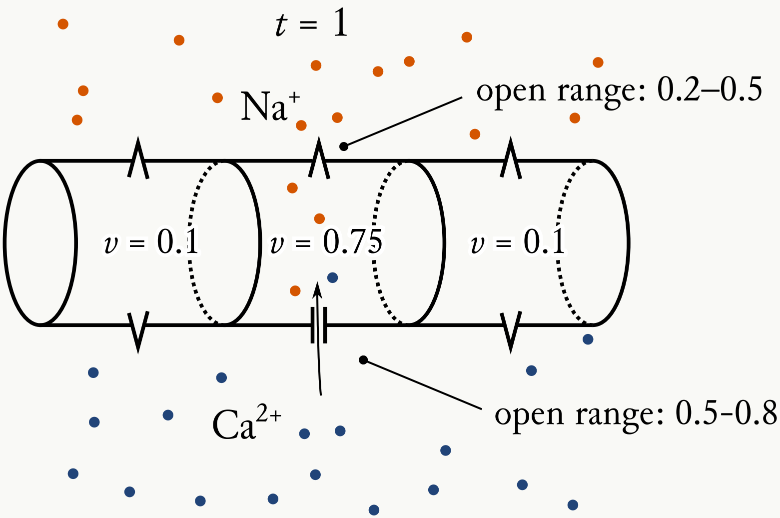


Drawings by
Santiago Ramón y Cajal

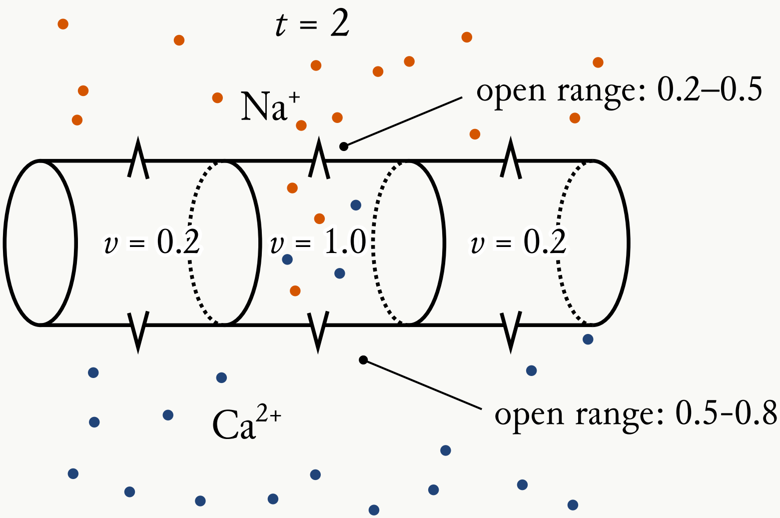
Synapses — *Biophysics*

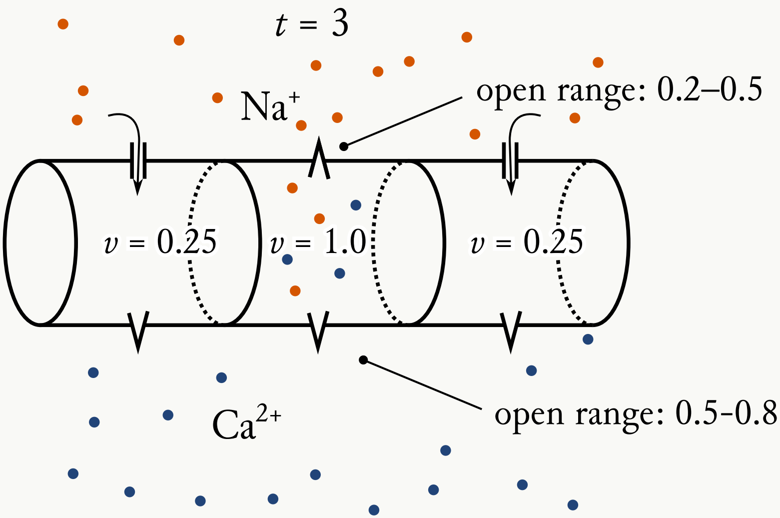



$t = 0$
Na+
open range: 0.2–0.5
$v = 0.0$
$v = 0.25$
$v = 0.0$
Ca2+
open range: 0.5-0.8



$t = 1$
Na^+
open range: 0.2–0.5
$v = 0.1$
$v = 0.75$
$v = 0.1$
Ca^{2+}
open range: 0.5-0.8


$t = 2$
Na^+
open range: 0.2–0.5
$v = 0.2$
$v = 1.0$
$v = 0.2$
Ca^{2+}
open range: 0.5-0.8


t = 3
Na+
open range: 0.2–0.5
v = 0.25
v = 1.0
v = 0.25
Ca2+
open range: 0.5-0.8

Comparing LIF and Biology

Complexity			NEF compatibility		
	LIF	*BIO*		*LIF*	*BIO*
Geometry	point	compartmental	**Tuning curve**	$A = f(\mathbf{x})$	$A = f(\mathbf{x}, t)$
Synapse	current	conductance	**Inputs**	linear filter	synaptic nonlinearity
Dynamics	integrate, leak	ion channels	**Dynamics**	synapse dominates	neuron dominates
	voltage reset	cable equation	**Decoders**	static	time-dependent
Simulation	fast	slow	**Estimates**	$\sum_j a_i(t) * \mathbf{d}_i^f$	?

PART II

Conductance-based n-LIF neurons

Reversal potentials

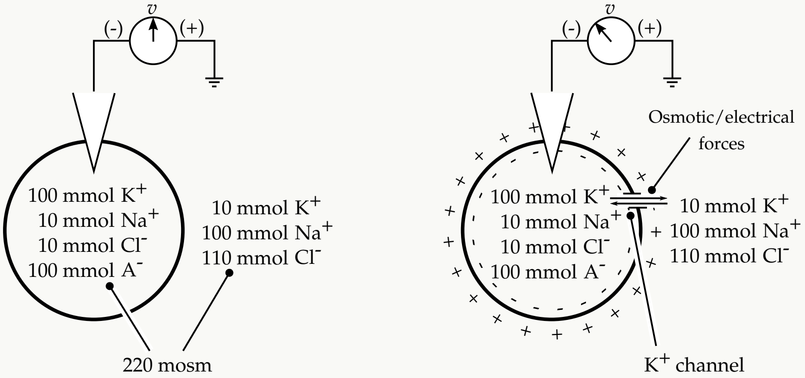
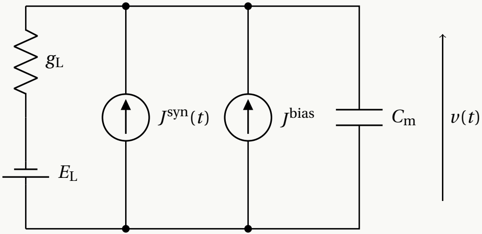


Illustration adapted from Reichert, 2000, Neurobiology.

Ion channels possess a specific *reversal potential* corresponding to the combination of *ion species* they are permeable for.

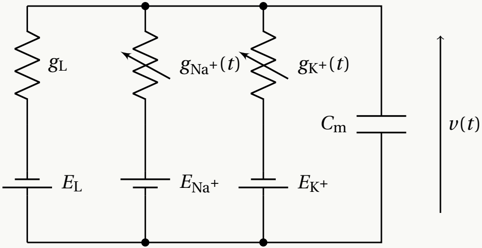
Current vs. Conductance-Based LIF

Current-based LIF



$$
\begin{aligned}
C_\mathrm{m}\dot{u}(t) = J^\mathrm{bias} \\
+ \alpha J^\mathrm{syn}(t) \\
+ g_\mathrm{L}(E_\mathrm{L} - u(t))
\end{aligned}
$$

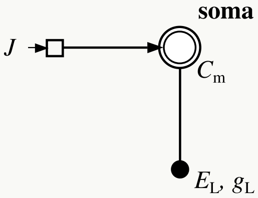
Conductance-based LIF



$$
\begin{aligned}
C_\mathrm{m}\dot{u}(t) = g_\mathrm{E}(t)(E_\mathrm{E} - u(t)) \\
+ g_\mathrm{I}(t)(E_\mathrm{I} - u(t)) \\
+ g_\mathrm{L}(E_\mathrm{L} - u(t))
\end{aligned}
$$

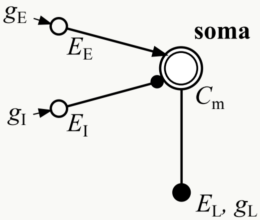
Current vs. Conductance-Based LIF

Current-based LIF



$$
\begin{aligned}
C_\mathrm{m}\dot{u}(t) = {} & J^\mathrm{bias} \\
& + \alpha J^\mathrm{syn}(t) \\
& + g_\mathrm{L}(E_\mathrm{L} - u(t))
\end{aligned}
$$

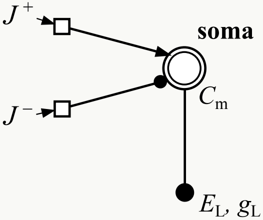
Conductance-based LIF



$$
\begin{aligned}
C_\mathrm{m}\dot{u}(t) = {} & g_\mathrm{E}(t)(E_\mathrm{E} - u(t)) \\
& + g_\mathrm{I}(t)(E_\mathrm{I} - u(t)) \\
& + g_\mathrm{L}(E_\mathrm{L} - u(t))
\end{aligned}
$$

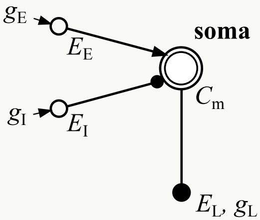
Current vs. Conductance-Based LIF

Current-based LIF



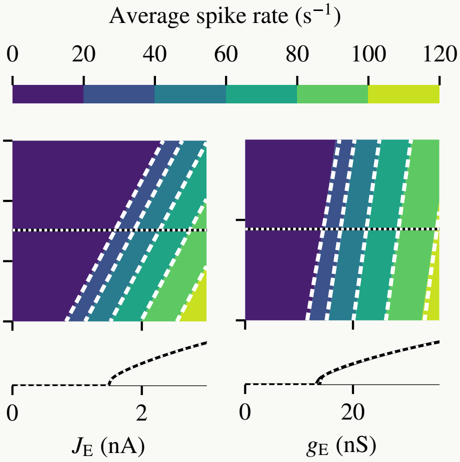
$$
\begin{aligned}
C_\mathrm{m}\dot{u}(t) = J^\mathrm{bias}& \\
+ \alpha\big(J^+(t) - J^-(t)\big)& \\
+ g_\mathrm{L}(E_\mathrm{L} - u(t))&
\end{aligned}
$$

Conductance-based LIF



$$
\begin{aligned}
C_\mathrm{m}\dot{u}(t) &= g_\mathrm{E}(t)(E_\mathrm{E} - u(t)) \\
&+ g_\mathrm{I}(t)(E_\mathrm{I} - u(t)) \\
&+ g_\mathrm{L}(E_\mathrm{L} - u(t))
\end{aligned}
$$

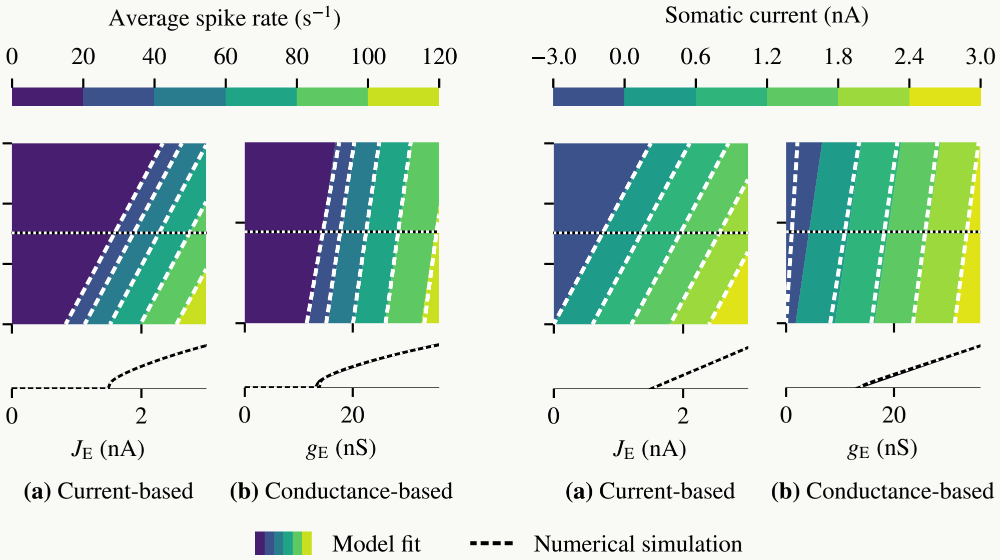
Current vs. Conductance-Based LIF Tuning Curves



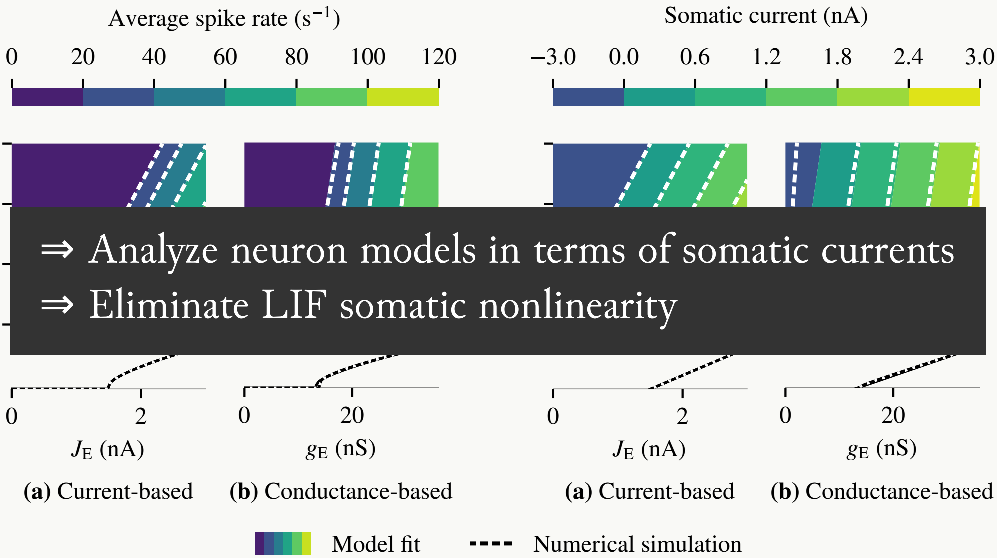
(a) Current-based **(b)** Conductance-based

Model fit Numerical simulation

Current vs. Conductance-Based LIF Tuning Curves



Current vs. Conductance-Based LIF Tuning Curves

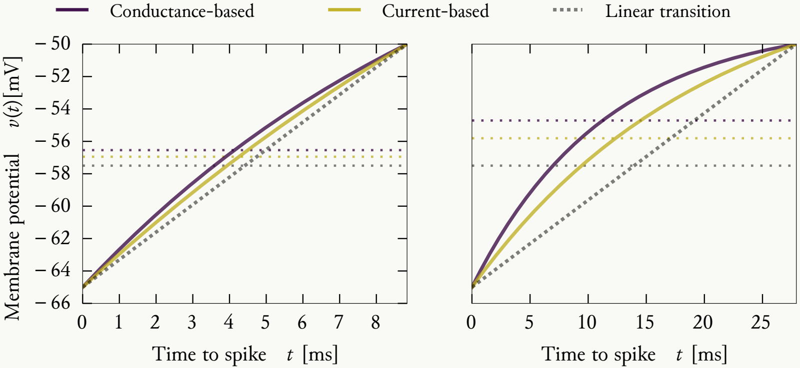


⇒ Analyze neuron models in terms of somatic currents

⇒ Eliminate LIF somatic nonlinearity

Single-compartment conductance-based LIF

- For firing rates $\gg 0$, conductance-based LIF is boring

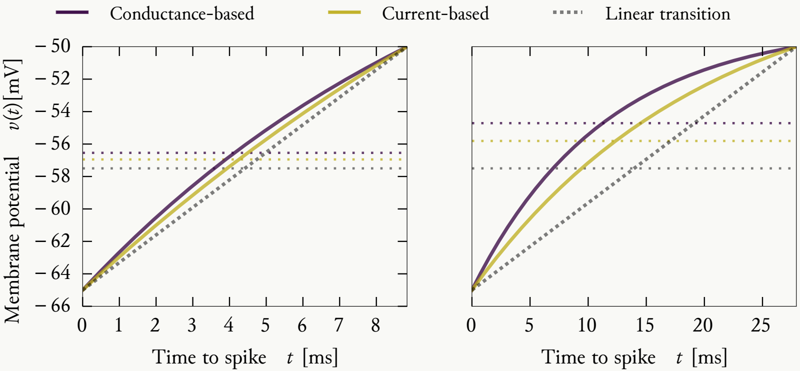


- Can assume average membrane potential $\bar{u}$

$$C_\mathrm{m}\dot{u}(t) = g_\mathrm{E}(t)(E_\mathrm{E} - u(t)) + g_\mathrm{I}(t)(E_\mathrm{I} - u(t)) + g_\mathrm{L}(E_\mathrm{L} - u(t))$$

Single-compartment conductance-based LIF

- For firing rates $\gg 0$, conductance-based LIF is boring

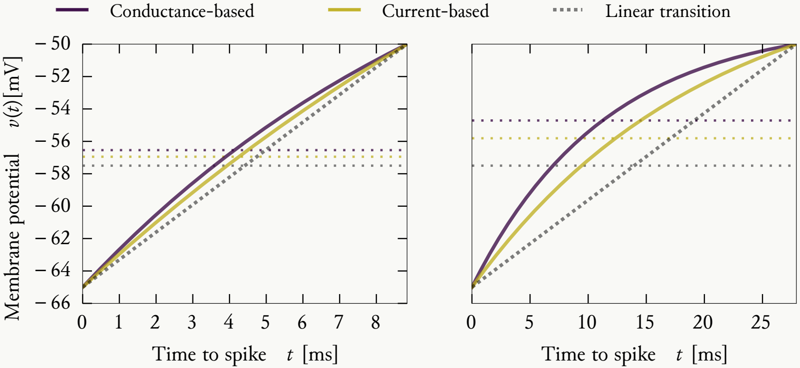


- Can assume average membrane potential $\bar{u}$

$$C_\mathrm{m}\dot{u}(t) = g_\mathrm{E}(t)(E_\mathrm{E} - \bar{u}) + g_\mathrm{I}(t)(E_\mathrm{I} - \bar{u}) + g_\mathrm{L}(E_\mathrm{L} - u(t))$$

Single-compartment conductance-based LIF

- For firing rates $\gg 0$, conductance-based LIF is boring



- Can assume average membrane potential $\bar{u} \Rightarrow$ just a skewed current-based LIF!

$$C_\mathrm{m}\dot{u}(t) = g_\mathrm{E}(t)\alpha_\mathrm{E} + g_\mathrm{I}(t)\alpha_\mathrm{I} + g_\mathrm{L}(E_\mathrm{L} - u(t))$$

Challenges and Extending the NEF

- **Challenges**

Challenges and Extending the NEF

- **Challenges**
  1. NEF assumes a constant *bias current*
     $\Rightarrow$ Reinterpretation of tuning curves; optimize weights in *current-space*

Challenges and Extending the NEF

- **Challenges**
  1. NEF assumes a constant *bias current*
     $\Rightarrow$ Reinterpretation of tuning curves; optimize weights in *current-space*
  2. NEF does not distinguish between *excitatory and inhibitory weights*
     $\Rightarrow$ Solve *non-negative* optimization problems

Challenges and Extending the NEF

- **Challenges**
  1. NEF assumes a constant *bias current*
     $\Rightarrow$ Reinterpretation of tuning curves; optimize weights in *current-space*
  2. NEF does not distinguish between *excitatory and inhibitory weights*
     $\Rightarrow$ Solve *non-negative* optimization problems
- **Extensions**

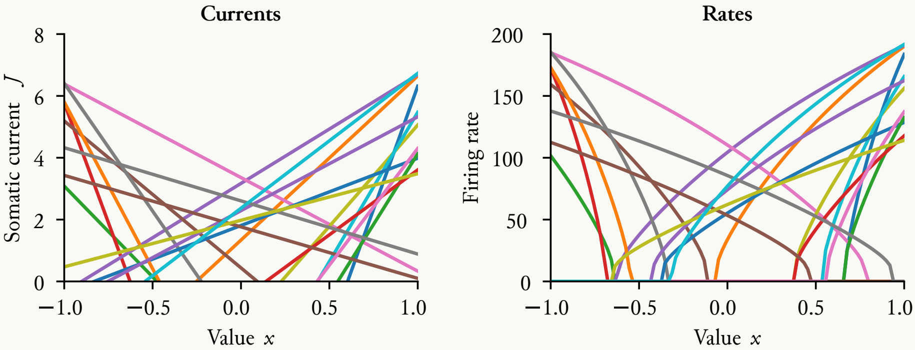
Challenges and Extending the NEF

- **Challenges**
  1. NEF assumes a constant *bias current*
     $\Rightarrow$ Reinterpretation of tuning curves; optimize weights in *current-space*
  2. NEF does not distinguish between *excitatory and inhibitory weights*
     $\Rightarrow$ Solve *non-negative* optimization problems
- **Extensions**
  3. Account for *sub-threshold currents* in the optimization process

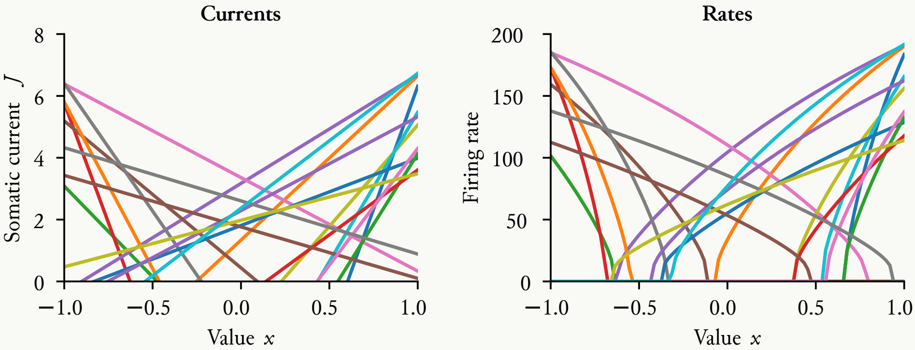
Challenges and Extending the NEF

- **Challenges**
  1. NEF assumes a constant *bias current*
     $\Rightarrow$ Reinterpretation of tuning curves; optimize weights in *current-space*
  2. NEF does not distinguish between *excitatory and inhibitory weights*
     $\Rightarrow$ Solve *non-negative* optimization problems
- **Extensions**
  3. Account for *sub-threshold currents* in the optimization process
  4. Take *dendritic non-linearity* into account

❶ Eliminating the Bias Current

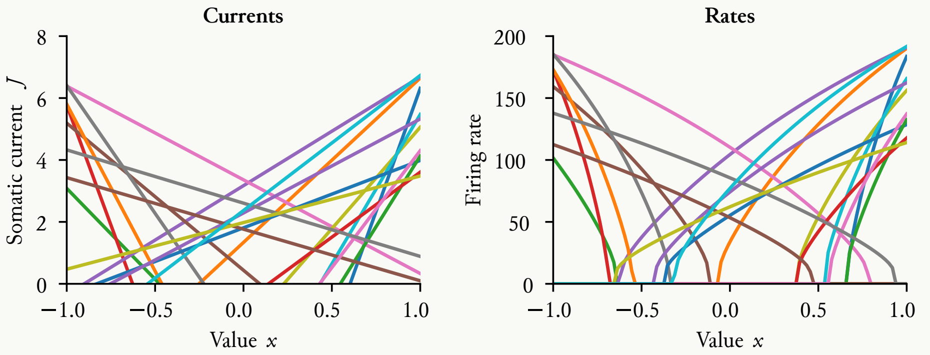


❶ Eliminating the Bias Current



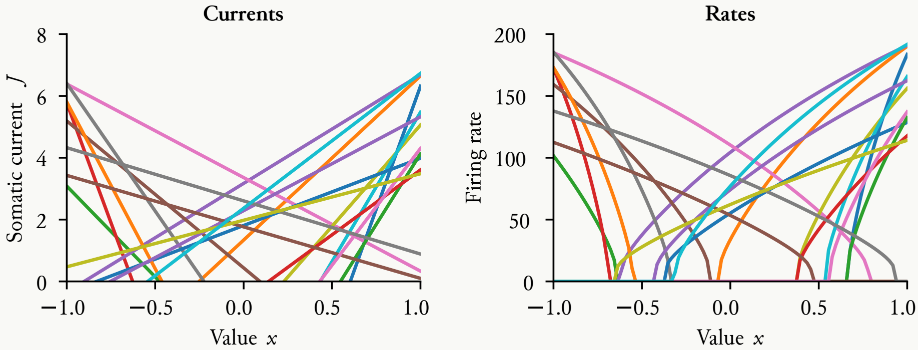
- Interpret tuning curve as normative: "We *want* firing rate a if the neuron represents x"

❶ Eliminating the Bias Current



- Interpret tuning curve as normative: "We *want* firing rate a if the neuron represents x"
- Firing rates correspond to a current J: "We *want* a current J if the neuron represents x"

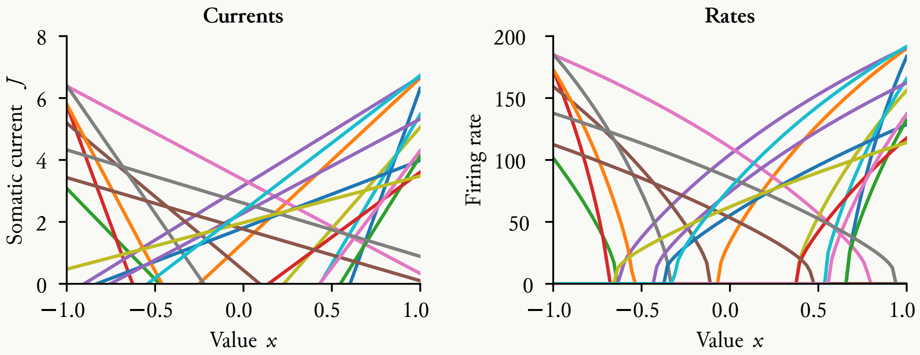
❶ Eliminating the Bias Current



- Interpret tuning curve as normative: "We *want* firing rate a if the neuron represents x"
- Firing rates correspond to a current J: "We *want* a current J if the neuron represents x"

$\Rightarrow$ Find a current-decoder for each individual post-neuron

❶ Eliminating the Bias Current



- Interpret tuning curve as normative: "We *want* firing rate a if the neuron represents x"
- Firing rates correspond to a current J: "We *want* a current J if the neuron represents x"

⇒ Find a current-decoder for **each individual post-neuron** (instead of population-wise)

❷ Nonnegative Weight Optimization

- Assume a post-neuron receives both excitatory and inhibitory input from each pre-neuron $i \Rightarrow$ weight vectors $\vec{w}_i^+, \vec{w}_i^-$.

$$\min_{\vec{w}_i^+, \vec{w}_i^-} \quad \frac{1}{2} \sum_{k=1}^{N} \left\| \vec{w}_i^+ \vec{a}_k^+ - \vec{w}_i^- \vec{a}_k^- - J\big(\langle \vec{e}_i, f(\vec{x}_k) \rangle\big) \right\|_2^2$$

$$= \frac{1}{2} \left\| \vec{w}_i' A' - \vec{j} \right\|_2^2 \quad \text{where } \vec{w}_i' = (\vec{w}_i^+, \vec{w}_i^-),\ A' = (A^+, -A^-)^T,$$

$$\text{and } (\vec{j})_k = J\big(\langle \vec{e}_i, f(\vec{x}_k) \rangle\big),$$

subject to $\vec{w}_i^+ \geq 0, \vec{w}_i^- \geq 0$

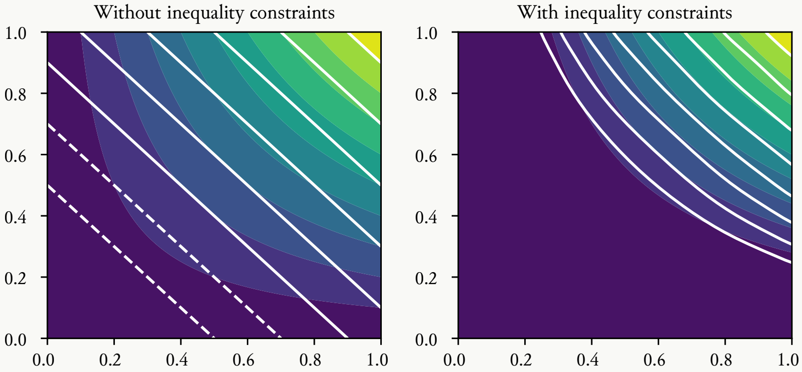
- Can remove rows/columns from A' and $\vec{w}_i$ to account for Dale's principle (e.g. only 20% of pre-neurons are inhibitory, 80% are excitatory)

❸ Account for sub-threshold currents

- Tuning curves are not injective (one-to-one): multiple x map onto a zero output rate

$\Rightarrow$ If the target rate is zero, we do not care about the current, as long as the rate is zero

$\Rightarrow$ Turn zero-rates into inequality instead of equality constraints (quadratic programming)



Target (colour) vs. decoded (white)

4 Take *dendritic nonlinearity* into account

- Generalise dendritic interaction between excitation and inhibition
- Dendritic nonlinearity H converts synaptic state to somatic current

❹ Take *dendritic nonlinearity* into account

- Generalise dendritic interaction between excitation and inhibition
- Dendritic nonlinearity H converts synaptic state to somatic current
- **Example:**
  Current-based model: $H(J^+, J^-) = J^+ - J^-$

$$\min_{\vec{w}_i^+, \vec{w}_i^-} \quad \frac{1}{2} \sum_{k=1}^{N} \left\| H(\vec{w}_i^+ \vec{a}_k^+, \vec{w}_i^- \vec{a}_k^-) - J\big(\langle \vec{e}_i, f(\vec{x}_k) \rangle\big) \right\|_2^2$$

subject to $\vec{w}_i^+ \geq 0, \vec{w}_i^- \geq 0$

❹ Take *dendritic nonlinearity* into account

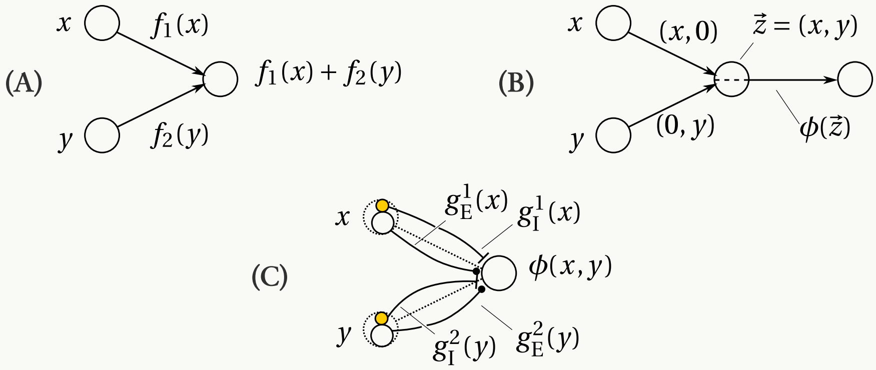
- Generalise dendritic interaction between excitation and inhibition
- Dendritic nonlinearity H converts synaptic state to somatic current
- **Example:**
  Current-based model: $H(J^+, J^-) = J^+ - J^-$

$$\min_{\vec{w}_i^+, \vec{w}_i^-} \quad \frac{1}{2} \sum_{k=1}^{N} \left\| H(\vec{w}_i^+ \vec{a}_k^+, \vec{w}_i^- \vec{a}_k^-) - J\big(\langle \vec{e}_i, f(\vec{x}_k) \rangle\big) \right\|_2^2$$

$$\text{subject to } \vec{w}_i^+ \geq 0, \vec{w}_i^- \geq 0$$

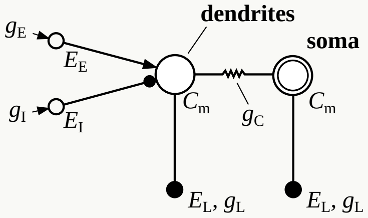
- **Questions:**
  - Can we find more interesting H?
  - Can we exploit H as a computational resource?
  - Under which conditions can we efficiently solve the optimization problem?

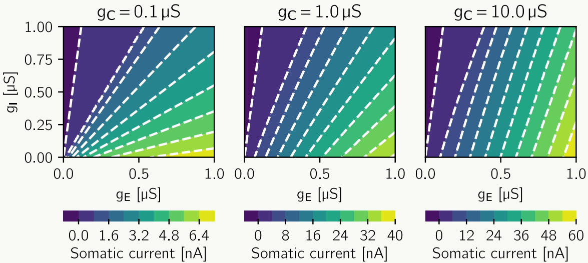
Computing Multivariate Nonlinear Functions



Example: Multiplication, compute $\phi(\vec{z}) = \phi(x, y) = x \cdot y$

Two-compartment LIF neuron

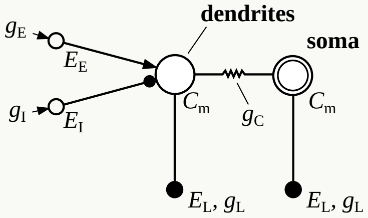


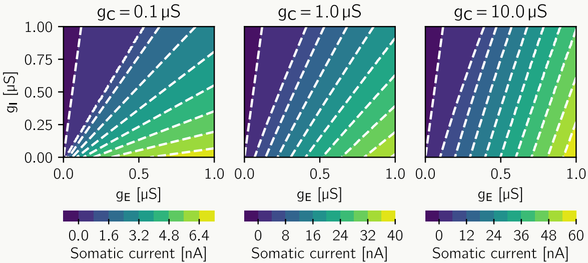


- Dendritic nonlinearity:

$$H(g_E, g_I) = \frac{b_1 + b_2 g_E + b_3 g_I}{a_1 + a_2 g_E + a_3 g_I}$$

Two-compartment LIF neuron



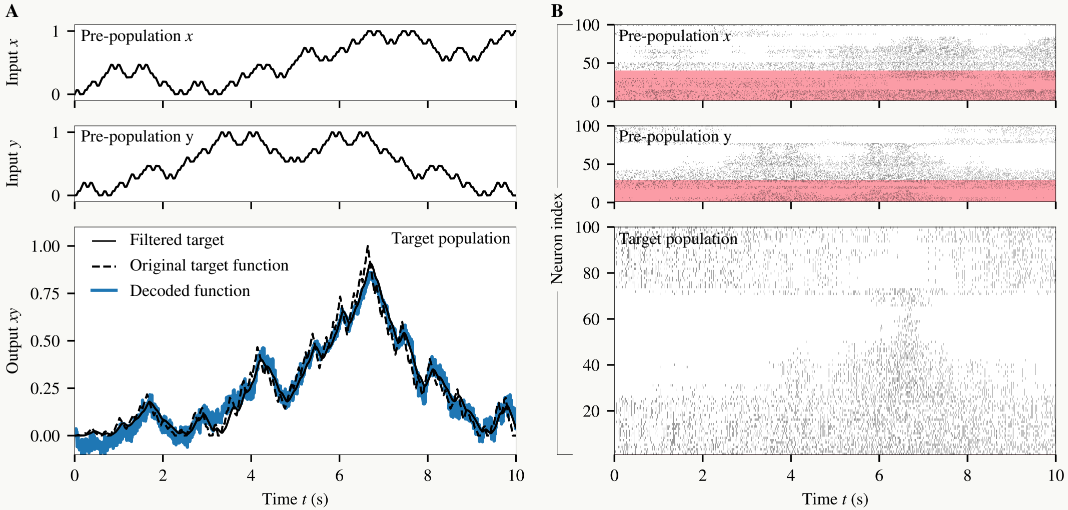


- Dendritic nonlinearity:

$$H(g_E, g_I) = \frac{b_1 + b_2 g_E + b_3 g_I}{a_1 + a_2 g_E + a_3 g_I}$$

- Can still formalize weight-optimization as convex quadratic programming problem, guaranteed to find global optimum

Results – *Dendritic computation* (I)

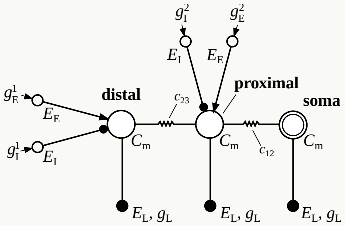


Results – *Dendritic computation* (II)

	Experiment setup						
	Standard LIF			Two comp. LIF $g_C = 50\,\mathrm{nS}$		Two comp. LIF $g_C = 100\,\mathrm{nS}$	
Target	no relaxation	Ⓐ standard	Ⓑ two-layer	Ⓒ standard	noise model	standard	noise model
$x + y$	5.1 ± 0.6%	5.5 ± 1.1%	11.0 ± 1.3%	**3.2 ± 1.1%**	9.1 ± 1.2%	5.1 ± 1.2%	11.5 ± 1.3%
$x \times y$	26.2 ± 0.4%	21.5 ± 6.6%	15.4 ± 4.0%	13.9 ± 2.9%	**11.9 ± 1.8%**	18.2 ± 4.0%	14.3 ± 2.1%
$\sqrt{x \times y}$	14.1 ± 0.4%	19.7 ± 6.1%	16.3 ± 3.0%	9.7 ± 2.6%	**7.1 ± 1.0%**	13.3 ± 4.2%	8.9 ± 1.7%
$(x \times y)^2$	44.5 ± 0.6%	33.0 ± 6.6%	**18.7 ± 6.7%**	27.7 ± 4.1%	27.4 ± 4.1%	34.3 ± 5.3%	30.3 ± 4.3%
$x/(1 + y)$	6.0 ± 0.4%	5.2 ± 0.7%	9.5 ± 0.8%	**3.4 ± 1.0%**	10.0 ± 1.6%	5.3 ± 1.3%	14.0 ± 1.9%
$\|(x, y)\|$	8.0 ± 0.4%	5.7 ± 1.1%	10.5 ± 1.0%	**3.1 ± 1.3%**	8.9 ± 1.2%	4.3 ± 1.8%	12.3 ± 1.8%
$\mathrm{atan}(x, y)$	10.3 ± 0.3%	8.6 ± 1.0%	13.4 ± 1.1%	**5.8 ± 1.3%**	8.4 ± 1.0%	7.0 ± 1.2%	12.7 ± 1.6%
$\max(x, y)$	14.9 ± 0.3%	10.0 ± 0.9%	11.3 ± 1.4%	**5.5 ± 0.9%**	7.7 ± 0.9%	7.3 ± 0.9%	9.7 ± 1.0%

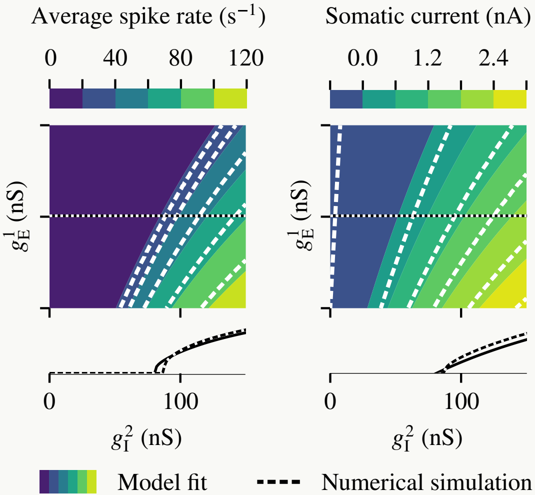
`http://arxiv.org/abs/1904.11713`

Three-compartment LIF neuron

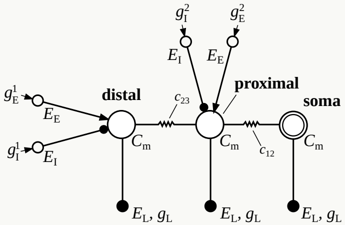


- Dendritic nonlinearity:

$$\left(\frac{b_1^1 + b_2^1 g_E + b_3^1 g_I}{a_1^1 + a_2^1 g_E + a_3^1 g_I}\right) \cdot \left(\frac{b_1^2 + b_2^2 g_E + b_3^2 g_I}{a_1^2 + a_2^2 g_E + a_3^2 g_I}\right)$$



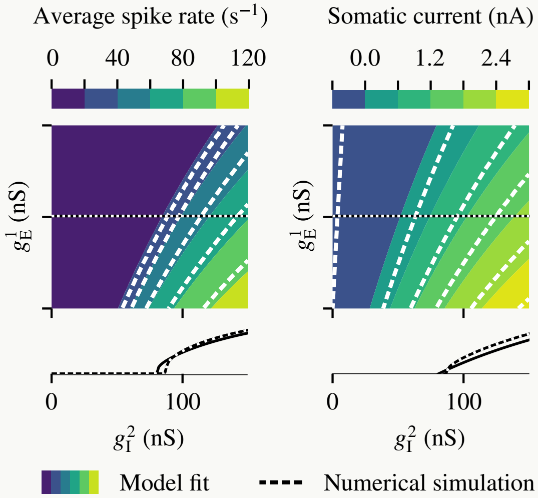
Three-compartment LIF neuron



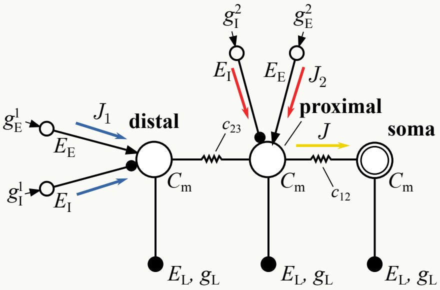
- Dendritic nonlinearity:

$$\left(\frac{b_1^1 + b_2^1 g_\mathrm{E} + b_3^1 g_\mathrm{I}}{a_1^1 + a_2^1 g_\mathrm{E} + a_3^1 g_\mathrm{I}}\right) \cdot \left(\frac{b_1^2 + b_2^2 g_\mathrm{E} + b_3^2 g_\mathrm{I}}{a_1^2 + a_2^2 g_\mathrm{E} + a_3^2 g_\mathrm{I}}\right)$$

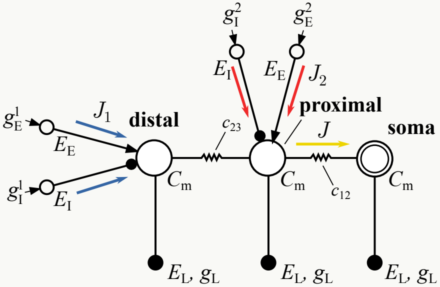
- For n-LIF, $n \geq 3$: can formalize weight-optimization as iterative trust-region-based optimization problem; *not* guaranteed to find global optimum

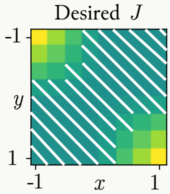


Three-compartment LIF neuron – *Example*

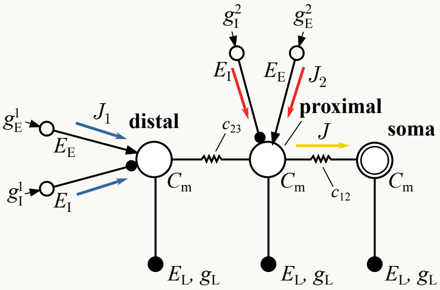


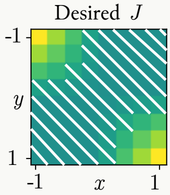
Three-compartment LIF neuron – *Example*

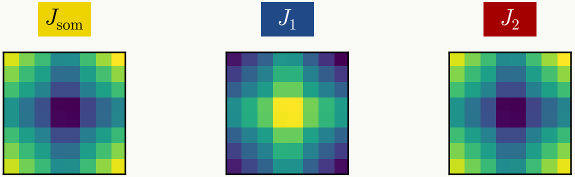




Three-compartment LIF neuron – *Example*

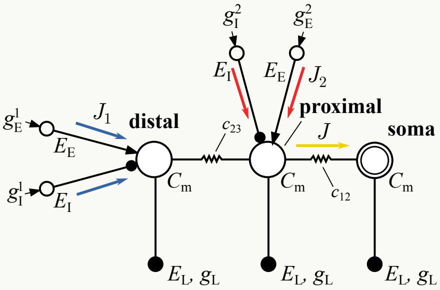


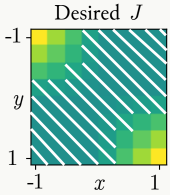


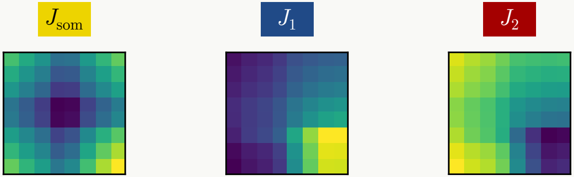


ITERATION 0

Three-compartment LIF neuron – *Example*

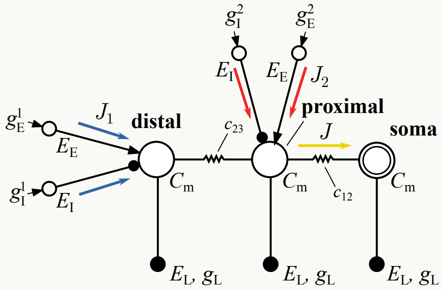


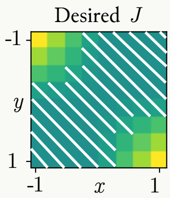


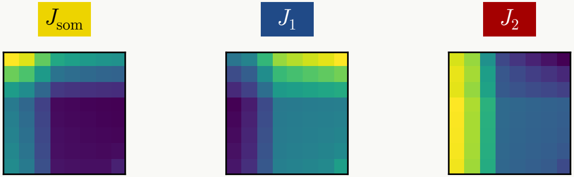


ITERATION 1

Three-compartment LIF neuron – *Example*

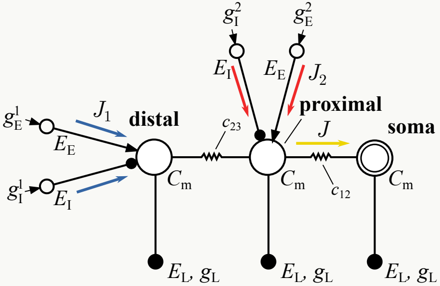


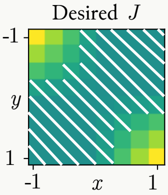




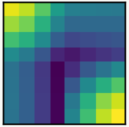
ITERATION 2

Three-compartment LIF neuron – *Example*

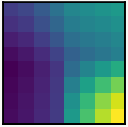




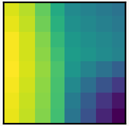
J_{som}



J_1

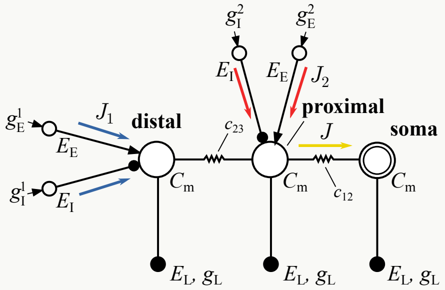


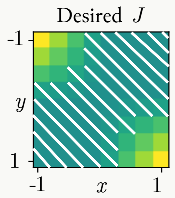
J_2

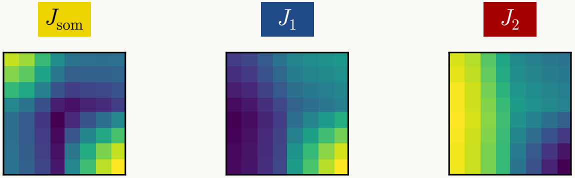


ITERATION 3

Three-compartment LIF neuron – *Example*

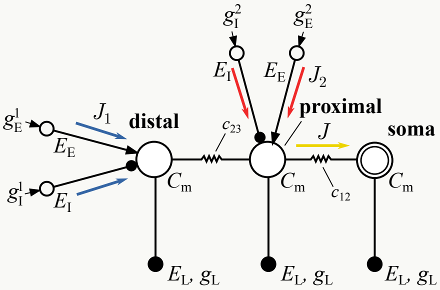


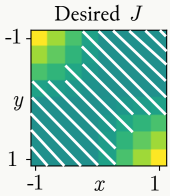


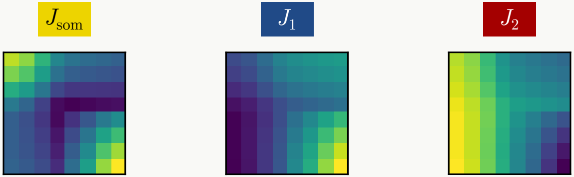


ITERATION 4

Three-compartment LIF neuron – *Example*

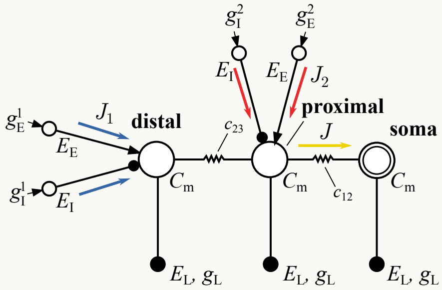


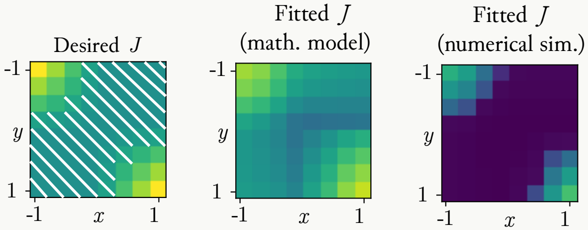


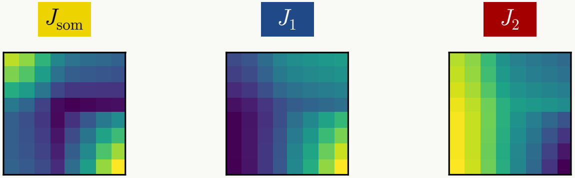


ITERATION 5

Three-compartment LIF neuron – *Example*







ITERATION 5

PART III

Detailed Neuron Models and the NEF

“Black Box” Modelling

Goal: methods to plug arbitrarily-complex neuron models into Nengo

Neuron Model

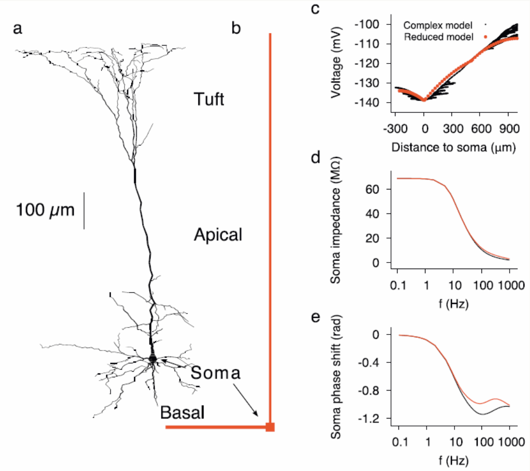


Figure: Morphology and electrophysiology for (a) a detailed reconstruction of a layer 5 pyramidal neuron and (b) a reduced model with 7 anatomical sections, 20 compartments, and 9 ion channels [?].

Training

Want: accurate state-space representation, transformation, dynamics

Training

Want: accurate state-space representation, transformation, dynamics

Setup:

- decompose synaptic $w_{ij} = \mathbf{d}_i \cdot \mathbf{e}_j$
- choose $\mathbf{e}_j$, α_j, β_j s.t. heterogeneous "tuning curves"
- emulate bias with randomly-weighted connection

Training

Want: accurate state-space representation, transformation, dynamics

Setup:

- decompose synaptic $w_{ij} = \mathbf{d}_i \cdot \mathbf{e}_j$
- choose $\mathbf{e}_j$, α_j, β_j s.t. heterogeneous "tuning curves"
- emulate bias with randomly-weighted connection

Training: calculate readout decoders $\mathbf{d}^{out}$:

- Simulate bioneuron network
- Collect activities $a(t)$ and targets $\mathbf{x}(t)$
- $\mathbf{d}^{out} = \text{solver}(a(t), \mathbf{x}(t))$

NEF Principles 0-2

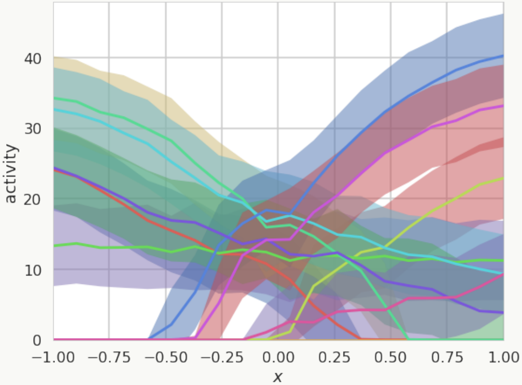


Figure: Heterogeneous bioneuron activities with randomly distributed $\mathbf{e}_j$, α_j, β_j.

NEF Principles 0-2

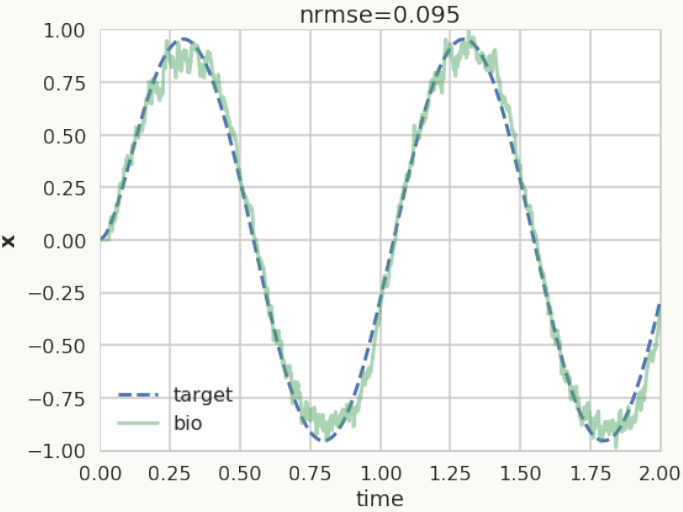


Figure: Principle 1: Representation

NEF Principles 0-2

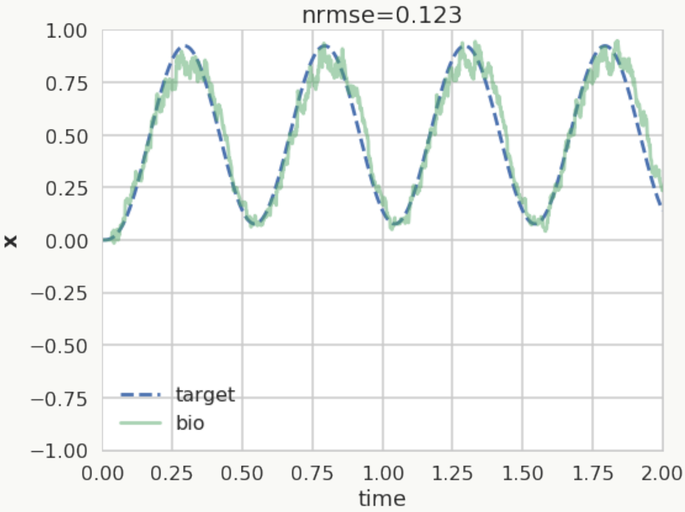


Figure: Principle 2: Transformation

Principle 3: Dynamics

Theory:

- $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$
- natural neural dynamics $\Rightarrow A'\mathbf{x} + B'\mathbf{u} \Rightarrow \mathbf{d}^{ff}, \mathbf{d}^{fb}$
- Problem: $A \Rightarrow A'$ assumes synapse dominates dynamics

Principle 3: Dynamics

Theory:

- $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$
- natural neural dynamics $\Rightarrow A'\mathbf{x} + B'\mathbf{u} \Rightarrow \mathbf{d}^{ff}, \mathbf{d}^{fb}$
- Problem: $A \Rightarrow A'$ assumes synapse dominates dynamics

Question: How to optimize feedback decoders $\mathbf{d}^{fb}$?

- $\mathbf{d}^{fb}$ affects synaptic weights $w^{fb} = \mathbf{d}^{fb} \cdot \mathbf{e}$
- w^{fb} affects $a(t)$
- $a(t)$ affects $\mathbf{d}^{fb} = \text{solver}(a(t), \mathbf{x}(t))$

Principle 3: Dynamics

Theory:

- $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$
- natural neural dynamics $\Rightarrow A'\mathbf{x} + B'\mathbf{u} \Rightarrow \mathbf{d}^{ff}, \mathbf{d}^{fb}$
- Problem: $A \Rightarrow A'$ assumes synapse dominates dynamics

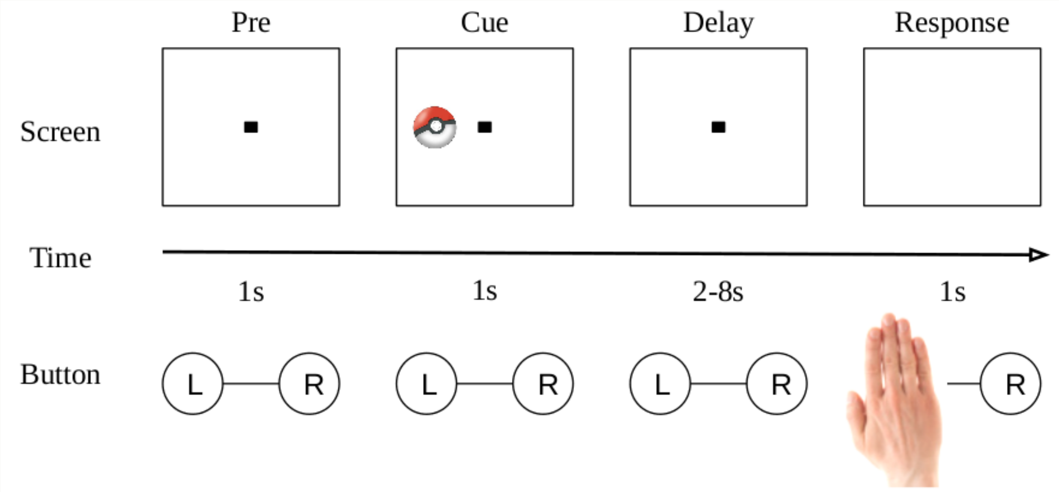
Question: How to optimize feedback decoders $\mathbf{d}^{fb}$?

- $\mathbf{d}^{fb}$ affects synaptic weights $w^{fb} = \mathbf{d}^{fb} \cdot \mathbf{e}$
- w^{fb} affects $a(t)$
- $a(t)$ affects $\mathbf{d}^{fb} = \text{solver}(a(t), \mathbf{x}(t))$

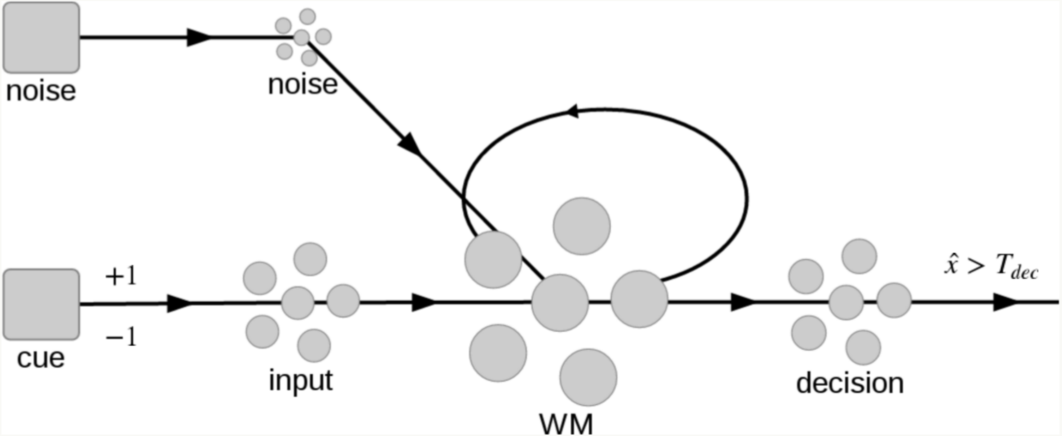
Possible approaches:

- supervise with training spikes “unrolling” the recurrence
- open up the “black box” to characterize internal dynamics
- project to/from higher dimensions with random weights

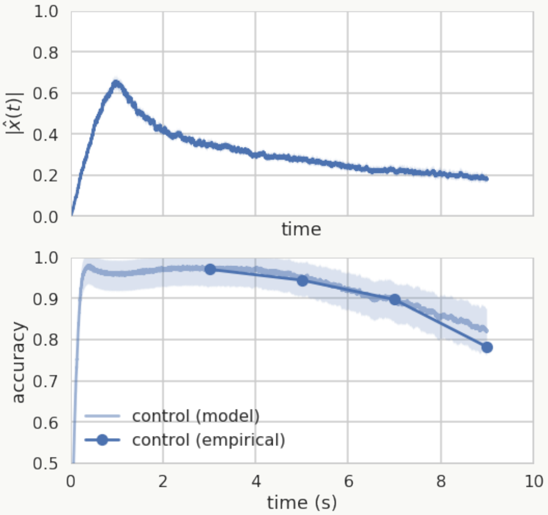
Application: Working Memory



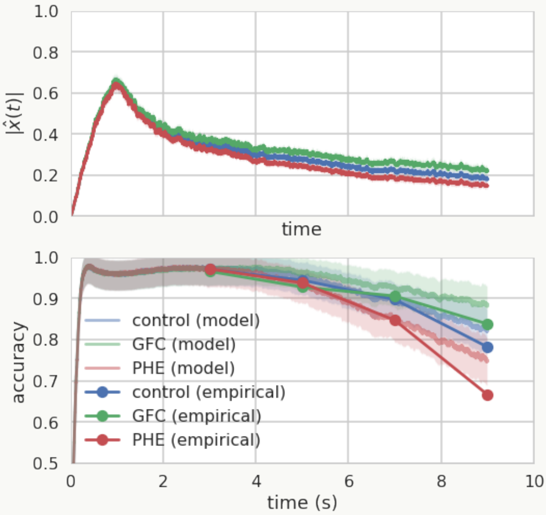
Working Memory Model



Drugs Affect Working Memory



Drugs Affect Working Memory



PART IV

`nengo-bio` hands-on

PART V

Summary & Conclusion

Summary

- Modelling with biologically plausible neurons is interesting for a variety of reasons

Summary

- Modelling with biologically plausible neurons is interesting for a variety of reasons
- Conductance-based synapses can be integrated into the NEF and increase the computational power per neuron

Summary

- Modelling with biologically plausible neurons is interesting for a variety of reasons

- Conductance-based synapses can be integrated into the NEF and increase the computational power per neuron

- Arbitrarily complex neuron models can be substituted into Nengo networks with mixed success (so far)

Summary

- Modelling with biologically plausible neurons is interesting for a variety of reasons

- Conductance-based synapses can be integrated into the NEF and increase the computational power per neuron

- Arbitrarily complex neuron models can be substituted into Nengo networks with mixed success (so far)

Summary

- Modelling with biologically plausible neurons is interesting for a variety of reasons

- Conductance-based synapses can be integrated into the NEF and increase the computational power per neuron

- Arbitrarily complex neuron models can be substituted into Nengo networks with mixed success (so far)

Thank you for your attention!